A Block Turbo Equalizer with Single-Parity Check Turbo Product Codes

D. A. Guimarães and A. F. Santos

***Abstract*— Combining a maximum a posteriori (MAP) equalizer and a D-dimensional Single-Parity Check Turbo Product Code (SPC-TPC) in a novel way, a Block Turbo Equalizer structure is proposed here. The proposed scheme has a potential for reduction in complexity, when compared to other block turbo equalizers. Simulations reveal attractive performance results.**

***Keywords*— Block Turbo Equalizer, Single-Parity Check Turbo Product Code.**

I. Introduction

Recently, the combination of equalization with channel decoding techniques, in a cooperative and iterative fashion, has been the subject of a lot of research, configuring what is generally named Turbo Equalization [1]. Most of the results reported so far in the literature consider turbo equalizer structures using the BCJR algorithm (or some of its variants) in the equalizer block [2][3], and well known soft-input, soft-output (SISO) algorithms in the channel decoding block, such as the Soft Output Viterbi Algorithm (SOVA) or variants of the BCJR algorithm, mainly for decoding convolutional codes [1][2][3] or convolutional turbo codes [4][6]. In the equalizer block, the use of algorithms with reduced complexity represents a challenging research effort in itself [3][5]. The use of block turbo codes in the turbo equalization process seems to be less investigated [6][7]. In these references, the channel decoding block uses the turbo decoding process described in [8] for decoding a BCH-based parallel-concatenated scheme and a BCH-based two-dimensional product code, respectively. In [6] and [7], one turbo decoding iteration is performed for each turbo equalizer iteration.

In this paper, a block turbo code is used in the channel-decoding block. However, instead of waiting for the end of a complete iteration of the turbo decoding process, only one dimension of a D-dimensional SPC-TPC is decoded at each iteration of the turbo equalization process, the complexity of such scheme being reduced, if compared to the block turbo equalizers analyzed in [6] and [7]. Furthermore, the delay of this process can be reduced in comparison to the case in which there is a complete iteration in the turbo decoding process per turbo equalizer iteration. Additionally, instead of using BCH-based codes with Chase-based turbo decoding, an SPC-TPC scheme is used, for which powerful codes can be constructed and a very simple Log-MAP-based turbo-decoding algorithm exists [9]. Therefore, further reduction in complexity can be achieved.

Simulations were carried out and revealed attractive results, as compared to those reported in [6]. Another feature of the proposed scheme is the opportunity of using high-rate codes in order to meet the objectives of the ever-increasing demand for spectrally efficient communication systems.

II. SPC-TPC codes revisited

A Product Code is an array code in which D component codes are serially concatenated. The resultant D-dimensional code has as codeword length, information word length, code rate and minimum Hamming distance the product of the corresponding codeword length, information word length, code rate and minimum Hamming distances of the component codes.

If the component codes are single-parity check codes $(n, n-1, 2)$, identical in each dimension, the resultant D-dimensional Single-Parity Check Product Code (SPC-PC) has the following parameters:

- length of the input block on the encoder:
$$K = (n-1)^D \tag{1}$$
- length of the output block on the encoder:
$$V = n^D \tag{2}$$
- code rate:
$$R = \left(\frac{n-1}{n}\right)^D \tag{3}$$
- minimum Hamming distance:
$$\delta_{min} = 2^D \tag{4}$$

Clearly, the SPC-PC has the potential for higher error correction capability than the SPC code alone.

When SPC-PC codes are turbo-decoded, we call them SPC-TPC (Single-Parity Check Turbo Product Codes). The algorithm used here in the tubo decoding of the SPC-TPC code is the algorithm described in [9], and it is briefly described here. This is a low-complexity decoding algorithm in which the *extrinsic information* is calculated for all symbols in the codeword, not only for the information symbols.

