

# Iterative Decoding of Concatenated Hadamard Codes<sup>1</sup>

Li Ping and Sammy Chan  
 Department of Electronics Engineering  
 City University of Hong Kong, Hong Kong  
 eeliping@cityu.edu.hk

## Abstract

We investigate the decoding technique and performance of concatenated Hadamard codes. Efficient soft-in-soft-out decoding algorithms based on the fast Hadamard transform are developed. Performance required by CDMA mobile or PCS speech services, e.g., BER=10<sup>-3</sup>, can be achieved at E<sub>b</sub>/N<sub>0</sub>=0.5dB using short interleaver length of 198 bits.

## I. Introduction

Recently it has been shown that concatenated convolutional and block codes employing iterative decoding technique can achieve remarkable error protection performance with modest decoding complexity [1-4]. The work along this direction has been particularly inspired by the discovery of the turbo codes [1]. Low rate, high performance concatenated codes have also been studied for CDMA mobile and PCS applications [5-7].

In this paper soft-in-soft-out MAP and Max-Log-MAP decoding algorithms based on the fast Hadamard transform (FHT) are developed and their applications in decoding low rate multi-dimensional concatenated Hadamard codes are examined. The proposed method has a notable advantage of achieving good performance with relatively short interleaver length. This is useful for some applications, such as mobile or PCS speech services where short frame length is mandatory [8] but relatively high BER (Bit Error Rate), e.g., 10<sup>-2</sup> ~ 10<sup>-3</sup>, is tolerable. (For a mixed speech and data network, extra coding and/or ARQ is normally used to protect data channels.) Simulation results show that the proposed method can meet the above requirement at E<sub>b</sub>/N<sub>0</sub> ≈ 0 dB using interleaving length of 196 bits and coding rate of 1/83.33.

## II. MAP and Max-Log-MAP decoding rules for the Hadamard codes

Consider a [2<sup>m</sup>, m+1, 2<sup>m-1</sup>] bi-orthogonal Hadamard code of elements {-1,+1}. The codeword set is {±h<sub>j</sub> : j=0,1,...,2<sup>m-1</sup>} with h<sub>j</sub> the j-th column of a Hadamard matrix **H**, Appendix A. Denote the transmitted codeword by **u**={u[i]} and the distorted vector by **x**=**u**+**n**, where **n**={n[i]} consists of independent Gaussian variables with zero mean and variance σ<sup>2</sup>. Construct a vector  $\tilde{\mathbf{L}}$  of *a priori* probability ratios conditioned on individual received symbols [3,4],

$$\begin{aligned} \tilde{L}[i] &= \log \left( \frac{\Pr\{u[i] = +1\}}{\Pr\{u[i] = -1\}} \right) \\ &= \log \left( \frac{\exp\left(\frac{-(x[i]-1)^2}{2\sigma^2}\right)}{\exp\left(\frac{-(x[i]+1)^2}{2\sigma^2}\right)} \right) = \frac{2x[i]}{\sigma^2} \end{aligned} \quad (1)$$

Assuming that all the codewords have independent and equal probability of occurrence, the output of the MAP decoder is given by [4,9,10],

$$L[i] = \log \frac{\sum_{H[i,j]=+1} \exp\left(\frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2}\right) + \sum_{H[i,j]=-1} \exp\left(\frac{-\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2}\right)}{\sum_{H[i,j]=-1} \exp\left(\frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2}\right) + \sum_{H[i,j]=+1} \exp\left(\frac{-\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2}\right)} \quad (2)$$

To reduce the computational cost, we can adopt the so-called Max-Log-MAP [11] rule using the dominating term to approximate each summation in (2),

$$\begin{aligned} L[i] \approx & \max \left\{ \max_{H[i,j]=+1} \left\{ \frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2} \right\}, \max_{H[i,j]=-1} \left\{ \frac{-\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2} \right\} \right\} \\ & - \max \left\{ \max_{H[i,j]=-1} \left\{ \frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2} \right\}, \max_{H[i,j]=+1} \left\{ \frac{-\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2} \right\} \right\} \end{aligned} \quad (3)$$

<sup>1</sup> This work is supported by the City University of Hong Kong Strategic Grant of number 7000703.

Efficient techniques for evaluating (2) and (3) are discussed in Appendix A and B.

