社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE
IT2003-6(2003-5)

信頼度情報に基づく置換生成行列を用いた最尤復号法の効率化
2元系列の順序関係を利用した計算量低減手法

八木 秀樹† 小林 学†† 平澤 茂一†

† 早稲田大学理工学部 〒169-8555 東京都新宿区大久保 3-4-1
†† 湘南工科大学工学部 〒251-8511 神奈川県藤沢市辻堂西海岸 1-1-25
E-mail: †yagi@hirasa.mgmt.waseda.ac.jp

あらまし 近年，受信系列から得られる信頼度が高い順に列置換された生成行列を用いて，候補符号語を繰り返し生成する最尤復号法や準最適な軟判定復号法が広く研究されている．本稿では，列置換された生成行列を用いた復号法において，不必要な候補符号語及び尤度計算を省略できる十分条件を導出し，復号計算量を低減する手法を提案する．提案する十分条件は2元系列間の順序関係を利用したもので，実数演算 (実数の加算と等価な演算) を必要としない．結果的に，最尤性を保証しつつ復号に支配的となる実数演算量を低減できること計算機シミュレーションにより示す．

キーワード 最尤復号法，軟判定復号，信頼度情報，置換生成行列，線形ブロック符号

An Improved Method of Maximum Likelihood Decoding Algorithms using the Most Reliable Basis based on a Order Relation among Binary Vectors

Hideki YAGI†, Manabu KOBAYASHI††, and Shigeichi HIRASAWA†

† Department of Industrial and Management Systems Engineering,
School of Science and Engineering, Waseda University, Tokyo, 169-8555 Japan.
†† Department of Information Science,
School of Engineering, Shonan Institute of Technology, Kanagawa, 251-8511 Japan.
E-mail: †yagi@hirasa.mgmt.waseda.ac.jp

Abstract Several reliability based maximum likelihood decoding (MLD) algorithms of linear block codes have been widely studied. These algorithms efficiently search the most likely codeword, using the most reliable basis of generator matrix whose leftmost k (the dimension of code) columns are the most reliable and linearly independent. In this paper, several sufficient conditions for eliminating unnecessary candidate codewords or their metrics computations are derived for MLD algorithms using the most reliable basis by utilizing a order relation among binary vectors. Under the certain assumption of generation order of candidate codewords, we devise an adaptive implementation of the derived conditions. Consequently, the MLD algorithm employing the derived conditions reduces the number of generated candidate codeword and of real number operations, compared to a conventional MLD using the MRB without the degradation in error performance.

Key words maximum likelihood decoding, soft decision decoding, most reliable basis, reliability, linear block codes

1. Introduction

Soft decision decoding for linear block codes reduces the block error probability by taking advantage of channel measurement information, compared with conventional hard decision decoding. Particularly, maximum likelihood decoding (MLD) achieves the best error performance when each codeword are transmitted with equal probability. Since the complexity of MLD becomes too complex to implement as the code length becomes larger, many researchers have been devoted to develop efficient MLD algorithms. Among such decoding algorithms, the *most reliable basis* (MRB) based MLD algorithms which iteratively generate candidate codewords by using the generator matrix of the code [1] ~ [4] (Sub-optimum versions of the MRB based soft decision decoding algorithms are found in [5] ~ [7]). The MRB based MLD algorithms averagely reduce the time complexity as well as the space one. Furthermore, these algorithms are applicable for any linear block codes [7].

In the MRB based decoding algorithms, test error patterns are iteratively generated to construct candidate codewords. In these algorithms, implicitly or explicitly, a sufficient condition for optimality is applied to generated candidate codeword. A sufficient condition for eliminating unnecessary test error patterns is also applied before they are encoded with the MRB (of generator matrix). As a result, the MRB based MLD algorithms require relatively small number of candidate codewords and their metrics computations. At low to moderate SNR and for moderate code rates and large code lengths, however, the time complexity of them for performing MLD is still very large.

In this paper, first, we define an order relation among binary vectors. Then we derive a sufficient condition for eliminating unnecessary test error patterns which cannot give the ML codeword by using the order relation. Although the basic concept of this approach has been presented in [8], we show that the derived condition is generalized and more stringent. With the similar principle, a sufficient condition for omitting unnecessary metrics computations of candidate codewords is derived. In accordance with the candidate codewords updated, an *adaptive implementation* of the algorithm, where the codeword referenced by the derived condition is adaptively altered, is considered to make the derived conditions more effective. Since the implementation of

the derived sufficient conditions require no real values, the number of real number addition-equivalent operations (we will call real number addition-equivalent operations real operations) is reduced in the MLD algorithm employing the proposed conditions. Finally, we show the effectiveness of the proposed methods without the degradation of the error performance.

This paper is organized as follows. In Sect. 2., the general principle of the MRB based MLD algorithm is reviewed as a preliminary. In Sect. 3., a generation condition of test error patterns in this paper is described. In Sect. 4., sufficient conditions for eliminating the unnecessary test error patterns or the unnecessary metrics computations are derived. Finally, some simulation results are shown in Sect. 5. and concluding remarks are stated in Sect. 6.

2. The MRB based MLD Algorithm

Let $\mathcal{C}$ be a binary linear (n, k, d) block code with length n, dimension k and minimum distance d. Let G be the generator matrix of $\mathcal{C}$. Assume that each codeword $\boldsymbol{c} = (c_1, c_2, \ldots, c_n) \in \mathcal{C}$ has equal probability to be transmitted through the Additive White Gaussian Noise (AWGN) channel with the signal to noise ratio (SNR) E_b/N_0 [dB]. The detector projects the received sequence $\boldsymbol{r} = (r_1, r_2, \ldots, r_n) \in \mathcal{R}^n$ into the reliability sequence $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$ such that $\theta_j = \ln \frac{P(r_j|c_j=0)}{P(r_j|c_j=1)}$, and inputs $\boldsymbol{\theta}$ into the decoder. Let $\mathcal{V}^n$ denote a set of all binary n dimensional vectors. The decoder estimates a transmitted codeword from both $\boldsymbol{\theta}$ and the hard decision received sequence $\boldsymbol{z} = (z_1, z_2, \ldots, z_n) \in \mathcal{V}^n$ of $\boldsymbol{\theta}$ such that

$$z_j = \begin{cases} 0, & \text{if } \theta_j \geqq 0; \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

An error probability of the symbol z_j, $P(z_j \neq c_j|r_j)$, is smaller as the value $|\theta_j|$, $j \in [1, n]$ becomes larger where $[j_\alpha, j_\beta]$ denotes a set of integers from j_α to j_β. Therefore, we call $|\theta_j|$ *reliability measure*.

For any $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathcal{V}^n$, let $L(\boldsymbol{x})$ be the function of the *reliability loss* with respect to a given (fixed) $\boldsymbol{z}$, defined as

$$L(\boldsymbol{x}) = \sum_{j=1}^{n} (x_j \oplus z_j)|\theta_j|, \tag{2}$$

where $\oplus$ represents the exclusive OR operation. The function $L(\boldsymbol{x})$ is also known as *discrepancy* [7], [9]. For a subset $\mathcal{X}$ of $\mathcal{V}^n$, let $\underline{L}[\mathcal{X}]$ be defined as

$$\underline{L}[\mathcal{X}] = \min_{\boldsymbol{x} \in \mathcal{X}} L(\boldsymbol{x}). \tag{3}$$

Then $L(\boldsymbol{c}_{\mathrm{ML}}) = \underline{L}[\mathcal{C}]$ if and only if $\boldsymbol{c}_{\mathrm{ML}} \in \mathcal{C}$ is the ML codeword [4], [7], [9].

In the MRB based decoder, positions of $\boldsymbol{\theta}$ are reordered in the nonincreasing order of reliability measure. We denote the resultant sequence $\bar{\boldsymbol{\theta}} = \lambda(\boldsymbol{\theta})$ where λ is the permutation function from $\boldsymbol{\theta}$ to $\bar{\boldsymbol{\theta}}$. i.e., $|\bar{\theta}_j| \geqq |\bar{\theta}_{j+1}|$, $1 \leqq j < n$. Let $\bar{G}$ be the permuted generator matrix according to the reordering of $\bar{\boldsymbol{\theta}}$ and $\bar{\mathcal{C}}$ be the code generated by $\bar{G}$. Let $\bar{\mathcal{V}}^n$ denote the set such that $\bar{\mathcal{V}}^n = \{\bar{\boldsymbol{x}} = \lambda(\boldsymbol{x}) | \ \boldsymbol{x} \in \mathcal{V}^n, \}$.

