

On MAP Decoding of High-Rate Non-Binary Linear Block Codes

Farzad Ghayour¹, Fambirai Takawira² and Hongjun Xu¹
School of Engineering,
University of KwaZulu-Natal, P. O. Box 4041, Durban, South Africa
Tel: +27 31 2602247, Fax: +27 31 2602740¹
and School of Electrical and Information Engineering
University of Witwatersrand
Tel: +27 11 7177243, Fax: +27 11 4031929²
email: {ghayour, xuh}@ukzn.ac.za¹; fambirai.takawira@wits.ac.za²

Abstract- Optimum or sub-optimum maximum a posteriori (MAP) algorithms are usually good candidates to be used in “soft-in/soft-out” decoders. However, the MAP algorithm, especially for the non-binary codes, is a computationally complicated decoding method. In this paper, an optimum MAP decoding rule for non-binary codes based on the dual space of the code is presented. Since, the complexity of this proposed algorithm is related to the inverse of the code-rate, it can be attractive for the codes with high coding rates.

Index Terms—MAP algorithm, Dual MAP, Non-binary MAP

I. INTRODUCTION

After invention of turbo codes [1], researchers became interested in the design of iteratively decode-able combined codes using “soft-in/soft-out” component decoders. The “soft-inputs” of the decoders are used as *a priori* information in order to calculate the “soft-output” values for the decoders. The “soft-outputs” are the symbols’ *a posteriori* probability (APP) values. In an iterative decoding process, the “soft-output” value of one decoder can be used as extrinsic-information, which alongside the channel-information are employed as the “soft-input” to the other decoders.

Optimum or sub-optimum MAP decoders are usually good candidates to be used as “soft-in/soft-out” decoders. The MAP decoding algorithm was first proposed by Bahl *et al.* to minimize the symbol-error probability of a decoded symbol [2] and it is an optimal method for calculating the symbols’ APP values. The Bahl’s algorithm, which is also known as BCJR, is based on the forward and backward recursions. There are two obstacles in using this algorithm, one is the complexity of MAP algorithm, which arises from the computation of its metric values and the other is the amount of memory requirement.

For reducing the complexity of the MAP algorithm, several optimal and suboptimal algorithms have been proposed. In [3-5] some of these algorithms, such as log-

MAP and max-log-MAP can be found. In all these algorithms, whether optimal or suboptimal, all the state metrics are required to be calculated and saved for the forward and backward recursions. In [5], it was shown that for a linear block code with length N and dimension K defined over a finite field of order q , the number of states based on the direct implementation of the MAP algorithm may reach up to q^K states. This number of states can be very large for the high-rate non-binary codes and thus direct implementation of the MAP decoding algorithm for such codes becomes impractical. On the other hand, in mobile communication systems, due to the increase in demand and growth in number of subscribers, bandwidth is scarce and expensive, and thus codes with high coding rates are required. Moreover, the quality of service, which requires fast encoding and decoding schemes, is another important issue. Therefore, simpler decoding algorithms, which impose less delay to the system, are required.

For high rate codes, the number of codewords of the codes’ dual spaces is less than the number of codewords of the codes’ spaces. Therefore, for a high rate code, implementing the MAP algorithm based on the dual space of the code is less complicated [4]. In this paper, an optimum MAP decoding rule for non-binary codes based on the dual space of the code is presented. The complexity of this algorithm is related to the inverse of the code-rate, and therefore it is practical for the high-rate codes. The paper is organized as follows. In Section II the problem formulation and a description of the MAP algorithm for block codes are given. In Section III, the proposed algorithm is described. The simulation results are presented in Section IV. The final section concludes this paper.

II. NOTATION AND PROBLEM FORMULATION

A. General Notation

In this paper, vectors and matrices are denoted in boldface letters and their elements in lower case, e.g., v_i is the i^{th} element of the vector $\mathbf{v} = (v_0, v_1, \dots, v_{N-1})$. Also an (N, K) code is referred to a linear block code with the length of N

and dimension of K with the parity-check matrix that is denoted by \mathbf{H} . The codes are defined over a finite field of order q , \mathbb{F}_q , therefore the elements of codewords belong to set $\{0, 1, \dots, q-1\}$. Codewords are transmitted over a discrete time memory-less, noisy channel and the soft decision received vector for a codeword is denoted by $\mathbf{r} = (r_0, r_1, \dots, r_{N-1})$.

