International Symposium on Information Theory and its Applications, ISITA2004
Parma, Italy, October 10–13, 2004

Efficient Reliability-based Soft Decision Decoding Algorithm over Markov Modulated Channel

Hideki YAGI†, Toshiyasu MATSUSHIMA†, and Shigeichi HIRASAWA†

† Department of Industrial and Management Systems Engineering,
School of Science and Engineering, Waseda University,
3-4-1 Ohkubo Shinjuku-ku, Tokyo, 169-8555 JAPAN
E-mail: yagi@hirasa.mgmt.waseda.ac.jp

Abstract

We discuss soft-decision decoding which achieves near-maximum likelihood decoding (MLD) of binary block codes over a Markov modulated channel. In this paper, a new soft-decision decoding algorithm using a generalized Expectation Maximization (EM) algorithm is proposed. Each iteration step of the proposed decoding algorithm can be regarded as performing MLD over an additive white Gaussian noise (AWGN) channel, so the proposed decoding algorithm can employ most of conventional efficient methods devised for the AWGN channel. The simulation results show that the proposed decoding algorithm achieves almost the same performance as that of MLD which needs exhaustive search of codewords.

1. Introduction

Soft-decision decoding (SDD) reduces the block error probability of decoding by taking advantage of information of channel noise. On additive white Gaussian noise (AWGN) channels, many researchers have devoted to develop efficient maximum likelihood decoding (MLD) and sub-optimum SDD algorithms for block codes [1, 2, 3, 7].

Recently, additive noise channels with a hidden Markov model have attracted great attentions due to its practicality. On these channels, several symbol-wise maximum a-posteriori probability (MAP) decoding algorithms of block codes have been proposed [6, 8, 9]. However, MLD or sub-optimum SDD algorithms that reduce the block error probability of decoding over these channels have not sufficiently studied. To the authors' knowledge, only sub ML (sequence-wise MAP) decoding algorithm for trellis codes has been devised [5]. To aim at reducing the block error probability of decoding, we need to consider much larger search space of the most likely codeword than that in an AWGN channel, since it is direct product of the spaces of codewords and channel states sequences.

In this paper, we propose a new SDD algorithm of block codes using a generalized Expectation Maximization (EM) algorithm [10] over Markov modulated Gaussian noise (MMGN) channels. In the EM principle, we can reduce the search space at each iteration step. The proposed decoding algorithm is an iterative algorithm and each iteration step can be conducted similar to an MLD algorithm of block codes over an AWGN channel. We then derive (i) reliability measure of binary symbols and (ii) a termination condition of the decoding algorithm. As a result, we show by simulations that the proposed SDD algorithm achieves near MLD with relatively small decoding complexity.

2. Preliminary

2.1. Channel Model

Assume that a channel is modeled as the Markov model with finite discrete states $\mathcal{S} = \{0, 1, \ldots, |\mathcal{S}|-1\}$. At a state $i \in \mathcal{S}$, a Gaussian noise with mean 0 and variance $\sigma^2(i)$ is generated. We denote a state of Markov model at time j with s_j and let $\boldsymbol{s}_j^{(m)} = (s_{j-m+1}, s_{j-m+2}, \ldots, s_j) \in \mathcal{S}^m$ for some integer $m \geq 1$. Then transition probability of the m-th order Markov model can be expressed as[1] $p(s_j = i|\boldsymbol{s}_{j-1}^{(m)})$. Assuming that the Markov model has stationary distribution, let $p(i)$ be the stationary probability at $i \in \mathcal{S}$. We assume that the order of Markov model, transition and stationary probabilities are known to the decoder. This channel is called an MMGN channel.

Let $\mathcal{C}$ be a binary linear (n, k) block code of length n, the dimension of code k with a generator matrix G. A codeword $\boldsymbol{c} = (c_1, c_2, \ldots, c_n) \in \{0, 1\}^n$ of $\mathcal{C}$ is mapped into $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, $x_j = (-1)^{c_j} \in \{-1, +1\}$ with equal probability and $\boldsymbol{x}$ is transmitted over an MMGN channel. At the decoder, we estimate

[1] We define that elements of $\boldsymbol{s}_{j-1}^{(m)}$ take no values if the time indices are negative, e.g., $p(s_1|s_{-1}, s_0) = p(s_1|s_0)$ for $m = 2$.

the transmitted codeword $\boldsymbol{c}$ from the received sequence $\boldsymbol{r} = (r_1, r_2, \ldots, r_n) \in \mathcal{R}^n$.

2.2. Reliability-based MLD Algorithm over AWGN channel

We describe reliability-based MLD algorithms with the column-permuted generator matrix [1, 2, 3, 7] that efficiently search the most likely codeword over an AWGN channel. In a decoder, the received sequence $\boldsymbol{r}$ is mapped into a sequence $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n), \theta_j = \ln \frac{P(r_j|c_j=0)}{P(r_j|c_j=1)}$, where $P(r_j|c_j)$ denotes the likelihood function of c_j. Then, we obtain the hard decision received sequence $\boldsymbol{y} = (y_1, y_2, \ldots, y_n) \in \{0,1\}^n$ by

$$y_j = \begin{cases} 0, & \text{if} \quad \theta_j \geq 0; \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

For $\forall j \in [1, n]$, $|\theta_j|$ is called j-th reliability (over the AWGN channel)[2]. In this paper, we call any measures which express the confidence value of y_j, j-th reliability.