Furthermore, columns of $\bar{G}$ are permuted such that the *most reliable and linearly independent* (MRI). The MRI columns are linearly independent k columns of generator matrix whose reliabilities are the largest among any other linearly independent k columns. For the resultant matrix, the leftmost $k \times k$ matrix is rearranged to be the identity matrix by the standard row operations. This identity matrix forms MRB. The resultant generator matrix is denoted by $\tilde{G}$. Let ϕ be this permutation function of columns from $\bar{G}$ to $\tilde{G}$. Therefore, $\tilde{\boldsymbol{\theta}} = \phi(\bar{\boldsymbol{\theta}})$. Note that $|\tilde{\theta}_j| \geqq |\tilde{\theta}_{j+1}|$, $1 \leqq j < k$, and $|\tilde{\theta}_{j'}| \geqq |\tilde{\theta}_{j'+1}|$, $k+1 \leqq j' < n$. Let $\tilde{\mathcal{C}}$ denote the code generated by $\tilde{G}$, which is equivalent to $\mathcal{C}$. Let $\tilde{\mathcal{V}}^n$ denote the set such that $\tilde{\mathcal{V}}^n = \{\tilde{\boldsymbol{x}} = \phi(\lambda(\boldsymbol{x})) | \ \boldsymbol{x} \in \mathcal{V}^n\}$.

Define that $\boldsymbol{b} = (b_1, b_2, \ldots, b_k)$ consists of k MRI symbols of $\tilde{\boldsymbol{z}} = (\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_n)$. i.e., $b_j = \tilde{z}_j$, $j \in [1, k]$. $\boldsymbol{b}$ is regarded as an information sequence and the decoder constructs the initial codeword $\tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}} = \boldsymbol{b}\tilde{G}$. Remark that $\tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}}$ is the ML codeword if $\tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}} = \tilde{\boldsymbol{z}}$ [4], [9]. If $L(\tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}}) > 0$, the decoder iteratively constructs candidate codewords by $\tilde{G}$ and searches the ML codeword.

For a binary vector $\boldsymbol{x}$, let $w_H(\boldsymbol{x})$ denote the Hamming weight of $\boldsymbol{x}$ and $supp(\boldsymbol{x}) = \{j | \ x_j = 1\}$ be its support. For a set $\mathcal{X}$, let $|\mathcal{X}|$ be the cardinality of $\mathcal{X}$.

[Definition 1] For a set of positive integers $\mathcal{J} = \{j_1, j_2, \ldots, j_l\}$ such that $|\mathcal{J}| = l$ and $1 \leqq j_1 < j_2 < \cdots < j_l \leqq k$, a *test error pattern* $\boldsymbol{t}_{\mathcal{J}} = (t_{\mathcal{J},1}, t_{\mathcal{J},2}, \ldots, t_{\mathcal{J},k}) \in \{0,1\}^k$ is defined such that $supp(\boldsymbol{t}_{\mathcal{J}}) = \mathcal{J}$. A codeword $\tilde{\boldsymbol{w}}_{\mathcal{J}} = (\tilde{w}_{\mathcal{J},1}, \tilde{w}_{\mathcal{J},2}, \ldots, \tilde{w}_{\mathcal{J},n}) = \boldsymbol{t}_{\mathcal{J}}\tilde{G}$ is called a *test error codeword* which gives a candidate codeword $\tilde{\boldsymbol{c}}_{\mathcal{J}} = \tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}} \oplus \tilde{\boldsymbol{w}}_{\mathcal{J}}$. A candidate codeword $\tilde{\boldsymbol{c}}_{\mathcal{J}}$ and a test error codeword $\tilde{\boldsymbol{w}}_{\mathcal{J}}$ are said to be *better* than $\tilde{\boldsymbol{c}}_{\mathcal{J}'}$ and $\tilde{\boldsymbol{w}}_{\mathcal{J}'}$, respectively, if $L(\tilde{\boldsymbol{c}}_{\mathcal{J}}) < L(\tilde{\boldsymbol{c}}_{\mathcal{J}'})$. For a subset $\tilde{\mathcal{C}}'$ of $\tilde{\mathcal{C}}$, a candidate codeword $\tilde{\boldsymbol{c}}_{\mathcal{J}}$ and a test error codeword $\tilde{\boldsymbol{w}}_{\mathcal{J}}$ are said to be the *best* in $\tilde{\mathcal{C}}'$ if $L(\tilde{\boldsymbol{c}}_{\mathcal{J}}) = \underline{L}[\tilde{\mathcal{C}}']$. □

By Def.1, $\boldsymbol{t}_{\boldsymbol{\emptyset}}$ is k dimensional all zero vector 0^k which is regarded as a test error pattern corresponding to the initial codeword $\tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}}$ since $\tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}} = \tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}} \oplus \boldsymbol{t}_{\boldsymbol{\emptyset}}\tilde{G} = \tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}} \oplus 0^n$. It is obvious that there is one to one correspondence between $\boldsymbol{t}_{\mathcal{J}}$ and $\tilde{\boldsymbol{c}}_{\mathcal{J}}$. Therefore the search order of candidate codewords is determined by the generation order of test error patterns. For simplicity of description, the location sets $\mathcal{J}_i$ are indexed with i in which we assume $\boldsymbol{t}_{\mathcal{J}_i}$ is generated in the increasing order of i such that $1 \leqq i < 2^k$.

Let $\tilde{\mathcal{C}}_{\mathcal{J}_s}$ be a set of codewords which includes all candidate codewords $\tilde{\boldsymbol{c}}_{\mathcal{J}_i} = \tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}} \oplus \tilde{\boldsymbol{w}}_{\mathcal{J}_i}$ such that $0 \leqq i < s$ before generating $\boldsymbol{t}_{\mathcal{J}_s}$, i.e.,

$$\tilde{\mathcal{C}}_{\mathcal{J}_s} = \{\tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}}\} \cup \left\{ \tilde{\boldsymbol{c}}_{\mathcal{J}_i} = \tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}} \oplus \tilde{\boldsymbol{w}}_{\mathcal{J}_i} \middle| \tilde{\boldsymbol{w}}_{\mathcal{J}_i} = \boldsymbol{t}_{\mathcal{J}_i}\tilde{G}, 1 \leqq i < s \right\}. \tag{4}$$

For a test error pattern $\boldsymbol{t}_{\mathcal{J}}$, let $\Delta(\boldsymbol{t}_{\mathcal{J}})$ be defined as

$$\Delta(\boldsymbol{t}_{\mathcal{J}}) = \sum_{j=1}^{k} t_{\mathcal{J},j}|\tilde{\theta}_j|, \tag{5}$$

which expresses the reliability loss with respect to $\boldsymbol{b}$.

At a decoding stage of generating $\boldsymbol{t}_{\mathcal{J}}$, we need not to encode $\boldsymbol{t}_{\mathcal{J}}$ if

$$\underline{L}[\tilde{\mathcal{C}}_{\mathcal{J}}] \leqq \Delta(\boldsymbol{t}_{\mathcal{J}_i}), \tag{6}$$

since a candidate codeword $\tilde{\boldsymbol{c}}_{\mathcal{J}} = \tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}} \oplus \tilde{\boldsymbol{w}}_{\mathcal{J}}$ such that $\tilde{\boldsymbol{w}}_{\mathcal{J}} = \boldsymbol{t}_{\mathcal{J}}\tilde{G}$ always satisfies $\Delta(\boldsymbol{t}_{\mathcal{J}}) \leqq L(\tilde{\boldsymbol{c}}_{\mathcal{J}})$. i.e., if eq.(6) holds, $\boldsymbol{t}_{\mathcal{J}}$ can never give the best candidate codeword.

Let $\tilde{\boldsymbol{e}}_{\boldsymbol{\emptyset}} = (\tilde{e}_{\boldsymbol{\emptyset},1}, \tilde{e}_{\boldsymbol{\emptyset},2}, \ldots, \tilde{e}_{\boldsymbol{\emptyset},n}) = \tilde{\boldsymbol{z}} \oplus \tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}}$. For $\boldsymbol{x} \in \tilde{\mathcal{V}}^n$, let $\Lambda(\boldsymbol{x})$ be defined as

$$\Lambda(\boldsymbol{x}) = \sum_{j \in supp(\boldsymbol{x})} (1 - 2\tilde{e}_{\boldsymbol{\emptyset},j})|\tilde{\theta}_j|. \tag{7}$$

Then, for $\tilde{\boldsymbol{c}}_{\mathcal{J}} = \tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}} \oplus \tilde{\boldsymbol{w}}_{\mathcal{J}}$, we can compute $L(\tilde{\boldsymbol{c}}_{\mathcal{J}})$ such that

$$L(\tilde{\boldsymbol{c}}_{\mathcal{J}}) = L(\tilde{\boldsymbol{c}}_{\boldsymbol{\emptyset}}) + \Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}}), \tag{8}$$

since from $\tilde{\boldsymbol{z}} \oplus \tilde{\boldsymbol{c}}_{\mathcal{J}} = \tilde{\boldsymbol{e}}_{\boldsymbol{\emptyset}} \oplus \tilde{\boldsymbol{w}}_{\mathcal{J}}$,

$$\begin{aligned} L(\tilde{\boldsymbol{c}}_{\mathcal{J}}) &= \sum_{j=1}^{n} (\tilde{e}_{\boldsymbol{\emptyset},j} \oplus \tilde{w}_{\mathcal{J},j})|\tilde{\theta}_j|, \\ &= \sum_{j=1}^{n} \tilde{e}_{\boldsymbol{\emptyset},j}|\tilde{\theta}_j| + \sum_{j=1}^{n} (1 - 2\tilde{e}_{\boldsymbol{\emptyset},j})\tilde{w}_{\mathcal{J},j}|\tilde{\theta}_j|. \end{aligned} \tag{9}$$

