Reduced Memory Turbo MAP Decoding Algorithm for Non-binary Orthogonal Signaling

Dae-Son Kim, Young-Joon Kim and Hong-Yeop Song
CITY-Center for Information Technology of Yonsei University.
Coding and Information Theory Lab, Department of Electrical and Electronic Engineering, Yonsei University
134 Sinchon-dong Seodaemun-gu, Seoul, Korea, 120-749

Email: {ds.kim, yj.kim, hy.song}@coding.yonsei.ac.kr

Abstract—In non-binary turbo decoding combined with non-binary orthogonal signaling such as $M$-ary FSK, the conditional channel probability densities of the non-binary symbols require $M$ values for each symbol. In this paper, we propose a method which can reduce the backup memory size of the channel values of non-binary orthogonal signaling. And we propose a new turbo decoding method which can reduce the memory for a non-binary turbo MAP decoder. We compare the memory requirements of proposed algorithms and the bit error rate performances.

Keywords—Turbo code, non-binary, memory size, MAP decoding

I. INTRODUCTION

Turbo codes have been introduced by Berrou et al. in [1], which consist of a parallel concatenation of recursive systematic convolutional (RSC) codes with interleaving between parallel branches. Their decoding algorithms are normally based on binary PSK signaling. In this paper, we concentrate on the non-binary turbo decoding algorithm for non-binary signaling. The non-coherent modems employing $M$-ary orthogonal signals such as MFSK signals are used in the frequency hopping (FH) satellite communications and many other systems. In order to use the conventional binary turbo decoder, they needed to convert the conditional probability of non-binary symbols to those of bits by using Bayes law. Through this process, however, some combining loss could be raised. This is why a non-binary codes are worth considering in an $M$-ary orthogonal signaling system. In non-binary turbo decoding, the log-likelihood ratios (LLR) for each symbol. In this paper, we propose a method which can reduce the number of the soft information values for each non-binary symbol. They normally use, for the turbo decoder, the windowing methods in the trellis. The decoding algorithm which we propose here can reduce the memory size irrespective of the windowing methods.

II. NON-BINARY MAP DECODING

A. Non-binary turbo code

When $d_k$ denotes the $k$-th $K$ input bits feeding into a non-binary turbo encoder for $k = 1, 2, ..., N$, where $N$ is assumed to be the length of the input symbol sequence, the systematic part and parity part of the output sequences are denoted respectively by $x^s = (x^s_1, x^s_2, ..., x^s_N)$ and $x^p = (x^p_1, x^p_2, ..., x^p_N)$. Such encoded symbols are assumed to be modulated with $2^K$-ary FSK signals and transmitted via AWGN channel. At the receiver, we denote the $k$-th demodulated symbols by $y^s_k$ and $y^p_k$ which are fed into the non-binary turbo decoder. If we define $y = (y_1, y_2, ..., y_N)$ and $y_k = (y^s_k, y^p_k)$.

The goal of the non-binary MAP algorithm is to provide us with the a posteriori probability (APP) of each symbol and decides the maximum APP value given by [1] [3]

$$\mathbb{M}(d_k = i) \triangleq \log(p(d_k = i|y))$$

$$= \log \left( \sum_{(s', s) \in S} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s) \right) \quad (1)$$

where $s_k$ is the state of an encoder at time $k$, $S_i$ is the set of ordered pairs $(s', s)$ corresponding to all state transition from $(s_{k-1} = s')$ to $(s_k = s)$ caused by data input $d_k = i$. The forward recursion and backward recursion of the MAP algorithm yield

$$\alpha_k(s) = \frac{p(s_k = s, y^s_1)}{p(y^s_1)}$$

$$\quad = \sum_{s'} \alpha_{k-1}(s') \gamma_k(s', s) \sum_{s'} \sum_{s' \in S} \alpha_{k-1}(s') \gamma_k(s', s) \quad (2)$$

and

$$\beta_{k-1}(s') = \frac{p(y^s_N | s_{k-1} = s')}{p(y^s_N)}$$

$$\quad = \sum_s \sum_{s'} \alpha_k(s) \gamma_k(s', s) \sum_s \sum_{s'} \alpha_k(s) \gamma_k(s', s). \quad (3)$$

