Efficient Box and Match Algorithm for Reliability-Based Soft-Decision Decoding of Linear Block Codes

Wenyi Jin
LSI Logic
765 Sycamore Drive
Milpitas, CA 95035
wenyi.jin@lsi.com

Marc Fossorier
EE Department
University of Hawaii at Manoa
Honolulu, HI 96822
marc@spectra.eng.hawaii.edu

***Abstract*— In this paper, efficient methods to improve the box and matching algorithm (BMA) are presented. Firstly, an efficient approach is introduced to terminate the decoding if a local optimal candidate satisfies a probabilistic sufficient condition. The false alarm probability associated with the use of the probabilistic sufficient condition is also derived. Secondly, by constructing a control band which is assumed error free, the matching capability of the BMA is enhanced. More precisely, the performance of BMA of order $(i+1)$ is nearly achieved with a small increase in complexity and no increase in memory with respect to the BMA of order i. A tight performance analysis is derived based on the theory of order statistics. An error floor associated either with false alarms or with errors in the control band is introduced, but this error floor can be controlled using the analysis in both cases. Simulation results show that the performance of the enhanced BMA for the decoding of the RS(255,239) code with BPSK signaling over an AWGN channel is about 0.1 dB away from that of maximum likelihood decoding at the word error rate (WER) 10^{-3}.**

I. Introduction

The box and matching algorithm (BMA)[3] is an efficient most reliable basis (MRB) based soft decision decoding algorithm. The BMA roughly reduces the computational cost of the ordered statistic decoding (OSD) algorithm [1] by its squared root at the expense of memory. In addition to considering all codewords associated with error patterns of Hamming weight at most i on the MRB, the BMA with order i also considers all codewords associated with error patterns of Hamming weight at most $2i$ on the s most reliable positions (MRPs), with $s > k$, where k is the dimension of the code. This algorithm is referred to as BMA($i, s-k$) and the $s-k$ values outside the MRB as the control band (CB).

In order to reduce the average complexity, it is desired to know before the computation of a candidate whether it can not be optimal or has a very low possibility to be optimal. This is the case when it does not satisfy some deterministic necessary condition (DNC) or probabilistic necessary condition (PNC) for optimality. It is also desired to know prematurely that a candidate is optimal or has a high probability to be optimal. This is the case when it satisfies some deterministic sufficient condition (DSC) or probabilistic sufficient condition (PSC) for optimality. DNCs and DSCs for MRB list decoding have been studied in [1][5][6][7][8][9] based on a principle first introduced in [4]. A PNC was proposed in [10] and its effectiveness was shown via simulations. DNCs and the DSCs can reduce the average complexity without degrading the error performance. However, the existing ones are not very efficient for long codes at practical signal to noise ratio (SNR) as based on the code minimum distance. PNCs can reduce the computation complexity at the price of performance degradation, which may be not negligible, but need to be properly analyzed [11].

To achieve better performance, BMA($i, s-k$) with larger order i is desired. However, not only the number of processed candidates but also the memory size increase exponentially with i. Hence it is desired to increase the decoding capability of BMA($i, s-k$) by considering an additional set of promising candidates, instead of resorting to BMA($i+1, s-k$).

In this paper, we first introduce a PSC which greatly reduces the average complexity in all SNR regions. This PSC is especially efficient for long codes. We also derive an upper bound for the false alarm probability associated with this PSC. Then we investigate a new method to efficiently improve the performance of BMA($i, s-k$), that can approach or even outperform BMA($i+1, s-k$) without increasing the memory size. We denote this type of algorithm as enhanced BMA($i, s-k$). The PSC is used in the enhanced BMA($i, s-k$) to reduce the average complexity.

II. Preliminary

Let C be a binary (n, k, d_H) linear block code of length n and dimension k with minimum Hamming distance d_H defined by its generator matrix G. Suppose BPSK signaling is used for transmission over an AWGN channel with variance $N_0/2$. Assume that each signal has unit energy. Let $\mathbf{v} = (v_0, v_1, ...v_{n-1})$ be a codeword in C. This codeword is mapped onto a sequence of BPSK signals $\mathbf{c} = (c_0, c_1, ...c_{n-1})$, where $c_i = (-1)^{v_i}$. This code sequence $\mathbf{c}$ is transmitted over the AWGN channel. Let $\mathbf{r} = (r_0, r_1, ...r_{n-1})$ be the received vector. In this case, the reliability of the hard decision $b(r_i) \in F_2$ associated with the received signal $r_i \in \Re$ is simply proportional to its amplitude $|r_i|$.

A. Most Reliable Basis

Finding the MRB is the first step to be performed in MRB reprocessing type algorithms. The basic procedure is the following:

1) Order the received symbols based on their reliability values in decreasing order. This order of received symbols defines a permutation π_1.

2) Permute the columns of generator matrix G based on π_1, which defines a permuted matrix G'. Gaussian elimination is then performed to put G' in reduced echelon form in order to determine the k most reliable independent positions (MRIPs). A second permutation π_2 may be necessary to make this reduced echelon form matrix into a matrix G_s in systematic form. The sequence $\mathbf{r}$ is permuted accordingly to form the vector $\mathbf{y}$ defined as follows:

$$\mathbf{y} = \pi_2[\pi_1[\mathbf{r}]]. \tag{1}$$

Define $\mathbf{y}_M$ as the vector corresponding to the MRB, and define $\mathbf{y}_L$ as the vector corresponding to the least reliable basis (LRB). Hence $|y_{M,i}| \geq |y_{M,j}|$ for $0 \leq i < j < k$, and $|y_{L,i'}| \geq |y_{L,j'}|$ for $0 \leq i' < j' < n-k$. Let $\mathbf{z}' = [\mathbf{z}'_M, \mathbf{z}'_L]$ be the hard decision of $\mathbf{y} = [\mathbf{y}_M, \mathbf{y}_L]$.

Let H be the parity check matrix defined by the MRB. The most right $n-k$ columns of H form the identity matrix and correspond to the LRB. The left k columns correspond to the MRB. In general, these k columns form a dense matrix, defined as D_k.

III. Probabilistic Sufficient Condition (PSC)

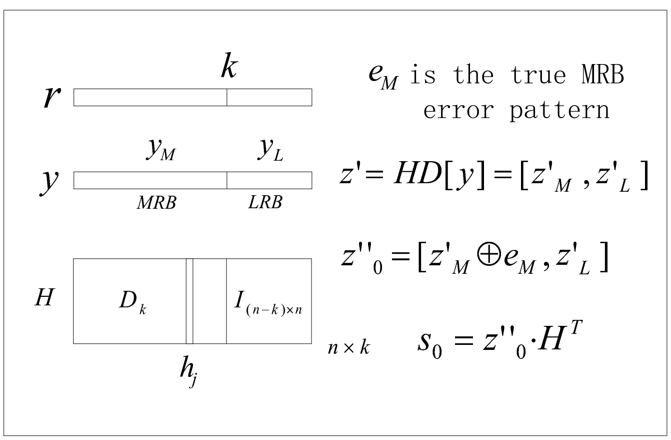

Fig. 1. Preliminaries

Let $\mathbf{e}_M$ be the error pattern corresponding to $\mathbf{y}$ within the MRB. Define vector $\mathbf{z}''_0 = [\mathbf{z}'_M \oplus \mathbf{e}_M, \mathbf{z}'_L]$. Define $\mathbf{s}_0 = \mathbf{z}''_0 \cdot H^T$. It is readily seen that $\mathbf{s}_0$ consists of the linear combination of the columns corresponding to error positions within the LRB. The Hamming weight of $\mathbf{s}_0$ is the number of error bits within the LRB.

A column of the dense matrix D_k can be assumed to follow a binomial distribution and with high probability has a weight close to $(n-k)/2$.