### III. Concatenated Hadamard codes

Low rate, high gain coding schemes can be constructed using multi-dimensional concatenated Hadamard codes. We adopted the following concatenation method. Let  $\mathbf{D}$  be an  $I \times (m+1)$  array of information bits. Let  $\mathbf{D}_n = \pi_n(\mathbf{D})$  be an  $I \times (m+1)$  array obtained from  $\mathbf{D}$  using interleaving  $\pi_n$ , where  $n=1,2,\dots,N$ . Apply a systematic length- $2^m$  bi-orthogonal Hadamard code row-wise to every  $\mathbf{D}_n$ ,  $n=1,2,\dots,N$ , producing  $N$  parity check arrays  $\{\mathbf{P}_n : n=1,2,\dots,N\}$ , each with dimension  $I \times (2^m - m - 1)$ . The overall codeword is formed by  $\mathbf{D}\mathbf{P}_1\mathbf{P}_2 \dots \mathbf{P}_N$ . We call  $\mathbf{D}_n\mathbf{P}_n$  the  $n$ -th dimension. The coding rate is  $(m+1)/(m+1+N(2^m-m-1))$ . An example of this concatenation scheme for  $N=3$  is shown in Fig.1.

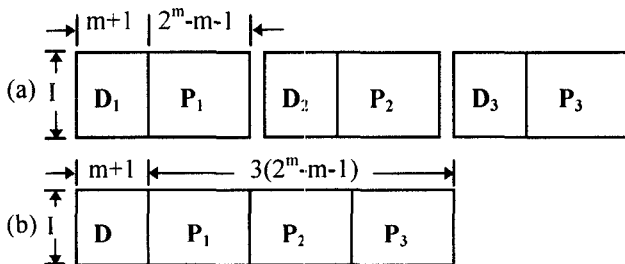


Fig.1 A three dimensional concatenation scheme. (a) The encoding process. (b) The overall codeword.

Discussions of the iterative turbo-type decoding structure for the multi-dimensional concatenated codes can be found in [3, 12].

### IV. Simulation results

Fig.2 contains the simulation results for a 3-dimensional concatenated length  $2^8=256$  bi-orthogonal Hadamard codes (C2). The interleaving length is  $N_d=9I=198$  with  $I=22$ . The rate is  $1/83.33$ . The performance of the standard [256, 128, 9] bi-orthogonal Hadamard code (C1) is included as reference. The interleavers used in constructing C2 are given by,  $D_n[i,j] = D[\alpha_n(i,j), \beta_n(i,j)]$  with  $\alpha_n(i,j) = (i + j \times J_n) \bmod 22$  and  $\beta_n(i,j) = (j + i \times I_n) \bmod 9$ . We have chosen  $\{J_n\} = \{0, 7, 17\}$  and  $\{I_n\} = \{0, 4, 7\}$  empirically.

It is seen that substantial performance improvement can be obtained by the concatenated

codes. With 10 iterations, C2 can achieve  $\text{BER}=10^{-3}$  at  $E_b/N_0=0.5$  dB using MAP decoding and at  $E_b/N_0=0.9$ dB using Max-Log-MAP decoding. This appears better than that of the turbo codes with similar frame lengths which can achieve  $\text{BER}=10^{-3}$  with  $E_b/N_0=1\sim 2$  dB [4-7]. For reducing SNR, the performance degradation of the proposed method is gradual until the channel capacity of  $-1.6$  dB. Even at  $E_b/N_0=-0.5$ dB, the MAP decoding still can achieve  $\text{BER}\approx 10^{-2}$ . This can be useful for preventing sudden channel drop in fading environments. For higher  $E_b/N_0$ , C2 has relatively lower rolling-off rate. Therefore it is more suitable for applications where relatively high BER can be tolerated, such as speech services. The performances of the MAP and Max-Log-MAP methods are close for high  $E_b/N_0$ . However, for very low  $E_b/N_0$ , the performance degrading of the MAP method is slower. Most coding gain is achievable within three iterations for both methods.

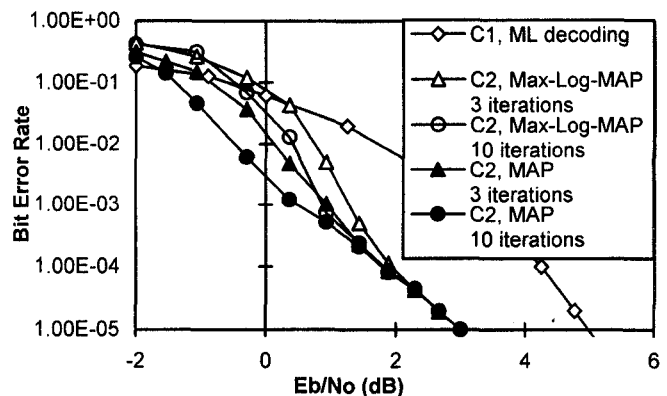


Fig.2. C1: Standard length-256 bi-orthogonal Hadamard code.

