

Single Parity Check Turbo Product Codes for the DVB-RCT Standard

Angelo Pinelli Martins Samia
National Institute of Telecommunications - Inatel
P.O. Box 05 - 37540-000
Santa Rita do Sapucaí - MG – Brazil
angelo.pinelli@inatel.br

Dayan Adionel Guimarães
National Institute of Telecommunications - Inatel
P.O. Box 05 - 37540-000
Santa Rita do Sapucaí - MG – Brazil
dayan@inatel.br

Abstract—This article suggests some modifications in conventional single-parity check turbo product code structures to fulfill the specifications of the DVB-RCT standard proposed for the return channel of DVB-T based Digital TV systems. The resultant codes are two and three-dimensional array codes, in which diagonal parity equations are applied. Simulation results are presented for the AWGN and Rayleigh fading channels and reveal a good trade-off between performance and complexity for the suggested channel coding schemes.

Index Terms—Array codes, Single Parity Check Product Codes, Block Turbo Codes, DVB-RCT, DVB-T.

I. INTRODUCTION

Two-way communications are essential to provide real-time interactivity between the subscriber and the digital television content provider. The so-called DVB-RCT (Return Channel-Terrestrial) standard [1] defines a wireless interface that can be shared by a large number of terminals, enabling them to send “return” signals to a base station, at the same time and through the same antenna that it receives IP data on the “forward” link, which are carried using the DVB-T [2] broadcast channel.

A discussion on the utilization of Single Parity Check Turbo Product Codes (SPC-TPC) [5] for the requirements of DVB-RCT is in order. Conventional two and three-dimensional (2D and 3D) SPC are modified through the use of different component codes in each dimension, diagonal parity equations and zero padding, in order to match the specifications of the DVB-RCT standard, concerning block sizes and code rates. The key feature of the channel-coding scheme suggested here is the low complexity of the coding and the decoding processes.

The remainder of this paper is organized as follows. Section II presents the channel-coding requirements according to the DVB-RCT standard, i.e. block sizes and code rates associated to each modulation scheme. In Section III, details about the code construction are presented, in order to fulfill the specifications listed in Section II. Section IV is devoted to the turbo-decoding algorithm, while Section V presents simulation results for the channel-coding schemes suggested here, for the AWGN and the Rayleigh fading channels. Finally, Section VI concludes the paper, highlighting some

open questions to be addressed in future work.

II. CODE REQUIREMENTS

According to the DVB-RCT, one of the following methods of channel-coding is used in a given RF channel: turbo codes or concatenated Reed-Solomon plus convolutional codes. The turbo code adopted for the standard uses parallel concatenation of Circular Recursive Systematic Convolutional (CRSC) codes [3] [9] and its performance is evaluated in [4] for a variety of input block sizes

No matter the channel-coding method used in the DVB-RCT standard, the data bursts produced after the coding and modulation process have a fixed length of 144 modulated symbols. Table 1 lists the original sizes of the useful data payloads to be encoded, given the selected modulation and code rate [1].

TABLE I
USEFUL DATA PAYLOAD OF A BURST (DVB-RCT STANDARD) [1]

Modulation	Rate = 1/2	Rate = 3/4
QPSK	144 bits	216 bits
16-QAM	288 bits	432 bits
64-QAM	432 bits	648 bits

In order to match the code structure with the requirements given by Table 1, the number of dimensions of a single parity check (SPC) product code and the length of the component codes proposed here are varied. In addition to the conventional multi-dimensional parity equations, diagonal parity [5] equations are also considered as a means to achieving these requirements. The resultant structure can be viewed as a form of Array Codes [12].

III. CODE CONSTRUCTION

The channel code can be constructed as shown in Figure 1, for the case where the input block size is 144 bits. First, the parity is calculated column-wise using a $(n,k) = (7,6)$ single parity check component code, which will make the original input block grow by 24 bits. The parity of the rows is then calculated using the same $(7,6)$ component code used in the first dimension. After that, the 49 parity bits corresponding to the encoding in the third dimension are calculated

using a (5,4) SPC code. Diagonal parity is applied to the resulting $(7 \times 7 \times 5, 6 \times 6 \times 4)$ code by computing 7 parity bits for each of the 5 planes, successively towards the “depth” of the structure. The final output block has 280 bits, which results in an overall code rate $r = 0.51$. Zero-padding the output with 8 symbols brings the code rate down to $\frac{1}{2}$ in this case, as required.

