Decoding Turbo Gallager Codes for the Erasure Channel

L.Yang, M.Ambroze and M.Tomlinson
Fixed and Mobile Communications Research
University of Plymouth
PL4 8AA, United Kingdom
Email: li.yang, marcel.ambroze, martin.tomlinson@plymouth.ac.uk

***Abstract*—Two new decoding schemes for Turbo Gallager codes transmitting over the binary erasure channel are presented. The first decoder uses an optimised iterative decoder based on a Look-Up Table improving on an earlier published result. The second decoder arrangement uses the first decoder followed by an "In-Place" matrix inversion algorithm designed to free the stopping sets. Results are presented which show that the latter arrangement achieves maximum likelihood performance. The complexity and performance of both decoders is analysed and evaluated.**

I. Introduction

The first iterative decoding algorithm for LDPC codes over the Binary Erasure Channel (BEC)[1] was proposed by Luby [2], who showed that this scheme approaches channel capacity arbitrarily closely. Finite-length analysis of LDPC codes over the BEC was described in [3]. Turbo codes also have near-capacity performance with low decoding complexity since their introduction by Berrou [4]. It is known that a code with a minimum Hamming distance of d can correct up to $d-1$ erasures and Rosnes showed that turbo codes are easy to design for a given minimum Hamming distance due to the existence of efficient weight spectrum algorithms [5]. He also showed that stopping sets exist for turbo codes and they characterise exactly the performance of turbo decoding on the BEC[6]. However asymptotic results [7], [8] indicate that LDPC codes could potentially have lower error floors than turbo codes. With this rationale, Colavolpe [9] proposed a new scheme that combines the advantages of turbo codes with those of LDPC codes, now known as Turbo Gallager codes (TGC).

This letter proposes two new decoding schemes for Turbo Gallager codes over BEC. The first decoder uses an optimised iterative decoder based on a Look-Up Table (LUT). It is shown that the LUT decoder described in [10] for convolutional codes may be simplified and applied to Turbo Gallager codes. Since this decoder like all iterative decoders suffers from stopping sets, a second decoding scheme referred to as the "Hybrid decoder" is proposed. This combines the LUT decoder with an "In-Place" matrix inversion algorithm [11] to free the stopping sets. The complexity and performance of both decoders is analysed and evaluated.

II. Optimised Iterative LUT Decoder

The first table look-up based approach for decoding based on the syndrome trellis was introduced by Schalkwijk [12] for convolutional codes over binary symmetric channel. A fast LUT decoding algorithm working on the decoding trellis was described in [10] for convolutional codes over erasure channel. The fast LUT decoding algorithm uses three LUTs to represent the forward recursion α_i, backward recursion β_i and output probability function, and is a simplification in complexity compared to the standard MAP decoder. However the LUT decoding algorithm may be further simplified. It is shown in Appendix A that the non-zero values of α or β are always identical. Consequently the values of α_i and β_i may be represented by a single bit with a "1" representing a non-zero value state and a "0" representing a zero value state. Thus two binary vectors can be used to represent the values of α and β at each trellis section and a more efficient decoding arrangement may be realised by constructing a Look-Up table, which includes two vectors of binary numbers to represent the final transitions, and one vector to represent the conditions of received bits. In the decoding process, the trellis section containing erased information is converted to obtain the final transition binary numbers. By looking up the table, it is possible to obtain directly the corrected information bits, or decoding failure.

Complexity Analysis: Let n_e be the number of data erasures at the input to the decoder, where $\bar{n_e} = (k \cdot \epsilon)$, k is the length of information bits and ϵ is the erasure probability. With each iteration of the iterative decoder, the number of erasures is reduced. Let n_e^i be the number of erasures at iteration i, and n_s is the number of states in the trellis, N_o represents the total number of operations, n_i is the number of iterations. For the standard turbo decoder, we consider the number of computations including the probability assignment, α, β and γ recursion computations. Thus the number of operations $N_o^{I_S}$ for one block with n_i iterations is obtained as:

$$N_o^{I_S}(k) = \sum_{i=1}^{n_i}(3k + 16kn_s) \tag{1}$$

For the proposed LUT decoder, the number of operations $N_o^{I_O}$ for one block with n_i iterations excluding the Look-Up table construction is :

