Turbo Decoding of Product Codes based on the Modified Adaptive Belief Propagation Algorithm

Christophe Jego
GET/ENST Bretagne
CNRS TAMCIC UMR2872
29238 Brest, France
Christophe.Jego@enst-bretagne.fr

Warren J. Gross
Department of Electrical and Computer Engineering
McGill University
Montreal, Quebec, H3A 2A7, Canada
Warren.Gross@ece.mcgill.ca

***Abstract*—This paper introduces the Modified Adaptive Belief Propagation (m-ABP) algorithm, an innovative method for the turbo decoding of product codes based on BCH component codes. The Adaptive Belief Propagation algorithm of Jiang and Narayanan is simplified by moving the matrix adaptation step outside of the iteration loop, significantly reducing the complexity. Performance in terms of the bit-error-rate (BER) of the novel turbo decoding algorithm is given. Simulation results for the turbo decoding of product codes show that compared to the Chase-Pyndiah algorithm no significant BER deviation is observed while the highly-parallelizable graph-based structure of the algorithm enables high-throughput decoding.**

***Index Terms*— iterative decoding of product codes, Adaptive Belief Propagation, BCH codes, parity check matrix.**

I. Introduction

Iterative Soft-Input Soft-Output (SISO) decoding, first proposed by C. Berrou et al. [1], performs very close to the Shannon limit. The general concept of the iterative SISO (turbo) decoding of concatenated convolutional codes has been extended to product codes and LDPC codes. In all cases, a SISO decoding process both receives and produces soft decisions. Iterative decoding based on Belief Propagation (BP) [2] is a SISO decoding algorithm for linear block codes that is based on the propagation of soft information along the edges of a graph defined by the parity check matrix associated with the code. This decoding algorithm has received great interest recently due to the rediscovery of low density parity check (LDPC) codes, a class of linear block codes defined by a sparse parity check matrix. The BP algorithm is considered to be the reference LDPC decoding algorithm and it exhibits a high degree of parallelism making it suitable for high data rate applications. It is commonly believed that BP algorithm is not suitable if the parity check matrix of the linear block code is not sparse. Traditional linear block codes such as BCH and Reed-Solomon codes do not have a sparse parity check matrix. For this reason, the BP algorithm has not been used for the decoding of BCH or Reed-Solomon codes.

Another soft decision decoding algorithm called the Chase-Pyndiah algorithm [3] is usually used for turbo decoding of product codes composed of BCH or Reed-Solomon component codes. In Chase-Pyndiah decoding, soft input information is used to find a set of test vectors, each of which is applied to a hard-decision algebraic decoder. The soft outputs are estimates of the log-likelihood ratios (LLR) of the binary decisions given during the algebraic decoding. The iterative decoding algorithm for a product code involves performing the successive decoding of all rows and all columns (two half-iterations). To increase the data rate, it is possible to perform all the rows (columns) in parallel during a half-iteration. In [4], an innovative architecture that enables the memory blocks between each half-iteration to be removed was proposed. The major limitation of the Chase-Pyndiah algorithm is that it does not allow a high degree of parallelism which does not make it suitable for high data rate applications. Therefore, the motivation of this paper is to find a BP-based algorithm to be applied to linear block codes with a non-sparse matrix for use in turbo product decoders.

Recently, however, the Adaptive Belief Propagation (ABP) algorithm was proposed for the decoding of Reed-Solomon codes [5]. This method adapts the parity check matrix at each iteration of the BP algorithm according to the bit reliabilities in order to sparsify the columns of the parity check matrix associated with the unreliable bits. The ABP algorithm is especially interesting for decoding linear block codes whose parity check matrix is not sparse, for example the component BCH codes of turbo product codes. The graph-based message passing step of the ABP algorithm is highly parallelizable, unlike the Chase-Pyndiah algorithm. However, due to the adaptation step that performs the sparsification at every iteration, the complexity of ABP becomes prohibitive for hardware implementation. In this paper we introduce a modified version of the ABP algorithm for BCH turbo product codes that significantly lowers the decoder complexity.

1-4244-1429-6/07/$25.00 ©2007 IEEE

First, we recall the basic principles of decoding of product codes: their construction and the turbo process. Then, the modified ABP algorithm dedicated to the turbo decoding of product codes is detailed in Section III. In Section IV, the bit error rate performance of the novel algorithm is given and a comparison with block turbo decoding based on the Chase-Pyndiah algorithm is done.