For $\boldsymbol{c} \in \mathcal{C}$, we define

$$L(\boldsymbol{c}) = \sum_{j|c_j \neq y_j} |\theta_j|. \tag{2}$$

Then, $\arg\max_{\boldsymbol{c}\in\mathcal{C}} \{P(\boldsymbol{r}|\boldsymbol{c})\} = \arg\min_{\boldsymbol{c}\in\mathcal{C}} \{L(\boldsymbol{c})\}$ [7]. An MLD algorithm searches the most likely codeword $\boldsymbol{c}_{\text{best}}$ such that $\boldsymbol{c}_{\text{best}} = \arg\min_{\boldsymbol{c}\in\mathcal{C}} L(\boldsymbol{c})$.

The reliability-based decoder first re-orders the most reliable and linearly independent (MRI) k columns of a generator matrix in non-increasing reliabilities. Then it performs standard row operations to make these k columns the identity matrix. We denote the resultant matrix by $\tilde{G}$. We obtain $\tilde{\boldsymbol{r}}$ and $\tilde{\boldsymbol{y}}$ by the same permutation of $\boldsymbol{r}$ and $\boldsymbol{y}$, respectively. Let $\tilde{\mathcal{C}}$ be the code obtained by the same permutation for $\mathcal{C}$.

Let $\boldsymbol{u} = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_k)$ be the MRI k symbols of $\tilde{\boldsymbol{y}}$ and the initial codeword of search is obtained by $\tilde{\boldsymbol{c}}_0 = \boldsymbol{u}\tilde{G}$. After obtaining $\tilde{\boldsymbol{c}}_0$, we generate k dimensional vectors $\boldsymbol{t} \in \{0,1\}^k$ to obtain candidate codewords by $\tilde{\boldsymbol{c}} = (\boldsymbol{u} \oplus \boldsymbol{t})\tilde{G}$, where we call $\boldsymbol{t} \in \{0,1\}^k$ test error patterns (TEPs)[3]. Then, we iteratively generate candidate codewords and compute their likelihood. The decoder outputs the most likely codeword $\tilde{\boldsymbol{c}}_{\text{best}}$.

The right hand side (r.h.s.) of eq. (2) is a sum of reliabilities (positive real number). Using this structure of eq. (2), (i) acceptance criteria of the most likely codeword and (ii) elimination criteria of unnecessary TEPs are proposed to make the MLD algorithm efficient [1, 2, 3].

[2] $[i, j]$ denotes the set of integers from i to j, for two integers i and j such that $i \leq j$.

[3] $\oplus$ represents Exclusive OR operation.

2.3. Generalized EM Algorithm

The EM algorithm is an iterative procedure for computing (near) maximum likelihood estimation of unknown parameter $\boldsymbol{\psi}$ from observed data $\boldsymbol{z}_{\text{obs}}$. Let $\boldsymbol{z}_{\text{mis}}$ and $\boldsymbol{z} = (\boldsymbol{z}_{\text{obs}}, \boldsymbol{z}_{\text{mis}})$ be "missing data" which cannot be observed directly and "complete data ", respectively. Even when it is practically infeasible to maximize the likelihood function $P(\boldsymbol{z}_{\text{obs}}|\boldsymbol{\psi})$, it is often easy to maximize the joint likelihood function $P(\boldsymbol{z}|\boldsymbol{c})$ that includes missing data [4, 10]. The EM algorithm iteratively maximizes an expectation of $P(\boldsymbol{z}|\boldsymbol{c})$ by the following steps. We here denote the estimated sequence in $(l-1)$-th iteration by $\boldsymbol{\psi}^{(l)}$ where $\boldsymbol{\psi}^{(1)}$ is chosen arbitrarily.

E step: Calculate the following function.

$$Q(\boldsymbol{\psi}|\boldsymbol{\psi}^{(l)}) = E_{\boldsymbol{z}}\Big[\ln P(\boldsymbol{z}|\boldsymbol{\psi})\big|\boldsymbol{z}_{\text{obs}}, \boldsymbol{\psi}^{(l)}\Big]. \tag{3}$$

M step: Search $\boldsymbol{\psi}^{(l+1)}$ such that

$$\boldsymbol{\psi}^{(l+1)} = \arg\max_{\boldsymbol{\psi}} Q(\boldsymbol{\psi}|\boldsymbol{\psi}^{(l)}). \tag{4}$$

Until $\boldsymbol{\psi}^{(l)} = \boldsymbol{\psi}^{(l+1)}$, the above steps are iteratively carried out. □

In M step, a generalized EM (GEM) algorithm sets $\boldsymbol{\psi}$, which satisfies $Q(\boldsymbol{\psi}|\boldsymbol{\psi}^{(l)}) \geq Q(\boldsymbol{\psi}^{(l)}|\boldsymbol{\psi}^{(l)})$ instead of maximizing $Q(\boldsymbol{\psi}|\boldsymbol{\psi}^{(l)})$, as estimated sequence $\boldsymbol{\psi}^{(l+1)}$. Namely, the EM algorithm is a specific instance of GEM algorithms [10].