$$N_o^{I_O}(n_e) = \sum_{i=1}^{n_i}(8kn_s + 8n_e^i n_s) \tag{2}$$

The comparison between the complexity of the standard turbo

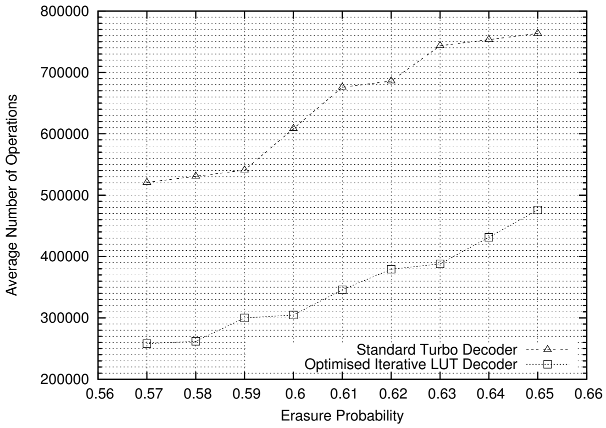

Fig. 1. Complexity Comparison between Standard Turbo and Optimised Iterative LUT Decoder

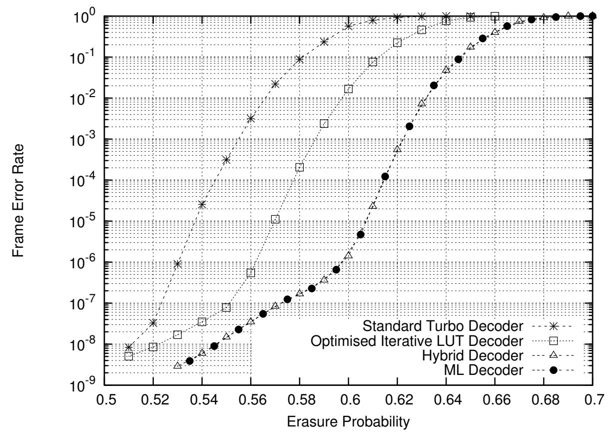

Fig. 2. Results of Turbo Gallager Codes (15/13) (1536,512)

TABLE I
NUMBER OF OPERATIONS FOR DECODED AND UNDECODED BLOCKS

Blocks	Number of Operations N_o
Decoded $N_o^D(n_e)$	N_o^{IO}
Undecoded $N_o^U(n_e)$	$N_o^D(n_e) + \frac{n(n_e^{n_i}-1)(2k-n_e^{n_i})}{64} + \frac{n(m-1)n_e^{n_i}}{32}$

decoder and the optimised iterative LUT decoder is shown in Fig. 1, and the reduction in complexity is apparent.

III. Hybrid Decoder

The idea of the hybrid decoder is to allow the iterative LUT decoder to decode the received codewords until either correct decoding is achieved or a stopping set is reached, in which case, the unsolved erasure pattern is passed to the "In-Place" decoder to attempt to free the stopping sets.

By considering the structure of Turbo Gallager codes and the usage of 32-bit integer as the storage media, the number of operations N_o^{IP} for the "In-Place" decoder is:

$$N_o^{IP}(n_e) = \frac{n(n_e-1)(2k-n_e)}{64} + \frac{n(m-1)n_e}{32} \quad (3)$$

where m is the memory of the component code shift register. Since the hybrid decoder includes the LUT decoder, the entire complexity is equal to the LUT decoder plus the complexity of the "In-Place" algorithm to solve the stopping sets, when stopping sets exist. The complexity of hybrid decoding is given in Table I, where the blocks decoded correctly by the iterative decoder are denoted as "Decoded", and the blocks which still contains erasures after LUT decoder are denoted as "Undecoded", and $n_e^{n_i}$ is the number of erasures left after LUT decoder.

IV. Results

The results for the TGC (15/13) (1536,512) code, with the standard iterative decoder and optimised iterative LUT decoder, hybrid decoder, and ML decoder are shown in Fig. 2. From the results, it can be seen that at any probability of erasure ϵ, the hybrid decoder achieves the same performance as the ideal, ML decoder. As ϵ is reduced, the performance of optimised iterative LUT decoder gradually reaches the ML decoder performance. As is well known, the ML decoder for the erasure channel is practically realisable but with complexity proportional to n_e^3. A recent example is described in [11]. A comparison in complexity between the ML decoder and the hybrid decoder is shown in Fig. 3. The complexity benefits of the hybrid decoder compared to ML decoding are a function of ϵ and the TGC parameters. For values of ϵ below 0.6, the complexity is similar to the iterative LUT decoder. For values of ϵ above 0.65, the complexity of the hybrid decoder is approaching that of full ML decoding.