The turbo decoder can be intepreted as composed by one SISO (Soft Input, Soft Output) decoder used D times for each turbo-decoding iteration, or can be viwed as a pipeline representation [5] in which there are D modules used, jointly, one time per turbo-decoding iteration. A SISO decoder or a SISO decoder module has two inputs and two outputs. The first input is the *channel information*, $L_c^D(y_n)$. The *channel*

Dayan Adionel Guimarães e André Fonseca dos Santos¸ Departamento de Telecomunicações, Instituto Nacional de Telecomunicações, Santa Rita do Sapucaí, Brasil, E-mails: dayan@inatel.br, andref@inatel.br.

information is a measure of the reliability of the output of the matched filter y_n. The other input is the a priori information, $L_{a(d)}(x_n)$. The *a priori information* of the dimension d is the sum of the *extrinsic information* from the other dimensions:

$$L_{a(d)}(x_n) = \sum_{\substack{i=1 \\ i \neq d}}^{D} L_{e(i)}(\hat{x}_n) \tag{5}$$

Throughout this paper it is assumed BPSK (Binary Phase-Shift Keying) signalling. When the binary mapping $\{0,1\} \rightarrow \{-1,1\}$ is used, the single dimension *extrinsic information* of the n_{th} bit in the SPC component code is given by [9]:

$$L_{e(d)}(\hat{x}_n) = (-1)^N \cdot 2 \cdot \arctan(u) \tag{6}$$

where

$$u = \prod_{\substack{i=1 \\ i \neq n}}^{N} \tanh\left(\frac{L_C^D(y_i) + L_{a(d)}(x_i)}{2}\right) \tag{7}$$

The a posteriori information, $L_d(\hat{x}_n)$, at the output of the decoder of the dimension d is:

$$L_d(\hat{x}_n) = L_C^D(y_n) + L_{a(d)}(x_n) + L_{e(d)}(\hat{x}_n) \tag{8}$$

At each iteration the *extrinsic information* is improved, giving a better estimation of the *a priori information*. The hard decision on the bits is taken based on the polarity of the a posteriori information of the turbo decoder:

$$L_{saida}^D = L_C^D(y) + \sum_{i=1}^{q} L_{ext(i)}(\hat{x}) \tag{9}$$

III. SPC-TPC CODES APLIED TO CONVENTIONAL TURBO EQUALIZATION AND TURBO DECODING

Before presenting the new turbo equalizer proposed here, the SPC-TPC code will be adapted to the conventional scheme of turbo equalization and turbo decoding used in [4], [7] and [6]. The communication system is depicted in Figure 1. The information bits b_k are encoded by a SPC-TPC encoder. The encoded symbols x_n are interleaved and modulated by a BPSK modulator, generating x_n^π. The symbols, x_n^π, are transmitted through a multipath channel with addition of AWGN noise, w_n. The received symbols from the corrupted channel, r_n, are processed by a turbo equalizer and the information bits are estimated.

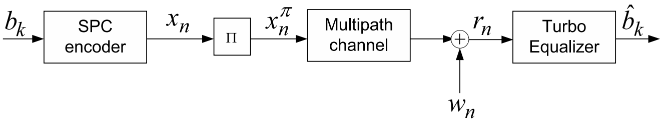

Fig. 1. The Novel Block Turbo Equalizer structure.

Figure 2 shows an example of a turbo equalizer using an SPC-TPC decoder of two dimensions (2D). It is composed by a SISO equalizer and an SPC-TPC decoder (pipeline representation). From the corrupted vector of received symbols, $\mathbf{r} = [r_1, r_2, \ldots, r_n, \ldots, r_N]$ where N is the codeword length, and from the *a priori information* $L_a^{Equ}(x_n^\pi)$, the equalizer block computes the a posteriori information of the interleaved symbols, $L^{Equ}(x_n^\pi)$, using the Log-MAP algorithm [2]. In the first Turbo Equalizer iteration, the *a priori information* of the equalizer block is set to 0 (a priori probability equal to $\frac{1}{2}$). The *a priori information* $L_a^{Equ}(x_n^\pi)$ is subtracted from the a posteriori information $L^{Equ}(x_n^\pi)$ at the output of the block equalizer, and only the *combined channel and extrinsic information* $L_e^{Equ}(x_n^\pi)$ of the coded symbols are transmitted to the decoder block, as in [3] and [6]. The *combined channel and extrinsic information* are de-interleaved and used as the channel log-likelihood ratio (LLR) $L_C^{Dec}(x_n)$ for the turbo decoder. Using the *channel information* provided by the block equalizer and the *a priori information* of the block decoder, the *extrinsic information* of each SISO decoder is computed using (6). The sum of the *extrinsic information* of all D dimensions is used as *a priori information* of the block equalizer in the next turbo equalizer iteration, while the *a priori information* of the decoder block is computed by (5). In this way, for each turbo equalizer iteration, one turbo decoding iteration is performed. When SPC-TPC codes composed by more dimensions than two are used, the same algorithm is used. The *extrinsic information* of all dimensions D are used as *a priori information*of the equalizer block.