3. Soft-Decision Decoding Algorithm over MMGN Channel

3.1. Reliability-based SDD algorithm via GEM algorithm

Let $\boldsymbol{s} = (s_0, s_1, \ldots, s_n) \in \mathcal{S}^{n+1}$ be the state sequence of the channel when a transmitted sequence $\boldsymbol{x}$ is input to the channel where $\boldsymbol{x}$ and $\boldsymbol{s}$ are mutually independent. Then, the likelihood of a codeword $\boldsymbol{c}$ is

$$P(\boldsymbol{r}|\boldsymbol{c}) = \sum_{\boldsymbol{s}\in\mathcal{S}^{n+1}} \prod_{j=1}^{n} p(s_j|\boldsymbol{s}_{j-1}^{(m)}) P(r_j|s_j, c_j). \tag{5}$$

To perform MLD, we have to find the codeword which maximizes eq. (5). However, since the r.h.s. of eq. (5) includes marginalization with all $\boldsymbol{s} \in \mathcal{S}^{n+1}$, the conventional efficient algorithm, such as the Viterbi algorithm, cannot be straightforwardly implemented[4]. Then, we will propose a new SDD algorithm via a GEM algorithm.

We regard a sequence $\boldsymbol{s} \in \mathcal{S}^{n+1}$ as missing data in the GEM algorithm. Similar to eq. (3), we define the

[4] The likelihood of a given $\boldsymbol{c}$ can be computed by the well-known Baum-Welch algorithm [4, 8] with the complexity of $O(n)$.

function $Q(\boldsymbol{c}|\boldsymbol{c}^{(l)})$ at l-th step of the GEM algorithm as

$$Q(\boldsymbol{c}|\boldsymbol{c}^{(l)}) = E_{\boldsymbol{S}}\Big[\ln P(\boldsymbol{r},\boldsymbol{s}|\boldsymbol{c})\Big|\boldsymbol{r},\boldsymbol{c}^{(l)}\Big]. \tag{6}$$

For $j \in [1,n]$, we obtain the hard-decision symbol y_j by eq. (1), where $P(r_j|c_j)$ can be computed by $P(r_j|c_j) = \sum_{i\in\mathcal{S}} p(s_j = i)P(r_j|s_j = i, c_j)$.

Theorem 1 For some $\boldsymbol{c}^{(l)} \in \mathcal{C}$, we define

$$M^{(l)}(\boldsymbol{c}) = \sum_{j|\, c_j\neq y_j}\sum_{i\in\mathcal{S}} \frac{p(s_j = i|\boldsymbol{r},\boldsymbol{c}^{(l)})}{\sigma^2(i)}\big|2r_j\big|. \tag{7}$$

Then, for $\boldsymbol{c}$ and $\boldsymbol{c}'$,

$$Q(\boldsymbol{c}|\boldsymbol{c}^{(l)}) \geq Q(\boldsymbol{c}'|\boldsymbol{c}^{(l)}) \quad \text{iff} \quad M^{(l)}(\boldsymbol{c}) \leq M^{(l)}(\boldsymbol{c}'). \tag{8}$$

Therefore, for arbitrary subcode $\mathcal{C}'$,

$$\arg\max_{\boldsymbol{C}\in\mathcal{C}'} Q(\boldsymbol{c}|\boldsymbol{c}^{(l)}) = \arg\min_{\boldsymbol{C}\in\mathcal{C}'} M^{(l)}(\boldsymbol{c}). \tag{9}$$

□

Proof: See appendix A.

By Theorem 1, maximizing $Q(\cdot|\boldsymbol{c}^{(l)})$ can be carried out by minimizing $M^{(l)}(\cdot)$. We here define

$$\phi_j^{(l)} = 2r_j\sum_{i\in\mathcal{S}} \frac{p(s_j = i|\boldsymbol{r},\boldsymbol{c}^{(l)})}{\sigma^2(i)}, \qquad j \in [1,n]. \tag{10}$$

Then eq. (7) is now $M^{(l)}(\boldsymbol{c}) = \sum_{j|\, c_j\neq y_j} |\phi_j^{(l)}|$ which has similar expression to eq. (2). Therefore, in the l-th M step, we can efficiently search $\boldsymbol{c}^{(l+1)}$ which decreases $M^{(l)}(\cdot)$ by the MLD algorithm described in Sect. 2.2, which minimizes $L(\cdot)$. In this search, $|\phi_j^{(l)}|, j \in [1,n]$, is regarded as j-th reliability.

