International Journal of Innovative Research in Advanced Engineering (IJIRAE)
Volume 1 Issue 2 (April 2014)

Iterative Decoding Algorithm for Turbo Product Codes

Megha.M.S
PG Scholar / ECE, Department of Electronics and Communication,
M S Ramaiah Institute of Technology, Bangalore
msmegha999@gmail.com

Abstract*—Turbo product codes (TPC) are very suitable for applications requiring a large code length, a high code-rate, and good error performance. This paper describes an iterative decoding algorithm for any product code built using linear block codes. It is based on soft-input/soft-output decoders for decoding the component codes so that near-optimum performance is obtained at each iteration. This soft-input/soft-output decoder is a Chase decoder which delivers soft outputs instead of binary decisions. The soft output of the decoder is an estimation of the log-likelihood ratio (LLR) of the binary decisions given by the Chase decoder. The iterative decoding of product codes is also known as block turbo code (BTC) because the concept is quite similar to turbo codes based on iterative decoding of concatenated recursive convolutional codes. The performance of product code over Gaussian channel using BPSK signaling is shown for different code words.

***Index Terms* –Block codes, turbo codes, product codes, iterative decoding, extended hamming codes.**

I. INTRODUCTION

Turbo codes are a class of high-performance forward error correction (FEC) codes developed in 1993.These are the first practical codes to closely approach the channel capacity.Turbo coding is now widely adopted in various communication systems. Of all practical error correction methods known to date, turbo Codes and low-density parity-check codes (LDPCs) come closest to approaching the Shannon limit (the theoretical limit of maximum information transfer rate over a noisy channel).

There are two major classes of turbo codes: convolution turbo codes (CTC) and block turbo codes (BTC). CTC is formed from the parallel concatenation of two constituent codes separated by an inter leaver. Each constituent code may be any type of FEC (Forward Error Correction) code used for conventional data communications. With convolution codes as its constituent codes, CTC have been selected for the FEC purpose in various 3G wireless mobile communication standards. Based on Turbo Codes, in 1994, Pyndiah and others applied the iterative idea to product codes then proposed the soft-input /soft-output algorithms of linear b lock codes based on Chase algorithm. Turbo product codes (short for TPC) came out. Not only do Turbo product codes inherit the advantages of Turbo codes but, because of their structures using linear block codes they can get a simpler decoding method than Turbo codes. In recent years, the TPC are being widely applied to various communications occasions and have great potential in poor channel conditions of wireless communication systems. A lot of papers have been published on turbo codes, but most of the authors have focused on convolution turbo codes (CTC's) and very few have considered the block turbo code (BTC). In fact, concatenated coding was first introduced for block codes. Unfortunately, the first algorithms proposed for decoding these codes gave rather poor results because they relied on hard-input/hard-output decoders and lacked the soft-input/soft-output decoders. Works by Lodge and Hagenauer have produced solutions with good performance. These solutions are based on the trellis of block codes and are limited to small block codes since the number of states in the trellis of a block code increases exponentially with the number of redundancy bits. This new BTC offers a good compromise between performance and complexity and is very attractive for implementation.

II.TWO-DIMENTIONAL EXTENDED HAMMING CODE-BASED TURBO PRODUCT CODES

Product codes (or iterated codes) are serially concatenated codes which were introduced by Elias in 1954. The concept of product codes is very simple and relatively efficient for building very long block codes by using two or more short block codes. Let us consider two systematic linear block codes C_1 (n_1, k1, d1) and C_2 (n_2, k_2, d_2), where n, k and d stand for codeword length, number of information bits, and minimum hamming distance, respectively. The product code is obtained (see Figure 1) by

$$P=C_1 \oplus C_2 \qquad (1)$$

(1) Placing (k1 x k_2) information bits in a array of k_1 rows and k_2 columns.
(2) Coding the k_1 rows using the code C_2
(3) Coding the n_2 columns using code C_1

The parameters of the product code P are $n=n_1\times n_2$, $k=k_1\times k_2$, $d=d_1\times d_2$ and the code rate $R=R_1\times R_2$. The result of coding product code first with rows is the same as that with columns, nothing to do with the coding sequence. Similarly, the two-dimensional product code can be extended to three-dimensional or multi dimensional product codes.

ISSN: 2278-2311

Thus, we can build very long block codes with large minimum Hamming distance by combining short codes with small minimum Hamming distance.