B. Computation of the MAP Decoding Algorithm

The APP algorithm for the block code can be developed as

$$\hat{v}_i = \arg \max_{v_i \in \{0, 1, \dots, q-1\}} \Pr(v_i | \mathbf{r}, \mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}) \quad (1)$$

Based on BCJR [2], (1) is equal to

$$\hat{v}_i = \arg \max_{v_i \in \{0, 1, \dots, q-1\}} \sum_{(s', s) \in \mathcal{E}_{v_i}^i} \alpha_i(s') \gamma_i(s', s) \beta_{i+1}(s) \quad (2)$$

where s is a state at the i th-level of the code trellis and s' is a state at the $(i+1)$ th-level of the code trellis. Moreover, $\mathcal{E}_{v_i}^i$ is the set of all edges of the trellis of the code, which connects the states s' to the states s and corresponds to the code symbol v_i . If $\mathbf{r}_{i,j} = (r_i, r_{i+1}, \dots, r_{j-1})$, for $0 \leq i \leq j \leq N$ we then have

$$\alpha_i(s) = \Pr(s_i = s, \mathbf{r}_{0,i}) \quad (3)$$

$$\beta_i(s) = \Pr(\mathbf{r}_{i,N} | s_i = s) \quad (4)$$

$$\gamma_i(s', s) = \Pr(s_{i+1} = s, r_i | s_i = s') \quad (5)$$

Let $\Omega_{i-1}^{(c)}(s)$ be as a set of all states at the $(i-1)$ th-level of the code trellis, which are adjacent to state s . According to the BCJR algorithm for state s at i th-level of the trellis, α_i and β_i can be calculated recursively.

For $0 \leq i \leq N$

$$\alpha_i(s) = \sum_{s' \in \Omega_{i-1}^{(c)}(s)} \alpha_{i-1}(s') \gamma_i(s', s) \quad (6)$$

By knowing the fact that $\alpha_0(s_0) = 1$, all α_i for $0 \leq i \leq N$ can be calculated. This is called the forward recursion. The backward recursion is similarly obtained.

For $0 \leq i \leq N$

$$\beta_i(s) = \sum_{s' \in \Omega_{i+1}^{(d)}(s)} \gamma_i(s, s') \beta_{i+1}(s') \quad (7)$$

where, $\Omega_{i+1}^{(d)}(s)$ is a set of all states at $(i+1)$ th-level of the code trellis, which these states are adjacent to state s . Using the fact that $\beta_N(s_f) = 1$, all β_i for $0 \leq i \leq N$ can be found. The branch transition probability in (5) for a block code with statistically independent information bits can be written as

$$\begin{aligned} \gamma_i(s', s) &= \Pr(s_{i+1} = s, r_i | s_i = s') \\ &= \Pr(s_{i+1} = s | s_i = s') \Pr(r_i | s_{i+1} = s, s_i = s') \\ &= \Pr(v_i) \Pr(r_i | v_i) \end{aligned} \quad (8)$$

Different methods have been suggested to carry out the MAP decoding algorithm, but in all of these methods, there are three major steps:

1. Perform the forward recursion process and store all the values calculated for α_i for $0 \leq i \leq N$.
2. Perform the backward recursion process and store all the values calculated for β_i for $0 \leq i \leq N$.
3. For each received codeword symbol, r_i , find the maximum a posteriori probability using (2). To calculate (2), the transition probabilities are needed.

Due to the independence of forward and backward recursions from each other, step 1 and step 2 can be done simultaneously.

III. DUAL IMPLEMENTATION OF THE MAP DECODING ALGORITHM

Based on the orthogonality of the dual code's vector space and the code's vector space, for any codeword belong to code C , (9) is satisfied.

$$\forall \mathbf{v}' \in C', \mathbf{v} \in C : \mathbf{v} \odot \mathbf{v}' = 0 \quad (9)$$

where C' is the dual code of C and \odot denote the inner product between two vectors that are defined over \mathbb{F}_q . Therefore, the APP value for symbol v_i of a codeword can be calculated from

$$\hat{v}_i = \arg \max_{\xi \in \{0, 1, \dots, q-1\}} \left\{ \frac{1}{q^{N-K}} \sum_{\substack{\mathbf{u} \in \mathbf{V}^N \\ u_i = \xi}} \left[\Pr(\mathbf{u} | \mathbf{r}) \sum_{\mathbf{v}' \in C'} e^{j \frac{2\pi}{q} \mathbf{u} \odot \mathbf{v}'} \right] \right\} \quad (10)$$

where \mathbf{V}^N is the vector space of all N -tuples defined over \mathbb{F}_q , and

$$\sum_{\mathbf{v}' \in C'} e^{j \frac{2\pi}{q} \mathbf{u} \odot \mathbf{v}'} = \begin{cases} q^{N-K}, & \mathbf{u} \in C \\ 0 & ; \text{ otherwise} \end{cases} \quad (11)$$