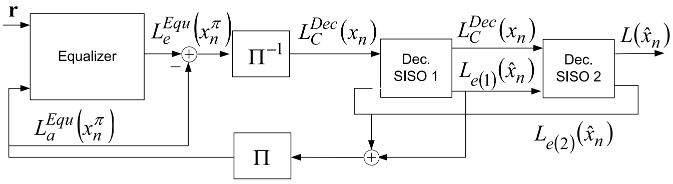

Fig. 2. Turbo Equalization using a 2D SPC-TPC code.

IV. DESCRIPTION OF THE PROPOSED TURBO EQUALIZER

The proposed turbo equalizer is depicted in Figure 3. The block equalizer is the same of the previous section. The modification introduced here is the use of the decoding of a single dimension SPC-TPC code on each iteration of the turbo equalizer.

Let $\{1,\ 2,\ \ldots,\ p,\ \ldots,\ P\}$ be the set of Turbo Equalizer iterations and $\{1,\ 2,\ \ldots,\ d,\ \ldots,\ D\}$ be the set of dimensions of the SPC-TPC. The relationship between the current turbo equalizer iteration and the current dimension of the SPC-TPC decoded by the decoder block is given by

$$q = [(p-1), mod(d)] + 1 \tag{10}$$

Since one turbo decoding iteration means decoding all dimensions of the SPC-TPC, here one turbo decoding iteration is performed in D turbo equalizer iterations. The single dimension *extrinsic information*, $L_{e(d)}(x_n)$, is computed by decoding dimension d of the SPC-TPC code in the turbo equalizer iteration p, using (6).

Taking into account the last D values of $L_{e(d)}(x_n)$ calculated in the previous single dimension decoding steps, the decoder *extrinsic information* is computed as follows:

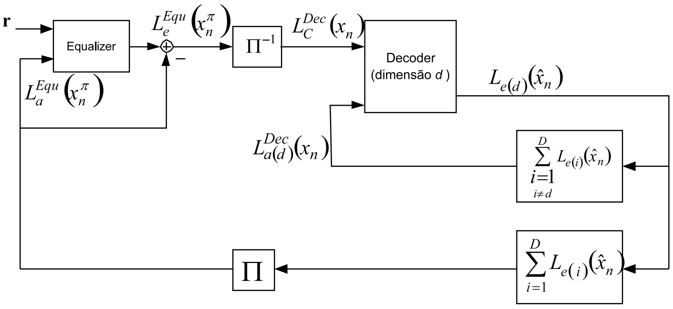

Fig. 3. The Novel Block Turbo Equalizer structure.

$$L_e^{Dec}(x_n) = \sum_{\substack{i=1 \\ i\neq d}}^{D} L_e(i)(\hat{x}_n) \tag{11}$$

The *extrinsic information* computed in (11) is used as an estimate of the *a priori information* for the decoder block in the next single dimension decoding.

At each turbo equalizer iteration, the estimate of the *a priori information* used by the equalizer block is computed as follows:

$$L_a^{Equ}(x_n) = \sum_{i=1}^{d} L_e(i)(\hat{x}_n) \tag{12}$$

V. Performance Results

Simulation results are presented in this section according to the following assumptions and configurations:

1) It is assumed that the channel is known, aiming at investigating the proposed idea with no influence of channel estimation imperfections.
2) Since the BCJR equalizer is used, the performance of the architecture proposed here must be viewed as a lower bound on the performance that can be achieved with other equalization algorithms.
3) Two channel impulse responses usually adopted in the literature were considered here, having tap weights: $h_1 = [0.671\ 0.5\ 0.387\ 0.316\ 0.224]$ [6] and $h_2 = [0.407\ 0.815\ 0.407]$ [3].
4) For comparison purposes, the codes and channel interleaver sizes were chosen, in order to approximately match the codeword lengths, code rates and channel interleaver sizes considered in [6]. The codes are three-dimensional SPC-TPC's with (6,5), (10,9) and (17,16) SPC component codes, denoted by $C_1 = (6,5)^3$, $C_2 = (10,9)^3$ and $C_3 = (17,16)^3$, having code rates 0.5787, 0.729 and 0.834, respectively. We have used S-random channel interleavers, with $S = 128$, with lengths 20.736 bits, 20.000 bits and 19.652 bits, when C_1, C_2 and C_3 are used, respectively.
5) As in [6], the number of iterations used here was defined as the number of iterations beyond which no significant iteration gains were obtained.This number was fixed around fifteen.
6) Each value of BER computed through the Monte Carlo simulations was estimated using at least fifty information bit errors. This number of errors was chosen to speed-up the simulations and was enough for a low variance in the curve position, given the steep nature of the BER waterfall region.

Figure 4 shows simulation results for the Turbo Equalizer with SPC-TPC turbo decoding, as in section IV, with the codes C_1, C_2 and C_3 over the channel h_2. Figure 5 shows simulation results for the proposed turbo equalizer with the codes C_1, C_2 and C_3 over the channels h_1 and h_2. The performances of C_1, C_2 and C_3 on the AWGN channel are also presented for comparisons. Comparing Figures 4 and 5, it can be seen that the proposed scheme outperforms the turbo equalizer with a complete turbo decoder iteration per turbo equalizer iteration. As can be seen from Figure 5, around the BER of 10^{-5} the gaps from the AWGN performance for C_1 and C_2 codes are less than 0.5 dB, regardless of the channel, and around 1.2 dB for C_3. This is quite a good result, almost eliminating the ISI. It can also be seen that, with C_2 and C_3, the proposed scheme outperforms the equalization schemes with Block Turbo (BT), Convolutional Turbo (CT) and Convolutional Code (CC) reported in [6], at a BER of 10^{-5}. However, with C_1 the results in [6] are superior. This can be explained by the fact that C_1 is a very short code with very small interleavers between each component code. In [6], code interleavers on the order of 15,000 bits were used.

Table I shows the approximate gain of the turbo equalizer proposed here compared with the results presented in [6].

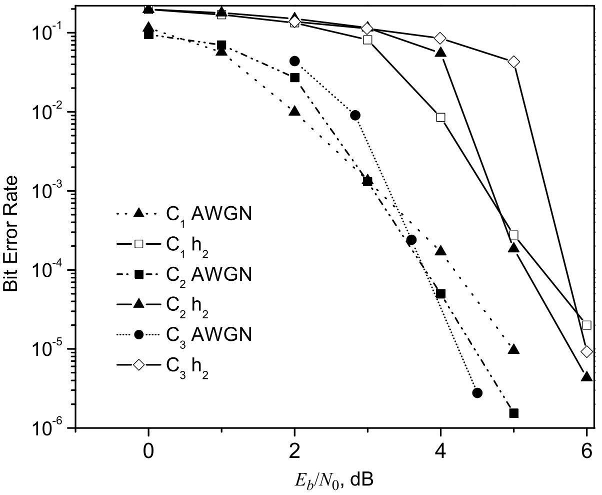

Fig. 4. Performance results for the turbo equalizer using a turbo SPC-TPC decoder. $C_1 = (6,5)^3$, $C_2 = (10,9)^3$, $C_3 = (17,16)^3$, $h_2 = [0.407\ 0.815\ 0.407]$.