In a MRB-reprocessing algorithm, a list L_{tot} of MRB error patterns $\mathbf{e}_i$ is used to produce candidates. We assume that the patterns $\mathbf{e}_i$ are processed in a predefined order and that a cost function has to be minimized. A pattern $\mathbf{e}_i$ which minimizes this cost function over all patterns processed before $\mathbf{e}_i$ is referred to as a local optimum of L_{tot}. For each $\mathbf{e}_i$, the vector $\mathbf{z}''_i = [\mathbf{z}'_M \oplus \mathbf{e}_i, \mathbf{z}'_L]$ is formed, and the related syndrome $\mathbf{s}_i = \mathbf{z}''_i \cdot H^T$ is computed. It follows that

$$\mathbf{s}_i = \sum_{j:\ \mathbf{e}_{i,j} \neq \mathbf{e}_{M,j}} \mathbf{h}_j \oplus \mathbf{s}_0 \tag{2}$$

Hence $\mathbf{s}_i$ is the summation of $\mathbf{s}_0$ and the columns where $\mathbf{e}_i$ and $\mathbf{e}_M$ are different. Since columns in the dense matrix D_k are assumed to follow a binomial distribution, then with high probability $\mathbf{s}_i$ has a much larger Hamming weight than $\mathbf{s}_0$ if $\mathbf{e}_i \neq \mathbf{e}_M$.

In this paper, we introduce a new PSC for MRB-reprocessing algorithms as follows. Suppose a local optimal candidate corresponding to the MRB error pattern $\mathbf{e}_i$ is found by a MRB-reprocessing algorithm during reprocessing. If the weight of $\mathbf{s}_i$ is smaller than a threshold T, this local optimal candidate is declared to be optimal, and decoding can be terminated. The advantages of this method are : (1) a miss event does not lead to a decoding error; (2) the PSC can greatly reduce the average complexity; (3) the miss probability and false alarm probability can be derived based on the theory of order statistics.

The fact that the syndrome corresponding to a vector with no error in the MRB has in general relatively small Hamming weight was also used in [12][13]. A DSC used for reliability-based syndrome decoding and a PSC used for minimum weight syndrome decoding were derived in [12]. However both this DSC and this PSC still depend on the minimum Hamming distance of the code. In [13], the syndrome weight of the original received vector was compared with a threshold before decoding to determine whether the MRB is error free or not. If the MRB was determined to be error free, then error positions in the LRB were flipped according to the syndrome. Otherwise, a method based on the OSD concept was used to find the error positions within the MRB by processing a list of candidates determined by the columns of the parity check matrix. The efficiency of this decoding algorithm highly depends on the threshold as not only a false alarm event but also a miss event can lead to a decoding error. Furthermore, all the candidates in a sublist are processed by that algorithm, and no performance analysis is provided.

IV. Performance Analysis

A PSC can efficiently reduce the average computation complexity. There are two types of events, however, which are not desirable when a PSC is used. One is the miss event, and the other is the false alarm event.

To analyze these two events, let us first define the PSC checking list L which contains only the MRB error patterns corresponding to all the local optimal candidates in L_{tot}. Hence L is a random set, which depends on the received vector. Define the number of MRB error patterns in L as $|L|$.

In the following analysis, we assume $\mathbf{e}_M \in L_{tot}$, which implies $\mathbf{e}_M \in L$ since only the degradation due to the PSC is considered (see [11] to relate this estimate to the performance of a MRB-reprocessing algorithm). Assume BMA($i, s-k$) is used. All the events defined below are conditioned on a given SNR value.

A. Miss Event

Define the Hamming weight of a vector $\mathbf{v}$ as $w_H[\mathbf{v}]$. Then the miss event E_m is defined as

$$\nexists \quad \mathbf{e}_i \in L \ : \ w_H[\mathbf{s}_i] \leq T \tag{3}$$

The miss event implies that although the optimal candidate is within the search list, it can not be declared optimal when it is processed. Furthermore, there exists no MRB error pattern $\mathbf{e}_i$ in L, $\mathbf{e}_i \neq \mathbf{e}_M$, which satisfies the PSC. As a result all the candidates in the list have to be processed. Hence the miss event does not degrade the error performance.

We provide a simple upperbound of the missing probability:

$$Pr\{E_m\} \leq Pr\{w_H[\mathbf{s}_0] > T\} \tag{4}$$

Recall that $w_H[\mathbf{s}_0]$ is the number of errors in the LRB, which decreases as the SNR increases. With a proper choice of the threshold T, $Pr\{E_m\}$ can become very small.

Since the miss event does not degrade the performance, we mainly focus on the false alarm event, which degrades the error performance.

B. False Alarm Event

Define the set of MRB error patterns A as follows

$$A = \{\mathbf{e}_i : \mathbf{e}_i \in L, \ \mathbf{e}_i \neq \mathbf{e}_M, \ w_H[\mathbf{s}_i] \leq T\} \tag{5}$$

The set A contains all MRB error patterns in L which are not the true MRB error pattern but satisfy the PSC. Define the event $E_{fa,1}$ as

$$|A| \geq 1 \ \text{ and } \ w_H[\mathbf{s}_0] > T \tag{6}$$

For $E_{fa,1}$, although the optimal candidate is in L, the related syndrome does not satisfy the PSC, and there exists at least one MRB error pattern $\mathbf{e}_i$ in L, $\mathbf{e}_i \neq \mathbf{e}_M$, which satisfies the PSC. Hence the related codeword is erroneously declared to be optimal.

Define the set of MRB error patterns B as follows

$$B = \{\mathbf{e}_i : \ \mathbf{e}_i \in A, \ \mathbf{e}_i \text{ is processed before } \mathbf{e}_M\} \tag{7}$$

Define the event $E_{fa,2}$ as

$$|B| \geq 1 \ \text{ and } \ w_H[\mathbf{s}_0] \leq T \tag{8}$$

For $E_{fa,2}$, although the optimal candidate is in L and the related syndrome satisfies the PSC, there exists at least one MRB error pattern $\mathbf{e}_i$ in L, $\mathbf{e}_i \neq \mathbf{e}_M$, which satisfies the PSC and is processed before $\mathbf{e}_M$. In this case too, a decoding error occurs although the optimal candidate is in L and satisfies the PSC. Define the false alarm event E_{fa} as

$$E_{fa} = E_{fa,1} \cup E_{fa,2} \tag{9}$$

Since events $E_{fa,1}$ and $E_{fa,2}$ are disjoint, it follows that

$$Pr\{E_{fa}\} = Pr\{E_{fa,1}\} + Pr\{E_{fa,2}\} \tag{10}$$

$Pr\{E_{fa}\}$ is not easy to be evaluated because the list of candidates considered by BMA($i, s-k$) is a random variable, which is not as structured as that of OSD(i). However the list of candidates considered by BMA($i, s-k$) is a subset of that considered by OSD($2i$), we can therefore set an upperbound of $Pr\{E_{fa}\}$ for BMA($i, s-k$) by deriving an upperbound of (10) for OSD($2i$), which is developed as follows.