II. Turbo Decoding of Product Codes

A. Product codes

The concept of product codes is a simple and efficient method to construct powerful codes with a large minimum Hamming distance, δ, using conventional linear block codes [6]. Let us consider a systematic linear block code C having parameters (n,k) as illustrated in Figure 1. The parameters n and k stand for the code length and the number of information symbols respectively. The product code $P=C\times C$ is obtained by placing k^2 information bits in a matrix of k rows and k columns and encoding the k rows and k columns using the code C. It can be shown that all n rows and all n columns are codewords of C. Furthermore, the parameters of the resulting product code P are given by $n_p=n^2$, $k_p=k^2$ and the code rate R_p is given by $R_p=R^2$. Thus, it is possible to construct powerful product codes based on linear block codes such as BCH codes. BCH codes are a class of linear cyclic block codes that have capabilities for multiple error detection and correction.

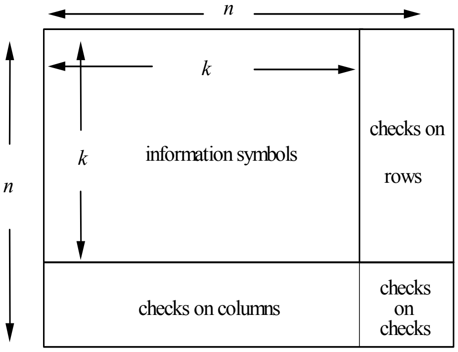

Fig. 1. Principle of product codes.

B. Iterative decoding of product codes

Product code decoding involves sequentially decoding rows and columns using a SISO decoding algorithm. The block turbo decoding process repeats this soft decoding for several iterations. Each decoding process computes soft information $\mathbf{R'}_{(it+1)}$ from the channel received information $\mathbf{R}$ and the information computed in the previous half-iteration, $\mathbf{R'}_{(it)}$, as shown in Figure 2. The extrinsic information $\mathbf{W}_{(it)}$ is obtained by subtracting the soft input information $\mathbf{R'}_{(it)}$ from the soft output information $\mathbf{F}_{(it)}$. The soft information $\mathbf{R'}_{(it+1)}$ is given by $\mathbf{R'}_{(it+1)} = \mathbf{R} + \alpha_{(it)}\mathbf{W}_{(it)}$ where $\alpha_{(it)}$ is a scaling factor that reduces the effect of the extrinsic information in the soft decoder during the first decoding steps.

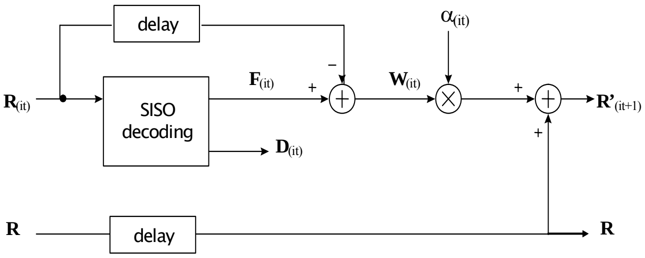

Fig. 2. The iterative block turbo decoding principle.

In 1972, Chase [7] proposed algorithms that approximate the optimum Maximum-Likelihood (ML) decoding of block codes with low computing complexity and small performance degradation. In 1994, R. Pyndiah et al. [8] presented a new iterative decoding algorithm for decoding product codes, based on the iterative SISO decoding of concatenated block codes. The Chase-Pyndiah algorithm is concisely summarized below:

1-) Search for the L_r least reliable binary bits and compute the syndrome S_0 of $\mathbf{R'}_{(it)}$,

2-) Generate T_v test vectors obtained by inverting some of the L_r least reliable binary symbols,

3-) Algebraic decoding of each test vector,

4-) For each test vector, compute the square Euclidian distance (metric) $M_i(i = 0, \ldots, T_{vn-1})$ between $\mathbf{R'}_{(it)}$ and the considered test vector,

5-) Select the Decided Word Dw having the minimal distance with $\mathbf{R'}_{(it)}$ and choose Cw concurrent words having the closest distance to $\mathbf{R'}_{(it),}$

6-) Compute reliability $\mathbf{F}_{(it)}$ for each symbol of the Dw,

7-) Compute extrinsic information $\mathbf{W}_{(it)} = \mathbf{F}_{(it)} - \mathbf{R'}_{(it)}$ for each symbol of the Dw.