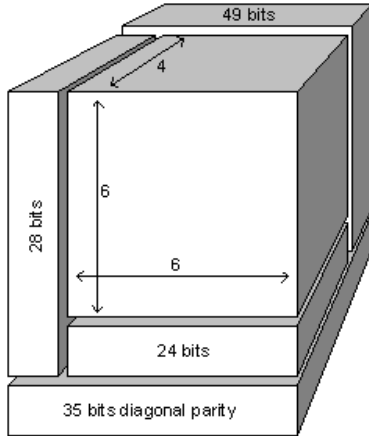


Fig. 1. Block shape matched to the specifications of DVB-RCT for QPSK ($R = \frac{1}{2}$).

An example of the procedure used to compute the diagonal parity on each plane is shown in Figure 2. The block formed with the information bits plus the symbols corresponding to the horizontal and vertical parities is first re-arranged to form a new 5×5 array, the columns of which corresponding to the left diagonals of the original block. One way to accomplish this is to do circular shifts on the lines of the structure in Figure 2(a). The first line is not shifted; the second line is shifted to the left by one bit and the l^{th} line of the original block is shifted by $l - 1$ bits. The left diagonal parity bits are then calculated by computing the parity bits along the columns of the array shown in Figure 2(b).

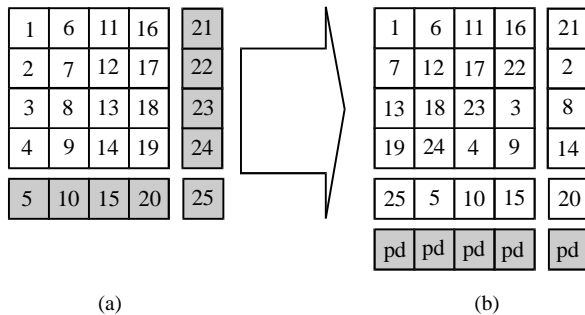


Fig. 2. Computation of the left diagonal parity for a 2D $(5,4)^2 = (5 \times 5, 4 \times 4)$ product code. (a) Array with information bits, horizontal and vertical parity bits; (b) Modification in (a) for left diagonal parity calculation.

The influence of diagonal parity is analyzed with more details in [8], where the best results were obtained with the restriction that the component codes

in each dimension being equal and having an odd number of bits. An independent initiative to compute the diagonal parity is proposed in [5] and, in this case, it is enough that the component codes have the same length. Because the requirements in terms of code rate and block length are defined by DVB-RCT, the structures suggested in this article were built so as to optimize these parameters, following the guidelines given in [5].

When the required code rate is $\frac{3}{4}$, a two-dimensional product code is used, as illustrated in Figure 3, for a 432-bit input block. Note that the last 9 positions of the array has to be filled with zeros so that the input bits are arranged as a perfect square and diagonal parity can be applied [5] [8]. Two rows of 16 bits are generated based on the calculation of the diagonal parity bits to the left and to the right of the array formed by the information bits plus vertical and horizontal parity bits.



Fig. 3. Block shape matched to the specifications of DVB-RCT for QPSK ($R = \frac{3}{4}$).

The structures designed for operation with 16-QAM (*Quadrature Amplitude Modulation*) and 64-QAM follow the same aforementioned principle. For each case, the number of dimensions and the size of the component codes had to be adjusted. The codes are designed so that the resulting code rate is equal or exceeds the one specified by the DVB-RCT standard.

It was found that 3D codes are viable only when the required code rate is $R = \frac{1}{2}$. When $R = \frac{3}{4}$ it is not possible to use more than two dimensions in the product code. As in QPSK mode, diagonal parity is applied to both 3D and 2D structures, but while in the first case the diagonal parity can only be applied left or right-wise due to rate restrictions, the latter case implements both left and right diagonal parity. In some configurations, both the input and output blocks had to be zero-padded. Figure 4 and 5 show the suggested structures for $\frac{1}{2}$ and $\frac{3}{4}$ code rates, respectively, when using 16-QAM modulation. For the 3D case, the resulting codeword has $288 + 48 + 56 + 49 + 63 = 504$ bits, and it has to be appended with 72 null symbols to comply with the $R = \frac{1}{2}$ specification.

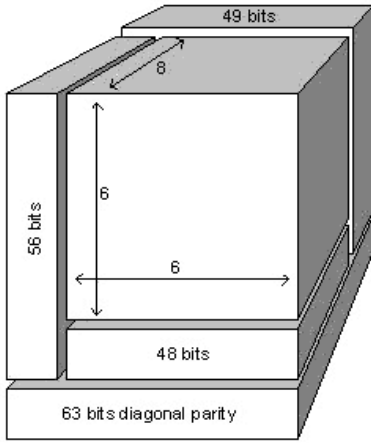


Fig. 4. Block shape matched to the specifications of DVB-RCT for 16-QAM ($R = \frac{1}{2}$).