For a subset $\tilde{\mathcal{X}}$ of $\tilde{\mathcal{V}}^n$ and $\tilde{\boldsymbol{c}}_{\boldsymbol{\theta}}$, let $\underline{\Lambda}[\tilde{\mathcal{X}}]$ be defined as

$$\underline{\Lambda}[\tilde{\mathcal{X}}] = \min_{\boldsymbol{x} \oplus \tilde{\boldsymbol{c}}_{\boldsymbol{\theta}} \in \tilde{\mathcal{X}}} \Lambda(\boldsymbol{x}). \tag{10}$$

At a decoding stage of generating $\boldsymbol{t}_{\mathcal{J}_s}$, we only need to search a test error codeword $\tilde{\boldsymbol{w}}_{\mathcal{J}_i} \in \tilde{\mathcal{C}} \backslash \tilde{\mathcal{C}}_{\mathcal{J}_s}$ such that $\Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}_i}) < \underline{\Lambda}[\tilde{\mathcal{C}}_{\mathcal{J}_s}]$. If there exists no test error pattern to be generated, the decoder outputs $\tilde{\boldsymbol{c}}_{\mathcal{J}_i} = \tilde{\boldsymbol{c}}_{\boldsymbol{\theta}} \oplus \tilde{\boldsymbol{w}}_{\mathcal{J}_i}$ as the estimated codeword. For $\boldsymbol{t}_{\mathcal{J}}$, let $f(\boldsymbol{t}_{\mathcal{J}})$ express any heuristic function such that $\Delta(\boldsymbol{t}_{\mathcal{J}}) \leqq f(\boldsymbol{t}_{\mathcal{J}}) \leqq L(\tilde{\boldsymbol{c}}_{\mathcal{J}})$. We describe the general version of the MRB based MLD algorithm below.

[The MRB based MLD Algorithm]

1) Generate $\tilde{\boldsymbol{c}}_{\boldsymbol{\theta}} := b\tilde{G}$, and set $\underline{L} := L(\tilde{\boldsymbol{c}}_{\boldsymbol{\theta}})$, $\tilde{\boldsymbol{w}}_{\mathrm{best}} := \boldsymbol{0}$, $\underline{\Lambda} := 0$ and $i := 1$.
2) Generate $\boldsymbol{t}_{\mathcal{J}_i}$ and calculate $f(\boldsymbol{t}_{\mathcal{J}_i})$. If $\underline{L} \leqq f(\boldsymbol{t}_{\mathcal{J}_i})$, then go to 4).
3) Generate $\tilde{\boldsymbol{w}}_{\mathcal{J}_i} := \boldsymbol{t}_{\mathcal{J}_i}\tilde{G}$ and calculate $\Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}_i})$. If $\Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}_i}) < \underline{\Lambda}$, then $\underline{\Lambda} := \Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}_i})$, $\underline{L} := L(\tilde{\boldsymbol{c}}_{\boldsymbol{\theta}}) + \underline{\Lambda}$ and $\tilde{\boldsymbol{w}}_{\mathrm{best}} := \tilde{\boldsymbol{w}}_{\mathcal{J}_i}$.
4) Set $i := i + 1$. If the certain terminating criterion holds or $i = 2^k$, then output $\tilde{\boldsymbol{c}}_{\mathrm{ML}} := \tilde{\boldsymbol{c}}_{\boldsymbol{\theta}} \oplus \tilde{\boldsymbol{w}}_{\mathrm{best}}$ and stop, otherwise go to 2). □

About a terminating criterion of the decoding algorithm in step 4), several criteria have been proposed in [2], [4], [5], [8].

We here state the complexity of the MRB based decoding algorithm. The time complexity of permuting $\boldsymbol{\theta}$ in the nonincreasing order is $O(n \log n)$ and the construction of $\tilde{G}$ requires $O(n \times \kappa^2)$ where $\kappa = \min\{k, n-k\}$ [1], [2], [5]. These procedures are carried out only once in a decoding algorithm. Contrary to the above procedures, generating $\boldsymbol{t}_{\mathcal{J}}$ and constructing $\tilde{\boldsymbol{w}}_{\mathcal{J}} = \boldsymbol{t}_{\mathcal{J}}\tilde{G}$ are carried out iteratively, where each encoding requires binary operations of $O(kn)$ with conventional encoding method [4], [5]. For each test error codeword constructed, calculating eq.(7) requires real operations of $O(n)$. Therefore, both encoding test error patterns and the real operations dominate mainly the whole decoding complexity.

3. Generation of Test Error Patterns

In the MRB based decoding algorithm, the time complexity for finding the ML codeword strongly depends on the search order of candidate codewords. In this section, we consider generation order of test error patterns.

[Definition 2] **(The Order Relation for Sets)** For two distinct sets $\mathcal{X} = \{j_1, j_2, \ldots, j_m\}$ and $\mathcal{X}' = \{j'_1, j'_2, \ldots, j'_{m'}\}$, we write "$\mathcal{X}' <_S \mathcal{X}$" if $m' \leqq m$ and there exists a subset $\{i_1, i_2, \ldots, i_{m'}\} \subseteqq \mathcal{X}$ such that $i_1 < i_2 < \cdots < i_{m'}$ and $i_h \leqq j'_h$, $1 \leqq {}^\forall h \leqq m'$. □

[Definition 3] **(The Order Relation for Binary Vectors)** For two distinct sets $X = \{j_1, j_2, \ldots, j_m\}$ and $\mathcal{X}' = \{j'_1, j'_2, \ldots, j'_{m'}\}$, let $\boldsymbol{v}_X$ and $\boldsymbol{v}_{\mathcal{X}'}$ be binary vectors such that $supp(\boldsymbol{v}_{\mathcal{X}}) = \mathcal{X}$ and $supp(\boldsymbol{v}_{\mathcal{X}'}) = \mathcal{X}'$, respectively. We write "$\boldsymbol{v}_{\mathcal{X}'} <_V \boldsymbol{v}_{\mathcal{X}}$" if and only if there is the order relation $\mathcal{X}' <_S \mathcal{X}$. □

In this paper, we assume that test error patterns are generated in accordance with the following condition.

[Generation Condition] During the decoding procedure, if $\boldsymbol{t}_{\mathcal{J}}, \mathcal{J} \subseteqq [1, k]$, is generated, a test error pattern $\boldsymbol{t}_{\mathcal{J}'}$ such that $\boldsymbol{t}_{\mathcal{J}'} <_V \boldsymbol{t}_{\mathcal{J}}$ has been already generated or eliminated from consideration at a preceding stage. □

This condition is similar to EG condition [9], in soft decision decoding algorithm using algebraic decoder.

For example, if test error patterns are generated in binary order, then the above condition is satisfied. The increasing order of $\Delta(\boldsymbol{t}_{\mathcal{J}})$ [3], [7] or the increasing order of Hamming weight $w_H(\boldsymbol{t}_{\mathcal{J}})$ [1], [5], [8] also satisfy Generation Condition(1). On the other hand, in the A* decoding algorithm [2], test error patterns are generated in the increasing value of a heuristic function where this order does not satisfy Generation Condition.

For the special case of Generation Condition, the generating order proposed by D. Gazelle and J. Snyders [1] is briefly described. Hereafter, we call the MRB based decoding algorithm employing the generation order in [1] the GS decoding algorithm.

For $l \in [1, \kappa]$, let $\mathcal{T}_l = \{\boldsymbol{t} \in \{0,1\}^k | w_H(\boldsymbol{t}) = l\}$ be the set of test error patterns. The GS decoder processes $\mathcal{T}_l$ in increasing order of l. The processing of $\mathcal{T}_l$ is referred to as *phase-l reprocessing* [5], [8]. In phase-l reprocessing, $l \in [1, \kappa]$, a test error pattern $\boldsymbol{t}_{\mathcal{J}_i}$ is generated in binary order.