C2: 3-dimensional concatenated length-256 bi-orthogonal Hadamard code with interleaving length  $N_d=198$ .

### V. Conclusion

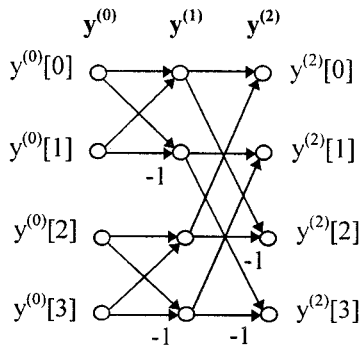
In traditional CDMA systems, the Hadamard code and the related Walsh and maximum length codes, are commonly used to construct spreading sequences. Although such codes can also provide certain coding gain, they are relatively weak when spreading ratio is less than 100. Therefore extra channel coding is usually adopted. The work in this paper suggests that it is possible to employ low-rate concatenated codes to provide both spreading and coding gains.

## Appendix A. Fast Hadamard Transform (FHT)

The required inner products in (2),  $\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle$ :  $j=0,1, \dots, 2^m-1$ , are the entries of  $\mathbf{H}^t \tilde{\mathbf{L}}$ . We adopt the following well known recursive construction of  $\mathbf{H}$ ,

$$\mathbf{H}^t = \begin{bmatrix} \mathbf{H} & \mathbf{H} \\ \mathbf{H} & -\mathbf{H} \end{bmatrix} \quad (4)$$

starting with a  $1 \times 1$  unity matrix. The resulting  $\mathbf{H}$  is symmetric and so  $\mathbf{H}^t \tilde{\mathbf{L}} = \mathbf{H} \tilde{\mathbf{L}}$ . This can be computed efficiently by the FHT [13]. As an example, a length-4 FHT in its flow graph form is given in Fig.3. There are  $m+1$  columns of nodes in the graph, representing  $m+1$   $2^m \times 1$  vectors,  $\{\mathbf{y}^{(m')}: m'=0,1,2, \dots, m\}$ . The overall transfer function is  $\mathbf{y}^{(m)} = \mathbf{H}\mathbf{y}^{(0)}$ . We will refer to the nodes representing  $\mathbf{y}^{(m')}$  as level  $m'$ . The cost of a length- $2^m$  FHT is  $m \times 2^m$  additions, compared with  $2^m \times (2^m - 1)$  additions for a direct matrix multiplication.

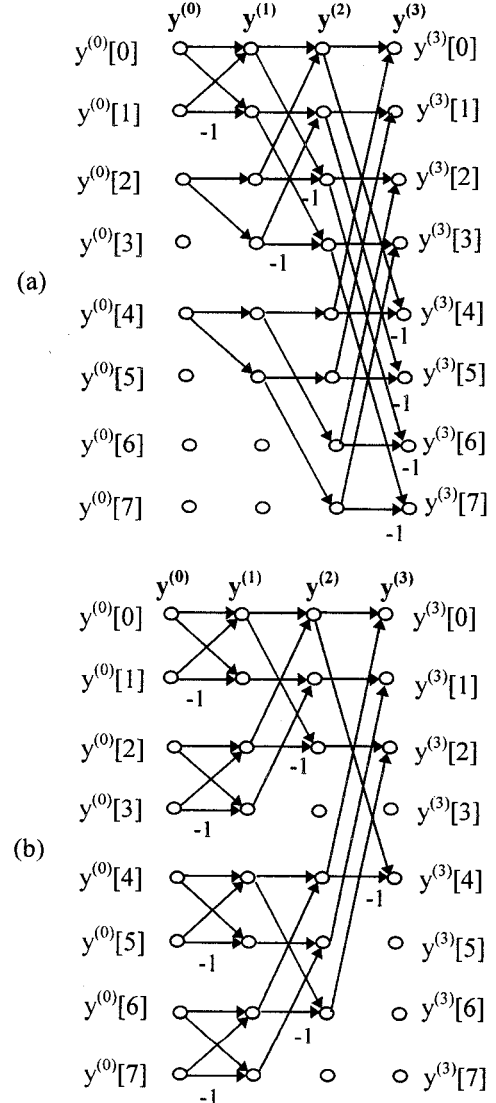


**Fig.3 A length-4 ( $m=2$ ) FHT flow graph for realizing  $\mathbf{y}^{(2)} = \mathbf{H}\mathbf{y}^{(0)}$ . Branches without labeling have unity gains.**