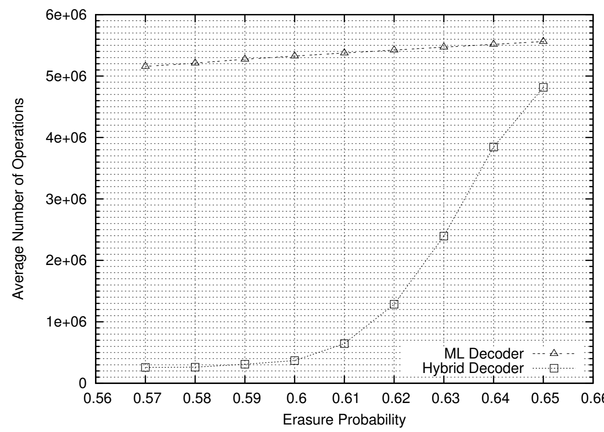

Fig. 3. Complexity Comparison between ML and Hybrid Decoder

V. Conclusion

Two new decoders using Turbo Gallager codes have been proposed for the erasure channel. The optimised LUT iterative decoder is shown to have better performance than the standard iterative Turbo decoder and to have significantly reduced complexity. Whilst neither of these low complexity decoders can equal ML decoding, it is shown that the proposed hybrid decoder does match the ML decoder in performance and at reduced complexity.

Appendix A
Proof that α and β non zero values are always equal for the erasure channel

The probability of a decoded bit d_t is computed as the sum of the probability of all transitions that are generated by $d_t = i$.

TABLE II
TRANSITIONS AND STATES RELATIONSHIP FOR S_{Start}=2^n, $n \in \mathbf{N}$

Data/ Parity	No. of End States	No. of Transitions	No. of Trans End State
d/d	$2^n \rightarrow 2^{n-1}$/ $2^n(2^{n+1})$	$2^{n-1}(2^{n+1})$	1
d/e	$2^n \rightarrow 2^n$	2^n	1
e/d	$2^n \rightarrow 2^n$	2^n	1
e/e	$2^n \rightarrow 2^n/2^{n+1}(2^{n+1})$	2^{n+1}	2/1(2^{n+1})

According to

$$P(d_t = i) = \frac{1}{P(\mathbf{R}_1^k)} \sum_{m',m|d_t(m',m)=i} \lambda_t(m', m) \qquad (4)$$

where

$$\lambda_t(m', m) = P(S_{t-1} = m', S_t = m, \mathbf{R}_1^k) \qquad (5)$$

is the joint probability of $S_t = m$ and $S_{t-1} = m'$. By eliminating the constant $P(\mathbf{R}_1^k)$, the joint probability is described as:

$$\lambda_t(m', m) = \alpha_{t-1}(m')\gamma_t(m', m)\beta_t(m) \qquad (6)$$

and based on the Bayes' rules, $\alpha_t(m)$ and $\beta_t(m)$ are obtained as:

$$\alpha_t(m) = \sum_{m'} \alpha_{t-1}(m') \cdot \gamma_t(m', m) \qquad (7)$$

$$\beta_t(m) = \sum_{m'} \beta_{t+1}(m') \cdot \gamma_{t+1}(m, m') \qquad (8)$$

For each possible transition, since $R_t = \{x_t, y_t\}$, and $x_t = i$, $y_t = i$ are mutually exclusive, the transition probability $\gamma_t(m', m)$ at time t for $d_t(m', m)$ is:

$$\begin{aligned} \gamma_t(m', m) \quad = \quad & P(x_t = i|S_t = m, S_{t-1} = m') \cdot \\ & P(y_t = i|S_t = m, S_{t-1} = m') \end{aligned} \qquad (9)$$