In phase-l reprocessing, a test error pattern $\boldsymbol{t}_{\mathcal{J}_{i+1}}$, $i \geqq 1$, is generated from $\boldsymbol{t}_{\mathcal{J}_i}$ in the following manner. Let $\mathcal{J}_i = \{j_1, j_2, \ldots, j_l\}$ and $\mathcal{J}_{i+1} = \{j'_1, j'_2, \ldots, j'_l\}$. The first element $\boldsymbol{t}_{\mathcal{J}_1}$ of $\mathcal{T}_l$ satisfies $j_h = k - l + h$, $1 \leqq {}^\forall h \leqq l$. For $1 \leqq i < \binom{k}{l}$, we find the position $I_{\mathrm{A}} = \max\{h |\ j_h - 1 \notin \mathcal{J}_i \text{ and } j_h \in \mathcal{J}_i\}$. The next test error pattern $\boldsymbol{t}_{\mathcal{J}_{i+1}}$ is obtained such that

$$j'_h = \begin{cases} j_h, & \text{for } 1 \leqq h < I_{\mathrm{A}}; \\ j_h - 1, & \text{for } h = I_{\mathrm{A}}; \\ k - l + h, & \text{for } I_{\mathrm{A}} < h \leqq l. \end{cases} \tag{11}$$

Hereafter we call this method, which generates $\boldsymbol{t}_{i+1}$ from $\boldsymbol{t}_i$, the *GS generation rule A*.

If a test error pattern $\boldsymbol{t}_{\mathcal{J}_i}$ satisfies eq.(6), then $\boldsymbol{t}_{\mathcal{J}}$ such that $\mathcal{J}_i <_S \mathcal{J}$ is also satisfy eq.(6) [7] and such $\boldsymbol{t}_{\mathcal{J}}$ need not to be encoded. Then the next test error pattern $\boldsymbol{t}_{\mathcal{J}_s} \in \mathcal{T}_l$ such that $\boldsymbol{t}_{\mathcal{J}_i} \not<_V \boldsymbol{t}_{\mathcal{J}_s}, i < s$, is generated in the following manner. Let $\mathcal{J}_i = \{j_1, j_2, \ldots, j_l\}$ and $\mathcal{J}_s = \{j'_1, j'_2, \ldots, j'_l\}$. First, we find the position such that $I_{\mathrm{B}} = \max\{h |\ j_h \in \mathcal{J}_i \text{ and } j_h + 1 \notin \mathcal{J}_i\}$. If $I_{\mathrm{B}} = 1$, then the rest of elements in $\mathcal{T}_l$ are eliminated from consideration and we enter the phase-$(l+1)$ reprocessing. Otherwise, for the temporary set $\mathcal{J}_{\mathrm{tmp}}$ such that $\mathcal{J}_{\mathrm{tmp}} = \{j_1, j_2, \ldots, j_{I_{\mathrm{B}}-1}\}$, we find the position such that $I_{\mathrm{tmp}} = \max\{h |\ j_h - 1 \notin \mathcal{J}_{\mathrm{tmp}} \text{ and } j_h \in \mathcal{J}_{\mathrm{tmp}}\}$. The next test error pattern $\boldsymbol{t}_{\mathcal{J}_s}$ is obtained such that

$$j'_h = \begin{cases} j_h, & \text{for } 1 \leqq h < I_{\mathrm{tmp}}; \\ j_h - 1, & \text{for } h = I_{\mathrm{tmp}}; \\ k - l + h, & \text{for } I_{\mathrm{tmp}} < h \leqq l. \end{cases} \tag{12}$$

Hereafter we call this method, which generates $\boldsymbol{t}_{\mathcal{J}_s}$ from $\boldsymbol{t}_{\mathcal{J}_i}$, the *GS generation rule B*.

[The GS Decoding Algorithm]

1) Generate $\tilde{\boldsymbol{c}}_{\boldsymbol{\theta}} := b\tilde{G}$, and set $\underline{L} := L(\tilde{\boldsymbol{c}}_{\boldsymbol{\theta}})$, $\tilde{\boldsymbol{w}}_{\mathrm{best}} := \boldsymbol{0}$, $\underline{\Lambda} := 0$ and $l := 1$.
2) a) Generate $\boldsymbol{t}_{\mathcal{J}_1} \in \mathcal{T}_l$ and calculate $\Delta(\boldsymbol{t}_{\mathcal{J}_1})$. If $\underline{L} < \Delta(\boldsymbol{t}_{\mathcal{J}_1})$, then output $\tilde{\boldsymbol{c}}_{\mathrm{ML}} := \tilde{\boldsymbol{c}}_{\boldsymbol{\theta}} \oplus \tilde{\boldsymbol{w}}_{\mathrm{best}}$ and stop. Otherwise generate $\tilde{\boldsymbol{w}}_{\mathcal{J}_1} := \boldsymbol{t}_{\mathcal{J}_1}\tilde{G}$.
   b) Calculate $\Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}_1})$. If $\Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}_1}) < \underline{\Lambda}$, then $\underline{\Lambda} := \Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}_1})$, $\underline{L} := L(\tilde{\boldsymbol{c}}_{\boldsymbol{\theta}}) + \underline{\Lambda}$, $\tilde{\boldsymbol{w}}_{\mathrm{best}} := \tilde{\boldsymbol{w}}_{\mathcal{J}_1}$ and $i := 2$.
3) a) Generate $\boldsymbol{t}_{\mathcal{J}_i}$ from $\boldsymbol{t}_{\mathcal{J}_{i-1}}$ by the GS generation rule A.
   b) Calculate $\Delta(\boldsymbol{t}_{\mathcal{J}_i})$. If $\underline{L} < \Delta(\boldsymbol{t}_{\mathcal{J}_i})$, then try to generate the next error pattern $\boldsymbol{t}_{\mathcal{J}_s}$ by the GS generation rule B, otherwise go to c). If there exists $\boldsymbol{t}_{\mathcal{J}_s}$, then set $\boldsymbol{t}_{\mathcal{J}_i} := \boldsymbol{t}_{\mathcal{J}_s}$ and go to b), otherwise go to 4).
   c) Set $\tilde{\boldsymbol{w}}_{\mathcal{J}_i} := \boldsymbol{t}_{\mathcal{J}_i}\tilde{G}$ and calculate $\Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}_i})$. If $\Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}_i}) < \underline{\Lambda}$, then $\underline{\Lambda} := \Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}_i})$, $\underline{L} := L(\tilde{\boldsymbol{c}}_{\boldsymbol{\theta}}) + \underline{\Lambda}$ and $\tilde{\boldsymbol{w}}_{\mathrm{best}} := \tilde{\boldsymbol{w}}_{\mathcal{J}_i}$. Set $i := i + 1$ and go to a).
4) Set $l := l + 1$. If $l \leqq \kappa$, then go to 2), otherwise output $\tilde{\boldsymbol{c}}_{\mathrm{ML}} := \tilde{\boldsymbol{c}}_{\boldsymbol{\theta}} \oplus \tilde{\boldsymbol{w}}_{\mathrm{best}}$ and stop. □

(1) : Among a set of test error patterns with the same Hamming weight, test error patterns are generated in binary order.

4. Sufficient Conditions based on a Order Relation

4.1 A Condition for Eliminating Unnecessary Test Error Patterns

Similar to eq.(6), if we find out that a test error pattern $\boldsymbol{t}_{\mathcal{J}}$ cannot give the best candidate codeword, the test error pattern is eliminated before encoded to $\tilde{\boldsymbol{w}}_{\mathcal{J}}$. Then the number of iterations (for constructing test error codewords) can be reduced and this saves the large number of computations. Furthermore, unlike eq.(6), if we find out that a test error pattern $\boldsymbol{t}_{\mathcal{J}}$ cannot give the best candidate codeword without any real operations, we can save the number of real operations.

For $\boldsymbol{t}_{\mathcal{J}}$, let a subset $\tilde{\mathcal{U}}(\mathcal{J})$ of $\tilde{\mathcal{V}}^n$ be defined such that

$$\tilde{\mathcal{U}}(\mathcal{J}) = \{\boldsymbol{x} \mid x_j = t_{\mathcal{J},j}, 1 \leqq j \leqq k \text{ and } w_H(\boldsymbol{x}) \geqq d\}. \quad (13)$$

For a nonempty set $\tilde{\mathcal{U}}(\mathcal{J})$, let $\tilde{\boldsymbol{u}}_{\mathcal{J}}$ denotes the binary vector such that $\Lambda(\tilde{\boldsymbol{u}}_{\mathcal{J}}) = \underline{\Lambda}[\tilde{\mathcal{U}}(\mathcal{J})]$.