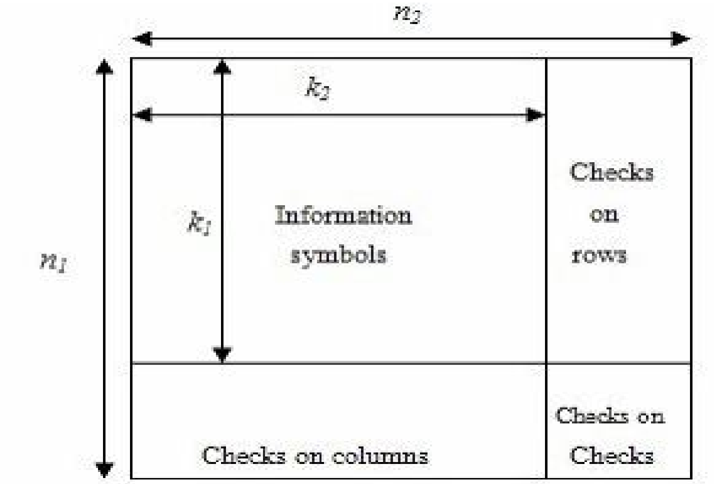

Figure1. Construction of Product Code

III. ITERATIVE DECODING OF TURBO PRODUCT CODES

Let us consider the decoding of the rows and columns of a product code transmitted on a Gaussian channel using QPSK signaling. On receiving matrix [R] corresponding to a transmitted codeword [E], the first decoder performs the soft decoding of the rows (or columns) of P using as input matrix [R].Soft-input decoding is performed using the Chase algorithm and the soft output is computed. By subtracting the soft input from the soft output we obtain the extrinsic information [W(2)] where index 2 indicates that we are considering the extrinsic information for the second decoding of P which was computed during the first decoding of P. The soft input for the decoding of the columns (or rows) at the second decoding of is given by

$$[R\,(2)]=[R]+\alpha\,(2)\,[W\,(2)] \qquad (2)$$

Where α (2) is the scaling factor which takes into account the fact that the standard deviation of samples in matrix [R] and matrix [W]. It is 0 at the first half-iteration and increases up to 1 as the half-iteration re-peats. The standard deviation of the extrinsic information is very high in the first decoding steps and decreases as we iterate the decoding. This scaling factor α is also used to reduce the effect of the extrinsic information in the soft decoder in the first decoding steps when the BER is relatively high. It takes small value in the first decoding steps and increases as the BER tends to zero. β is used in order to convert the hard decision output of an algebraic decoder to a soft decision value when the competing codeword does not exist in the decoding process. The decoding procedure described above is then generalized by cascading elementary decoders illustrated in Figure. 2**.**

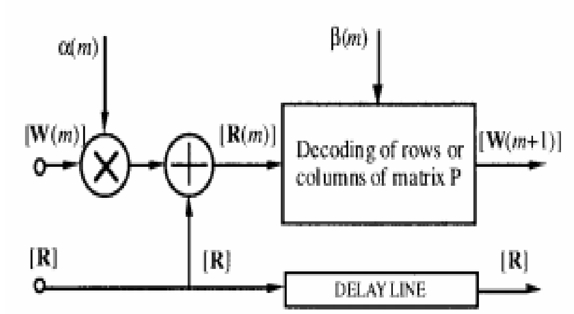

Figure2. Block diagram of elementary block turbo decoder.

IV CHASE-PYNDIAH DECODING ALGORITHM

The elementary decoding procedure consists of eight steps described as follows.

Step1: Generate the reliability $\mathbf{R_{abs}}=(|r_1|,|r_1|,\ldots|r_n|)$ and the sequence $\mathbf{Y}=(y_1,y_2,\ldots y_n)$ from the observation **R**

$$= \begin{cases} 1\,, if\ r1 > 0 \\ 0, otherwise \end{cases}$$

Step 2: Find p least reliable positions from $\mathbf{R_{abs}}$ obtained at Step 1.