The transition probability $\gamma_k(s', s)$ is

$$\gamma_k(s', s) = p(s_k = s, y_{k-1} = s')$$

$$= P(d_k)p(y^s_{k+1}|d_k)p(y^p_k|d_k). \quad (4)$$

$P(d_k)$ in (4) is a priori probability delivered from previous decoder and $p(y_{k}|d_k)$ is the conditional channel probability density function.
B. Modified Channel Value (MCV) for memory reduction

The \( m \)-th matched-filtered and squared output of the demodulator is denoted by \(|r_m|^2\). The conditional channel probability density function of the \( m \)-th output is [4] [5]

\[
p(r_m|d_k = i) = \begin{cases} 
\frac{1}{2\pi\sigma^2} \exp \left( - \frac{|r_m|^2 + E_s}{\sigma^2} \right) & \text{if } m = i \\
\frac{1}{2\pi\sigma^2} \exp \left( - \frac{|r_m|^2}{2\sigma^2} \right) & \text{otherwise}
\end{cases}
\]

(5)

where \( I_n(x) \) is the modified Bessel function of the first kind, \( \sigma^2 \) is the average noise variance and \( E_s \) is the expected received signal energy. MFSK receiver demodulates \( M \) received symbol values which are independent each other. Then the product of all of the probabilities for \( m = 1, 2, \ldots, M \) in (5) leads to

\[
p(y_k|d_k = i) = A \cdot I_{\alpha} \left( \frac{|r_m|\sqrt{E_s}}{\sigma^2} \right),
\]

(6)

where

\[
A \triangleq \left( \frac{1}{2\pi\sigma^2} \right)^M \exp \left( - \frac{|r_1|^2 + \cdots + |r_M|^2 + E_s}{2\sigma^2} \right). \tag{7}
\]

Non-binary turbo decoder using MFSK signaling needs to remember \( M \) conditional channel probability density functions, or channel values, \( p(y_k|d_k = i) \), \( i = 1, 2, \ldots, M \), for each non-binary symbol \( y_k \). Usually, in this case, the only one value out of these \( M \) channel values contains the transmitted energy and all the others contain the contribution only form the noise, which can be neglected. For this, we define \( i_{\max} \) and \( p_{\max} \) as

\[
p(y_k|d_k = i_{\max}) \geq p(y_k|d_k = j), \quad \forall j = 1, 2, \ldots, M \tag{8}
\]

and

\[
p_{\max}(y_k|d_k = i_{\max}) \triangleq \max \{p(y_k|d_k = 1), \ldots, p(y_k|d_k = M)\}. \tag{9}
\]

We call \( p_{\max} \) as the maximum channel value and \( i_{\max} \) as the side value about it. We can normalize the maximum channel value for further reduction as

\[
p_{\max}(y_k|d_k = i_{\max}) \triangleq \frac{p_{\max}}{p_{\max} + \sum_{i, i \neq i_{\max}} p(y_k|d_k = i)/(M-1)} \tag{10}
\]

The normalized value of the other \( M-1 \) channel values can be derived as follows:

\[
p(y_k|d_k = j) = 1 - p_{\max}(y_k|d_k = i_{\max}), \quad j \neq i_{\max}. \tag{11}
\]

Here, we propose a method that remembers only \( i_{\max} \) and \( p_{\max} \) in the entire decoding algorithm instead of all the \( p(y_k|d_k = i), i = 1, 2, \ldots, M \).

