A Class of Product Codes and its Iterative (Turbo) Decoding

D. A. Guimarães[1] and J. Portugheis[2]

[1]Inatel, Av. João de Camargo, 510, CEP 37540-000, Santa Rita do Sapucaí, MG, Brasil
[2]DECOM, FEEC, Unicamp, CP 6101, CEP 13083-970, Campinas, SP, Brasil
[1]E-mail: dayani@inatel.br, [2]E-mail: jaime@decom.fee.unicamp.br

Abstract: *This paper describes a class of D-dimensional product codes of length n^D, rate $(½)^D$ and minimum distance 4^D. The key feature of this class is that its iterative (turbo) decoding with a soft-input soft-output (SISO) algorithm is based on a very simple minimum distance (MD) decoding of the component codes. Simulation results are compared to bounds on MD performance for both the AWGN and flat Rayleigh fading channels. They show the main advantage of such a coding/decoding scheme: a good tradeoff between complexity and performance.*

Keywords: Channel coding, multidimensional product codes, SISO algorithms, iterative (turbo) decoding.

1. INTRODUCTION

This paper considers a class of low rate multidimensional product codes. This class is formed by using the same non-systematic component code in each dimension. The component code is a specific example of application of the generalized code concatenation technique [1]. The description given here is based on a similar description given in [2] for the construction of generalized code concatenation over rings. The key feature of this component code is that it is possible to derive for it a very simple MD decoding algorithm based on applying Wagner decoding [3] twice. This feature guarantees a decoding complexity for the product code similar to that of the single parity check product codes described in [4]. In order to achieve an even lower complexity, a modified form of Pyndiah´s SISO decoding algorithm [5] is applied.

The paper is organized as follows: Section 2 describes the considered class, whereas Section 3 explains briefly its iterative decoding algorithm. Section 4 analyzes the performance results, and, finally, Section 5 is devoted to some conclusions and final comments.

2. DESCRIPTION OF THE CLASS

Let $\boldsymbol{c}_1$ be a codeword of the binary repetition code $\boldsymbol{C}_1 = (n/2, 1, n/2)$ and $\boldsymbol{c}_2$ be a codeword of the binary single parity-check code $\boldsymbol{C}_2 = (n/2, n/2\text{-}1, 2)$. Then, a codeword $\boldsymbol{c}$ of the non-systematic code $\boldsymbol{C} = (n, k, d_{\min}) = (n, n/2, 4)$ can be expressed as

$$\boldsymbol{c} = [01]\boldsymbol{c}_1 \oplus [11]\boldsymbol{c}_2 \tag{1}$$

where the sum ⊕ is over GF(2) and the product $[01]\boldsymbol{c}_1$ is calculated by substituting a 0 in $\boldsymbol{c}_1$ by 00 and a 1 by 01. The same is done for $[11]\boldsymbol{c}_2$, where now a 1 in $\boldsymbol{c}_2$ becomes 11.

By using the same non-systematic code $\boldsymbol{C}$ as the component code in each of the D dimensions, a product code of length n^D, rate $(½)^D$ and minimum distance 4^D is obtained. Figure 1 shows how to construct such a code when D=3 and n=8. The three-dimensional cube of 64 information bits consists of four 4×4 two-dimensional arrays. In the first step, each array is encoded row-by-row, yielding four 4×8 two-dimensional arrays and a total of 128 coded bits. Then, in the second step, each of these arrays is encoded column-by-column, yielding four 8×8 two-dimensional arrays and a total of 256 coded bits. Finally, in the third step, these arrays are interpreted as consisting of eight 4×8 two-dimensional arrays and encoded in the direction of dimension d=3 to yield a block of 512 coded bits.

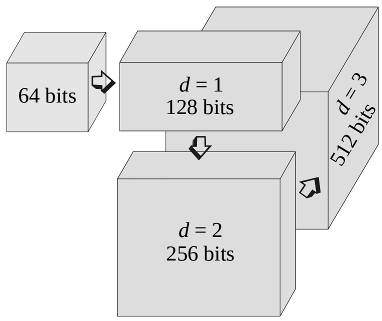

Figure 1: Construction of a three-dimensional product code with (8,4) component codes.