In a systematic encoding, the index set  $J_m = \{0, 1, 2, 4, \dots, 2^{m-1}\}$  can be chosen as information positions for a length- $2^m$  Hadamard code. We now consider the costs of two special cases of FHT, which will be used in the discussion in Appendix B. The first one, referred to as the reduced input FHT, is that all the parity entries of  $\mathbf{y}^{(0)}$ , i.e.,  $\{y^{(0)}[j] : j \notin J_m\}$ , are zeros. It can be verified that in a reduced input FHT, additions are only necessary for nodes  $\{y^{(m')}[0], y^{(m')}[1], \dots, y^{(m')}[2^{m'}-1]\}$  at level  $m'$ , Fig.4(a). The number of additions required is  $2^1 + 2^2 + \dots + 2^m = 2(2^m - 1) \approx 2 \times 2^m$ .

The second special case, referred to as the reduced output FHT, is that the outputs are only required at information positions. It can be verified

that in this case additions are only necessary for nodes  $\{y^{(m')}[k2^{m'}+j] \mid j \in J_m, k=0,1, \dots, 2^{m-m'}-1\}$  at level  $m'$ , Fig.4(b), where  $J_m$  is the set of the information positions of a length- $2^m$  Hadamard code. The total addition number required is  $2 \times 2^{m-1} + 3 \times 2^{m-2} + 4 \times 2^{m-3} + \dots + (m+1) \times 2^0 = 3(2^m - 1) - m \approx 3 \times 2^m$ .



**Fig.4. Flow graphs for special cases of FHT (a) Length-8 reduced input FHT. Branches delivering zero values are deleted. (b) Length-8 reduced output FHT. Branches not leading to the required outputs are deleted. The information positions are  $\{0, 1, 2, 4\}$ .**

## Appendix B. Efficient soft-in-soft-out algorithms for the Hadamard codes

This appendix concerns the computation involved in (2) and (3), assuming that the values of

$\{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle\}$  are available. It is convenient to adopt the concept of the compound vector and matrix. Denote by  $\mathbf{a}$  and  $\mathbf{b}$  two  $2^m \times 1$  compound vectors, whose entries are  $2 \times 1$  real blocks given below, respectively,

$$\mathbf{a}[j] = \begin{bmatrix} \exp\left(\frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2}\right) \\ \exp\left(-\frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2}\right) \end{bmatrix} \quad j=0,1, \dots, 2^m-1 \quad (5a)$$

$$\mathbf{b}[i] = \begin{bmatrix} \sum_{H[i,j]=+1} \exp\left(\frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2}\right) + \sum_{H[i,j]=-1} \exp\left(-\frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2}\right) \\ \sum_{H[i,j]=-1} \exp\left(\frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2}\right) + \sum_{H[i,j]=+1} \exp\left(-\frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2}\right) \end{bmatrix} \quad i=0,1, \dots, 2^m-1 \quad (5b)$$

Clearly  $\mathbf{b}$  contains all the summations in (2). The following relationship is a direct consequence of (5),

$$\mathbf{b}[i] = \begin{cases} \mathbf{I} \times \mathbf{a}[j] + \text{contributions of other entries of } \mathbf{a} & \text{if } H[i,j]=+1 \\ \hat{\mathbf{I}} \times \mathbf{a}[j] + \text{contributions of other entries of } \mathbf{a} & \text{if } H[i,j]=-1 \end{cases} \quad (6)$$

where matrixes  $\mathbf{I}$  and  $\hat{\mathbf{I}}$  are defined by,

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \hat{\mathbf{I}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (7)$$

We now construct a compound matrix  $\mathbf{Q}$  by substituting 1 and -1 in  $\mathbf{H}$  with  $\mathbf{I}$  and  $\hat{\mathbf{I}}$ , respectively. Then from (6),

$$\mathbf{b} = \mathbf{Q}\mathbf{a} \quad (8)$$

Examples of  $\mathbf{H}$  and  $\mathbf{Q}$  are ( $m=1$ ),

$$\mathbf{H} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & \hat{\mathbf{I}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \quad (9)$$