We describe the proposed SDD algorithm below, where $\boldsymbol{c}^{(1)}$ is some binary sequence.

[The proposed SDD algorithm]

E step: For $j \in [1,n]$ and $i \in \mathcal{S}$, compute $p(s_j = i|\boldsymbol{r},\boldsymbol{c}^{(l)})$ by the Baum-Welch algorithm [4, 8]. For $j \in [1,n]$, we obtain $|\phi_j^{(l)}|$ by eq. (10).

M step: Search $\boldsymbol{c}^{(l+1)}$ satisfying the following equation via an MLD algorithm[5] of Sect. 2.2.

$$M^{(l)}(\boldsymbol{c}^{(l+1)}) \leq M^{(l)}(\boldsymbol{c}^{(l)}). \tag{11}$$

The above steps are iterated until $\boldsymbol{c}^{(l)} = \boldsymbol{c}^{(l+1)}$, then $\boldsymbol{c}^{(l)}$ is output as $\boldsymbol{c}_{\text{best}}$. □

The proposed SDD algorithm is not guaranteed to converge the global maxima because of the EM principal [10].

[5]We will describe the algorithm in detail in the next section.

In the l-th M step, the reliability $|\phi_j^{(l)}|, \forall j \in [1,n]$, satisfies

$$\begin{aligned}\phi_j^{(l)} &= E_{s_j}\Big[\ln \frac{P(r_j|s_j, c_j = 0)}{P(r_j|s_j, c_j = 1)}\Big|\boldsymbol{r},\boldsymbol{c}^{(l)}\Big] \\ &= \sum_{i\in\mathcal{S}} p(s_j = i|\boldsymbol{r},\boldsymbol{c}^{(l)}) \ln \frac{P(r_j|s_j = i, c_j = 0)}{P(r_j|s_j = i, c_j = 1)}.\end{aligned} \tag{12}$$

i.e., $\phi_j^{(l)}$ is expectation of joint log likelihood ratio of s_j and c_j with $p(s_j = i|\boldsymbol{r},\boldsymbol{c}^{(l)})$. The derivation of eq. (12) will be given in Appendix B.

3.2. The l-th M Step of the Proposed Decoding Algorithm

In this section, we describe the l-th M step of the proposed SDD algorithm.

Let $\tilde{G}^{(l)}$ be a permuted generator matrix whose left-most k columns are MRI in non-increasing reliabilities $|\phi_j^{(l)}|$. We denote the best codeword obtained so far by $\tilde{\boldsymbol{c}}^*$. The algorithm searches candidate codewords satisfying $M^{(l)}(\tilde{\boldsymbol{c}}) \leq M^{(l)}(\tilde{\boldsymbol{c}}^*)$ and the most likely codeword among them is set to $\tilde{\boldsymbol{c}}^{(l+1)}$.

For a TEP $\boldsymbol{t} = (t_1, t_2, \ldots, t_k)$, we define a evaluation function of $\boldsymbol{t}$ as

$$\Delta^{(l)}(\boldsymbol{t}) = \sum_{j|t_j=1} |\tilde{\phi}_j^{(l)}|. \tag{13}$$

Lemma 1 (Elimination of a TEP) In the l-th M step, assume that a generated TEP $\boldsymbol{t}$ satisfies

$$\Delta^{(l)}(\boldsymbol{t}) \geq M^{(l)}(\tilde{\boldsymbol{c}}^*). \tag{14}$$

Then the candidate codeword $\tilde{\boldsymbol{c}} = (\boldsymbol{u} \oplus \boldsymbol{t})\tilde{G}^{(l)}$ given by $\boldsymbol{t}$ satisfies $M^{(l)}(\tilde{\boldsymbol{c}}) \geq M^{(l)}(\tilde{\boldsymbol{c}}^*)$.

Proof: From the definitions of $M^{(l)}(\boldsymbol{c})$ and $\Delta^{(l)}(\boldsymbol{t})$,

$$\begin{aligned}M^{(l)}(\tilde{\boldsymbol{c}}) &= \sum_{j|t_j=1} |\tilde{\phi}_j^{(l)}| + \sum_{j=n-k+1}^{n} (\tilde{y}_j \oplus \tilde{c}_j)|\tilde{\phi}_j^{(l)}| \\ &= \Delta^{(l)}(\boldsymbol{t}) + \sum_{j=n-k+1}^{n} (\tilde{y}_j \oplus \tilde{c}_j)|\tilde{\phi}_j^{(l)}|.\end{aligned} \tag{15}$$

Therefore, $M^{(l)}(\tilde{\boldsymbol{c}}) \geq \Delta^{(l)}(\boldsymbol{t})$. Eq. (14) implies $M^{(l)}(\tilde{\boldsymbol{c}}) \geq M^{(l)}(\tilde{\boldsymbol{c}}^*)$. □

From Lemma 1, we need not encode a TEP satisfying eq. (14) and the next TEP is generated[6]. In order to judge if the l-th M step can be terminated, we have the following theorem which is readily proven by Lemma 1.