Starting with k^D information bits, it is possible to construct a block of coded bits in D dimensions. In this case, one should interpret the D-dimensional product code as a serial concatenation of D codes separated by D-1 block interleavers in which the number of columns is $N_c = n$ and the number of rows of the interleaver between the code d and d+1 is

$$N_r = n^{d-1}k^{D-d} \tag{2}$$

If (2) is accomplished, it is possible to verify [6] that all the $n^{D\text{-}1}$ n-element vectors oriented in the “direction” of each dimension of the D-dimensional hypercube of n^D coded bits are codewords of $\boldsymbol{C}$.

The key feature of this specific component code $\boldsymbol{C}$ is that it is possible to derive for it a very simple MD algorithm. Set $\boldsymbol{c}_1=\mathbf{0}$ (the all-zero codeword) and apply Wagner decoding for a single parity-check code of length $n/2$ over the binary alphabet {00, 11}. The decision is $\hat{\boldsymbol{c}}$. Then set $\boldsymbol{c}_1=\mathbf{1}$ (the all-one codeword) and again apply Wagner decoding for a parity-check code over the alphabet {01, 10}. The decision is $\hat{\boldsymbol{c}}'$. Compare the Euclidean distances from $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{c}}'$ to the received codeword $\boldsymbol{r}$ and choose as the final decision the shortest one.

3. DECODING ALGORITHM

The iterative decoding algorithm uses the same three main decoding phases of the algorithms for single parity-check multidimensional product codes, described in [4]. These phases are: initialization, decoding in each dimension, and repetition. The main difference lies on the SISO algorithm used for decoding the codewords in each dimension. Instead of using the BCJR MAP algorithm, a modified form of Pyndiah´s SISO decoding algorithm is applied.

In the initialization phase, the channel likelihood ratios for all received noisy symbols are defined as

$$\Lambda(c \mid r, g) = gr \tag{3}$$

where g is the fading amplitude, c is the transmitted codeword symbol and r is the received channel value. For the AWGN channel, g is set equal to one.

As mentioned above, the decoding algorithm is essentially Pyndiah´s algorithm. However, instead of using Chase Algorithm to decode the codewords in each dimension, the algorithm described in Section 2 is applied in phase two. Due to the fact that it is based on applying Wagner algorithm twice, the output of this algorithm is a unique decision, $\hat{\boldsymbol{c}}_d = \hat{\boldsymbol{c}}$ (or $\hat{\boldsymbol{c}}'$), that is, no list of concurrent codewords is generated.

The repetition phase consists of repeating decoding iterations as long as required. Figure 2 shows a block diagram, representing operations for the j-th decoding step, where the maximum value of j, say $j_{\max}$, is the total number of iterations multiplied by D. The vector $\boldsymbol{R}$ represents all n^D received noisy symbols and $\overrightarrow{g\boldsymbol{R}}$ represents the symbol-by-symbol multiplication by the respective fading amplitude. The expression "decoding in one dimension" in this figure means decoding n^{D-2} $n \times n$ two-dimensional arrays in the "direction" of one of the dimensions. Decoding an array consists of decoding n rows (or n columns). Hence, "decoding in one dimension" implies applying the SISO decoding algorithm n^{D-1} times.

The model for the parameter $\beta(j)$ was chosen so as to follow a linear rule similar to that of [7], that is, $\beta(j)=K_1(j+1)/j_{\max}$. However, the model for the parameter $\alpha(j)$ was a logarithmic one, that is, $\alpha(j)=K_2\log(j)/j_{\max}$, where K_1 and K_2 were empirically chosen to be, respectively, 8 and 6 for any 2D code and 15 and 8 for any 3D code.