Based on Bayes' rule (10) is equal to

$$\hat{v}_i = \arg \max_{\xi \in \{0, 1, \dots, q-1\}} \left\{ \frac{q^{K-N}}{\Pr(\mathbf{r})} \sum_{\substack{\mathbf{u} \in \mathbf{V}^N \\ u_i = \xi}} \left[\Pr(\mathbf{r}; \mathbf{u}) \sum_{\mathbf{v}' \in C'} e^{j \frac{2\pi}{q} \mathbf{u} \odot \mathbf{v}'} \right] \right\} \quad (12)$$

Equation (12) can be written as (13) where \oplus denote addition defined for \mathbb{F}_q and $\delta_{i,j}$ denote the Kronecker delta. Kronecker delta is equal to one if $i = j$ and zero otherwise.

$\mathbf{e}_i = (\delta_{0,i}, \delta_{1,i}, \delta_{2,i}, \dots, \delta_{N-1,i})$ is a vector with 1 at the i^{th} position and zero elsewhere. Based on the fact that $\left\{e^{j\frac{2\pi}{q}li}\right\}$ for $i = 0, 1, \dots, q-1$ forms a set of orthogonal bases. (13) can be written as (14).

Let $\mathbf{v}' = (v'_0, v'_1, v'_2, \dots, v'_{N-1})$ denote a codeword of the dual code C' . Since, the codewords are transmitted over a memoryless channel and the message symbols are statistically independent, therefore (14) can be written as (15) where \otimes denote multiplication defined for \mathbb{F}_q .

From the definition of Kronecker delta we can conclude that

$$\frac{1}{q} \sum_{l=0}^{q-1} e^{j\frac{2\pi}{q}l\delta_{im}(n\otimes\xi)} = 1 \quad 0 \leq m \leq N-1; m \neq i \quad (16)$$

and from the orthogonality of the basis $\left\{e^{j\frac{2\pi}{q}li}\right\}$ for $i = 0, 1, \dots, q-1$, we know that for $m=i$, (16) is equal to zero unless $(n\otimes\xi) = 0$.

This means that for $m = i$, (15) is maximized if $n = \xi$. Therefore, (15) can be re-written as (17). Equation (17) is a general formula to find the most likely symbol for \hat{v}_i when the codewords are transmitted over any memoryless channel. For reducing the computational complexity of (17), we can use the log-likelihood ratio (LLR) value for each symbol. LLR of a random variable like v_i is defined as

$$L_\xi(v_i) = \ln \left(\frac{\Pr(v_i = \xi)}{\Pr(v_i = 0)} \right) \quad (18)$$

Moreover, LLR can be defined for the joint random variables. For example $L_\xi(v_i; \mathbf{r})$ is given by (19), where v_i is a random variable and \mathbf{r} is a vector of random variables

$$L_\xi(v_i; \mathbf{r}) = \ln \left(\frac{\Pr(v_i = \xi; \mathbf{r})}{\Pr(v_i = 0; \mathbf{r})} \right) \quad (19)$$

and by using the LLR values, (17) can be written as (20).

$$\hat{v}_i = \arg \max_{\xi \in \{0, 1, \dots, q-1\}} \left\{ \frac{q^{K-N}}{\Pr(\mathbf{r})} \sum_{\mathbf{u} \in \mathbb{V}^N} \left[\Pr(\mathbf{r}; \mathbf{u}) \left(\sum_{\mathbf{v}' \in C'} e^{j\frac{2\pi}{q}\mathbf{u} \odot \mathbf{v}'} \right) (\delta_{0, (\mathbf{e}_i \odot \mathbf{u}) \otimes \xi}) \right] \right\} \quad (13)$$