Define m as the number of errors in the MRB. Using total probability and based on the assumption that OSD(i) is used, (10) can be written as

$$Pr\{E_{fa}\} = \sum_{j=0}^{i} \big[Pr\{E_{fa,1} \mid m=j\} + Pr\{E_{fa,2} \mid m=j\}\big] \cdot Pr\{m=j\} \tag{11}$$

From (6), we obtain

$$\begin{aligned} & Pr\{E_{fa,1} \mid m=j\} \\ & = Pr\{|A| \geq 1 \ \text{ and } \ w_H[\mathbf{s}_0] > T \mid m=j\} \end{aligned} \tag{12}$$

From (8), we obtain

$$\begin{aligned} & Pr\{E_{fa,2} \mid m=j\} \\ & = Pr\{|B| \geq 1 \ \text{ and } \ w_H[\mathbf{s}_0] \leq T \mid m=j\} \end{aligned} \tag{13}$$

For a MRB error pattern $\mathbf{e}_i \in L, \ \mathbf{e}_i \neq \mathbf{e}_M$, define

$$\begin{aligned} & P_{fa}(j) = Pr\{w_H[\mathbf{s}_\mathrm{i}] \leq T \mid w_H[\mathbf{s}_0] > T, \ m=j\} \\ & = Pr\{w_H[\mathbf{u} \oplus \mathbf{s}_0] \leq T \mid w_H[\mathbf{s}_0] > T, \ m=j\} \end{aligned} \tag{14}$$

where $\mathbf{u}$ is defined as

$$\mathbf{u} = \sum_{j: \ \mathbf{e}_{i,j} \neq \mathbf{e}_{M,j}} \mathbf{h}_j \tag{15}$$

Define the binomial distribution as

$$P(n|N,p) = \binom{N}{n} p^n (1-p)^{N-n} \tag{16}$$

The column weight of $\mathbf{h}_j$ in D_k can be well approximated by $P(w_H[\mathbf{h}_j]|n-k, 0.5)$ if $n-k$ is large enough. This was verified by simulation. Then $w_H[\mathbf{u}]$ can also be approximated by $P(w_H[\mathbf{u}]|n-k, 0.5)$ regardless of how many columns in D_k are involved in the summation of (15).

Note that the weight of a column in D_k can not be smaller than $d_H - 1$, where d_H is the minimum Hamming distance of the code. In fact the weight of $\mathbf{u}$ should range from $d_H - l$ to $n-k$ if l columns are involved in the summation of (15). In the following, these boundary effects are neglected as of minor influence and allowing columns of weight smaller than $d_H - 1$ even increases the probability of a false alarm.

From simulations, we observe that $w_H[\mathbf{s}_0]$ is also well approximated by a binomial distribution $P(w_H[\mathbf{s}_0]|n-k,p)$, where the parameter p depends on the SNR, the number of errors m in the MRB, and is conditioned on the event $\{w_H[\mathbf{s}_0] > T\}$. However, to obtain the distribution of $w_H[\mathbf{u}\oplus\mathbf{s}_0]$, we do not need to know the value of p as shown in the following.

Let $\mathbf{w} = \mathbf{u}\oplus\mathbf{s}_0 = (w_1, w_2, ..., w_{n-k})$, where $w_i = \mathbf{u}_i\oplus\, \mathbf{s}_{0,i}$. It follows that

$$Pr\{\mathbf{u}_i = 1\} = \frac{1}{2} \tag{17}$$

$$Pr\{\mathbf{s}_{0,i} = 1\} = p \tag{18}$$

$$Pr\{\mathbf{w}_i = 1\} = \frac{1}{2} \tag{19}$$

Hence, since $w_H[\mathbf{u}]$ is distributed as $P(w_H[\mathbf{u}]|n-k, 0.5)$, $w_H[\mathbf{u}\oplus\mathbf{s}_0]$ has the same distribution as $w_H[\mathbf{u}]$ if $w_H[\mathbf{s}_0]$ is distributed as $P(w_H[\mathbf{s}_0]|n-k,p)$, regardless of the value of p.

Then based on (2) and (15), (14) can be simplified as

$$P_{fa} = P_{fa}(j) = Pr\{w_H[\mathbf{u}] \le T\} \tag{20}$$

For order i reprocessing, define

$$|L_{tot}(j)| = \sum_{l=0}^{j} \binom{k}{l}, \quad j = 0, 1, ..., i \tag{21}$$

so that

$$\begin{aligned} &Pr\{|A| \ge 1 \mid w_H[\mathbf{s}_0] > T,\ m = j\} \\ &\le 1 - \prod_{i:\ \mathbf{e}_i \in L_{tot}} (1 - P_{fa}) \\ &= 1 - [1 - P_{fa}]^{|L_{tot}(i)|-1} \\ &= P_{|L_{tot}(i)|} \end{aligned} \tag{22}$$

The inequality in (22) follows from the fact that we consider all the MRB error patters in L_{tot} instead of just those in L. Then (12) can be upperbounded by

$$\begin{aligned} &Pr\{E_{fa,1}|\ m = j\} \\ &\le P_{|L_{tot}(i)|} \cdot Pr\{w_H[\mathbf{s}_0] > T|\ m = j\} \end{aligned} \tag{23}$$

To calculate $Pr\{|B| \ge 1 \mid w_H[\mathbf{s}_0] \le T,\ m = j\}$, we use similar approximations. Note that when $m = 0$, $\mathbf{e}_M$ is processed first and no other MRB error pattern can be processed before $\mathbf{e}_M$. In this case $|B| = 0$. Otherwise, we still consider the worst case, in which all the MRB error patterns processed from phase(0) to phase(j) are in L.

Following the analysis methods used above, we obtain

$$\begin{aligned} &Pr\{E_{fa,2}|\ m = j\} \\ &\le P_{|L_{tot}(j)|} \cdot Pr\{w_H[\mathbf{s}_0] \le T|\ m = j\} \end{aligned} \tag{24}$$

Then (11) can be upperbounded as

$$\begin{aligned} &Pr\{E_{fa}\} \\ &\le P_{|L_{tot}(i)|} \cdot Pr\{w_H[\mathbf{s}_0] > T|\ m = 0\} \cdot Pr\{m = 0\} \\ &\quad + \sum_{j=1}^{i} \big[P_{|L_{tot}(i)|} \cdot Pr\{w_H[\mathbf{s}_0] > T|\ m = j\} \\ &\quad + P_{|L_{tot}(j)|} \cdot Pr\{w_H[\mathbf{s}_0] \le T|\ m = j\}\big] \cdot Pr\{m = j\}, \end{aligned} \tag{25}$$

$Pr\{m = j\}$ is computed from ordered statistics [14][15]. for example, we readily obtain

$$\begin{aligned} &Pr\{E_{fa}|OSD(2)\} \\ &\le P_{|L_{tot}(2)|} \cdot Pr\{w_H[\mathbf{s}_0] > T|\ m = 0\} \cdot Pr\{m = 0\} \\ &\quad + \big[P_{|L_{tot}(2)|} \cdot Pr\{w_H[\mathbf{s}_0] > T|\ m = 1\} \\ &\quad + P_{|L_{tot}(1)|} \cdot Pr\{w_H[\mathbf{s}_0] \le T|\ m = 1\}\big] \cdot Pr\{m = 1\} \\ &\quad + P_{|L_{tot}(2)|} \cdot Pr\{m = 2\} \end{aligned} \tag{26}$$

It follows that (26) can be used as an upperbound of $Pr\{E_{fa}\}$ for BMA$(1, s-k)$.

The error probability P_A of a MRB reprocessing type algorithm-A with PSC can be upper bounded by the union bound as

$$P_A \le P_{MLD} + P_{list} + Pr\{E_{fa}\}, \tag{27}$$

where P_{MLD} is the probability of an MLD error and P_{list} is the probability that the transmitted codeword is not in the list considered by algorithm-A. It is desired that

$$Pr\{E_{fa}\} \ll P_{MLD} + P_{list} \tag{28}$$

V. Enhanced BMA

Define e_m as the number of errors in the MRB, and define e_c as the number of errors in the control band. Fig.2 depicts the concept of BMA$(i, s-k)$.

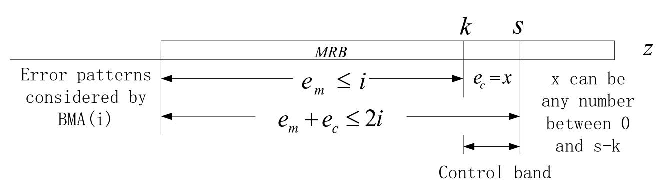

Fig. 2. Concept of BMA$(i, s-k)$.