The required input block for 16-QAM mode and $R = \frac{3}{4}$ has 432 bits and the 2D product code with diagonal parity suggested for this case is shown in Figure 5. Here the resulting codeword has 528 bits and it is zero-padded with 43 symbols at the output to bring the code rate down to $\frac{3}{4}$.

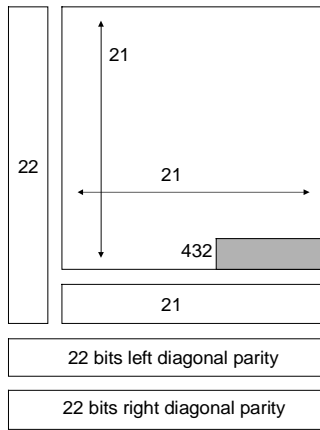


Fig. 5. Block shape matched to the specifications of DVB-RCT for 16-QAM ($R = \frac{3}{4}$).

The structure designed for $\frac{1}{2}$ code rate and 64-QAM modulation is depicted in Figure 6. Concerning this 3D case, it was necessary to pad the input block with 16 zeros in order for the diagonal parity of the eight successive planes to be properly calculated from perfect 8×8 squares.

For the same 64-QAM modulation scheme, and when the required code rate is $\frac{3}{4}$, a 2D structure like the one shown in Figure 7 is suggested. This time, 28 bits were appended to the input block for the diagonal parity computation.

IV. TURBO DECODING

A turbo decoding process was used to decode the channel-coding schemes described in Section III. For simplicity, such schemes were evaluated using BPSK (*Binary Phase Shift Keying*) modulation over AWGN (*Additive White Gaussian Noise*) and Rayleigh fading

channels, for the specified block lengths and rates listed on Table 1, Section II. Simulation of these schemes strictly in accordance with DVB-RCT is beyond the scope of this work. Similar approaches have been adopted in [4] [7] to evaluate the Circular Recursive Convolutional Code (CRSC) performance, which is specified by DVB-RCT as an alternative to the serial concatenation of Reed-Solomon with non-recursive convolutional codes, as mentioned in Section II.

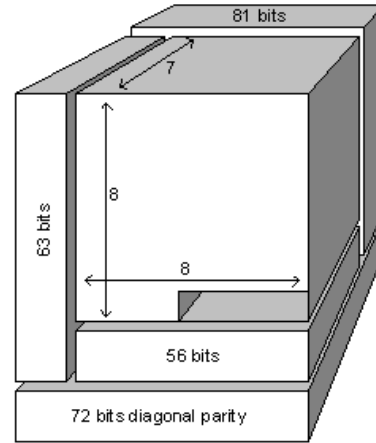


Fig. 6. Block shape matched to the specifications of DVB-RCT for 64-QAM ($R = \frac{1}{2}$).

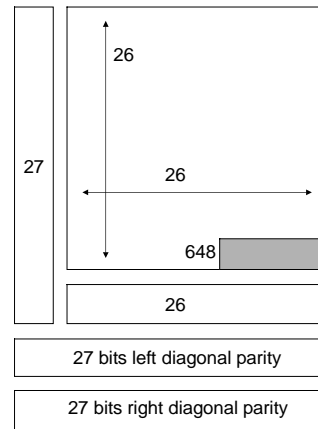


Fig. 7. Block shape matched to the specifications of DVB-RCT for 64-QAM ($R = \frac{3}{4}$).

A symbol-by-symbol MAP algorithm was used for turbo-decoding of the SPC product code, as in [13, pp. 57-60], in which the extrinsic information is factored out and passed between decoders in each dimension, referred to as a SISO (soft-input, soft-output) decoding process. A simple extension of this algorithm is done here for the case where diagonal parity is used. In this case, the diagonal encoding can be considered as a form of a new “dimension” of the code. The decoding steps are as follows:

- (i) Calculate the channel log-likelihood ratios (LLR) for all received symbols and set the extrinsic information to zero for all bits in the code;

(ii) Calculate the extrinsic information for all bits and each dimension. The *a priori* information for each bit in a given dimension is the sum of the extrinsic information from the other dimensions in the code;

(iii) The decoding of all dimensions one time is called iteration. Repeat this process for as long as required.