[Definition 4] For two vectors $\tilde{\boldsymbol{v}} = (\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_n) \in \tilde{\mathcal{V}}^n$ and $\tilde{\boldsymbol{x}} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n) \in \tilde{\mathcal{V}}^n$, define two location sets $\mathcal{D}_\alpha(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{x}})$ with $\alpha \in \{0, 1\}$ as

$$\mathcal{D}_\alpha(\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{x}}) = \{j \mid \tilde{v}_{\phi(j)} = 1 \text{ and } \tilde{x}_{\phi(j)} = \alpha\},\ \alpha \in \{0,1\}, \quad (14)$$

where ϕ denotes the permuation function such that $\tilde{\boldsymbol{x}} = \phi(\bar{\boldsymbol{x}}) \in \tilde{\mathcal{V}}^n$, i.e., $\tilde{\theta}_{\phi(j)} = \tilde{\theta}_j$ and $|\tilde{\theta}_{\phi(j)}| \geqq |\tilde{\theta}_{\phi(j+1)}|$ for $1 \leqq j < n$. □

[Theorem 1] For a test error pattern $\boldsymbol{t}_{\mathcal{J}}$, assume that there is a order relation such that

$$\mathcal{D}_1(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset) <_S \mathcal{D}_0(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset), \quad (15)$$

then $\tilde{\boldsymbol{c}}_{\mathcal{J}}$ cannot be better than $\tilde{\boldsymbol{c}}_\emptyset$ where $\tilde{\boldsymbol{c}}_{\mathcal{J}} = \tilde{\boldsymbol{c}}_\emptyset \oplus \boldsymbol{t}_{\mathcal{J}}\tilde{G}$.

Proof: Assume that $\mathcal{D}_0(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset) = \{j_1, j_2, \ldots, j_m\}$ and $\mathcal{D}_1(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset) = \{j'_1, j'_2, \ldots, j'_{m'}\}$ such that $j_1 < j_2 < \cdots < j_m$ and $j'_1 < j'_2 < \cdots < j'_{m'}$. By Def.4, for each element of $\mathcal{D}_\alpha(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset)$, $\alpha \in \{0, 1\}$ satisfies

$$|\tilde{\theta}_{\phi(j_1)}| \geqq |\tilde{\theta}_{\phi(j_2)}| \geqq \cdots \geqq |\tilde{\theta}_{\phi(j_m)}|, \quad (16)$$

$$|\tilde{\theta}_{\phi(j'_1)}| \geqq |\tilde{\theta}_{\phi(j'_2)}| \geqq \cdots \geqq |\tilde{\theta}_{\phi(j'_{m'})}|. \quad (17)$$

By the order relation $\mathcal{D}_0(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset) <_S \mathcal{D}_1(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset)$, $m' \leqq m$ and $|\tilde{\theta}_{\phi(j_h)}| \geqq |\tilde{\theta}_{\phi(j'_h)}|$ for $1 \leqq h \leqq m'$.

Equation (7) expands as follows:

$$\Lambda(\tilde{\boldsymbol{u}}_{\mathcal{J}}) = \sum_{j \in \mathcal{D}_0(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset)} |\tilde{\theta}_{\phi(j)}| - \sum_{j \in \mathcal{D}_1(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset)} |\tilde{\theta}_{\phi(j)}| > 0. \quad (18)$$

Since $\Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}}) \geqq \Lambda(\tilde{\boldsymbol{u}}_{\mathcal{J}})$, from eq.(8)

$$L(\tilde{\boldsymbol{c}}_{\mathcal{J}}) \geqq L(\tilde{\boldsymbol{c}}_\emptyset) + \Lambda(\tilde{\boldsymbol{u}}_{\mathcal{J}}) \geqq L(\tilde{\boldsymbol{c}}_\emptyset). \quad (19)$$

Then $\tilde{\boldsymbol{c}}_{\mathcal{J}}$ is not better than $\tilde{\boldsymbol{c}}_\emptyset$. □

By Theorem 1, if eq.(15) holds for a test error pattern $\boldsymbol{t}_{\mathcal{J}}$, then we can eliminate $\boldsymbol{t}_{\mathcal{J}}$ without encoding. Therefore we will call eq.(15) *Elimination Criterion*.

[Lemma 1] Assume that there is a order relation such that $\mathcal{D}_1(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset) <_S \mathcal{D}_0(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset)$. For any $\boldsymbol{t}_{\mathcal{J}'} \neq \boldsymbol{t}_{\mathcal{J}}$ such that $\boldsymbol{t}_{\mathcal{J}} <_V \boldsymbol{t}_{\mathcal{J}'}$, $\tilde{\boldsymbol{c}}_{\mathcal{J}'}$ cannot be better than $\tilde{\boldsymbol{c}}_\emptyset$ where $\tilde{\boldsymbol{c}}_{\mathcal{J}'} = \tilde{\boldsymbol{c}}_\emptyset \oplus \boldsymbol{t}_{\mathcal{J}'}\tilde{G}$.

Proof: The order relation such that $\boldsymbol{t}_{\mathcal{J}} <_V \boldsymbol{t}_{\mathcal{J}'}$ indicates

$$\Lambda(\tilde{\boldsymbol{u}}_{\mathcal{J}'}) > \Lambda(\tilde{\boldsymbol{u}}_{\mathcal{J}}). \quad (20)$$

Then $\Lambda(\tilde{\boldsymbol{u}}_{\mathcal{J}'}) > 0$. By eq.(20) and the equation such that $\Lambda(\tilde{\boldsymbol{w}}_{\mathcal{J}'}) > \Lambda(\tilde{\boldsymbol{u}}_{\mathcal{J}'})$, $L(\tilde{\boldsymbol{c}}_{\mathcal{J}'}) > L(\tilde{\boldsymbol{c}}_\emptyset)$. □

In [8], a criterion for eliminating any test error patterns with Hamming weight l called *Covering Test* has been proposed in the Least Reliable Basis based soft decision decoding. Elimination Criterion can be regarded as a dual version of Covering Test. Covering Test is a termination criterion which is applied to only the first element $\boldsymbol{t}_{\mathcal{J}_1} \in \mathcal{T}_l$, generated by the GS generation rule A. If $\boldsymbol{t}_{\mathcal{J}_1} \in \mathcal{T}_l$ satisfies the Covering Test (or Elimination Criterion), then all elements in $\mathcal{T}_l$ can be eliminated by Lemma 1 because there is order relations $\boldsymbol{t}_{\mathcal{J}_1} <_V \boldsymbol{t}_{\mathcal{J}_i}$ for $\boldsymbol{t}_{\mathcal{J}} \in \mathcal{T}_l, i > 1$. On the other hand, Elimination Criterion in this paper can be used for any test error patterns since it is not an explicit termination condition. For $\boldsymbol{t}_{\mathcal{J}_1} \in \mathcal{T}_l$, Covering Test is based on the generalized Hamming weight [11] and $\tilde{\boldsymbol{x}}_{\mathcal{J}_1}$ such that $\tilde{\boldsymbol{x}}_{\mathcal{J}_1} <_V \tilde{\boldsymbol{u}}_{\mathcal{J}_1}$ is used in the notation of this paper. Although we consider the MLD for arbitrary long codes whose Hamming weights are rarely known, the Elimination Criterion is more stringent than Covering Test since $\tilde{\boldsymbol{x}}_{\mathcal{J}_1} <_V \tilde{\boldsymbol{u}}_{\mathcal{J}_1}$. However, test error patterns cannot be predetermined as in [8].

For $\boldsymbol{t}_{\mathcal{J}}$ and $\tilde{\boldsymbol{c}}_{\text{ref}} \in \mathcal{C}$ which has been already obtained bafore generating $\boldsymbol{t}_{\mathcal{J}}$, let a subset $\tilde{\mathcal{U}}(\mathcal{J}, \tilde{\boldsymbol{c}}_{\text{ref}})$ of $\tilde{\mathcal{V}}^n$ be defined such that

$$\tilde{\mathcal{U}}(\mathcal{J}, \tilde{\boldsymbol{c}}_{\text{ref}}) = \{\boldsymbol{x} \mid x_j = t_{\mathcal{J},j}, 1 \leqq j \leqq k, \quad (21)$$
$$w_H(\boldsymbol{x}) \geqq d \text{ and } d_H(\boldsymbol{x}, \tilde{\boldsymbol{c}}_{\text{ref}}) \geqq d\}.$$

For a nonempty set $\tilde{\mathcal{U}}(\mathcal{J}, \tilde{\boldsymbol{c}}_{\text{ref}})$, let $\tilde{\boldsymbol{u}}_{\mathcal{J}}(\tilde{\boldsymbol{c}}_{\text{ref}})$ denotes the binary vector such that $\Lambda(\tilde{\boldsymbol{u}}_{\mathcal{J}}(\tilde{\boldsymbol{c}}_{\text{ref}})) = \underline{\Lambda}[\tilde{\mathcal{U}}(\mathcal{J}, \tilde{\boldsymbol{c}}_{\text{ref}})]$.