Step 3: Create the set of $q(2^p)$ test patterns T^q which includes every composition of 0 and 1 on the searched positions, and all zeros on the other positions.
Step4: Generate the test sequence set

$$Z^q = Y \bigoplus T^q$$

Step 5: Calculate the syndrome $\mathbf{S_i}= (S_1^i, S_2^i \ldots S_q^i)$ using equation (3) for each sequence $\mathbf{Z^i}$ in the test sequence set $\mathbf{Z^q}$ and the parity check matrix H of the Hamming code.

$$\mathbf{S^i = Z^i . H^T} \qquad (3)$$

If the calculated syndrome is not zero which implies that there are errors, construct the candidate codeword set Cq by flipping the expected error position under the assumption that there are no more errors than one.
Step 6: Compute the Euclidean distance from the observation R to each candidate codeword C_i, in the candidate codeword set Cq obtained at Step 5, and decide the maximum-likelihood (ML) codeword

$$\mathbf{D} = \mathbf{C^i} \text{ if } |\mathbf{R} - \mathbf{Ci}|^2 = \min \{|\ \mathbf{R} - \mathbf{C}^l|2\}\ \forall l \in [1, 2p], \text{ Where } |R - Ci|^2 = \sum_{i=0}^{n} ((r_l - c_l^i)^2 \quad C_l^i \text{ is mapped from } \{0, 1\}$$

Step 7: Construct S_j^{+1} and S_j^{-1} which are the candidate codeword sets whose j-th bit is +1 and -1, respectively. Then, search the code words C^{+1j} and C^{-1j} that have the smallest metric calculated at the previous step in S_j^{+1} and S_j^{-1} respectively.
Step 8: One of C_j^{+1} and C_j^{-1} , whose distance metric is smaller, is the codeword **D** determined in Step 5, and the other is decided as the competing codeword **C**. Whereas the code word **D** always exists, the competing codeword **C** may not at some cases. According to the existence of the competing codeword, equation (4) defined by Pyndiah is used to calculate the updated LLR of r_j,

$$\begin{cases} \left(\frac{|\mathbf{R}-\mathbf{C}|^2 - |\mathbf{R}-\mathbf{D}|^2}{4}\right) dj. & \text{if } \mathbf{C} \text{ exists,} \\ \beta \times 1. & \text{Otherwise,} \end{cases}$$

where dj is the j-th element of . The extrinsic information wj, the output of the SISO decoder, is then updated using equation (4)

$$wj = r_j' - r_j \qquad (4)$$

V SIMULATION RESULTS

The turbo decoding algorithm described in this paper applies to any product code based on linear block codes. Before proceeding to the simulation results, we shall now give the different parameters used by this turbo decoding algorithm.
The characteristics of the turbo decoding algorithm are as follows.

1) *Test sequences*: The number of test patterns is 16 and are generated by the four least reliable bits *(p=4)*.
2) *Weighting factor* α: To reduce the dependency of α on the product code, the mean absolute value of the extrinsic information |w| is normalized to one. The evolution of α with the decoding number is
α (m)=[0.0, 0.2, 0.3, 0.5,0.7,0.9,1.0, 1.0]
3) *Reliability factor β:* To operate under optimal conditions, the reliability factor should be determined as a function of the BER. For practical considerations, we have fixed the evolution of β with the decoding step to the following values:
β(m)=[0.2,0.4, 0.6, 0.8, 1.0, 1.0, 1.0]

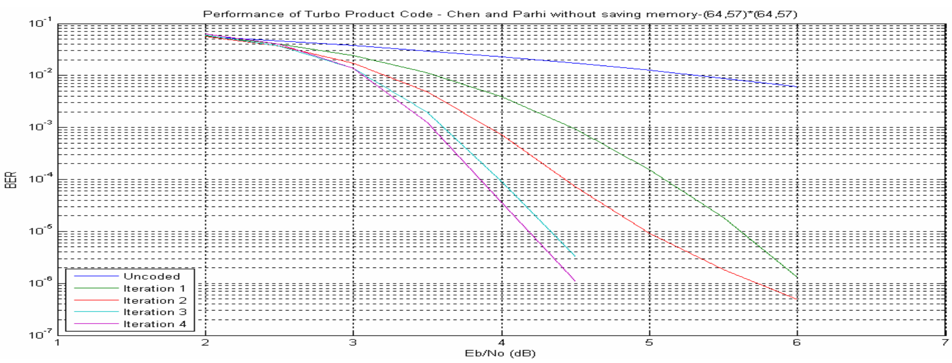

Figure.3. BER versus Eb/N0 of product code [(64, 57, 4)] ^{2} on a Gaussian channel using BPSK signaling for 10,000 code words.