[6]Although more effective evaluation functions than $\Delta^{(l)}(\cdot)$ are devised in [1, 2, 3], we will not describe them for simplicity. However, they can be also applicable to the proposed decoding algorithm in this paper.

Theorem 2 (Condition of Local Termination)
In the l-th M step, let $\mathcal{T}^{(l)}$ be a set of TEPs not generated yet. If

$$\min_{\boldsymbol{t}\in\mathcal{T}^{(l)}} \left\{\Delta^{(l)}(\boldsymbol{t})\right\} \geq M^{(l)}(\tilde{\boldsymbol{c}}^*), \tag{16}$$

then the algorithm outputs $\tilde{\boldsymbol{c}}^{(l+1)} = \tilde{\boldsymbol{c}}^*$. □

We can use eqs. (14) and (16) to reduce the number of searched candidate codewords.

We describe the the l-th M Step of the proposed SDD algorithm.

[The l-th M Step of the Proposed Algorithm]

S1) Construct $\tilde{G}^{(l)}$ and generate the initial codeword $\tilde{\boldsymbol{c}}_0 := \boldsymbol{u}\tilde{G}^{(l)}$. If $P(\boldsymbol{r}|\boldsymbol{c}^{(l)}) \geq P(\boldsymbol{r}|\boldsymbol{c}_0)$, then set $\tilde{\boldsymbol{c}}_{\text{best}} := \tilde{\boldsymbol{c}}^{(l)}$, otherwise set $\tilde{\boldsymbol{c}}_{\text{best}} := \tilde{\boldsymbol{c}}_0$. Set $P_{\text{best}} := P(\boldsymbol{r}|\boldsymbol{c}_{\text{best}})$, $M_{\text{best}} := M^{(l)}(\tilde{\boldsymbol{c}}_{\text{best}})$.
S2) Generate a TEP $\boldsymbol{t} \in \mathcal{T}^{(l)}$ and set $\mathcal{T}^{(l)} := \mathcal{T}^{(l)} \setminus \boldsymbol{t}$.
  a) If eq. (14) holds for $\boldsymbol{t}$, then go to step S3).
  b) Set $\tilde{\boldsymbol{c}} := (\boldsymbol{u} \oplus \boldsymbol{t})\tilde{G}$. If $P_{\text{best}} \geq P(\boldsymbol{r}|\boldsymbol{c})$, then set $P_{\text{best}} := P(\boldsymbol{r}|\boldsymbol{c}), M_{\text{best}} := M^{(l)}(\tilde{\boldsymbol{c}}), \tilde{\boldsymbol{c}}_{\text{best}} := \tilde{\boldsymbol{c}}$.
S3) If eq. (16) is satisfied or $\mathcal{T}^{(l)} = \emptyset$, then output $\tilde{\boldsymbol{c}}^{(l+1)} := \tilde{\boldsymbol{c}}_{\text{best}}$ and terminate the l-th M step. Otherwise go to step S2). □

The proposed SDD algorithm can be terminated by the following conditions: assume that eq. (16) holds and $\tilde{\boldsymbol{c}}^{(l)} = \tilde{\boldsymbol{c}}^*$, then $\tilde{\boldsymbol{c}}^{(l+1)} = \tilde{\boldsymbol{c}}^*$ from Theorem 2. i.e., the SDD algorithm is converged and $\tilde{\boldsymbol{c}}^{(l)}$ is output as the estimated codeword.

4. Evaluation by Simulations

4.1. Conditions

We compare four decoding algorithms: (i) an ideal MLD algorithm given the information of the real state transition sequences (denoted by "Ideal")[7], (ii) the MLD algorithm by exhaustive search (denoted by "MLD"), (iii) the proposed SDD algorithm (denoted by "Proposed"), (iv) the MLD algorithm by regarding noises are AWGN at average SNR [dB] (denoted by "Conventional" or "Conv."). In M step of the proposed SDD algorithm, "Ideal" and "Conventional" algorithms, we generate TEPs according to the method of Gazelle et al. [2]. The initial sequence in the proposed algorithm is set as $\boldsymbol{c}^{(1)} = \boldsymbol{y}$. For each decoding algorithm, at least 10,000 codewords are transmitted until 100 decoding errors occur. We evaluate them by (i) decoding performance (block error rate) and (ii) decoding complexity (the number of searched candidate codewords for each decoding algorithm and the number of iteration for the proposed SDD algorithm[8]).

[7]It is obvious that results of the "Ideal" algorithm can never be obtained in an actual decoder.

We assume the first-order Markov Model with $\mathcal{S} = \{0,1\}$, $p(0) = 0.9$ and $p(1|0) = 0.1$ for the MMGN channel. Furthermore, we assume the variances of the Gaussian distribution of two states satisfy $\sigma^2(1) = \rho\sigma^2(0), \rho \geq 1$.