[Theorem 2] For a test error pattern $\boldsymbol{t}_{\mathcal{J}}$, assume that there is a order relation such that

$$\mathcal{D}_1(\tilde{\boldsymbol{u}}_{\mathcal{J}}(\tilde{\boldsymbol{c}}_{\text{ref}}), \tilde{\boldsymbol{e}}_\emptyset) <_S \mathcal{D}_0(\tilde{\boldsymbol{u}}_{\mathcal{J}}(\tilde{\boldsymbol{c}}_{\text{ref}}), \tilde{\boldsymbol{e}}_\emptyset), \quad (22)$$

then $\tilde{\boldsymbol{c}}_{\mathcal{J}}(\tilde{\boldsymbol{c}}_{\text{ref}})$ cannot be better than $\tilde{\boldsymbol{c}}_\emptyset$ where $\tilde{\boldsymbol{c}}_{\mathcal{J}} = \tilde{\boldsymbol{c}}_\emptyset \oplus \boldsymbol{t}_{\mathcal{J}}\tilde{G}$. □

[Lemma 2] For a test error pattern $\boldsymbol{t}_{\mathcal{J}}$ and $\tilde{\boldsymbol{c}}_{\text{ref}} \neq 0^n$, there is a order relation $\tilde{\boldsymbol{u}}_{\mathcal{J}} <_V \tilde{\boldsymbol{u}}_{\mathcal{J}}(\tilde{\boldsymbol{c}}_{\text{ref}})$. □

[Theorem 3] For a test error pattern $\boldsymbol{t}_{\mathcal{J}}$ and $\tilde{\boldsymbol{c}}_{\text{ref}} \neq 0^n$, Elimination Criterion holds for $\boldsymbol{t}_{\mathcal{J}}$ only if eq.(22) holds.

Proof: By Lemma 2 and the definitions of $\tilde{\boldsymbol{u}}_{\mathcal{J}}$ and $\tilde{\boldsymbol{u}}_{\mathcal{J}}(\tilde{\boldsymbol{c}}_{\text{ref}})$, there are relations such that

$$\mathcal{D}_0(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset) \leqq_S \mathcal{D}_0(\tilde{\boldsymbol{u}}_{\mathcal{J}}(\tilde{\boldsymbol{c}}_{\text{ref}}), \tilde{\boldsymbol{e}}_\emptyset), \quad (23)$$

$$\mathcal{D}_1(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset) = \mathcal{D}_1(\tilde{\boldsymbol{u}}_{\mathcal{J}}(\tilde{\boldsymbol{c}}_{\text{ref}}), \tilde{\boldsymbol{e}}_\emptyset). \quad (24)$$

Hence, Elimination Criterion holds only if $\mathcal{D}_1(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset) <_S \mathcal{D}_0(\tilde{\boldsymbol{u}}_{\mathcal{J}}, \tilde{\boldsymbol{e}}_\emptyset)$. □

Theorem 3 indicates eq.(22) is the necessary condition of Elimination Criterion. Therefore, eq.(22) is more stringent version of Elimination Criterion.

We consider the case when Elimination Criterion is applied to the GS decoding algorithm. In the GS decoding algorithm, each time a test error pattern $\boldsymbol{t}_{\mathcal{J}_i}$ is generated, we test if eq.(6) holds for $\boldsymbol{t}_{\mathcal{J}_i}$. If eq.(6) holds, several successive test error patterns $\boldsymbol{t}_{\mathcal{J}_s}$ such that $\boldsymbol{t}_{\mathcal{J}_i} <_V \boldsymbol{t}_{\mathcal{J}_s}, i < s$, as well as $\boldsymbol{t}_{\mathcal{J}_i}$ are eliminated (see the GS generation rule B). When $\boldsymbol{t}_{\mathcal{J}_i}$ is generated, we suppose that Elimination Criterion is tested with respect to $\boldsymbol{t}_{\mathcal{J}_i}$. If Elimination Criterion holds for $\boldsymbol{t}_{\mathcal{J}_i}$, Elimination Criterion also holds for $\boldsymbol{t}_{\mathcal{J}_{s'}}$ such that $\boldsymbol{t}_{\mathcal{J}_i} <_V \boldsymbol{t}_{\mathcal{J}_{s'}}$. Then if Elimination Criterion holds for $\boldsymbol{t}_{\mathcal{J}_i}$, we can also adopt the GS generation rule B. Remark that no real operations are required in this case, because we need not to compute $\Delta(\boldsymbol{t}_{\mathcal{J}_i})$.

4.2 Conditions for Omitting Unnecessary Metrics Computations

Hereafter we consider the case in which $\boldsymbol{t}_{\mathcal{J}}$ does not satisfy the Elimination Condition and is encoded to $\tilde{\boldsymbol{w}}_{\mathcal{J}}$. If we find out $\tilde{\boldsymbol{c}}_{\mathcal{J}}$ cannot be the best candidate codeword without any real operations, than the computation of eq.(7) can be omitted.

[Theorem 4] For a test error codeword $\tilde{\boldsymbol{w}}_{\mathcal{J}}$, assume

that there is a order relation such that $\mathcal{D}_1(\tilde{w}_\mathcal{J}, \tilde{e}_\emptyset) <_S \mathcal{D}_0(\tilde{w}_\mathcal{J}, \tilde{e}_\emptyset)$. Then $\tilde{c}_\mathcal{J}$ cannot be better than $\tilde{c}_\emptyset$ where $\tilde{c}_\mathcal{J} = \tilde{c}_\emptyset \oplus w_\mathcal{J}$.

Proof: Assume that $\mathcal{D}_0(\tilde{w}_\mathcal{J}, \tilde{e}_\emptyset) = \{j_1, j_2, \ldots, j_m\}$ and $\mathcal{D}_1(\tilde{w}_\mathcal{J}, \tilde{e}_\emptyset) = \{j'_1, j'_2, \ldots, j'_{m'}\}$ such that $j_1 < j_2 < \cdots < j_m$ and $j'_1 < j'_2 < \cdots < j'_{m'}$. Then eq.(16) and (17) hold. Equation (7) is now

$$\Lambda(\tilde{w}_\mathcal{J}) = \sum_{j \in \mathcal{D}_0(\tilde{w}_\mathcal{J}, \tilde{e}_\emptyset)} |\tilde{\theta}_{\phi(j)}| - \sum_{j \in \mathcal{D}_1(\tilde{w}_\mathcal{J}, \tilde{e}_\emptyset)} |\tilde{\theta}_{\phi(j)}| > 0. \quad (25)$$

Therefore $L(\tilde{c}_\mathcal{J}) = L(\tilde{c}_\emptyset) + \Lambda(\tilde{w}_\mathcal{J}) \geq L(\tilde{c}_\emptyset)$. □

The order relation such that $\mathcal{D}_1(\tilde{w}_\mathcal{J}, \tilde{e}_\emptyset) <_S \mathcal{D}_0(\tilde{w}_\mathcal{J}, \tilde{e}_\emptyset)$ can be used to find the test error codewords whose reliability loss need not to be computed. Hereafter, we call the test of this order relation *Omitting Criterion A*.

Theorem 4 implies that $\tilde{w}_\emptyset$ is compared with $\tilde{w}_\emptyset = 0^n$ and judeged if $\Lambda(\tilde{w}_{\mathcal{J}_i}) > \Lambda(\tilde{w}_\emptyset) = 0$. At a decoding stage of constructing $\tilde{w}_{\mathcal{J}_i}$, the best test error codeword $\tilde{w}_*$ in $\tilde{C}_i$ is not necessarily equal to $\tilde{w}_\emptyset$. Since $\Lambda(\tilde{w}_*) < 0$ for $\tilde{w}_* \neq 0^n$, if we can compare $\tilde{w}_{\mathcal{J}_i}$ with $\tilde{w}_*$, we can expect that the Omitting Criterion A be more effective.

At a decoding stage of constructing $\tilde{w}_\mathcal{J}$, let $\tilde{c}_*$ and $\tilde{e}_*$ be such that $\tilde{c}_* = \tilde{c}_\emptyset \oplus \tilde{w}_*$ and $\tilde{e}_* = \tilde{z} \oplus \tilde{c}_*$.

[Lemma 3] At a decoding stage of constructing $\tilde{w}_\mathcal{J}$, $L(\tilde{c}_\mathcal{J})$ is calculated as follows:

$$L(\tilde{c}_\mathcal{J}) = L(\tilde{c}_*) + \Lambda(\tilde{w}_* \oplus \tilde{w}_\mathcal{J}). \quad (26)$$

Proof: Since $\tilde{c}_\emptyset = \tilde{c}_* \oplus \tilde{w}_*$ and $\tilde{e}_* = \tilde{z} \oplus \tilde{c}_*$, the left hand side of eq.(26) expands in the following way.

$$\begin{aligned} L(\tilde{c}_\mathcal{J}) &= L(\tilde{c}_\emptyset \oplus \tilde{w}_\mathcal{J}) = L(\tilde{c}_* \oplus \tilde{w}_* \oplus \tilde{w}_\mathcal{J}) \\ &= \sum_{j=1}^{n} (\tilde{e}_{*,j} \oplus \tilde{w}_{*,j} \oplus \tilde{w}_{\mathcal{J},j}) |\tilde{\theta}_j| \\ &= \sum_{i=1}^{n} \tilde{e}_{*,j} |\tilde{\theta}_j| + \sum_{j=1}^{n} (1 - 2\tilde{e}_{*,j})(\tilde{w}_{*,j} \oplus \tilde{w}_{\mathcal{J},j}) |\tilde{\theta}_j| \\ &= L(\tilde{c}_*) + \Lambda(w_* \oplus \tilde{w}_\mathcal{J}). \quad (27) \end{aligned}$$

Hence we have eq.(26) and the proof is completed. □

Equation (26) shows the relation between $\tilde{c}_*$ and $\tilde{c}_\mathcal{J}$.