ISSN: 2278-2311

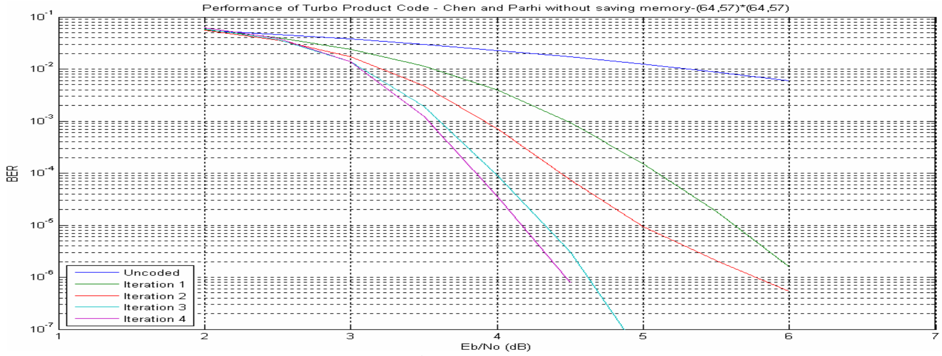

Figure.4. BER versus Eb/N0 of product code [(64, 57, 4)] ^{2} on a Gaussian channel using BPSK signaling for 20,000 code words.

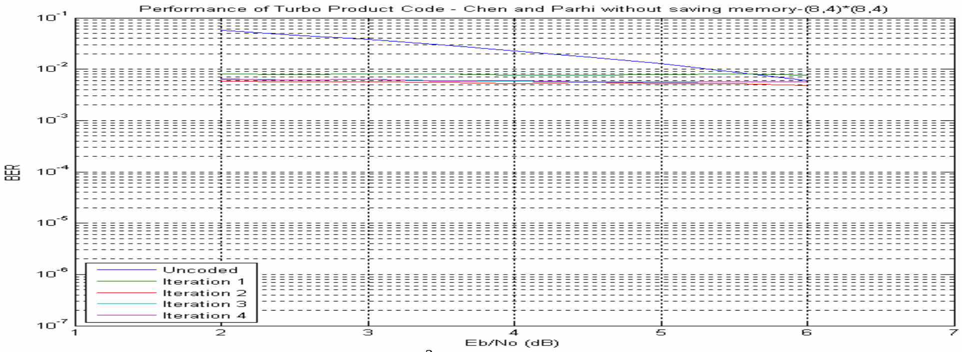

Figure.5. BER versus Eb/N0 of product code [(8, 4, 4)]^2 on a Gaussian channel using BPSK signaling for 10,000 code words.

VI CONCLUSION

The performance achieved by the BTC's presented in this paper indicates that they are the most efficient known codes for high code rate applications. We have presented a block turbo code which is constructed using extended hamming codes. The block turbo codes (product codes) are decoded using an iterative decoding algorithm (block turbo decoder) based on soft decoding and a soft decision of the component codes. A lot of exciting work remains to be done on BTC. Among others, joint error detection and error correction seems very promising, as well as unequal error protection and BTC for small data blocks.

VII REFERENCES

[1] R. Pyndiah, A. Glavieux, A. Picart and S. Jacq, "Near Optimum decoding of product codes",IEEE Transactions on Communications,VOL.46,No.8,August 1998.

[2] Junhee Cho and Wonyong Sung, "Reduced complexity chase-pyndiah decoding algorithm for turbo product codes", Department of Electrical Engineering Seoul National University,2011.

[3] Frederic Lehmann and Gian Maggio, "Analysis of the Iterative Decoding of LDPC and Product Codes Using the Gaussian Approximation", IEEE Transactions on Information Theory, VOL.49, NO.11, November 2003.

[4] Jessica pursley, "Turbo product codes and channel capacity", IEEE Transactions on Information Theory, March 2000.

[5] P. Adde, R. Pyndiah, O. Raoul, and J. R. Inisan, "Block turbo decoder design," in Proc. IEEE Int. Symp. Turbo Codes & Related Topics, vol.1/1, Brest, France, Sept. 1997, pp. 166–169.

[6] R. Pyndiah, "Iterative decoding of product codes: Block turbo codes,"in Proc. IEEE Int. Symp. Turbo Codes & Related Topics, vol. 1/1, Brest,France, pp. 71–79, Sept. 1997.

[7] A. Goalic and R. Pyndiah, "Real-time turbo decoding of product codes on a digital signal processor," in Proc. IEEE Int. Symp. Turbo Codes &Related Topics, Brest, France, Sept. 1997, pp. 267–270.

[8] O. Aitsab and R. Pyndiah, "Performance of Reed-Solomon block turbo codes," in Proc. IEEE GLOBECOM'96 Conf., vol. 1/3, London, U.K.,Nov. 1996, pp. 121–125