The *a posteriori Log-Likelihood Ratio* (LLR) of the symbol at the output of the detector (soft-input of the SISO decoder) can be expressed as [6]:

$$L(\hat{x}) = L(x|y) = \ln \left[\frac{P(x = +\sqrt{E_c} | y, g)}{P(x = -\sqrt{E_c} | y, g)} \right] \quad (1)$$

where g accounts for the fading and is called *channel gain*. Using the Bayes Theorem, from Equation (1) it is obtained:

$$L(\hat{x}) = \ln \left[\frac{P(y|x = +\sqrt{E_c}, g)}{P(y|x = -\sqrt{E_c}, g)} \right] + \ln \left[\frac{P(x = +\sqrt{E_c})}{P(x = -\sqrt{E_c})} \right] \quad (2)$$

where the first term on the right of Equation (2), which will henceforth be designated as $Lc(y)$, corresponds to the channel measurement or channel state information at the detector output, given the alternate conditions that a 1 or a 0 may have been transmitted. The second term on the right of Equation (2) will be called $L(x)$ and is the *a priori* value of the input symbol, whose value is not known until the first decoding iteration is completed. That is why, in general, the *a priori* values of the input symbols are initially set to zero.

Substituting the numerator and the denominator of $Lc(y)$ by the equations that describe the Gaussian distribution, considering the expected values in each case, according to (1), yields:

$$Lc(y) = \ln \left[\frac{\exp \left[-\frac{1}{2} \left(\frac{y - g\sqrt{E_c}}{\sigma} \right)^2 \right]}{\exp \left[-\frac{1}{2} \left(\frac{y + g\sqrt{E_c}}{\sigma} \right)^2 \right]} \right] = 4g \frac{\sqrt{E_c}}{N_0} y \quad (3)$$

where $\sigma^2 = N_0/2$ is the average noise power at the matched filter output and $g = 1$ if the channel is purely AWGN.

From the LLR at the output of the detector expressed in Equation (2), it can be shown that, for systematic codes, the log-likelihood ratio at the output of the decoder (soft output), at a given dimension q , can be written as [10]:

$$L_q(\hat{x}) = Lc(\bar{y}) + L_q(\bar{x}) + Le_q(\hat{x}) \quad (4)$$

where $L_q(\bar{x})$ is the vector or array that bear the *a priori* information that feeds the input of the decoder at the q^{th} dimension, and $Le_q(\hat{x})$ is the *extrinsic information*, or additional information due to parity, that is obtained after the decoding process at the q^{th} dimension. The extrinsic information represents the amount of information added to the soft value at the input of the decoder to form the soft output. The extrinsic information for the i^{th} bit ($i = 1, 2, \dots, n_q$) of a received codeword is given by [5] [13]:

$$Le_q(\hat{x}_i) = (-1)^{n_q} 2 \operatorname{atanh} \left[\prod_{\substack{j=1 \\ j \neq i}}^{n_q} \tanh \left(\frac{L_q(x_j) + Lc(y_j)}{2} \right) \right] \quad (5)$$

where n_q is the number of coded bits in the q^{th} dimension, $Lc(y_j)$ is the channel state information (CSI) at bit position j and $L_q(x_j)$ is the input LLR of bit j at the q^{th} dimension. Equation (5) differs from the one shown in [13] by the term $(-1)^{n_q}$, which is necessary when the logical element "0" is represented by symbol $-\sqrt{E_c}$, as it is the case of this article and of the example given in [14, p. 483-488]. When the bit "0" is represented by the symbol $+\sqrt{E_c}$, the term $(-1)^{n_q}$ is not used [11] [13]. The vector that holds the *a priori* values at the input of the decoder, according to step (ii) of the algorithm described above, can be expressed as:

$$L_q(\bar{x}) = \sum_{\substack{i=1 \\ i \neq q}}^D Le_i(\hat{x}) \quad (6)$$

where D is the number of dimensions of the code. Note that diagonal parity vectors or arrays are considered as additional dimensions in the schemes proposed here. The total LLR at the output of decoder is then calculated through:

$$L_T(\hat{x}) = Lc(\bar{y}) + \sum_{i=1}^D Le_i(\hat{x}) \quad (7)$$

where $L_T(\hat{x})$ is a vector with real numbers whose sign of each element yields a hard decision about the corresponding transmitted symbol and whose magnitude reveals the reliability of that decision. Practical aspects require the total values of the extrinsic information to be limited to avoid numerical problems during implementation. According to what has been suggested in [13], the results of the calculations to find the extrinsic information, at each dimension, were limited within the range $-100 \leq L_T(\hat{x}) \leq +100$. An estimation on the

transmitted codeword, \hat{x} , can be carried out by using the following decision criteria:

$$\hat{x} = \begin{cases} 1 & L_T(\hat{x}) \geq 0 \\ 0 & L_T(\hat{x}) < 0 \end{cases} \quad (8)$$

V. SIMULATION RESULTS

The performance as a function of the number of iterations is shown in Figure 8 for the $(10 \times 9 \times 8, 8 \times 8 \times 7)$ code depicted in Figure 6. Each decoding cycle improves the performance, with decreasing returns, but after a number of iterations equivalent to the number of dimensions of the code, this improvement, as expected [13], becomes only marginal.