The algebraic decoding that is used by each test vector involves different major steps. These steps depend on the product code parameters and, in particular, on the error correction power. In this paper, the Chase-Pyndiah algorithm is replaced by a new SISO decoding algorithm: the modified ABP algorithm. The proposed algorithm will be detailed in the next section.

III. The Modified Adaptive Belief Propagation Algorithm Applied to Iterative Product Decoding

Let us consider the transmission of binary symbols encoded by a product code on a Gaussian channel using BPSK mapping {0 → -1, 1 → 1}. The BCH component code of the product code has a parity check matrix $\mathbf{H}$ with n-k rows and n columns. In this section, we propose the modified ABP algorithm (m-ABP) to be used as the SISO decoder during the iterative decoding of BCH product codes. A detailed description of the proposed algorithm is given in Figure 3. The m-ABP algorithm is inspired by the standard ABP algorithm [5]. Each iteration of the ABP algorithm consists of two sub-steps: matrix sparsification and belief propagation. The m-ABP algorithm is motivated by the high cost of the Gaussian elimination required for adaptation in each iteration of the ABP algorithm. For the case of BCH product codes, we propose eliminating the adaptation from the iteration loop leaving only one adaptation in the initialization phase. The modified algorithm is composed of four steps. First, the received vector $\mathbf{R'}_{(\mathrm{it})}$ is ordered according to the absolute value of the soft input symbols in an ascending order (from the least reliable value to the most reliable value). Then, the *n-k* columns of the original parity check matrix **H** corresponding to the least reliable bits in $\mathbf{R'}_{(\mathrm{it})}$ are reduced to obtain an identity matrix by applying Gaussian elimination. The objective is to decrease the number of ones in the part of the parity check matrix that are associated with the least reliable symbols. Adapting the parity check matrix makes it suitable for the standard BP algorithm. The standard BP algorithm is applied in Step 3 to generate the soft output information $\mathbf{F}_{(\mathrm{it})}$. We note that few iterations (3 to 5) are necessary during the BP algorithm in the proposed algorithm. We call these local iterations to distinguish them from the iterations of the turbo process which we call global iterations. Finally, the hard decision $\mathbf{D}_{(\mathrm{it})}$ is computed during the last global iteration.

Modified Adaptive Belief Propagation for BCH codes

Step1. *Order the* $\mathbf{R'}_{(it)}$ *vector according to the absolute value of the soft input symbols* $|\mathbf{R'}_{(it)}|$

Step2. *Apply Gaussian elimination to the* n-k *columns of the original parity check matrix* **H** *according to the ordered indices obtained during the previous step.*

Step3. *Apply the standard BP algorithm to generate the soft output information* $\mathbf{F}_{(it)}$

Step4. *Compute hard decision* $\mathbf{D}_{(it)}$

Fig. 3. Proposed algorithm for iterative decoding of BCH codes.

An original algorithm called m-ABP that is an innovative method for turbo decoding of product codes is proposed. A major improvement of the m-ABP algorithm for iterative decoding of BCH product codes lies in the fact that the parity check matrix adaptation is done before a standard BP process. The adapted parity check matrix is the same during all the local iterations of the BP process. The matrix updating stage and the bit-reliability updating stage are not in an iterative loop as in the standard ABP algorithm. This feature enables a significant decrease of the ABP algorithm complexity. In addition, no damping coefficient [5] is necessary for the m-ABP algorithm. Instead, the reduction of the extrinsic information effect is done during the soft information computation as shown in Figure 2.