In this section, we develop the enhanced BMA$(i, s-k)$. To describe the procedure clearly, we assume $i = 2$. We apply the PSC to all the simulations below. The threshold can be selected based on (27) and (28) such that a controlled error floor is allowed. In the following simulations, we select the threshold such that no false alarm event is observed.

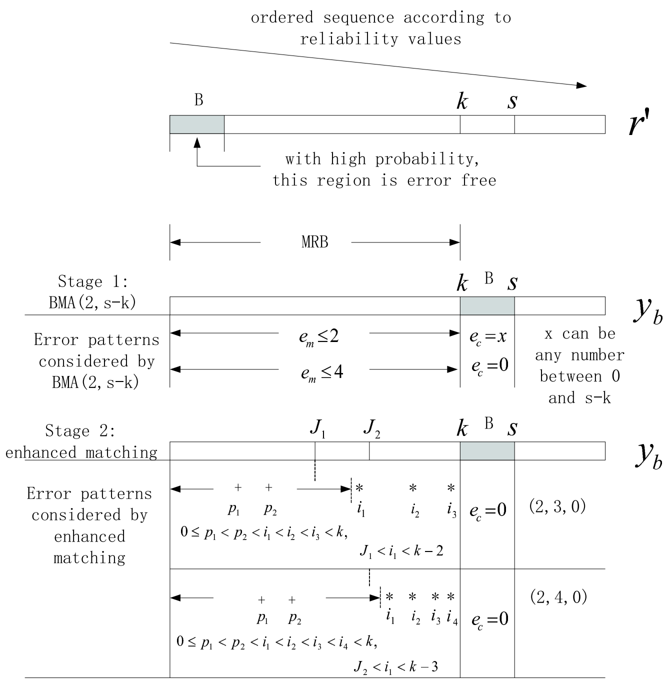

Fig. 3. Concept of $BMA(2, s-k)$ with enhanced matching.

A. BMA($2, s-k$) with Enhanced Matching $\left(EBMA(2, s-k)\right)$

1) Algorithm: Figure 3 depicts an enhanced $BMA(2, s-k)$ $\left(EBMA(2, s-k)\right)$. Define the region of the first $s-k$ positions in the ordered sequence $\mathbf{r}'$ as B. If $s-k$ remains small, with high probability, B is error free, especially for long codes. We first find the MRB of $\mathbf{r}'$ without including any position in B. Then we construct a vector $\mathbf{y}_b$ by placing this MRB in the first k positions of $\mathbf{y}_b$, moving region B to positions from $k+1$ to s of $\mathbf{y}_b$ (as the control band), and placing the remaining positions of $\mathbf{r}'$ in the positions from $s+1$ to n of $\mathbf{y}_b$. Denote the corresponding generator matrix as G_b and the hard decision of $\mathbf{y}_b$ as $\mathbf{z}_b$. It follows that with high probability, the control band of $\mathbf{y}_b$ is error free (i.e $e_c = 0$).

Based on $\mathbf{y}$, $EBMA(2, s-k)$ is performed in two stages. Define e_m as the number of errors in the MRB, and define e_c as the number of errors in the control band. In stage 1, $BMA(2, s-k)$ is performed to correct all the error patterns with $\{e_m \le 2\}$ or $\{e_m \le 4; e_c = 0\}$. In this stage, all the MRB error patterns with Hamming weight 2 are stored in the corresponding boxes following the procedures described in [3]. These boxes are accessed in stage 2, where an enhanced matching is used to correct most of the error patterns with $\{e_m = 5;\ e_c = 0\}$ or $\{e_m = 6;\ e_c = 0\}$. Define $J1$ and J_2 as two parameters used for enhanced matching, where $0 \le J_1 \le k-3$ and $0 \le J_2 \le k-4$. Define p_1 and p_2 as the position index of the MRB error pattern that has been stored in a box in stage 1. The enhanced matching consists of two steps.

Define i_1, i_2, i_3 as the error positions of a MRB error pattern with Hamming weight 3. In the first step, we generate the set A of all the MRB error patterns with Hamming weight 3 such that $J_1 < i_1 < i_2 < i_3 < k$. For each $\mathbf{e}_3 \in A$, there may be a box Φ initialized in stage 1, such that for any MRB error pattern $\mathbf{e}_\Phi$ stored in Φ, the positions from k to s of $\mathbf{z}_b \oplus \left[(\mathbf{e}_3 \oplus \mathbf{e}_\Phi) \otimes G_b\right]$ are all zero. We denote this type of matching as 0-matching. We then uniquely process each of the MRB error pattern $\mathbf{e}_3 \oplus \mathbf{e}_\Phi$. It is readily observed that if $e_m = 5$ and $e_c = 0$, then $\mathbf{y}$ is correctable provided that $pos_3 > J_1$, where pos_3 is the position of the third MRB error of $\mathbf{y}_b$.

Define i_1, i_2, i_3, i_4 as the error positions of a MRB error pattern with Hamming weight 4. In the second step, we generate all the MRB error patterns with Hamming weight 4 such that $J_2 < i_1 < i_2 < i_3 < i_4 < k$. We perform the same 0-matching as described above. It follows that if $e_m = 6$ and $e_c = 0$, then $\mathbf{y}_b$ is correctable provided $pos_3 > J_2$.

Note that in stage 2 of the enhanced matching, we try to approach the decoding capability of BMA(3) with much less candidates processed than for BMA(3). Furthermore, the memory used for the enhanced matching is the same as that of BMA(2). It is readily seen that both the decoding capability and the computation complexity of the second stage depend on J_1 and J_2. The smaller J_1 and J_2 are, the better the performance is. However, the computation complexity of stage 2 increases rapidly as J_1 and J_2 decrease. These values are selected so that the complexity of stage 2 remains close to that of stage 1.

2) Computation Complexity: Define the maximum number of boxes visited in stage 1 as $n_{b,1}$. Define the maximum number of boxes visited in the first step and second step of stage 2 as $n_{b,21}$ and $n_{b,22}$, respectively. Define the maximum number of boxes visited in EBMA($2, s-k$) as $n_{b,EEBMA(2,s-k)}$. It follows that

$$n_{b,EBMA(2,s-k)} = n_{b,1} + n_{b,21} + n_{b,22}, \tag{29}$$

where $n_{b,1} = \binom{k}{1} + \binom{k}{2}$, $n_{b,21} = \binom{k-J_1}{3}$, and $n_{b,22} = \binom{k-J_2}{4}$. As expected, $n_{b,21}$ and $n_{b,22}$ increase exponentially with $k-J_1$ and $k-J_2$.

3) Performance Analysis: In the following analysis, we use definitions closely following those of [14], [15]. Define $\mathbf{u} = (u_0, u_1, ..., u_{n-1})$ as the hard decision of the received vector $\mathbf{r}$, where $u_i = 1$ if $r_i \le 0$ and $u_i = 0$ otherwise. The reliability of this hard decision is taken as $\alpha_i = |r_i|$. In the received sequence $\mathbf{r}$, assume t transmission errors have occurred and the corresponding reliability values are reordered in decreasing order. For $1 \le j \le t$, let $\beta_j(t)$ represent the j-th ordered reliability value among t hard decision errors in a received sequence of length n, so that $\beta_1(t) \ge \beta_2(t) \ge ...\beta_t(t)$. The remaining $n-t$ reliability values corresponding to the correct hard decisions are also reordered in decreasing order. For $1 \le l \le n-t$, let $\gamma_l(n-t)$ represent the l-th ordered reliability value among the remaining $n-t$ correct hard decisions in the received sequence of length n. It follows that $\gamma_1(n-t) \ge \gamma_2(n-t) \ge ...\gamma_{n-t}(n-t)$. The density functions of $\beta_j(t)$ and $\gamma_l(n-t)$ have been expressed in [14] and allow to evaluate probabilities of the form $P(\beta_j(t) \le \gamma_l(n-t))$.