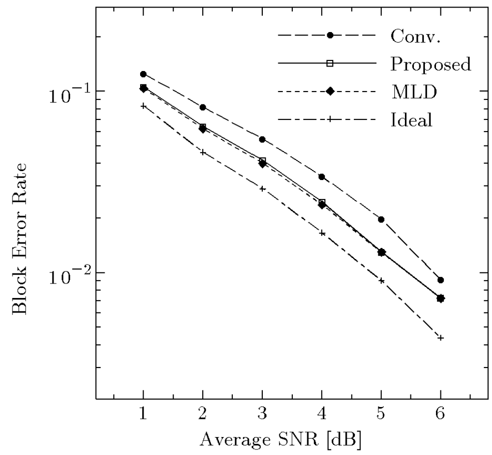

Figure 1: Decoding results for the (24, 12) extended Golay code at $\rho = 5.5$.

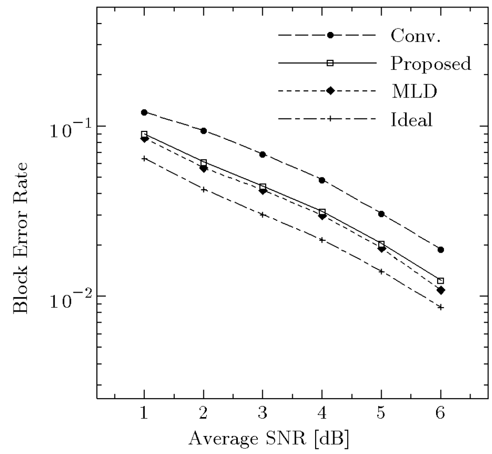

Figure 2: Decoding results for the (24, 12) extended Golay code at $\rho = 8.5$.

4.2. Simulation Results

(Decoding Performance)

In Figs. 1, 2 and 3, we show results of decoding performances at $\rho = 5.5, 8.5$ and average SNR 6.0 [dB], respectively, for the (24,12) extended Golay code. In Fig. 4, we show results for the (63,30) BCH code at $\rho = 8.5$

By Fig. 1, block error rates of the MLD and of the proposed SDD algorithms are almost the same at each SNR when $\rho = 5.5$. It can be expected that from small to medium value of ρ, posterior probability $P(s_j|\boldsymbol{r}, \boldsymbol{c}^{(l)})$

[8]One iteration consists of E and M steps.

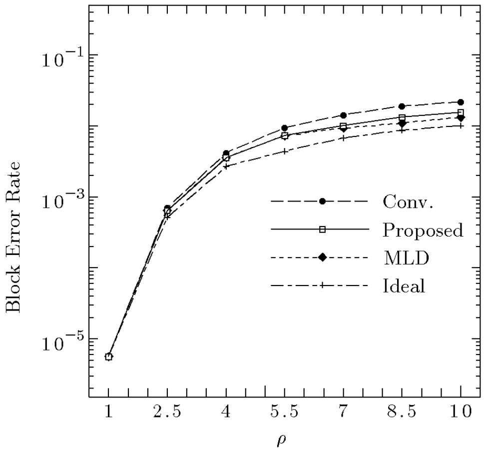

Figure 3: Decoding results for the (24, 12) extended Golay code at average SNR 6.0 [dB].

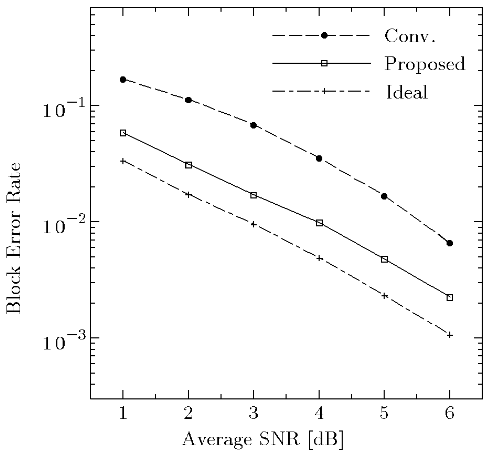

Figure 4: Decoding results for the (63, 30) BCH code at $\rho = 8.5$.

in eqs. (10) and (12) can be estimated highly accurately. By Fig. 2, although we see slight degradation of the performance of the proposed algorithm from MLD at $\rho = 8.5$, it is greatly improved compared with the conventional algorithm. Fig. 3 also shows that the proposed SDD algorithm performs as well as MLD algorithm at each ρ. Note that at $\rho = 1$, the MMGN channel is reduced to the AWGN channel.

By Fig. 4, we see the similar result for the (63,30) BCH code. Remark that the gain of the proposed algorithm from the conventional algorithm is larger than that for the (24,12) extended Golay code. The similar results have obtained at $\rho = 5.5$ and 6.0 [dB] for the (63,30) BCH code.