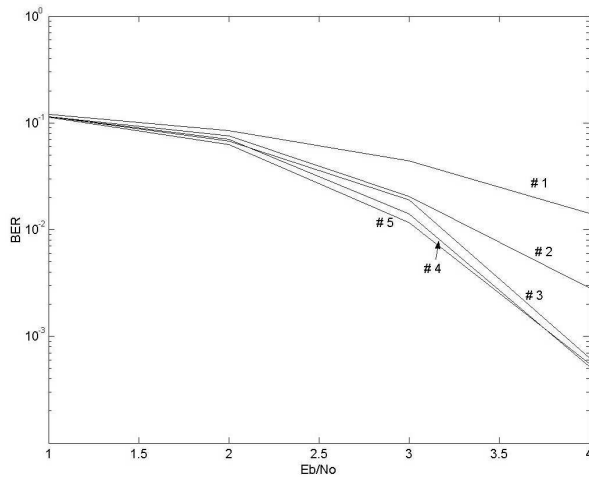


Fig. 8. BER as a function of the number of iterations for the 54-byte input block 3D product codes with diagonal parity.

The distribution of $L_T(\hat{x})$, the total LLR out of the turbo decoder after one to five iterations is shown in Figures 9(a) to 9(e), for equally likely input symbols and for $E_b/N_0 = 4$ dB.

The center of the horizontal axis of the histograms in Figure 9 is the decision threshold, on the right of which the decision is favorable to bit “1”. Otherwise, bit “0” will be chosen. The soft values of the LLR have greater magnitude the more distant they are from the threshold. It can be seen the similarity between the amplitude distribution after four and five iterations, which is in agreement with the convergence observed in Figure 8. This is the reason why the simulation results presented here for BER estimation were obtained using a fixed number of five iterations. Figure 9 also reveals that the soft values of the LLR are concentrated at regular intervals. This behavior is similar to what happens for all codes mentioned in this work, and represents an opportunity for future investigation.

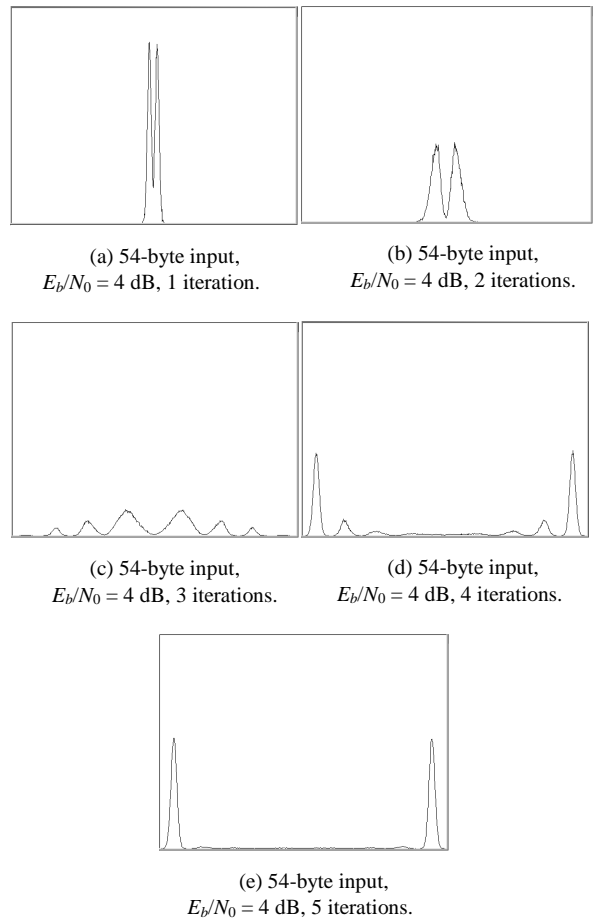


Fig. 9. Normalized histograms resulting from the measurements of LLR soft values at the output of the turbo decoder for the code $(10 \times 9 \times 8, 8 \times 8 \times 7)$ @ $E_b/N_0 = 4$ dB.

A. Performance in AWGN channel

The performance results from the simulation of the 3D product codes with diagonal parity over AWGN channel are shown in Figure 10.