[Theorem 5] For a test error codeword $w_{\mathcal{J}_i}$, assume that there is a order relation such that

$$\mathcal{D}_1((\tilde{w}_* \oplus \tilde{w}_{\mathcal{J}_i}), \tilde{e}_*) <_S \mathcal{D}_0((\tilde{w}_* \oplus \tilde{w}_{\mathcal{J}_i}), \tilde{e}_*), \quad (28)$$

then $\tilde{c}_{\mathcal{J}_i}$, $\tilde{c}_{\mathcal{J}_i} = \tilde{c}_\emptyset \oplus w_{\mathcal{J}_i}$, cannot be better than $\tilde{c}_*$. □

In general, $w_H(\tilde{e}_*)$ such that $\tilde{e}_* \neq \tilde{e}_\emptyset$, is smaller than $w_H(\tilde{e}_\emptyset)$ [8]. Consequently, we have the more stringent criterion for omitting unnecessary computations of eq.(7) than the Omitting Criterion A since $|\mathcal{D}_1((\tilde{w}_* \oplus \tilde{w}_{\mathcal{J}_i}), \tilde{e}_*)|$ tends to be small and $(\tilde{w}_* \oplus \tilde{w}_\mathcal{J}) \not<_V \tilde{w}_*$ (this equation is led by Generation Condition of test error patterns). We will call the order relation eq.(28) *Omitting Criterion B*.

[Corollary 1] For $\tilde{w}_\mathcal{J}$ and $\tilde{w}_*$, Omitting Criterion B holds only if $t_* <_V t_\mathcal{J}$ where $\tilde{w}_* = t_* G$. □

This result is from Generation Condition of test error patterns. Therefore, for $\tilde{w}_\mathcal{J}$, we adopt Omitting Criterion A if $t_* \not<_V t_\mathcal{J}$, and adopt Omitting Criterion B otherwise.

5. Simulation Results

In this section, we present simulation results for the binary (63,30,13) BCH code and the binary (127,64,21) BCH code in order to evaluate effectiveness of the proposed criteria. The results are obtained by simulating 10000 codewords for each SNR (E_b/N_0 [dB]) and the average values are shown in tables. The proposed criteria are applied to the GS decoding algorithm and we compare the results with the original GS decoding algorithm.

Table 1 The number of constructed test error codewords for the (63, 30, 13) BCH code and the (127, 64, 21) BCH code

E_b/N_0	(63,30) code		E_b/N_0	(127,64) code	
	GS	EC		GS	EC
2.00	$8.08 \cdot 10^2$	$8.07 \cdot 10^2$	3.00	$1.78 \cdot 10^5$	$1.78 \cdot 10^5$
2.50	$3.77 \cdot 10^2$	$3.76 \cdot 10^2$	3.50	$2.05 \cdot 10^4$	$2.05 \cdot 10^4$
3.00	$1.60 \cdot 10^2$	$1.59 \cdot 10^2$	4.00	$2.69 \cdot 10^3$	$2.67 \cdot 10^3$
3.50	$6.10 \cdot 10^1$	$5.96 \cdot 10^1$	4.50	$4.15 \cdot 10^2$	$3.98 \cdot 10^2$
4.00	$2.39 \cdot 10^1$	$2.24 \cdot 10^1$	5.00	$7.92 \cdot 10^1$	$6.75 \cdot 10^1$
4.50	9.91	8.67	5.50	$1.86 \cdot 10^1$	$1.2 \cdot 10^1$
5.00	3.71	2.86	6.00	4.95	2.07
5.50	1.42	0.90	6.50	1.30	0.27

Table 2 The number of real operations for the (63, 30, 13) BCH code

E_b/N_0	GS	OC_{fix}	OC_{adapt}	$EC + OC_{adapt}$
2.00	$2.04 \cdot 10^4$	$6.92 \cdot 10^3$	$5.94 \cdot 10^3$	$5.94 \cdot 10^3$
2.50	$9.36 \cdot 10^3$	$3.08 \cdot 10^3$	$2.57 \cdot 10^3$	$2.57 \cdot 10^3$
3.00	$3.88 \cdot 10^3$	$1.21 \cdot 10^3$	$8.80 \cdot 10^2$	$8.77 \cdot 10^2$
3.50	$1.43 \cdot 10^3$	$4.26 \cdot 10^2$	$2.99 \cdot 10^2$	$2.96 \cdot 10^2$
4.00	$5.49 \cdot 10^2$	$1.42 \cdot 10^2$	$9.20 \cdot 10^1$	$8.91 \cdot 10^1$
4.50	$2.26 \cdot 10^2$	$5.71 \cdot 10^1$	$4.01 \cdot 10^1$	$3.76 \cdot 10^1$
5.00	$8.39 \cdot 10^1$	$2.03 \cdot 10^1$	$1.27 \cdot 10^1$	$1.06 \cdot 10^1$
5.50	$3.31 \cdot 10^1$	7.63	5.76	4.23

Table 3 The number of real operations for the (127, 64, 21) BCH code

E_b/N_0	GS	OC_{fix}	OC_{adapt}	$EC + OC_{adapt}$
3.00	$8.04 \cdot 10^5$	$2.38 \cdot 10^6$	$1.63 \cdot 10^6$	$1.63 \cdot 10^6$
3.50	$8.80 \cdot 10^5$	$2.69 \cdot 10^5$	$1.55 \cdot 10^5$	$1.55 \cdot 10^5$
4.00	$1.10 \cdot 10^5$	$3.38 \cdot 10^4$	$1.59 \cdot 10^4$	$1.59 \cdot 10^4$
4.50	$1.62 \cdot 10^4$	$4.03 \cdot 10^3$	$1.85 \cdot 10^3$	$1.82 \cdot 10^3$
5.00	$2.99 \cdot 10^3$	$5.83 \cdot 10^2$	$2.89 \cdot 10^2$	$2.70 \cdot 10^2$
5.50	$6.86 \cdot 10^2$	$8.25 \cdot 10^1$	$4.73 \cdot 10^1$	$3.82 \cdot 10^1$
6.00	$1.82 \cdot 10^2$	$1.93 \cdot 10^1$	$1.33 \cdot 10^1$	8.78
6.50	$4.94 \cdot 10^1$	5.59	4.80	2.74

Table 4 The number of computations of eq.(7) for the (63, 30, 13) BCH code

E_b/N_0	GS	OC_{fix}	OC_{adapt}
2.00	$8.08 \cdot 10^2$	$1.95 \cdot 10^2$	$1.48 \cdot 10^2$
2.50	$3.77 \cdot 10^2$	$8.89 \cdot 10^1$	$6.39 \cdot 10^1$
3.00	$1.60 \cdot 10^2$	$3.56 \cdot 10^1$	$1.96 \cdot 10^1$
3.50	$6.10 \cdot 10^1$	$1.32 \cdot 10^1$	6.85
4.00	$2.39 \cdot 10^1$	4.24	1.76
4.50	9.91	1.65	$7.96 \cdot 10^{-1}$
5.00	3.71	$5.42 \cdot 10^{-1}$	$1.56 \cdot 10^{-1}$
5.50	1.42	$1.36 \cdot 10^{-1}$	$4.02 \cdot 10^{-2}$

Table 5 The number of computations of eq.(7) for the (127, 64, 21) BCH code

E_b/N_0	GS	OC_{fix}	OC_{adapt}
3.00	$1.78 \cdot 10^5$	$3.43 \cdot 10^4$	$1.49 \cdot 10^4$
3.50	$2.05 \cdot 10^4$	$4.56 \cdot 10^3$	$1.50 \cdot 10^3$
4.00	$2.69 \cdot 10^3$	$6.52 \cdot 10^2$	$1.63 \cdot 10^2$
4.50	$4.15 \cdot 10^2$	$7.97 \cdot 10^1$	$1.86 \cdot 10^1$
5.00	$7.92 \cdot 10^1$	$1.16 \cdot 10^1$	3.19
5.50	$1.86 \cdot 10^1$	1.40	$3.77 \cdot 10^{-1}$
6.00	4.95	$2.61 \cdot 10^{-1}$	$8.40 \cdot 10^{-2}$
6.50	1.30	$2.86 \cdot 10^{-2}$	$5.40 \cdot 10^{-3}$

Consider the following four modifications of the GS decoding algorithms.

(1) [the algorithm-EC]: each time a test error pattern $t_\mathcal{J}$ is generated, first Elimination Criterion is tested. If Elimination Criterion holds for $t_\mathcal{J}$, then we generate the next test error pattern following to the GS generation rule B. Otherwise, eq.(6) is tested as well as the original GS decoding algorithm. In tables, this algorithm is denoted with EC.