Define the event E_{be} as

$$E_{be} = \{\text{region } B \text{ contains at least one error}\}, \quad (30)$$

and the event E_{list} as

$$E_{list} = \{\text{optimal candidate is not in the list considered by EBMA}(2, s-k)\}, \quad (31)$$

and the event E_{de} as

$$E_{de} = \{\text{ EBMA}(2, s-k) \text{ fails}\} \quad (32)$$

Based on the union bound and total probability, we obtain

$$Pr\{E_{de}\} \leq P_{MLD} + Pr\{E_{list}; E_{be}\} + Pr\{E_{list}; \bar{E}_{be}\} \quad (33)$$

It follows that (see Fig.3)

$$Pr\{E_{list}; E_{be}\} = Pr\{e_m \geq 3,\ e_c \geq 1\} \quad (34)$$

$$\begin{aligned} Pr\{E_{list}; \bar{E}_{be}\} &= Pr\{e_m \geq 7,\ e_c = 0\} \\ &+ Pr\{e_m = 6,\ e_c = 0,\ \ pos_3 > J_1\} \\ &+ Pr\{e_m = 5,\ e_c = 0,\ \ pos_3 > J_2\}, \end{aligned} \quad (35)$$

From (34)-(35), the union bound in (33) can be computed from the joint ordered statistics of $\gamma_i' s$ and $\beta_j' s$ [14], [15].

B. Biased-MRB-EBMA$(2, s-k, a)$

1) Algorithm: For the selected values J_1 and J_2, the biasing method of chapter 4 can be used to further improve performance while the computation complexity increases linearly with the number of biasing iterations.

Figure 4 depicts the concept of biased MRB $EBMA(2, s-k)$, which basically consists of two steps. In the first step, $EBMA(2, s-k)$ defined in Figure 3 is performed. In the second step, we bias the MRB of $\mathbf{y}$ and repeat stage 2 of $EBMA(2, s-k)$ iteratively as follows.

Define the bias as the following binary random variable:

$$\theta = \begin{cases} -a, & p = 1/2 \\ +a, & p = 1/2 \end{cases} \quad (36)$$

where a is a positive real value.

Define H_b as the systematic parity check matrix generated in step 1 of biased $EBMA(2, s-k)$. The first k columns of H_b defines the MRB of $\mathbf{y}_b$. The last $n-k$ columns of H_b represent the identity matrix. Define $\mathbf{w} = (w_0, w_1, ..., w_{k-1})$ with $w_i = y_i + \theta$. The elements of $\mathbf{w}$ are permuted according to the reliability values $|w_i|$, which defines a permutation π_1. The first k columns of the matrix H_b are then permuted according to π_1, which defines a new matrix H_b' . Note that the permuted first k columns of H_b' still defines the MRB. The corresponding generator matrix G_b' can be directly obtained from H_b' without Gaussian elimination . Using the same permutation as that defined by H_b' and H_b, we obtain $\mathbf{y}_b'$ from $\mathbf{y}_b$. There is the same number of errors in the MRB of $\mathbf{y}_b$ and $\mathbf{y}_b'$, but in different positions due to permutation. The error pattern $\{e_m = 6; e_c = 0; pos_3 < J_2\}$ or $\{e_m = 5; e_c = 0; pos_3 < J_1\}$ can not be corrected by $EBMA(2, s-k)$. However, after permutation, it is possible that pos_3 of $\mathbf{y}_b'$ is changed such

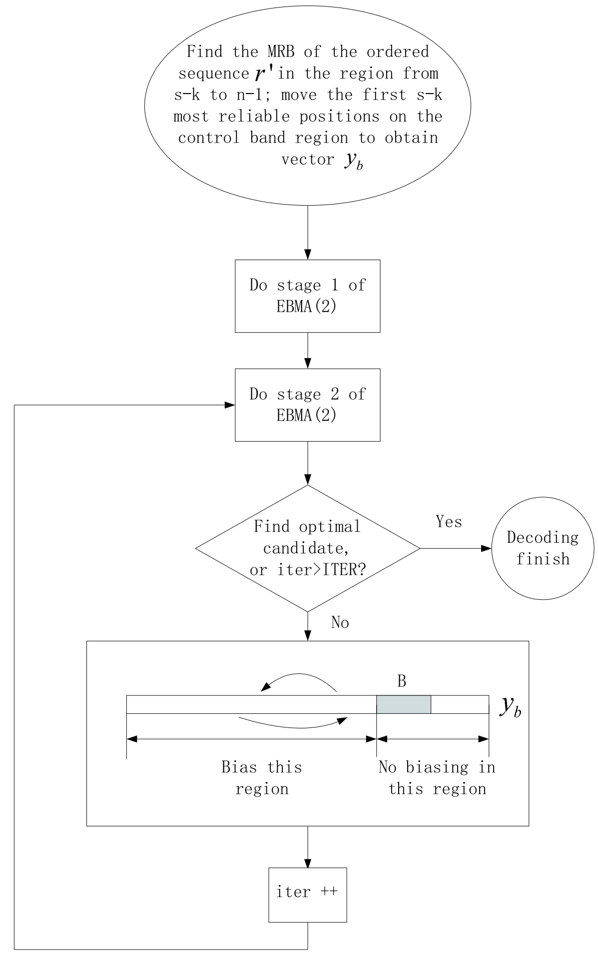

Fig. 4. Concept of $Biased-MRB-EBMA(2, s-k, a)$.

that $\mathbf{y}_b'$ becomes correctable. Since we only bias the MRB, we call this algorithm $Biased-MRB-EBMA(2, s-k, a)$ (or $BM-EBMA(2, s-k, a)$).

2) Performance Analysis: The bias amplitude and the number of iterations determine the decoding capability of BM-EBMA$(2, s-k, a)$. We derive a lower bound $P_{low,BME}$ of BM-EBMA$(2, s-k, a)$ by assuming that all the error blocks with $\{e_m = 6; e_c = 0\}$ or $\{e_m = 5; e_c = 0\}$ are correctable. This is the best performance that BM-EBMA$(2, s-k, a)$ can achieve with any a and a large enough number of iterations.

It follows that

$$\begin{aligned} & P_{low,BM-EEBMA(2,s-k)} \\ = \ & Pr\{e_m \geq 3; e_c \geq 1\} + Pr\{e_m \geq 7; e_c = 0\} \end{aligned} \quad (37)$$

As examples of EBMA$(2, s-k)$ and BM-EBMA$(2, s-k, a)$, we consider the binary image of the (255,239) Reed Solomon (RS) code so that $n = 2040$ and $k = 1912$. Let $s-k = 22$, $J_1 = k - 200$ and $J_2 = k - 100$.

The maximum number of boxes visited by BMA$(2, 22)$ is

$$n_{b,1} = \binom{k}{1} + \binom{k}{2} \approx 2^{20.80} \quad (38)$$

and the maximum number of boxes visited in the second stage of EBMA$(2, 22)$ is

$$n_{b,21} + n_{b,22} = \binom{k - J_1}{3} + \binom{k - J_2}{4} \approx 2^{20.33} + 2^{21.90} = 2^{22.32} \quad (39)$$

We observe that $n_{b,21} + n_{b,22} \approx 3n_{b,1}$, so that the complexity of EBMA$(2, 22)$ remains of the same order as that of BMA$(2, 22)$ $\big($note that $n_{b,1} \approx 2^{30.12}$ for BMA(3,22)$\big)$.

In Fig.5, we plot (33) for EBMA$(2, 22)$ and (37) for BM-EBMA$(2, 22, a)$. We observe that the performance of EBMA$(2, 22)$ is between that of BMA$(2, 22)$ and BMA$(3, 22)$, while the lower bound of BM-EBMA$(2, 22, a)$ can approach that of BMA$(3, 22)$.