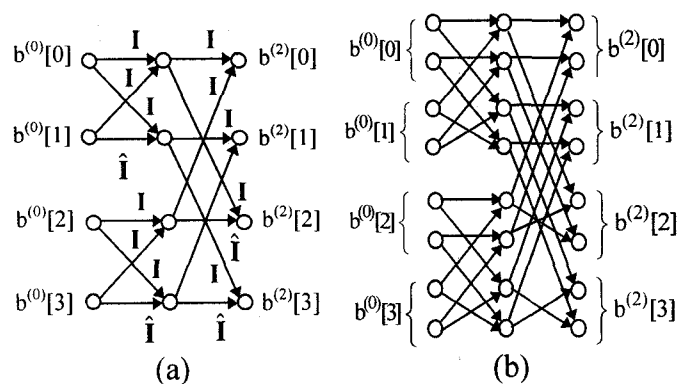
The evaluation of eqn.(8) would require  $2 \times 2^m \times (2^m - 1)$  additions in a straightforward manner but the following technique can reduce the cost to only  $2m \times 2^m$  additions.

**Remark:** Let  $G_H$  be a flow graph representing a length- $2^m$  FHT, Appendix A. Let  $G_M$  be obtained from  $G_H$  by applying the mapping  $\{1 \rightarrow \mathbf{I}, -1 \rightarrow \hat{\mathbf{I}}\}$  to the branch gains of  $G_H$  and redefining the nodes to represent  $2 \times 1$  blocks. Let the nodes of level  $m'$  represent a  $2^m \times 1$  compound vector  $\mathbf{b}^{(m')}$ ,

$m'=0,1,2,\dots,m$ . Then the overall transfer function of  $G_M$  is  $\mathbf{b}^{(m)} = \mathbf{Q}\mathbf{b}^{(0)}$ .

**Proof:** It can be shown that there is at most one forward path between any pairs of nodes in either  $G_M$  or  $G_H$ . Thus the transfer function between any two nodes, if it is not zero, is the product of the branch gains along the forward path between them. Since  $\hat{\mathbf{I}} \times \hat{\mathbf{I}} = \mathbf{I}$  and  $\hat{\mathbf{I}} \times \mathbf{I} = \hat{\mathbf{I}}$ , groups  $\{\hat{\mathbf{I}}, \mathbf{I}\}$  and  $\{-1, +1\}$  are isomorphic under multiplication. If a forward path in  $G_H$  has a gain of +1 (res. -1), the corresponding forward path in  $G_M$  must have a gain of  $\mathbf{I}$  (resp.  $\hat{\mathbf{I}}$ ). This one-to-one mapping is exactly the same as the mapping between  $\mathbf{H}$  and  $\mathbf{Q}$ . Hence if  $\mathbf{y}^{(m)} = \mathbf{H}\mathbf{y}^{(0)}$  in  $G_H$ , then  $\mathbf{b}^{(m)} = \mathbf{Q}\mathbf{b}^{(0)}$  in  $G_M$ .

We will refer to the algorithm represented by  $G_M$  as the MAP FHT. An example of  $G_M$  in its direct form is shown in Fig.5(a) for  $m=2$ . Every node in Fig. 5(a) can be split into two, representing two real variables. This leads to the expanded form in Fig.5(b), in which the number of nodes, and so the number of additions, double those of the corresponding  $G_H$ , respectively. A complete MAP FHT thus costs  $2m \times 2^m$  additions. Similar to the discussions in Appendix A, we can define the reduced output MAP FHT, costing  $6(2^m - 1) - 2m \approx 6 \times 2^m$  additions, which will be used below.



**Fig.5 An example of a length-4 MAP FHT.**

(a) Direct form where every node represents a  $2 \times 1$  block.

(b) Expanded form. Every node represents a real number and every branch has a unity gain.

For the Max-Log-MAP rule (3), let  $G_L$  be obtained from the expanded form of  $G_M$  by replacing additions at all the nodes with maximizations. Every

entry of  $\mathbf{b}^{(m)}$  should be the maximum term among those in the summations in (5b). Redefine the input vector as,

$$b^{(0)}[j] = \begin{bmatrix} \frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2} \\ -\frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2} \end{bmatrix} \quad j=0,1,\dots, 2^m-1 \quad (10)$$

Then the output vector should be,

$$b^{(m)}[j] = \begin{bmatrix} \max \left\{ \max_{H[i,j]=+1} \left\{ \frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2} \right\}, \max_{H[i,j]=-1} \left\{ \frac{-\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2} \right\} \right\} \\ \max \left\{ \max_{H[i,j]=-1} \left\{ \frac{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2} \right\}, \max_{H[i,j]=+1} \left\{ \frac{-\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle}{2} \right\} \right\} \end{bmatrix} \quad i=0,1, \dots, 2^m-1 \quad (11)$$

which is the wanted output for (3).