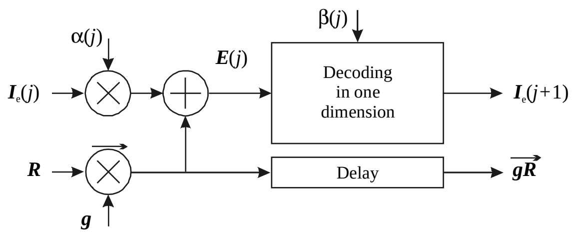

Figure 2: Decoding structure for the j-th decoding step.

As mentioned above, concurrent codewords are unavailable due to the fact that soft decoding of component code $\boldsymbol{C}$ does not generate a list of test patterns. Therefore, the reliabilities of decisions, r_d, are always obtained through

$$r_d = \beta \hat{c}_d \tag{4}$$

where $\hat{c}_d$ represents a symbol of the final decision $\hat{\boldsymbol{c}}_d$.

It is important to emphasize that the fading amplitudes are used only in computing the channel likelihood ratios in Equation (3). No other operation of Figure 2 for the j-th decoding step considers knowledge of the fading amplitudes.

4. PERFORMANCE RESULTS

This section presents some simulation and analytical performance results for two-dimensional (2D) and three-dimensional (3D) product codes constructed with component codes $\boldsymbol{C}$ = (8,4,4) and $\boldsymbol{C}$ = (12,6,4). The product codes are denoted as $(n, n/2, 4)^D$, where D is the dimension. All simulation results are for a number of iterations equal to 10. Two types of upper bounds on the bit error probability for MD decoding with BPSK signalling are also considered: true upper bounds and average upper bounds. They were obtained according to the methods for analysing serial concatenated codes described in [8], [9].

Figure 3 shows the performance for 2D and 3D product codes with (8,4,4) component codes on the AWGN channel. It can be seen that for an E_b/N_0 about 5.5 dB, the performance curve of the 2D code crosses the corresponding average bound, but it remains above the true bound, since the turbo decoding algorithm is not MD decoding. It can also be seen in Figure 3 that the performance of the 3D code is slightly better than that of the 2D one, especially for low values of E_b/N_0. Moreover, it is observed that, for both codes, the performance curves tend to follow the rate of decay of the bit error rate estimated by the bound, but for the 3D code the performance of the iterative decoding algorithm is still far from the average bound. These last observations are also valid for the results shown in Figure 4.

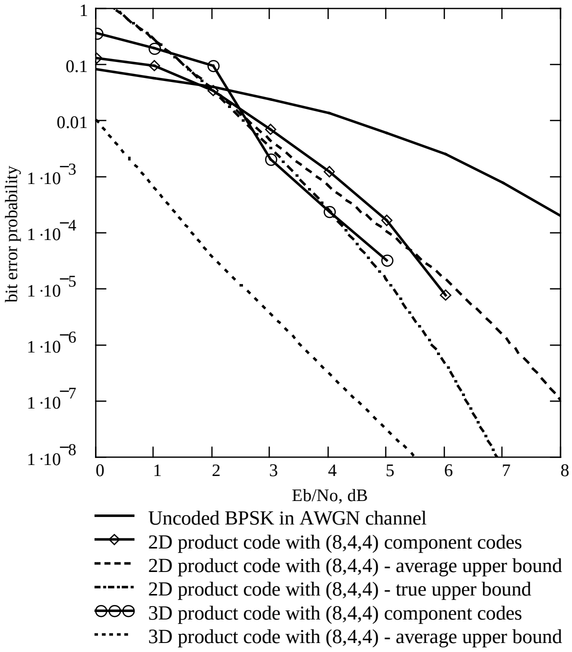

Figure 3: Performance results for the $(8,4,4)^2$ and $(8,4,4)^3$ product codes on AWGN channel.