$$\hat{v}_i = \arg \max_{\xi \in \{0, 1, \dots, q-1\}} \left\{ \frac{q^{K-N}}{\Pr(\mathbf{r})} \sum_{\mathbf{u} \in \mathbb{V}^N} \left[\Pr(\mathbf{r}; \mathbf{u}) \left(\sum_{\mathbf{v}' \in C'} e^{j\frac{2\pi}{q}\mathbf{u} \odot \mathbf{v}'} \right) \left(\frac{1}{q} \sum_{l=0}^{q-1} e^{j\frac{2\pi}{q}l((\mathbf{e}_i \odot \mathbf{u}) \otimes \xi)} \right) \right] \right\} \quad (14)$$

$$\begin{aligned} \hat{v}_i &= \arg \max_{\xi \in \{0, 1, \dots, q-1\}} \left\{ \frac{q^{K-N}}{\Pr(\mathbf{r})} \prod_{m=0}^{N-1} \sum_{n=0}^{q-1} \left[\Pr(r_m; n) \left(\sum_{\mathbf{v}' \in C'} e^{j\frac{2\pi}{q}n \otimes \mathbf{v}'_m} \right) \left(\frac{1}{q} \sum_{l=0}^{q-1} e^{j\frac{2\pi}{q}l\delta_{im}(n\otimes\xi)} \right) \right] \right\} \\ &= \arg \max_{\xi \in \{0, 1, \dots, q-1\}} \left\{ \frac{q^{K-N}}{\Pr(\mathbf{r})} \prod_{m=0}^{N-1} \sum_{n=0}^{q-1} \left[\sum_{\mathbf{v}' \in C'} \left[\left(\frac{1}{q} \sum_{l=0}^{q-1} e^{j\frac{2\pi}{q}l\delta_{im}(n\otimes\xi)} \right) \Pr(r_m; n) e^{j\frac{2\pi}{q}n \otimes \mathbf{v}'_m} \right] \right] \right\} \\ &= \arg \max_{\xi \in \{0, 1, \dots, q-1\}} \left\{ \frac{q^{K-N}}{\Pr(\mathbf{r})} \sum_{\mathbf{v}' \in C'} \left[\prod_{m=0}^{N-1} \sum_{n=0}^{q-1} \left[\left(\frac{1}{q} \sum_{l=0}^{q-1} e^{j\frac{2\pi}{q}l\delta_{im}(n\otimes\xi)} \right) \Pr(r_m; n) e^{j\frac{2\pi}{q}n \otimes \mathbf{v}'_m} \right] \right] \right\} \quad (15) \end{aligned}$$

$$\hat{v}_i = \arg \max_{\xi \in \{0, 1, \dots, q-1\}} \left\{ \frac{q^{K-N} \Pr(r_i; \xi)}{\Pr(\mathbf{r})} \sum_{\mathbf{v}' \in C'} \left[\left(e^{j\frac{2\pi}{q}\mathbf{v}'_i \otimes \xi} \right) \left(\prod_{\substack{m=0 \\ m \neq i}}^{N-1} \sum_{n=0}^{q-1} \left[\Pr(r_m; n) e^{j\frac{2\pi}{q}n \otimes \mathbf{v}'_m} \right] \right) \right] \right\} \quad (17)$$

$$\hat{v}_i = \arg \max_{\xi \in \{0, 1, \dots, q-1\}} \left[L_\xi(v_i; r_i) + \ln \left(\frac{\sum_{\mathbf{v}' \in C'} \left[\left(e^{j\frac{2\pi}{q}\mathbf{v}'_i \otimes \xi} \right) \left(\prod_{\substack{m=0 \\ m \neq i}}^{N-1} \sum_{n=0}^{q-1} \left(e^{L_n(n; r_m)} e^{j\frac{2\pi}{q}(n \otimes \mathbf{v}'_m)} \right) \right) \right]}{\sum_{\mathbf{v}' \in C'} \left[\prod_{\substack{m=0 \\ m \neq i}}^{N-1} \sum_{n=0}^{q-1} \left(e^{L_n(n; r_m)} e^{j\frac{2\pi}{q}(n \otimes \mathbf{v}'_m)} \right) \right]} \right) \right] \quad (20)$$

IV. SIMULATION RESULTS

We simulated the MAP decoding for a two-dimensional (64,49) product code defined over \mathbb{F}_4 , where single parity-check codes are used as its component codes. We are not making any use of non-binary modulation techniques and we assume that all the codewords are being transmitted over a binary input AWGN (BI-AWGN) channel, using binary phase-shift keying (BPSK) modulation. Therefore, the symbols need to be mapped to binary sequences prior to their transmission.