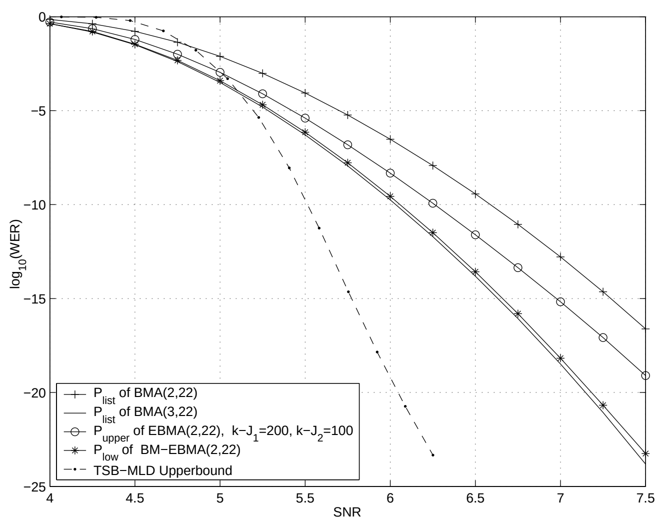

Fig. 5. Error performance analysis of EBMA$(2, 22)$ and BM-EBMA$(2, 22, a)$ for RS (255,239).

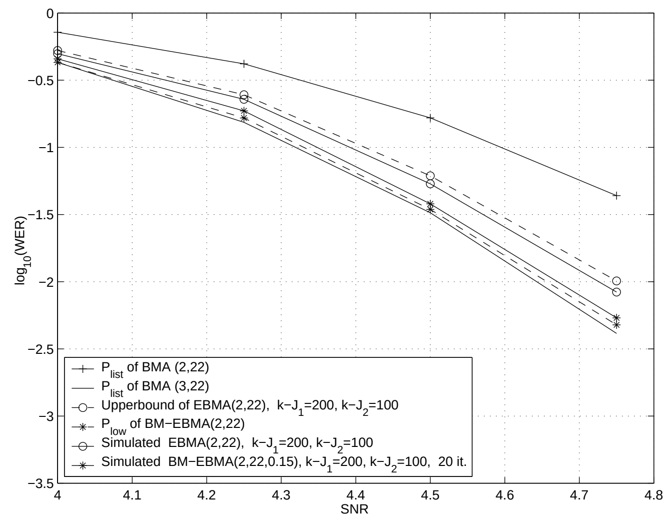

Fig. 6. Simulated error performance of EBMA$(2, 22)$ and BM-EBMA$(2, 22, 0.15)$ for RS(255,239).

In Fig.6, we plot the corresponding simulation results and (33) for EBMA$(2, 22)$. We also plot the simulation results of BM-EBMA$(2, 22, 0.15)$ with 20 iterations.

We observe that the upperbound of EBMA$(2, 22)$ is tight. Simulation results show that EBMA$(2, 22)$ becomes much better than BMA(2,22) with enhanced matching. The performance of BM-EBMA$(2, 22, 0.15)$ with 20 iterations approaches the lower bound of BM-EBMA$(2, 22, a)$, and is very close to the performance of BMA(3). Note that not only the number of candidates considered by BM-EBMA$(2, 22, 0.15)$ with 20 iterations is much smaller than that of BMA(3), but the memory used by BM-EBMA$(2, 22, 0.15)$ is just a small fraction of that used by BMA(3).

C. Biased-Block-EBMA$(2, s - k, a)$

1) Algorithm: In this section, enhanced matching is performed with the biasing method. At each iteration, all the positions of $\mathbf{y}_b$ are biased except the CB. The biased symbols of $\mathbf{y}_b$ outside the CB are reordered in decreasing reliability values, which defines a permutation π_1. The columns of G_b are permuted based on π_1 which defines a permuted matrix G'_b. Gaussian elimination is then performed to put G'_b in reduced echelon form in order to determine the k MRIPs. A second permutation π_2 may be necessary to make this reduced echelon form matrix into a matrix G_s in systematic form. The sequence $\mathbf{y}_b$ is permuted accordingly to form the vector $\mathbf{y}'_b$ defined as follows:

$$\mathbf{y}'_b = \pi_2[\pi_1[\mathbf{y}_b]]. \quad (40)$$

Then $BMA(2, 22)$ with 0-matching and enhanced 0-matching can be performed on $\{G_s, \mathbf{y}'_b, \pi_1, \pi_2\}$. We denote this algorithm as $Biased-block-EBMA(2, s-k, a)\big(BB-EBMA(2, s-k, a)\big)$.

Note that in both BM-EBMA$(2, s-k, a)$ and BB-EBMA$(2, s-k, a)$, each iteration is independent of the others so that it can be performed in parallel, offering a tradeoff with respect to latency. The potential improvement brought by each iteration depends on the magnitude a of θ, J_1 and J_2.

2) Performance Analysis: We derive the list error probability $P_{list,BB-EEBMA(2,s-k,a)}$ of BB-EBMA$(2, s-k, a)$ with the biasing magnitude a, assuming enough iterations are performed.

BB-EBMA$(2, s-k, a)$ contains two stages. In the first stage, BMA$(2, s-k)$ is performed on $\mathbf{z}$. In the second stage, BMA(2) with 0-matching and enhanced 0-matching are performed on the biased sequence $\mathbf{y}'$ iteratively. Define E_{BB1} as

$$E_{BB1} = \{\text{optimal candidate is not in the list considered by Stage 1 of BB-EBMA}(2, s-k, a)\} \quad (41)$$

and E_{BB2} as

$$E_{BB2} = \{\text{optimal candidate is not in the list considered by Stage 2 of BB-EBMA}(2, s-k, a)\} \quad (42)$$

It follows that

$$P_{list,BB-EEBMA(2,s-k,a)} = Pr\{E_{BB1} \text{ and } E_{BB2}\} \quad (43)$$

Define E_{BB11} and E_{BB12} as

$$E_{BB11} = \{E_{BB1} \ \text{ and } \ E_{be}\} \quad (44)$$

$$E_{BB12} = \{E_{BB1} \ \text{ and } \ \bar{E}_{be}\}, \quad (45)$$

where E_{be} was defined in (30).

It follows from total probability that

$$P_{list,BB-EEBMA(2,s-k,a)} = Pr\{E_{BB11} \ \text{ and } \ E_{BB2}\} + Pr\{E_{BB12} \ \text{ and } \ E_{BB2}\} \quad (46)$$

Assume j transmission errors have occurred in $\mathbf{z}$. It is readily derived that

$$\begin{aligned} &Pr\{E_{BB12} \ \ and \ \ E_{BB2}\} = \\ &Pr\{\beta_3(j) > \gamma_{k-2}(n-j) \text{ and } \ \beta_5(j) > \gamma_{s-4}(n-j) \\ &\quad \text{and } \ \gamma_{s-k}(n-j) > \beta_1(j) \\ &\quad \text{and } \ \beta_7(j) - a > \gamma_{s-6}(n-j) + a\}, \end{aligned} \quad (47)$$

where $\{\beta_3(j) > \gamma_{k-2}(n-j) \ \text{ and } \ \beta_5(j) > \gamma_{s-4}(n-j)\}$ defines the event that BMA($2, s-k$) is in error, $\{\gamma_{s-k}(n-j) > \beta_1(j)\}$ defines the event that first $s-k$ most reliable positions are error free and $\{\beta_7(j) - a > \gamma_{s-6}(n-j) + a\}$ indicates that at each iteration, the first s positions always contain more than 7 errors when the bias amplitude is a. As a result, this error block can not be corrected by BB-EBMA($2, s-k, a$).