C. The Reduced Memory (RM) MAP algorithm

In the turbo decoding, channel, \( \alpha \) and LLR values need to be stored. In the iterative decoding of turbo code, the one of the probabilities \( \alpha \) converges 1 and the other probabilities converge 0. The probabilities \( \alpha \) can be modified by the same method as in the previous section. If the memory size of RSC encoder is \( K \), there are \( M(=2^K) \) states in the trellis structure. The \( \alpha_k(s) \) needs \( M \) values for each state respectively but we will store only the max value of \( \alpha_k(s) \). The forward recursions of the MAP can be expressed as [3]

\[
\tilde{\alpha}_k(s) = \sum_{s'} \tilde{\alpha}_{k-1}(s') \gamma_k(s', s). \tag{12}
\]

We define \( i_{\max} \) and \( \tilde{\alpha}_{\max} \) as

\[
\tilde{\alpha}_k(s = i_{\max}) \geq \tilde{\alpha}_k(s = j), \quad \forall j = 1, 2, \ldots, M \tag{13}
\]

and

\[
\tilde{\alpha}_{\max}(s = i_{\max}) \triangleq \max \{\tilde{\alpha}_k(s = 1), \ldots, \tilde{\alpha}_k(s = M)\}. \tag{14}
\]

We can change the maximum forward recursion value to the normalized form (2) as follows:

\[
\tilde{\alpha}_{\max}^\text{max}(s = i_{\max}) \triangleq \frac{\tilde{\alpha}_{\max}(s = i_{\max})}{\sum_s \tilde{\alpha}_k(s)}. \tag{15}
\]

And the other \( M-1 \) values can be derived as follows:

\[
\tilde{\alpha}_k(s = j) = \frac{(1 - \tilde{\alpha}_{\max}^\text{max}(s = i_{\max}))}{(M-1)}, \quad j \neq i_{\max}. \tag{16}
\]

General MAP algorithms have to normalize \( \alpha \) for every state but the proposed algorithm only needs to normalize \( \alpha \) only once at the final step.

In Table I, we compare the memory requirements of each case about channel values and \( \alpha \) values where \( N \) is the length of the input symbol sequence or is the interleaver size, and \( M \) is the number of states in the trellis and also is representing \( M \) signals. If the non-binary symbol size increases, then the required memory size dramatically increases. But proposed algorithm is not critical about the non-binary symbol size.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>MEMORY COMPARISON BETWEEN NORMAL MAP AND PROPOSED MAP ALGORITHMS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Channel value</td>
</tr>
<tr>
<td>Normal MAP</td>
<td>N: M: 3</td>
</tr>
<tr>
<td>RM MAP</td>
<td>N: M</td>
</tr>
<tr>
<td>Normal MAP with MCV</td>
<td>N: 2: 3</td>
</tr>
<tr>
<td>RM MAP with MCV</td>
<td>N: 2: 3</td>
</tr>
</tbody>
</table>

III. SIMULATION RESULT AND CONCLUSIONS

In this paper, we will only use the dual-\( K \) turbo code as a non-binary encoder for simulation [6]. The block diagram of the dual-\( K \) turbo encoder for \( K = 2 \) is depicted in Fig. 1. The code rate of dual-\( K \) RSC encoder is 1/2 and the constraint length is 2K. When \( K \) bits are fed into the encoder at a time, the encoder produces an \( M \)-ary systematic symbol and
an M-ary parity symbol where \( M = 2^K \). Two such dual-K RSC encoders can be parallel concatenated via symbolwise interleaver.

Figure 2 shows the iterative decoding performance of the non-binary dual-K turbo code with normal MAP, RM MAP, normal MAP with MCV, and RM MAP with MCV. The code rate is 1/2 obtained by puncturing. The number of iterations is 5 and its performance is sufficiently merged to the decoding performance limit. The interleaver is a random interleaver and its size is 10000. The 8-ary FSK non-coherent modem and the 8-state symbol-by-symbol MAP decoder are considered. Proposed MAP decoder with MCV can save a lot memories (Table I) but has loss about 1dB at BER = 10^{-5}. Proposed RM MAP decoder with MCV can reduce memories further, with an additional loss that can be neglected.

We conclude that the proposed MAP decoding algorithms are practically suitable for decoding turbo codes with non-binary orthogonal signaling.

**ACKNOWLEDGMENT**

This work was supported by the Electronics and Telecommunications Research Institute (ETRI) of the Korea.

**REFERENCES**