The number of bytes in the legend of Figure 10 corresponds to the input block size. Coding gain is approximately 4.8 dB for the 54-byte input block size code and its distance from capacity (≈ 0.2 dB for $R = 1/2$ and BPSK modulation [6]) is around 4.6 dB, both results considering a BER = 10^{-5} . This is a quite good result, given the low complexity of the coding and the decoding processes and the relatively short codeword length.

When the required code rate is $3/4$, 2D product codes with diagonal parity are used. The performance for the corresponding codes is shown in Figure 11. If compared to the 3D case, the 2D code that operates with 54 bytes at the input requires an additional 0.6 dB of normalized signal-to-noise ratio to yield the same bit error rate of 10^{-5} .

Still referring to Figures 10 and 11, it is worthy noting that the shorter codes outperforms the longer ones for low signal to noise ratios, a fact that can be attributed to the longer sequences of zero-padding used in the longer codes. However, the asymptotic gain of the longer codes seems to make the

corresponding curves steeper, probably leading to lower BER for higher values of signal-to-noise ratios, when compared to the shorter codes.

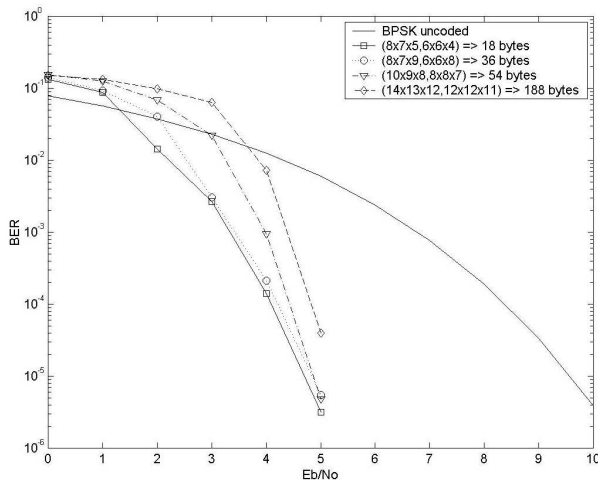


Fig. 10. Simulation results for rate $\frac{1}{2}$ 3D product codes with diagonal parity, after 5 decoding iterations: AWGN channel.

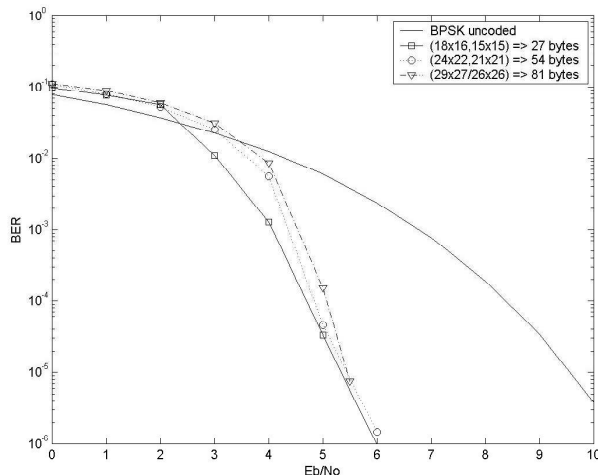


Fig. 11. Simulation results for rate $\frac{3}{4}$ 2D product codes with diagonal parity, after 5 decoding iterations: AWGN channel.

B. Performance over the Rayleigh channel

The performance of the 3D product codes *without* diagonal parity, after 5 decoding iterations, and over the *Rayleigh* channel is illustrated in Figure 12. The curves show the simulation results when the channel state information (CSI) is not available to the decoder. In this case, the bit error rate seems to be affected by an error floor after a certain value of E_b/N_0 and this behavior was also observed for codes that implement diagonal parity. The code that operates with 18 bytes at its input needs a signal-to-noise ratio of about 11 dB to yield $BER = 10^{-5}$.

For the case when the receiver is implemented with some sort of mechanism that allows the CSI to be estimated and fed to the decoder input, the performance of the 3D product codes with diagonal parity is like the one shown in Figure 13. In this case, there seems to be no error floor, at least for the signal-

to-noise ratios under consideration, and the $(7 \times 7 \times 5, 6 \times 6 \times 4)$ code now requires only 10 dB to operate with $BER = 10^{-5}$. This improvement is due to both the diagonal parity and the availability of the CSI. Because of the error floor, a larger difference on the required E_b/N_0 should be expected if the analysis is carried out at a lower BER.