(Decoding Complexity)

In Tables 1 and 2, we show results of the number of generated candidate codewords for each decoding algorithm at $\rho = 8.5$ for the (24,12) extended Golay code and the (63,30) BCH code, respectively. We also show results of the average (denoted by "ave") and maximum (denoted by "max") number of iterations for the proposed SDD algorithm.

Table 1: The number of candidate codewords for each decoding algorithm and the number of iteration for the (24, 12) extended Golay code at $\rho = 8.5$

SNR	codewords			iterations	
[dB]	MLD	Conv.	Proposed	ave	max
1.0	$3.55 \cdot 10^3$	5.93	9.67	1.74	3
2.0	$3.17 \cdot 10^3$	4.96	7.52	1.55	3
3.0	$2.66 \cdot 10^3$	4.24	5.79	1.30	3
4.0	$2.09 \cdot 10^3$	3.42	4.48	1.02	3
5.0	$1.54 \cdot 10^3$	2.61	3.10	0.754	3
6.0	$1.14 \cdot 10^3$	1.86	1.99	0.558	3

Table 2: The number of candidate codewords for each decoding algorithm and the number of iteration for the (63, 30) BCH code at $\rho = 8.5$

SNR	codewords		iterations	
[dB]	Conv.	Proposed	ave	max
1.0	$4.44 \cdot 10^3$	$1.47 \cdot 10^4$	1.99	3
2.0	$3.86 \cdot 10^3$	$9.55 \cdot 10^3$	1.97	3
3.0	$3.19 \cdot 10^3$	$6.67 \cdot 10^3$	1.88	3
4.0	$2.19 \cdot 10^3$	$3.49 \cdot 10^3$	1.69	3
5.0	$1.33 \cdot 10^3$	$1.26 \cdot 10^3$	1.42	3
6.0	$5.94 \cdot 10^2$	$3.94 \cdot 10^2$	1.14	3

Table 1 shows that the number of generated candidate codewords for the proposed SDD algorithm at 1.0 [dB] is less than 10 which is less than twice that of the conventional algorithm. As for the number of iterations, the average and maximum numbers are no more than two and three, respectively.

Table 2 also shows the effectiveness of the proposed SDD algorithm, whose searched codewords is at most less than four times that for the conventional algorithm, for the (63,30) BCH code at each average SNR. It is noteworthy that, at high average SNRs, the number of searched codewords in the proposed SDD algorithm is less than that for the conventional algorithm. Remark that the exhaustive MLD algorithm cannot be conducted because of its large dimension. As for the number of iterations, the behavior is not different from the case of (24,12) extended Golay code, so we can say that the proposed algorithm is also effective for large codes.

We have the similar results at $\rho = 5.5$ for both codes. These results indicate the effectiveness of Lemma 1 and Theorem 2.

5. Concluding Remarks

In this paper, we propose a new SDD algorithm of block codes over the MMGN channel via a GEM algorithm. The proposed algorithm has an effective termination condition in each iteration step. We show by simulations that the proposed algorithm achieves near MLD with relatively small complexity.

As for further works, we need to devise a method for eliminating unnecessary iterations of the proposed algorithm. Since the proposed SDD algorithm largely depends on the initial sequence $\boldsymbol{c}^{(1)}$, a measure for selecting a good initial sequence is also needed.

Acknowledgment

The one of the authors, H. Yagi, would like to thank Dr. M. Kobayashi at Shonan Institute of Technology and Mr. T. Ishida and Mr. G. Hosoya at Waseda University for their valuable comments.

Appendix A: Proof of Theorem 1

The function $Q(\boldsymbol{c}|\boldsymbol{c}^{(l)})$ for some $\boldsymbol{c}^{(l)} \in \mathcal{C}$ is expanded as:

$$\begin{aligned} Q(\boldsymbol{c}|\boldsymbol{c}^{(l)}) &= \sum_{\boldsymbol{s} \in \mathcal{S}^{n+1}} p(\boldsymbol{s}|\boldsymbol{r}, \boldsymbol{c}^{(l)}) \ln P(\boldsymbol{r}, \boldsymbol{s}|\boldsymbol{c}) \\ &= \sum_{\boldsymbol{s} \in \mathcal{S}^{n+1}} p(\boldsymbol{s}|\boldsymbol{r}, \boldsymbol{c}^{(l)}) \Big\{ \ln P(\boldsymbol{r}|\boldsymbol{s}, \boldsymbol{c}) + \ln p(\boldsymbol{s}) \Big\}, \quad (17) \end{aligned}$$

where the last equation is lead by $p(\boldsymbol{s}|\boldsymbol{c}) = p(\boldsymbol{s})$. The first term of eq. (17), which only depends on a choice of $\boldsymbol{c}$, is further expanded as follows:

$$\begin{aligned} &\sum_{\boldsymbol{s} \in \mathcal{S}^{n+1}} p(\boldsymbol{s}|\boldsymbol{c}^{(l)}, \boldsymbol{r}) \ln P(\boldsymbol{r}|\boldsymbol{s}, \boldsymbol{c}) \\ &\quad = \sum_{j=1}^{n} \sum_{i \in \mathcal{S}} p(s_j = i|\boldsymbol{c}^{(l)}, \boldsymbol{r}) \ln P(r_j|s_j = i, c_j). \quad (18) \end{aligned}$$