IV. Bit Error Rate performance

In this section, Bit Error Rate (BER) vs. Signal to Noise Ratio (SNR) performance results for iterative decoding of product codes are presented. First, results for a $(16,11)^2$ BCH product code over a Gaussian channel using BPSK mapping are shown in Figure 4. For this product code, standard BP decoding, standard ABP decoding, modified ABP decoding and Chase-Pyndiah decoding are applied. Five local iterations and 6 global iterations are chosen for the turbo decoding process with a standard BP algorithm. Five local iterations and 10 global iterations are necessary for the standard ABP algorithm. On the other hand, 3 local iterations and 6 global iterations are sufficient for the modified ABP algorithm. Six global iterations are chosen for the Chase-Pyndiah algorithm. For comparison, the uncoded BPSK and the simulated ML lower bound [9] curves are also plotted on Figure 4. As predicted, the error performance of standard BP algorithm for iterative decoding of product codes is poor. The standard ABP algorithm outperforms the standard BP algorithm. But, the Chase-Pyndiah algorithm outperforms the standard ABP algorithm by about 0.5 dB at BER of 10^{-6}. Theses results were also observed for other BCH codes. For the m-ABP algorithm, no significant BER difference (< 0.1 dB) is observed for $(16,11)^2$ BCH product codes if a comparison between the proposed algorithm and the classical Chase-Pyndiah algorithm is done.

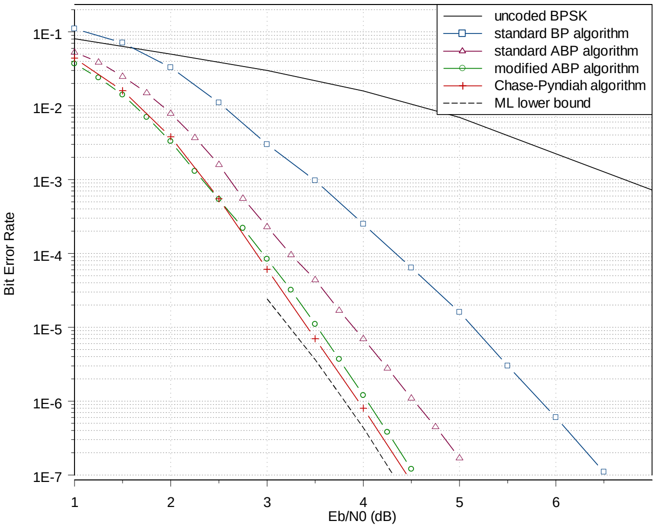

Fig. 4. Bit error rate performance of turbo decoding of a $(16,11)^2$ BCH product code on a Gaussian channel.

The performance of the m-ABP algorithm is compared with that of the Chase-Pyndiah algorithm for different product codes using BCH component codes. Three local iterations and 8 global iterations are chosen for the m-ABP algorithm and 8 global iterations are chosen for the Chase-Pyndiah. Bit error rate performance of turbo decoding of BCH product codes on Gaussian channel for two error correction powers (t=1 and t=2) are depicted in Figures 5 and 6 respectively. Performance results are very close for the two algorithms. The Chase-Pyndiah algorithm outperforms the m-ABP algorithm by about 0.135 dB at BER of 10^{-6} for $(32,26)^2$ BCH product code. In contrast, the m-ABP algorithm outperforms Chase-Pyndiah algorithm by about 0.125 dB at 10^{-3} for a $(128,113)^2$ BCH product code. These two cases are the greatest deviations observed in favor of each algorithm.

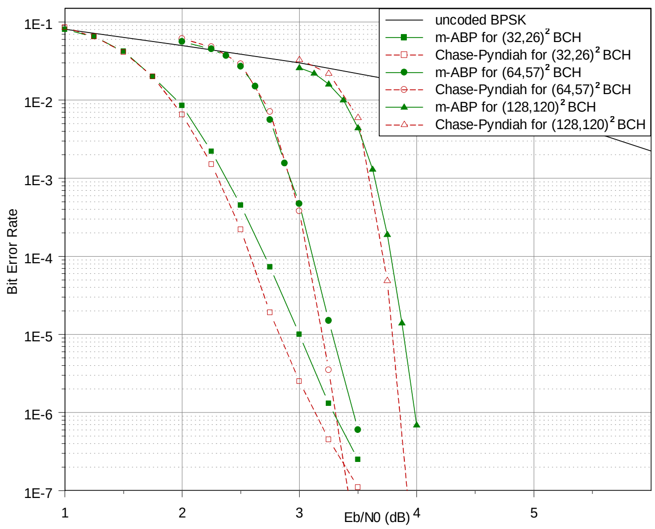

Fig. 5. Bit error rate performance of turbo decoding of BCH product codes (t=1) on a Gaussian channel.