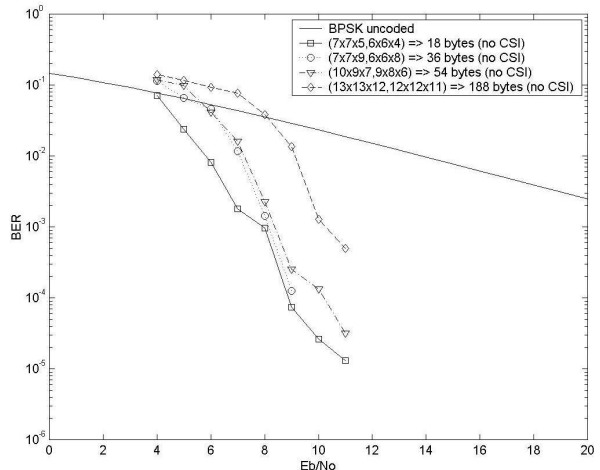


Fig. 12. Simulation results for rate $\frac{1}{2}$ 3D product codes with diagonal parity, after 5 decoding iterations: Rayleigh channel, no CSI available for the decoder.

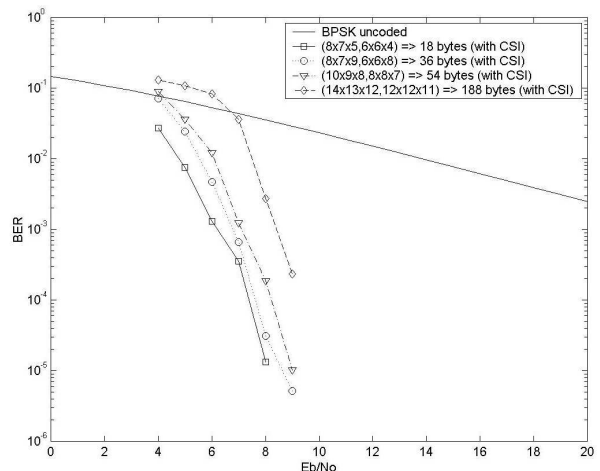


Fig. 13. Simulation results for rate $\frac{1}{2}$ 3D product codes with diagonal parity, after 5 decoding iterations: Rayleigh channel, with CSI available for the decoder.

According to Figure 13, coding gains greater than 14 dB for the 18-byte input block size code can be achieved (and greater than 11 dB for all block sizes) for bit error rates less than 10^{-3} . A distance of approximately 6.5 dB from capacity ($\cong 1.5$ dB for $R = \frac{1}{2}$ and BPSK modulation [6]) is obtained for the 18-byte input block size code, for a BER equal to 10^{-5} .

The simulation results presented herein for the fading channel were obtained under the ideal condition in which a sufficiently large interleaver is employed, in a manner that the fading that takes place in a given symbol is uncorrelated with the fading that affects the other symbols in the same codeword. Therefore, the

simulated fading was generated as independent and identically distributed random variables for each symbol transmitted through the channel.

Figure 14 shows the performance results for the case where the required code rate is $\frac{3}{4}$ and 2D product codes with diagonal parity are used. An E_b/N_0 around 10.5 dB is needed for the 54-byte input block size product code to operate with a 10^{-3} bit error rate.

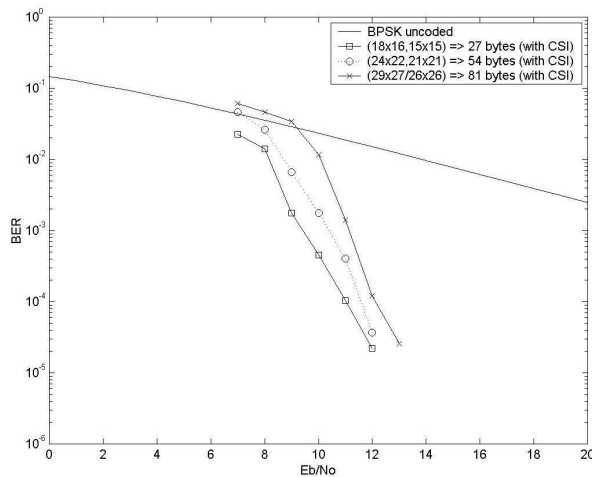


Fig. 14. Simulation results for rate $\frac{3}{4}$ 2D product codes with diagonal parity, after 5 decoding iterations: Rayleigh channel, with CSI available for the decoder.

VI. CONCLUDING REMARKS

It was found through simulation that, for low signal-to-noise ratios, the performance of the product codes is better the shorter is the block length. When E_b/N_0 increases, codes that operate with larger blocks tend to perform better. Therefore, it is expected that there is a crossing point in the curve, from which longer codes will have better performance.