(2) [the algorithm-OC_{fix}]: each time a test error codeword

$\tilde{w}_{\mathcal{J}}$ is constructed, Omitting Criterion A is tested if $\tilde{w}_{\mathcal{J}}$ gives a better candidate codeword than $\tilde{c}_{\emptyset}$. If Omitting Criterion A holds for $\tilde{w}_{\mathcal{J}}$, then we omit the calculation of eq.(7) and generate the next error pattern. In tables, this algorithm is denoted with $\mathrm{OC}_{\mathrm{fix}}$.

(3) [the algorithm-$\mathrm{OC}_{\mathrm{adapt}}$]: each time $\tilde{w}_{\mathcal{J}}$ such that $\tilde{w}_{\mathcal{J}} = t_{\mathcal{J}}\tilde{G}$ is constructed, either Omitting Criterion A or B is adaptively tested. As stated in Sect.4.2, Omitting Criterion A is applied if $t_* \not< _V t_{\mathcal{J}}$, and Omitting Criterion B is applied otherwise. In tables, this algorithm is denoted with $\mathrm{OC}_{\mathrm{adapt}}$.

(4) [the algorithm-$(\mathrm{EC} + \mathrm{OC}_{\mathrm{adapt}})$]: each time a test error pattern $t_{\mathcal{J}}$ is generated, first Elimination Criterion is tested. If Elimination Criterion holds for $t_{\mathcal{J}}$, then we generate the next test error pattern following to the GS generation rule B. Otherwise, eq.(6) is tested as well as the original GS decoding algorithm. When $\tilde{t}_{\mathcal{J}}$ is constructed, then either Omitting Criterion A or B is adaptively applied, similar to the the algorithm-$\mathrm{OC}_{\mathrm{adapt}}$. In tables, this algorithm is denoted with $\mathrm{EC} + \mathrm{OC}_{\mathrm{adapt}}$.

In tables, we show the results of the following simulations.

(1) In order to evaluate the effectiveness of Elimination Criterion, we compare the average number of constructing test error codewords in the original GS decoding algorithm and the algorithm-EC. The results are shown in Table 1.

(2) In order to evaluate the effectiveness of Elimination Criterion, Omitting Criterion A and B, we compare the average number of real operations in each decoding algorithm. The results are shown in Table 2 and 3.

(3) In order to evaluate the effectiveness of Omitting Criterion A and B, we compare the average number of computations of eq.(7) in the original GS decoding algorithm, the algorithm-$\mathrm{OC}_{\mathrm{fix}}$ and the algorithm-$\mathrm{OC}_{\mathrm{adapt}}$. The results are shown in Table 4 and 5.

We show the results of Table 1 as follows. At high SNR, for both (63,30,13) BCH code and (127,64,21) BCH code, the number of candidate codewords in the algorithm-EC is from 1/2 to 2/3 that in the original GS decoding algorithm. This values indicate that Elimination Condition works well. At middle to low SNR, however, the number of candidate codewords is almost the same in both decoding algorithm. The reason is, $w_H(e_{\emptyset})$ is generally large at middle to low SNR, then eq.(6), which is updated each time the best candidate is obtained, works better than Elimination Criterion, whose effectiveness largely depends on $w_H(e_{\emptyset})$.

From table 2 and 3, the number of real operations in the original GS decoding algorithm is the largest and that in the algorithm-$\mathrm{OC}_{\mathrm{fix}}$ is the second largest. Even the algorithm-$\mathrm{OC}_{\mathrm{fix}}$ reduces the number of real operations about 1/3 for the (63,30,13) code and 1/4 for the (127,64,21) code compared with the original GS decoding algorithm. The number of real operations for the algorithm-$\mathrm{OC}_{\mathrm{adapt}}$ is the third largest and for the algorithm-$(\mathrm{EC} + \mathrm{OC}_{\mathrm{adapt}})$ is the smallest at each SNR for both codes. At low SNR, however, there is no difference between the numbers of real operations for the algorithm-$\mathrm{OC}_{\mathrm{adapt}}$ and the algorithm-$\mathrm{EC} + \mathrm{OC}_{\mathrm{adapt}}$, since, as we see the result of Table 1, Elimination Criterion hardly works and the efficiency of both algorithms are totally depends of Omitting Criteria. It is noteworthy that, even at low SNR, the numbers of real operations in the algorithm-$\mathrm{OC}_{\mathrm{adapt}}$ and algorithm-$(\mathrm{EC} + \mathrm{OC}_{\mathrm{adapt}})$ are about 1/4 for the (63,30,13) code and 1/5 for the (127,64,21) code, respectively, compared with the original GS decoding algorithm.

From Table 4, Omitting Criterion A in algorithm-$\mathrm{OC}_{\mathrm{adapt}}$ holds for more than 3/4 test error codewords for (63,30,13) code the at each SNR. From Table 5, the rate which Omitting Criterion A in algorithm-$\mathrm{OC}_{\mathrm{adapt}}$ holds increases up to 4/5 for the (127,64,21) code at each SNR. These results imply that $\tilde{c}_{\emptyset}$ is a good candidate as the initial codeword in the MRB based decoding algorithms. The values for the algorithm-$\mathrm{OC}_{\mathrm{adapt}}$ show the effectiveness of adaptive procedure in which the reference codewords are selected in accordance with an order relation. It is noteworthy that, at high SNR, the number of computation of eq.(6) in the algorithm-$\mathrm{OC}_{\mathrm{adapt}}$ is almost negligible for both (63,30,13) code and (127,64,21) code.

6. Concluding Remarks

In this paper, we derived two types of criteria for the MRB based MLD algorithm. The first one is a criterion for eliminating test error patterns for which it is impossible to give the ML codeword. The second one is a criterion for omitting metrics computations of candidate codewords which cannot be the ML codeword. For implementation of these criteria, we need not real number operations and, in consequence, it can be concluded that the algorithm applying criteria always reduce (or at most the same as) the computational complexity compared to the conventional algorithm. The results of computer simulation show the effectiveness of the proposed criteria for the (63,30,13) and the (127,64,21) BCH code in the GS decoding algorithm. The proposed criteria is applicable to other MRB based MLD algorithm such as one in [3]. As future improvements, more stringent criteria that eliminate unnecessary test error patterns need to be derived.

Acknowledgement

This work is supported by Japan Society for the Promotion of Science under Grants-in-Aid for Scientific Research No. 1576-0281 and Waseda University Grant for Special Research Project No. 2001A-566.

Reference

[1] D. Gazelle and J. Snyders, "Reliability-based code-search algorithm for maximum likelihood decoding of block codes," IEEE Trans. Inf. Theory vol.43, pp.239-249, Jan. 1997.

[2] Y. S. Han, C. P. R. Hartman and C. C. Chen, "Efficient priority-first search maximum-likelihood soft decision decoding of linear block codes," IEEE Trans. Inf. Theory vol.39, pp.1514–1523, Sept. 1993.

[3] A. Valembois and M. P. C. Fossorier, "An improved method to compute lists of binary vectors that optimize a given weight function with application of soft-decision decoding," IEEE Commn. Letters, vol.5, pp.456–458, Nov. 2001.

[4] T. Okada, M. Kobayashi and S. Hirasawa, "An efficient heuristic search method for maximum likelihood decoding of linear block codes using dual codes," IEICE Trans. Fundamentals, vol.E85–A, No.2, pp.485–489, Feb. 2002.

[5] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," IEEE Trans. Inf. Theory vol.41, pp.1379-1396, Sept. 1995.

[6] C. C. Shih, C. R. Wulff, C. R. P. Hartmann and C. K. Mohan, "Efficient heuristic search algorithms for soft-decision decoding of linear block codes," IEEE Trans. Inf. Theory, vol.44, pp.3023–3038, Nov. 1998.

[7] A. Valembois and M. P. C. Fossorier, "A comparison between "most-reliable-basis reprocessing" strategies," IEICE Trans. Fundamentals, vol.E85-A, pp.1727–1741, July 2001.

[8] M. P. C. Fossorier, S. Lin and J. Snyders, "Reliability-based syndrome decoding of linear block codes," IEEE Trans. Inf. Theory, vol.44, pp.388–398, Jan. 1998.

[9] T. Koumoto, T. Kasami and S. Lin, "A Sufficient Condition for Ruling Out Some Useless Test Error Patterns in Iterative Decoding Algorithms," IEICE Trans. Fundamentals, Vol. E81–A, No.2, pp.321–326, Feb. 1998.

[10] D. J. Taipale and M. B. Pursley, "An improvement to generalized minimum-distance decoding," IEEE Trans. Inform. Theory, vol.37, pp.167 – 172, Jan. 1991.

[11] V. K. Wei, "Generalized Hamming Weights for Linear Codes", IEEE Trans. Inform. Theory, vol.37, pp.1412–1413, Sept. 1991.