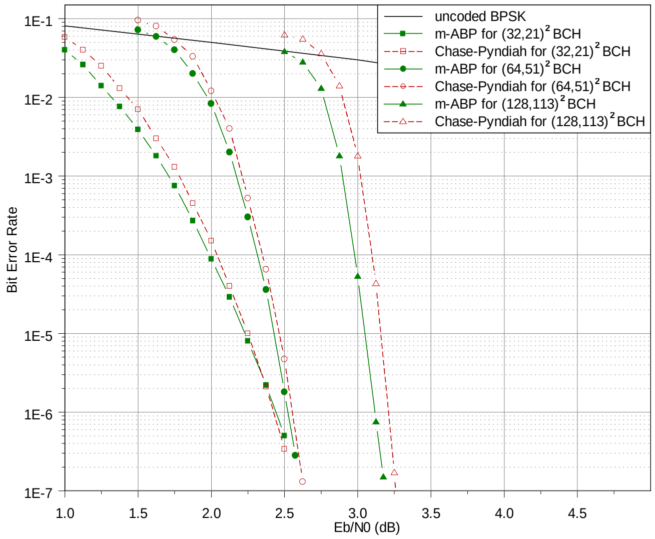

Fig. 6. Bit error rate performance of turbo decoding of BCH product codes (t=2) on a Gaussian channel.

V. Conclusion

In this paper, a simplification of the ABP algorithm has been proposed as an innovative method for the turbo decoding of product codes using BCH component codes. The novel algorithm has a lower decoding complexity and requires fewer iterations than the standard ABP algorithm for the turbo decoding of product codes based on BCH component codes. Simulation results have shown that the m-ABP algorithm outperforms the standard ABP algorithm and provides similar performance in terms of bit error rate to the classical Chase-Pyndiah algorithm. The advantages of the m-ABP algorithm are its high degree of parallelism for high data rate applications and the possibility of applying to it a new decoding method called "stochastic decoding". Indeed, stochastic decoding previously applied to the BP decoding of LDPC codes [10], has the potential to be applied to product codes decoded with the m-ABP algorithm.

Acknowledgment

The authors would like to thank R. Le Bidan for precious suggestions and interesting comments. They are grateful to P. Adde for providing the bit error performance of iterative decoding of product codes based on Chase-Pyndiah algorithm.

References

[1] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: Turbo Codes", *IEEE Int. Conf. on Communications, ICC'93*, v.2, May 1993, pp.1064-1070.

[2] F. R. Kschischang, B. J. Frey, H.-A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," IEEE Trans. Inform. Theory, vol. 47, no. 2, Feb. 2001, pp. 498-519.

[3] R. Pyndiah, "Near optimum decoding of product codes: Block Turbo Codes", *IEEE Trans. on Communications*, vol. 46, August 1998, pp. 1003–1010.

[4] C. Jego, P. Adde, C. Leroux, "Fully-parallel architecture for turbo decoding of product codes", *Electronics Letters* Vol. 42, Issue 18, August 2006, pp. 55–56.

[5] J. Jiang, K. R. Narayanan, "Iterative Soft-Input-Soft-Output decoding of Reed-Solomon codes by adapting the parity check Matrix", *IEEE Trans. on Inform. Theory*, vol. 52, no. 8, August 2006. pp. 3746-3756.

[6] P. Elias, "Error-free coding", *IRE Trans. on Inf. Theory*, vol. IT-4, September 1954, pp. 29-37.

[7] D. Chase, "A class of algorithms for decoding block codes with channel measurement information", *IEEE Trans. Inform. Theory*, vol IT-18, January 1972, pp 170-182.

[8] R. Pyndiah, A. Glavieux, A. Picart, S. Jacq, "Near optimum decoding of product codes", *GLOBECOM94*, November 1994.

[9] Franco Chiaraluce and Roberto Garello, "Extended Hamming Product Codes Analytical Performance Evaluation for Low Error Rate Applications", *IEEE Transactions on Wireless Communications*, vol. 3, November 2004, pp. 2353-2361.

[10] S. Sharifi Tehrani, W. J. Gross, S. Mannor, "Stochastic decoding of LDPC codes", *IEEE Communication Letters*, vol. 10, no. 10, October 2006, pp. 716 – 718.