

Unit-Memory Hamming Turbo Codes

Jung-Fu Cheng

Robert J. McEliece

Electrical Engineering Dept., California Institute of Technology, Pasadena, CA 91125, USA

I. INTRODUCTION

Several parallel concatenated coding schemes (turbo codes) based on multi-memory (MM) convolutional codes (more specifically, a (2, 1, 4, 7) code) were recently proposed to achieve near Shannon-limit error correction performance with reasonable decoding complexity [1]-[3]. On the other hand, in many cases of interest, unit-memory (UM) codes have been demonstrated to have larger free distances than the MM codes with the same rate and the same number of memory elements [4]. In this paper, new turbo codes based on the (8, 4, 3, 8) UM Hamming code [4] will be developed and shown to possess better performance potential in some senses. The standard turbo decoding algorithms, however, do not appear to achieve this potential.

II. ENCODER

An equivalent systematic recursive generator matrix for the UM Hamming code can be obtained by first properly permuting the columns and then multiplying on the left by the inverse of the left-most 4×4 sub-matrix of the original generator matrix:

$$G = [I|P] = \begin{bmatrix} 1 & 0 & 0 & 0 & \frac{1}{1+D} & 1 & \frac{D}{1+D} & 1 \\ 0 & 1 & 0 & 0 & 1 & \frac{1}{1+D} & 1 & \frac{D}{1+D} \\ 0 & 0 & 1 & 0 & \frac{D}{1+D} & 1 & 1 & \frac{1}{1+D} \\ 0 & 0 & 0 & 1 & 1 & \frac{D}{1+D} & \frac{1}{1+D} & 1 \end{bmatrix}$$

The corresponding encoder can be implemented with three memory elements. The encoder for the UM turbo (UMT) code is similar to those for the MMT codes [1]-[3], except that there are multiple inputs to the encoder of the component codes. The trellis is terminated using the method of [3]. Since the systematic bits from the second encoder are discarded, the overall code rate is $K/3(K+4)$, where K is the interleaver size.

III. THE MAP ALGORITHM FOR MULTI-INPUT RECURSIVE TRELLIS CODES

In this section, a modified MAP algorithm is presented to deal with multiple inputs. Let the state of the encoder for the (n, k, ν) code at time t be $S_t \in \{0, 1, \dots, 2^\nu - 1\}$, for $t = 0, \dots, L = K/k$, where the initial and final states, S_0 and S_L , are known. The input block $\mathbf{u}_t = (u_{t,1}, \dots, u_{t,k})$ causes a transition from S_{t-1} to S_t , and the corresponding output codeword $\mathbf{x}_t = (x_{t,1}, \dots, x_{t,n})$ is observed over an AWGN channel as $\mathbf{y}_t = (y_{t,1}, \dots, y_{t,n})$, for $t = 1, \dots, L$. The log likelihood ratios of the *a posteriori* probabilities can be computed as:

$$\Lambda(u_{t,j}) = \log \frac{\sum_s \sum_{s'} \gamma_{t,j}^{+1}(s', s) \alpha_{t-1}(s') \beta_t(s)}{\sum_s \sum_{s'} \gamma_{t,j}^{-1}(s', s) \alpha_{t-1}(s') \beta_t(s)}$$

$$\alpha_t(s) = \frac{\sum_{s'} \Gamma_t(s', s) \alpha_{t-1}(s')}{\sum_s \sum_{s'} \Gamma_t(s', s) \alpha_{t-1}(s')}, \quad \text{for } t = 1, \dots, L$$

$$\beta_t(s) = \frac{\sum_{s'} \Gamma_{t+1}(s, s') \beta_{t+1}(s')}{\sum_s \sum_{s'} \Gamma_{t+1}(s, s') \alpha_t(s')}, \quad \text{for } t = L-1, \dots, 0$$

where, if the transition $s' \rightarrow s$ is allowed by input $u_{t,j} = i$,

$$\gamma_{t,j}^i(s', s) = \Pr \{u_{t,j} = i\} \Pr \{\mathbf{y}_t | S_t = s, u_{t,j} = i, S_{t-1} = s'\}$$

$$\Gamma_t(s', s) = \sum_{\mathbf{i}: s' \rightarrow s} \Pr \{\mathbf{u}_t = \mathbf{i}\} \Pr \{\mathbf{y}_t | S_t = s, \mathbf{u}_t = \mathbf{i}, S_{t-1} = s'\}$$

IV. DECODER AND PERFORMANCE

The decoder structure used is similar to that in [2] except that the MAP algorithm in III is applied instead. Numerical results are shown in Fig. 1 and summarized as follows:

- The minimum distance of the (60, 16) UMT code with the best known interleaver is 14. Maximum-likelihood decoding simulation of this code shows a gain of 0.5 dB over the (80, 16) MMT code [3] which has the same minimum distance. The use of turbo decoding introduces a loss of about 1.5 dB.
- For large block lengths, simulation results show that the turbo decoding algorithm converges faster than that for MMT codes, but the performance is not as good. Comparing these with the transfer bounds computed with a double recursion method and a random averaging argument [5], a gap of coding gain with turbo decoding as in the previous case can be observed again.

REFERENCES

- [1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," *Proc. IEEE Int'l Conf. Comm.*'93.
- [2] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," *Proc. IEEE Global Telecomm. Conf.*'94.
- [3] D. Divsalar and F. Pollara, "Turbo Codes for Deep-Space Communications," *JPL TDA Progress Report 42-120*, Feb. 1995.
- [4] K. Abdel-Ghaffar, R. J. McEliece and G. Solomon, "Some partial-unit-memory convolutional codes," *JPL TDA Progress Report 42-107*, Nov. 1991.
- [5] S. Benedetto and G. Montorsi, "Performance evaluation of turbo-codes," *IEE Elec. Letters*, Feb. 1995.

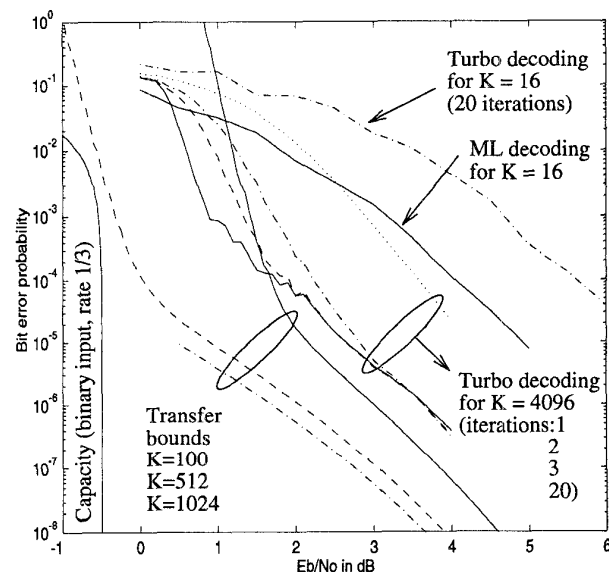


Figure 1: Performance of unit-memory turbo codes.