From the assumption of Gaussian distribution at each state $s_j \in \mathcal{S}$,

$$\ln P(r_j|s_j = i, c_j) = \Big\{ -\frac{(r_j - (-1)^{c_j})^2}{2\sigma^2(i)} \Big\} + D \quad (19)$$

where D is the independent term of $\boldsymbol{c}$. Substituting the r.h.s. of eq. (19) into eq. (18),

$$\begin{aligned} &\sum_{\boldsymbol{s} \in \mathcal{S}^{n+1}} p(\boldsymbol{s}|\boldsymbol{c}^{(l)}, \boldsymbol{r}) \ln P(\boldsymbol{r}|\boldsymbol{s}, \boldsymbol{c}) \\ &\quad = \sum_{j=1}^{n} \sum_{i \in \mathcal{S}} p(s_j = i|\boldsymbol{c}^{(l)}, \boldsymbol{r}) \Big\{ -\frac{(r_j - (-1)^{c_j})^2}{2\sigma^2(i)} + D \Big\} \\ &\quad = \sum_{j=1}^{n} \sum_{i \in \mathcal{S}} \frac{p(s_j = i|\boldsymbol{c}^{(l)}, \boldsymbol{r})}{\sigma^2(i)} r_j (-1)^{c_j} + D', \quad (20) \end{aligned}$$

where D' is the independent term of $\boldsymbol{c}$. Then from eqs. (17) and (20), we have

$$\begin{aligned} Q(\boldsymbol{c}|\boldsymbol{c}^{(l)}) &= \sum_{j=1}^{n} \sum_{i \in \mathcal{S}} \frac{p(s_j = i|\boldsymbol{c}^{(l)}, \boldsymbol{r})}{\sigma^2(i)} r_j (-1)^{c_j} + D'', \\ &= \sum_{j=1}^{n} \sum_{i \in \mathcal{S}} \frac{p(s_j = i|\boldsymbol{c}^{(l)}, \boldsymbol{r})}{\sigma^2(i)} |r_j| - M^{(l)}(\boldsymbol{c}) + D'', \quad (21) \end{aligned}$$

where the second term D'' is independent of $\boldsymbol{c}$ and eq. (21) indicates eq. (9). The proof of eq. (8) is straightforward from eq. (9).

Appendix B: Derivation of eq. (12)

From Gaussian distribution of eq. (19), we have

$$\begin{aligned} &\sum_{i \in \mathcal{S}} p(s_j = i|\boldsymbol{r}, \boldsymbol{c}^{(l)}) \ln \frac{P(r_j|s_j = i, c_j = 0)}{P(r_j|s_j = i, c_j = 1)} \\ &\quad = \sum_{i \in \mathcal{S}} p(s_j = i|\boldsymbol{r}, \boldsymbol{c}^{(l)}) \\ &\qquad \times \Big\{ -\frac{1}{2\sigma^2(i)}(r_j - 1)^2 + \frac{1}{2\sigma^2(i)}(r_j + 1)^2 \Big\} \\ &\quad = \sum_{i \in \mathcal{S}} p(s_j = i|\boldsymbol{r}, \boldsymbol{c}^{(l)}) \frac{2r_j}{\sigma^2(i)}. \quad (22) \end{aligned}$$

Eq. (22) and the definition of $\phi_j^{(l)}$ prove eq. (12).

References

[1] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inform. Theory*, vol.41, no.5, pp.1379 – 1396, Sept. 1995.

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

[3] 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. Inform. Theory*, vol.39, no.5, pp.1514 – 1523, Sept. 1993.

[4] A. Logothetis and V. Krishnamurthy, "Expectation Maximization algorithms for MAP estimation of jump Markov linear systems," *IEEE Trans. Signal Pro.*, vol.47, no.8, pp.2139 – 2156, Aug. 1999.

[5] W. Turin, "MAP decoding in Channels with Memory," *IEEE Trans. Commn.*, vol.48, no.5, pp.757 – 763, May 2000.

[6] W. Turin, "MAP symbol decoding in Channels with Error Bursts," *IEEE Trans. Inform. Theory*, vol.47, no.5, pp.1832 – 1838, July 2001.

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

[8] T. Wadayama, "An iterative decoding algorithm of low density parity check codes for hidden Markov noise channel," *Proc. of Int. Symposium on Information Theory and Its Applications*, Hawaii, U.S.A., Nov. 2000.

[9] T. Wadayama, "An iterative decoding algorithm for channels with Gibbs distributed noise," *Proc. of IEEE Int. Symposium on Information Theory*, Yokohama, Japan, June 2003.

[10] C. F. J. Wu, "On the convergence properties of the EM algorithm," *Ann. Statist.*, vol.11, no.1, pp.95 – 103, Jan. 1983.