When region B contains errors, enhanced 0-matching in stage 2 always fails. Hence an error block is correctable only if BMA(2) of stage 1 succeeds, or BMA(2) with 0-matching of stage 2 succeeds. Define $E_{B,t}$ as the event that there are t errors in the region B, $1 \le t \le s-k$.

It follows that

$$\begin{aligned} & Pr\{E_{BB11} \ \text{ and } \ E_{BB2}\} \\ = & \sum_{t=1}^{s-k} Pr\{E_{BB1} \ \text{ and } \ E_{B,t} \ \text{ and } \ E_{BB2}\} \\ > & \ Pr\{E_{BB1} \ \text{ and } \ E_{B,1} \ \text{ and } \ E_{BB2}\} \\ = & \ Pr\{ \ \beta_3(j) > \gamma_{k-2}(n-j) \ \text{ and } \ \beta_5(j) > \gamma_{s-4}(n-j) \\ & \quad \text{and } \ \beta_1(j) > \gamma_{s-k}(n-j) > \beta_2(j) \\ & \quad \text{and } \ \beta_4(j) - a > \gamma_{s-3}(n-j) + a \ \} \end{aligned} \quad (48)$$

In (48), we use the fact that $E_{B,1}$ is the dominant event when region B contains errors, since B is the most reliable region with small width $s-k$.

$\{\beta_1(j) > \gamma_{s-k}(n-j) > \beta_2(j)\}$ is the dominant event that the region B contains errors; $\{\beta_4(j) - a > \gamma_{s-3}(n-j) + a\}$ indicates that BMA(2) with 0-matching fails at each iteration.

Fig.7 depicts $P_{list,BB-EBMA(2,s-k,a)}$ for the binary image of RS(255,239), with $a = 0.05, 0.1, 0.15$, respectively, and $s-k = 22$. We observe that the larger the bias amplitude is, the smaller the list error probability is. However, the convergence of the iterative approach with a large bias is slower than that with a smaller bias [18].

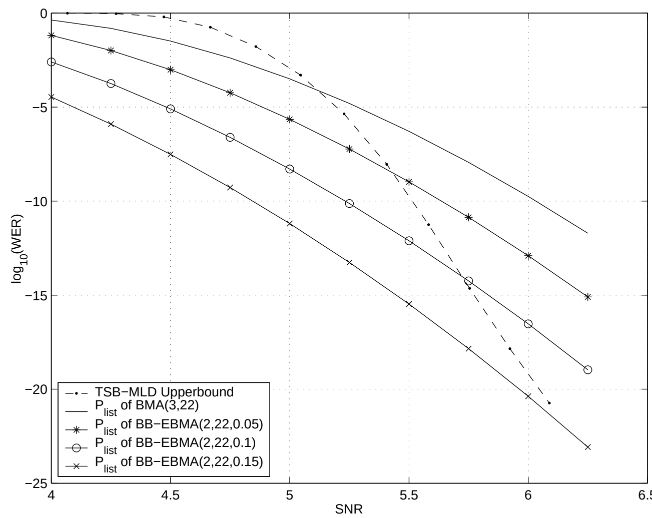

Fig. 7. List error probability of BB-EBMA($2, s-k, a$) for RS(255,239).

In Fig.8, we plot the simulation results of BB-EBMA($2, s-k, 0.1$) with 100 iterations, for the binary image of RS(255,239), with $k - J_1 = 200$, $k - J_2 = 100$. For comparison, we plot the simulation result of the BIAS(it_b)-IISR(it_i)-BMA($2, s-k, p, a$), with biasing iteration number $it_b = 20$, iterative information set reduction (IISR)[17] iteration number $it_i = 3$, control band length $s - k = 22$, IISR shift width $p = 10$, and biasing amplitude $a = 0.1$ [18]. This algorithm achieves the nearest MLD performance reported so far. We also plot the simulation results of the ADP(it_{outer}, it_{inner}) combined with a hard decision decoder (HDD), with the maximum outer iteration number $it_{outer} = 80$, inner iteration number $it_{inner} = 50$ and the damping coefficient $\alpha = 0.08$ [19]. This algorithm is the most efficient reported soft decoding different from direct MRB approaches.

We observe that after 10 iterations, BB-EBMA($2, s-k, 0.1$) outperforms BMA(3), BIAS(20)-IISR(3)-BMA(2,22,10,0.1) and ADP(80, 50). After 100 iterations, we observe many MLD errors as recorded in Table I. To efficiently reduce the average computation complexity, we used the PSC of Section III with the threshold $T = 20$. The average computation complexity of BIAS(20)-IISR(3)-BMA(2,22,10,0.1) and BB-EBMA($2, s-k, 0.1$) are recorded in Table II, where the same PSC threshold is used for both algorithms. Since MRB reprocessing algorithms are list decoding algorithms, the complexities are defined as the maximum and average numbers of candidates (or list sizes) per received word $\mathbf{r}$ processed by the algorithm at a given SNR value. We observe from Fig.8 and Table II that the average computation complexity of BIAS(20)-IISR(3)-BMA(2,22,10,0.1) is close to that of BB-EBMA($2, s-k, 0.1$) with 50 iterations. However the performance of BB-EBMA($2, s-k, 0.1$) with 10 iterations is already better than that of BIAS(20)-IISR(3)-BMA(2,22,10,0.1). Note that both algorithms use the same size of memory. Furthermore, BIAS(20)-IISR(3)-BMA(2,22,10,0.1) needs 80 Gaussian eliminations, while BB-EBMA($2, s-k, 0.1$) with it iterations needs it Gaussian eliminations.

We also conducted simulations with enhanced BMA for the decoding of binary image of (460,420) Reed Solomon code defined on the field $GF(2^{10})$. The concept of enhanced BMA with order 1 is depicted in Figure 9 in a similar manner as that in Figure 3. In Figure 10, we plot the simulation results of $BM-EBMA(1,22,0.15)$ with 10 iterations and $BB-EBMA(1,22,0.15)$ with 15 iterations, with $k-J_1=1700$, $k-J_2=200$ and the threshold $T=100$. We observe that after 10 iterations, the performance of $BM-EBMA(1,22,0.15)$ can approach that of $BMA(2,22)$, and the performance of $BB-EBMA(1,22,0.15)$ with 15 iterations has already been better than that of $BMA(2,22)$. Note that the memory used by the an enhanced $BMA(1,s-k)$ algorithm is the same as that of $BMA(1,s-k)$, which is much smaller than that used by $BMA(2,s-k)$.

References

[1] M. Fossorier and S. Lin, "Soft Decision Decoding of Linear Block Codes Based on Ordered Statistics," *IEEE Trans. Inform. Theory.*, vol. 41, pp. 1379-1396, Sept. 1995.

[2] D. Gazelle and J. Snyders, "Reliability-Based Code Search Algorithms for Maximum Likelihood Decoding of Block Codes", *IEEE Trans. Inform. Theory*, vol. 43, pp. 239-249, Jan. 1997.

[3] A. Valembois and M. Fossorier, "Box and Match Techniques Applied to Soft Decision Decoding," *IEEE Trans. Inform. Theory.*, vol. 50, pp. 796-810, May 2004.

[4] D. Taipale and M. Pursley, "A Improvement to Generalized-minimum-distance Decoding", *IEEE Trans. Inform. Theory.*, vol. 37, pp. 167-172, Jan. 1991.

[5] A. Valembois and M. Fossorier, "A Comparison Between Most Reliable Basis Reprocessing Strategies", *IEICE Trans. on Fundam.*, vol. E85-A, pp. 1727-1741, July 2002.

[6] T. Kaneko, T. Nishijima, H. Inazumi, and S. Hirasawa, "An Efficient Maximum Likelihood Decoding of Linear Block Codes with Algebraic Decoder", *IEEE Trans. Inform. Theory.*, vol. 40, pp. 320-327, March. 1994.