Finally we briefly estimate the computational costs involved in evaluating (2) and (3). The required values of  $\{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle\}$  can be generated together in

$\mathbf{H}\tilde{\mathbf{L}}$  using an FHT with  $m \times 2^m$  additions. This cost can be further reduced in a turbo-type iterative decoding process after the first iteration. Let  $\tilde{\mathbf{L}}_D$  and  $\tilde{\mathbf{L}}_P$  represent the contributions of the information and parity check bits in  $\tilde{\mathbf{L}}$ , so that their entries are non-zero only at information and parity check positions respectively. Now,

$$\mathbf{H}\tilde{\mathbf{L}} = \mathbf{H}\tilde{\mathbf{L}}_D + \mathbf{H}\tilde{\mathbf{L}}_P \quad (12)$$

In a turbo-type decoding,  $\tilde{\mathbf{L}}_P$  is not updated [1,3,4,12]. The computation of  $\mathbf{H}\tilde{\mathbf{L}}_D$  costs about  $2 \times 2^m$  additions by a reduced input FHT (Appendix A) with another  $2^m$  additions to complete (12) using stored  $\mathbf{H}\tilde{\mathbf{L}}_P$ . Thus (12) involves only about  $3 \times 2^m$  additions except for the first iteration. Whence  $\{\langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle\}$  are available, computing  $\exp(\pm \langle \mathbf{h}_j, \tilde{\mathbf{L}} \rangle / 2)$  requires  $2 \times 2^m$  exponential functions. Then the evaluation of the summations in (2) can be accomplished with about  $6 \times 2^m$  additions by a reduced output MAP FHT.

For the Max-Log-MAP algorithm,  $\mathbf{H}\tilde{\mathbf{L}}$  is evaluated as in the MAP algorithm. The exponential and logarithm functions are not necessary. The additions in evaluating the summations in (2) are replaced by the comparisons in (3), and the divisions by subtractions. The total cost is approximately  $9 \times 2^m$  addition equivalent operations except for the first iteration. This complexity is linear with respect to the codeword length.

## References

- [1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes", Proc. IEEE ICC-93, pp. 1064-1070, 1993.
- [2] P. Robertson, "Improving decoder and code structure of parallel concatenated recursive systematic (turbo) codes", Proc ICUPC'94, pp.183-187, 1994.
- [3] D. Divsalar and F. Pollara, "Multiple turbo codes", Proc IEEE Milcom-95, pp.279-285, 1995.
- [4] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes", IEEE Trans. Inform. Theory, vol. IT-42, pp.429-445, March 1996.
- [5] D. Divsalar and F. Pollara, "Turbo codes for PCS applications", Proc. IEEE ICC-95, pp.54-59, 1995.
- [6] P. Komulainen and K. Pehkonen, "A low-complexity superorthogonal turbo-code for CDMA applications", Proc. IEEE PIMRC'96, pp.369-373, 1996.
- [7] J. Hagenauer "Forward error correcting for CDMA systems", Proc. IEEE 4<sup>th</sup> Intern. symp. on Spread Spectrum Techniques and Applications. pp.566-569, 1996.
- [8] P. Jung, "Comparison of turbo-code decoders applied to short frame transmission systems", IEEE J. Selected Areas in Communications, vol-14, pp. 530-537, April 1996.
- [9] R. Herzog, J. Hagenauer and A. Schmidbauer, "Soft-in/soft-out Hadamard despreader for iterative decoding in the IS-95(A) system", Proc IEEE VTC-97, pp.1219-1222, 1997.
- [10] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate", IEEE Trans. Inform. Theory, IT-20, pp.284-287, March 1974.
- [11] P. Robertson, E. Villebrun and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain", Proc. IEEE ICC-95, pp. 1009-1013, 1995.
- [12] Li Ping, S. Chan and K.L. Yeung, "Iterative decoding of multidimensional concatenated single parity check codes", in this Proceedings.
- [13] M. Kunt, "On computation of the Hadamard transform and the R transform in ordered form", IEEE Trans. Comput., vol. C-24, pp.1120-1121, Nov. 1975.