The behavior for low E_b/N_0 seems to be aggravated when zeros are inserted at the output. To make matters worse, the suggested structures that have more null symbols at the output are exactly those with the larger block length. The inconveniences brought by zero-padding were partially overcome by replacing the null symbols with new parity equations, computed towards the left and right diagonal of the 2D product code and only towards the left diagonal of the several planes that compose the 3D product codes. One of the reasons why right diagonal parity has not been applied to the 3D product codes is because one of the structures (with 18 bytes at the input) does not support the additional parity bits while keeping the specified code rate at $\frac{1}{2}$.

A comparison between BER performances between the product codes presented in this article and the turbo code defined in DVB-RCT standard indicates that the latter has better performance. Although the decoding complexity of the product codes is clearly low, more investigation is needed for this complexity/performance trade-off to be exactly

quantified.

Among the future work that would explore further what was covered so far could be to investigate the performance of these codes using higher order modulation schemes, specifically 16-QAM and 64-QAM. It would also be worth investigating why the LLR's have such a concentration of amplitudes at regular intervals, according to what was seen in Figure 9. Another suggestion for future work is to confirm and investigate the reasons for the error floor, observed in a Rayleigh channel when no CSI is available for the decoder. Finally, an investigation using bounds on bit error probabilities would be useful to validate (or not) the conjecture stating that the longer codes considered here will outperform the short codes for high values of the signal-to-noise ratio.

REFERENCES

- [1] ETSI EN 301 958 V1.1.1, *Digital Video Broadcasting (DVB); Interaction channel for Digital Terrestrial Television (RCT) incorporating Multiple Access OFDM*, European Standard (Telecommunications series), December, 2001.
- [2] ETSI EN 300 744 V1.4.1, *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television*, European Standard (Telecommunications series), January 2001.
- [3] Berrou, C., Douillard, C. and Jézéquel, M. *Multiple parallel concatenation of circular recursive convolutional (CRSC) codes*, Annals of Telecommunications, Vol. 54, N°3-4, pp. 166-172, March-April 1999.
- [4] Berrou, C., Jézéquel, M., Douillard, C., Brengarth, N., Tausch, J. and Pham, N., *The Turbo Code Standard for DVB-RCS*. 2nd Int'l Symposium on Turbo Codes & Related Topics: September 2000. pp. 535-538.
- [5] Guimarães, D. A., *Turbo Decoding of Single-Parity Check Product Codes*, *Revista Telecomunicações - Inatel*, June 2002 (original title in Portuguese: Decodificação Turbo de Código Produto de Paridade Simples).
- [6] Guimarães, D. A., *A Class of Product Codes and its Turbo Decoding Applied to a Multicarrier CDMA System*. Ph.D. Thesis, Universidade Estadual de Campinas, SP, Brazil, June 2003. (original title in Portuguese: Uma Classe de Códigos Produto e sua Decodificação Turbo Aplicada em um Sistema CDMA Multiportadora).
- [7] Guinand, Paul; Kabal, Peter; Ould-Cheikh-mouhamedou, Youssouf. *Enhanced Max-Log-APP and Enhanced Log-APP Decoding for DVB-RCS*. Proc. Int. Symp. Turbo Codes, Brest: September 2003. pp. 259-263.
- [8] Hunt, Andrew W., *Hyper-codes: High-performance Low-complexity Error-correcting Codes*, Master's Thesis, Ottawa-Carleton Institute of Electrical Engineering, May 1998.
- [9] Berrou, C., Douillard, C., Jézéquel, M. *Designing turbo codes for low error rates*. *IEE Colloquium on Turbo Codes in Digital Broadcasting - Could It Double Capacity?* Electronics Letters, v.35 n.1. January 1999. p. 39-40.
- [10] Berrou, C., Glavieux, A., Thitimajshima, P., Near Shannon limit error-correcting coding and decoding: Turbo-codes. *IEEE International Conference on Communications*, v.2 pp.1064-1070. ICC 1993.
- [11] Hagenaue, J.; Offer, E., Papke, L. *Iterative Decoding of Binary Block and Convolutional Codes*. *IEEE Transactions on Information Theory*, v. 42, n. 2, March 1996. pp. 429-445.
- [12] Honary, B. and Markarian, G., *Trellis Decoding of Block Codes: A practical approach*, Kluwer, 1997.
- [13] Rankin, David M., *Single Parity Check Product Codes and Iterative Decoding*, Ph.D. Thesis, University of Canterbury, Australia, May 2001.
- [14] Sklar, Bernard. *Digital Communications - Fundamentals and Applications*. 2nd. ed. Upper Saddle River: Prentice Hall, 2001.