[7] M. Fossorier, T. Koumoto, T. Takata, T. Kasami, and S. Lin, "The Least Stringent Sufficient Condition on the Optimality of a Suboptimally Decoded Codeword Using the Most Reliable Basis", *The Proc IEEE ISIT*, pp. 430, Ulm, Germany, June 1997

[8] T Kasami, "On Integer Programming Problems Related to Soft Decision Decoding Algorithms", *Applied Algebra, Algebic Algorithms and Error-Correcting Codes.*, vol. 1719, pp. 43-54, Springer-Verlag LNCS, Nov. 1999.

[9] T Kasami, Y. Tang, T. Koumoto, and T. Fujiwara, "Sufficient Conditions for Ruling Out Useless Iterative Steps In a Class of Iterative Decoding Algorithms", *IEICE Trans. on Fundam*, vol. E82-A, pp. 2061-2073, Oct 1999.

[10] Y. Wu, R. Koetter and C. Hadjicostis, "Soft-Decision Decoding of Linear Block Codes Using Preprocessing and Diversification", Submitted to *IEEE Trans. Inform. Theory*, 2005

[11] M. Fossorier, "Average and Maximum Computational Complexities for Information Set Decoding of Block Codes," *The Proc Third Asia-Europe Workshop on Coding and Information Theory*, Kamogawa, Japan, June 2003.

[12] M. Fossorier, S. Lin and J. Snyders, "Reliability-Based Syndrome Decoding of Linear Block Codes," *IEEE Trans. on Info. Theory*, vol. IT-44, pp. 388- 398, Jan. 1998.

[13] A. Ahmed, R. Koetter, and N. R. Shanbhag, "Performance Analysis of the Adaptive Parity Check Matrix Based Soft-Decision Decoding Algorithm," *The Proc Asilomar Conf.*, Pacific Grove, USA, Nov. 2004

[14] D. Agrawal and A. Vardy. "Generalized Minimum Distance Decoding in Euclidean-Space:Performance Analysis", *IEEE Trans. on Inform. Theory*, vol. 46, pp. 60-83, Jan.2000

[15] M. Fossorier and S. Lin, "Error Performance Analysis for Reliability-Based Decoding Algorithms", *IEEE Trans. on Inform. Theory*, vol. 48, pp. 287-293, Jan.2002

[16] A. Papoulis "Probability, Random Variables, and Stochastic Processes", 3rd ed New York:McGraw-Hill,1991

[17] M. Fossorier, "Reliability-Based Soft-Decision Decoding with Iterative Information Set Reduction", *IEEE Trans. on Info. Theory*, vol. 48, pp. 3101-3106, Dec.2002.

[18] W. Jin and M. Fossorier, "Reliability-Based Soft-Decision Decoding with Multiple Biases," *IEEE Trans. Inform. Theory.*, to appear.

[19] J. Jiang and K. R. Narayanan, "Iterative Soft Input Soft Output Decoding of Reed-Solomon Codes by Adapting the Parity Check Matrix," *IEEE Trans. Inform. Theory.*, pp. 3746- 3756, Aug.2006.

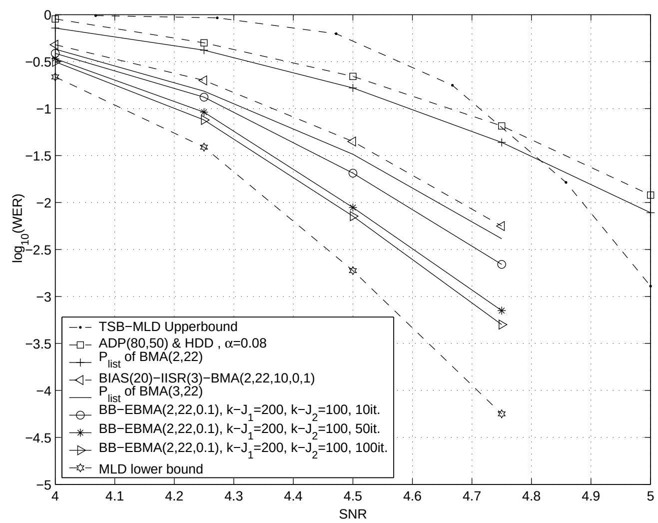

Fig. 8. BB-EBMA$(2,s-k,0.1)$ for the decoding of RS(255,239).

TABLE I

PERCENTAGE OF MLD ERRORS

SNR	4.0	4.25	4.5	4.75
Percentage of errors which are MLD errors	70%	50%	25%	10%

TABLE II

AVERAGE COMPUTATION COMPLEXITY

SNR	Average computation complexity			
	4.0	4.25	4.5	4.75
BIAS(20)-IISR(3)-BMA(2, 22, 10, 0. 1)	138, 867, 900	99, 101, 977	37, 745, 250	15, 267, 122
BB-EBMA(2, 22, 0. 1) Iteration = 10	36, 602, 064	26, 520, 225	9, 637, 408	4, 620, 123
Iteration = 50	158, 687, 831	112, 816, 329	38, 325, 993	19, 189, 098
Iteration = 100	310, 253, 129	217, 908, 499	73, 651, 295	37, 269, 003

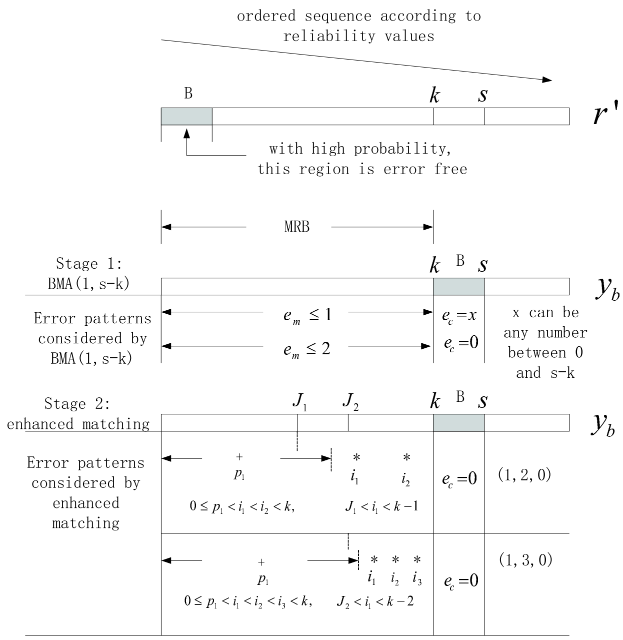

Fig. 9. Concept of $BMA(1, s-k)$ with enhanced matching.

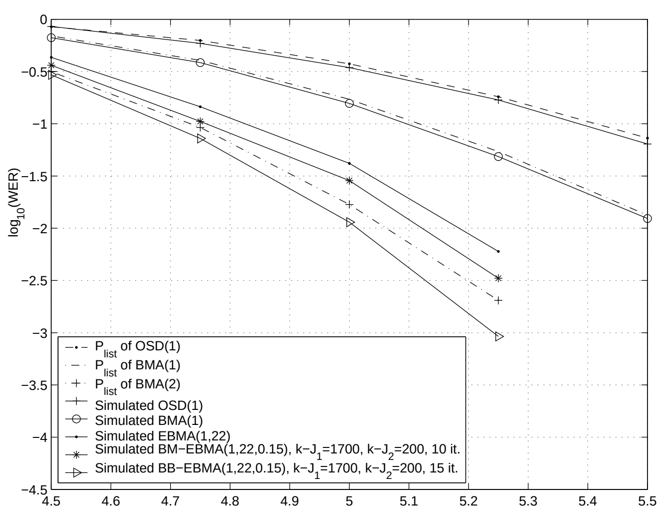

Fig. 10. Simulation results for the decoding of $(460, 420)$ Reed Solomon code over $GF(2^{10})$ with enhanced $BMA(1)$.