For $\lambda_t(m', m)$ to be non-zero in (6), the three terms of $\alpha_{t-1}(m')$, $\beta_{t+1}(m')$ and $\gamma_t(m, m')$ must be non-zero. From (7), it is clear that $\alpha_t(m)$ only depends on the value $\gamma_t(m', m)$, if $\alpha_{t-1}(m')$ is non-zero. Similarly, from (8), $\beta_t(m)$ only depends on the value $\gamma_{t+1}(m, m')$, if $\beta_{t+1}(m')$ is non-zero. In order to make sure $\gamma_t(m, m')$ is non-zero, firstly the transition should be possible, which means $P(S_t = m|S_{t-1} = m') = 1$. In addition, the probabilities for both data and parity bit over the transition should be non-zero for $d_t(m', m) = i$, which means $P(x_t = i|S_t = m, S_{t-1} = m') \neq 0$ and $P(y_t = i|S_t = m, S_{t-1} = m') \neq 0$. The relationship between transitions and states are shown in Table II, where S_{Start} is the number of start states, "d" represents a known bit and "e" an erasure, **N** is the set of nature number.

For all conditions of d/d, d/e, e/d and e/e, since $\alpha_{t-1}(m')$ or $\beta_{t+1}(m')$ are known as equal, the values of $\alpha_t(m)$ or $\beta_t(m)$ only depend on the product of $P(x_t = i)$ and $P(y_t = i)$ for $d_t(m', m) = i$ which are the transition probabilities. All possible transition probabilities are shown in Table III. For all conditions of the received data and parity bits, the transition probabilities take only the values of "0" or the same nonzero value which is from the set of "$\{1, 0.5, 0.25\}$". According to (7) and (8), $\alpha_t(m)$ or $\beta_t(m)$ if non zero , will be computed with the same value. Hence during the α or β recursions, the non zero values of α or β, at any condition, are equal at any given trellis section.

TABLE III
PRODUCT OF POSSIBLE TRANSITION PROBABILITIES

Data/Parity	Data $(x_t = i)$	Parity $(y_t = i)$	Transition Probability
d/d	0,1	0,1	0,1
d/e	1,0	0.5	0,0.5
e/d	0.5	0,1	0,0.5
e/e	0.5	0.5	0,0.25

REFERENCES

[1] P.Elias, "Coding for two noisy channels," in *Proc. 3rd London Symposium on Information Theory*, London, England, 1955, pp. 61–76, Academic Press, New York.
[2] M. Shokrollahi M. Luby, M. Mitzenmacher and D. Spielman, "Efficient erasure correcting codes," *IEEEIT*, vol. 47, Feb 2001.
[3] I.E. Telatar C. Di, D. Proietti, T. Richardson, and R. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Transactions on Information Theory*, vol. 48, pp. 1570–1579, Jun 2002.
[4] A. Glavieux C. Berrou and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE International Conference on Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.
[5] E. Rosnes and O. Ytrehus, "Improved algorithms for the determination of turbo-code weight distributions," *IEEECOMMS*, 2005.
[6] E. Rosnes and O. Ytrehus, "Turbo decoding on the binary erasure channel: Finite-length analysis and turbo stopping sets," in *IEEEISIT*, Feb 2006.
[7] A. Pennoti and S. Benedetto, "A new upper bound on the minimum distance of turbo codes," *IEEE Transactions on Information Theory*, vol. 50, no. 12, Dec 2004.
[8] R.G. Gallager, *Low-Density Parity-Check Codes*, Cambridge, MA: MIT Press, 1963.
[9] G.Colavolpe, "Design and performance of turbo gallager codes," *IEEECOMMS*, vol. 52, no. 11, Nov 2004.
[10] P.H. Siegel B.M. Kurkoski and J.K. Wolf, "Exact prbability of erasure and a decoding algorithm for convolutional codes on the binary erasure channel," in *Proc. IEEE GLOBECOM*, San Francisco, CA, Dec 2003, vol. 3, pp. 1741–1745.
[11] M. Tomlinson M. Ambroze J. Cai, C. Tjhai and M.Z. Ahmed, "An efficient solution to packet loss: Erasure correcting codes," in *Proc. 4th IASTED International Conference Communication Systems and Networks*, Spain, 2005, pp. 224–229, ACTA Press.
[12] J.P.M.Schalkwijk and A.J. Vinck, "Syndrome decoding of convolutional codes," *IEEECOMMS*, vol. 23, Jul 1975.