VI. Concluding Remarks

A novel block turbo equalizer scheme is proposed here. For high-rate codes this scheme seems to be more attractive than

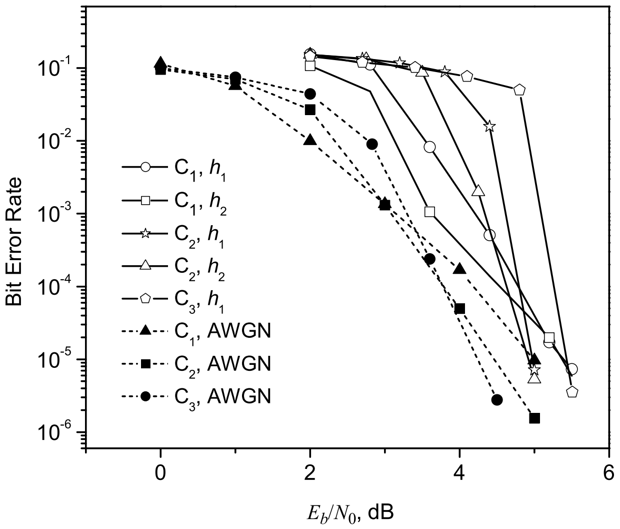

Fig. 5. Performance results for the proposed scheme. $C_1 = (6,5)^3$, $C_2 = (10,9)^3$, $C_3 = (17,16)^3$, $h_1 = [0.671\ \ 0.5\ \ 0.387\ \ 0.316\ \ 0.224]$, $h_2 = [0.407\ \ 0.815\ \ 0.407]$.

TABEL I

APPROXIMATE GAIN OF THE TURBO EQUALIZER PROPOSED HERE COMPARED WITH THE RESULTS PRESENTED IN [6].

Code used in [Yea02]	approximate gain (dB) @10^{-5}	
	C_2	C_3
convolutional turbo	0,2	0,3
BCH turbo	0,7	0,8
convolutional	1	0,8

those considered in [6]. Suggestions for further investigations include, but are not limited to: 1) EXIT chart analysis, 2) quantitative complexity analysis in order to permit the complexity of the proposed scheme with those reported in [6], 3) performance analysis with sub-optimal equalizer blocks for adaptive turbo equalization in time-varying ISI channels, 4) performance analysis with non-gaussian noise, e. g. impulsive noise and 5) investigation about the performance of a turbo equalizer with convolutional turbo decoding, performing the SISO decoding of a single component code for each turbo equalizer iteration.

VII. ACKNOWLEDGMENT

The authors are grateful to the anonymous reviewers for their valuable comments and suggestions. Thanks also to the finantial support by FINEP - Financiadora de Estudos e Projetos, contract number 22.02.0431.00 .

REFERENCES

[1] Douillard, C., M. Jezequel, C. Berrou, A Picart, P. Didier, and A Glavieux, *Iterative Correction of Intersymbol Interference: Turbo Equalization*. European Transactions on Telecommunication, vol. 6, pp. 507-511, Sep-Oct 1995.

[2] Bauch, G. and Franz, V.,*A Comparison of Soft-in/Soft-out Algorithms for Turbo detection*, , Intern. Conf. on Telecommunication, pp. 259-263, Jun 1998.

[3] Koetter, R., A. Singer and M. Tüchler, *Turbo Equalization*. IEEE Signal Processing Magazine, Feb 2003.

[4] Raphaeli, D. and Y. Zarai, *Combined Turbo Equalization And Turbo Decoding*. Proc. Global Telecommunications Conf., Phoenix, AZ, pp. 639-641, 1997.

[5] Glavieux, A., C. Laot, and J. Labat, *Turbo Equalization Over a Frequency Selective Channel*. Proc. of the Intern. Symposium on Turbo codes, Brest, France, pp. 96-102, Sep 1997.

[6] Yeap, B. L., T. H. Liew, J. Hamorsky, and L. Hanzo, *Comparative Study of Turbo Equalization Schemes Using Convolutional, Convolutional Turbo, and Block-Turbo Codes*. IEEE Transactions on Wireless Communications, vol. 1, N^o 2, Apr 2002.

[7] Noorbakhsh, M. and K. Mohamed-Pour, *Combined Turbo Equalization and Block Turbo Coded Modulation*. IEE proceedings on communications, vol. 150, n^{o}3, pp.149-152, 2003.

[8] Pyndiah, R. M., *Near-Optimum Decoding of Product Codes: Block Turbo Codes*. IEEE Transactions on Communications, pp. 41-46, Vol. 49, n^o 8, Aug 1998.

[9] Rankin, D. M. and T. A. Gulliver, *Single Parity Check Product Codes, IEEE Transactions on Communication*. vol. 49, no. 8, pp. 1354-1362, Aug 2001.