In [6], the theoretical upper bound for the bit-error probability of an (N, K) binary block code with the minimum distance of d_{\min} , transmitted over an AWGN channel, with coherent detection, and, using maximum likelihood decoding, is given as

$$p_e \leq \sum_{h=d_{\min}}^N \sum_{\omega=1}^K \frac{\omega}{K} A_{\omega,h} Q\left(\sqrt{2R_c h E_b/N_0}\right) \quad (21)$$

where R_c is the code rate and $A_{\omega,h}$ are number of codewords of the block code with output weight h associated with an input sequence of weight ω . Based on (21) the theoretical upper bound for the (64,49) product code, under the given conditions would be equal to

$$p_e \leq \sum_{h=4}^{64} \sum_{\omega=1}^{49} \frac{\omega}{49} A_{\omega,h} Q\left(\sqrt{1.53h E_b/N_0}\right) \quad (22)$$

In Figure.1 the performance of the (64,49) product code decoded with the direct MAP algorithm is compared to the performance of the same code when it is decoded with the dual MAP algorithm. The performances are also compared with the theoretical upper bound, which exists for soft decoding of this code.

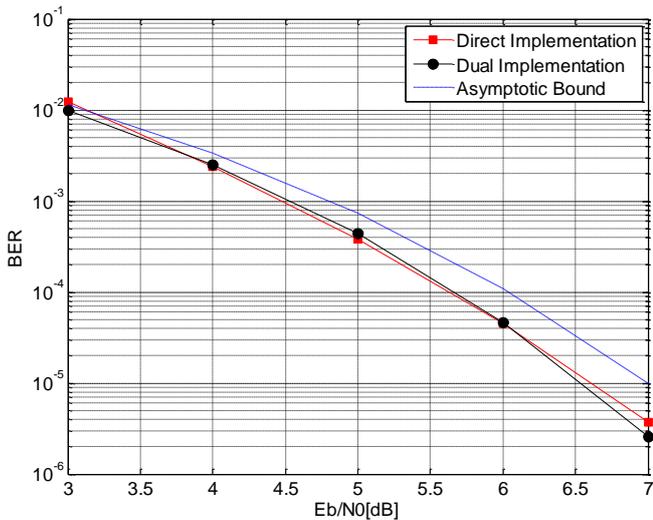


Figure 1. Simulated BER performance of the two-dimensional (64,49) product code defined over \mathbb{F}_4 with single parity-check codes as component codes, AWGN channel.

In Table I, the decoding complexity of the direct MAP algorithm is compared to the decoding complexity of the dual MAP algorithm for the (64,49) product code.

TABLE I. Complexity comparison between the direct MAP algorithm and the dual MAP algorithm for the (64,49) product code defined over \mathbb{F}_4

	<i>Direct MAP Algorithm</i>	<i>Dual MAP Algorithm</i>
<i>Number of Additions</i>	64×4^{50}	64×4^{16}
<i>Number of Multiplication</i>	4096×4^{50}	4096×4^{16}

V. CONCLUSION

In this paper, an optimum MAP decoding rule for non-binary codes based on the dual space of the code have been presented. Based on this proposed algorithm, the complexity of the MAP algorithm significantly reduces for the high-rate codes. The numerical results of the proposed algorithm have been given.

REFERENCES

- [1] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon-limit error-correcting coding and decoding: Turbo codes," in *Proc. 1993 IEEE Int. Communication Conf.* (Geneva, Switzerland, May 1993), pp.1046-1070.
- [2] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284-287, Mar. 1974.
- [3] R. Lucas, M. Bossart, and M. Breitbach, "On iterative soft decision decoding of linear binary block codes and product codes," *IEEE J. Select. Areas Commun.*, vol. 16, No. 2, pp.276-298, Feb. 1998.
- [4] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional code," *IEEE Trans. Inf. Theory*, vol. 42, No. 2, pp. 429-445, Mar. 1996.
- [5] R. J. McEliece, "On the BCJR trellis for linear block codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 4, pp. 1072-1092, July. 1996.
- [6] J. G. Proakis and M. Salehi, "Digital Communication", 5th ed., McGraw Hill, 2008.

Farzad Ghayour received the BSc and MSc degree in electronic engineering from the Isfahan University of technology, Isfahan, Iran in 2002 and 2006, respectively. He is currently working towards the PhD degree in electronic engineering at the University of KwaZulu-Natal. His research interests include information theory and error control coding for wireless communication systems.