Figure 4 shows results for the $(12,6,4)^2$ and $(12,6,4)^3$ product codes on the AWGN channel. It can be seen that for an E_b/N_0 of about 5.7 dB, the performance curve of the 2D code also crosses the corresponding average bound. For the $(12,6,4)^3$ code, one notes a significant performance improvement, in comparison with the $(8,4,4)^3$ code performance shown in Figure 3. Moreover, going from 2 to 3 dimensions with the (12,6,4) component code results in much more additional improvement in performance than that obtained with the (8,4,4) component code. In spite of the fact that the performance is around 5.2 dB away from capacity (-1,2 dB for rate 1/8 and BPSK signalling), a coding gain of approximately 6 dB is obtained with the $(12,6,4)^3$ code, for a bit error rate equal to 10^{-5}. This is quite a good result, given the low complexity of the coding/decoding process and the relatively short length of the code.

As mentioned above, for 3D codes, there is a significant gap in performance between the iterative decoding algorithm and the average upper bound. In order to reduce this gap, the iterative algorithm described in Section 3 was modified. Instead of calculating the reliabilities of decisions, r_d, by using only equation (4), the method of [5], which first searches for a concurrent codeword, was considered. Figure 5 compares the two methods for 3D codes. The method of [5] is denoted with “reliability B” whereas the method of Section 3 is denoted with “reliability A”. For low error rates, significant improvements in performance are obtained.

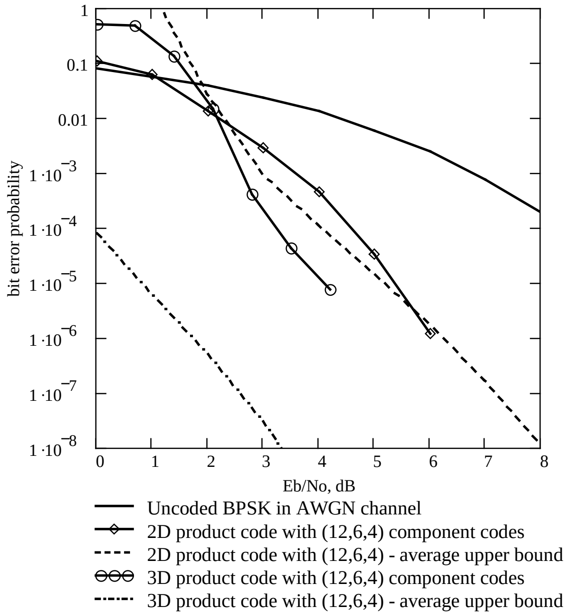

Figure 4: Performance results for the $(12,6,4)^2$ and $(12,6,4)^3$ product codes on AWGN channel.

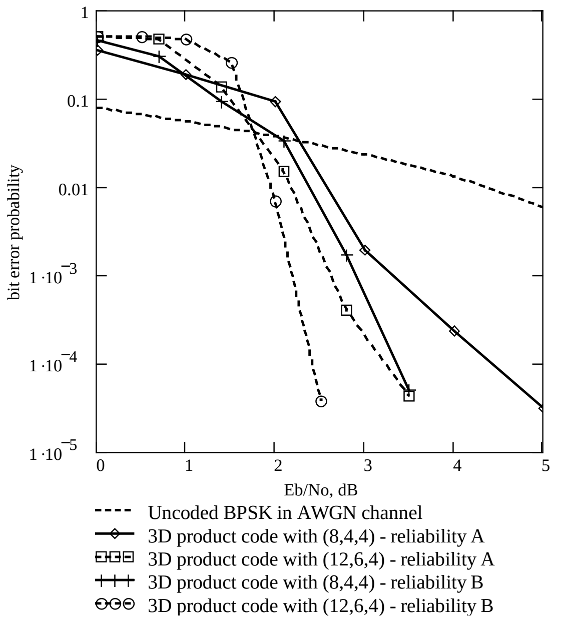

Figure 5: Performance results for the $(8,4,4)^3$ and $(12,6,4)^3$ product codes on AWGN channel with different reliability calculations.

Figures 6 and 7 show performance results on a flat Rayleigh fading channel, for product codes with (8,4,4) and (12,6,4) component codes, respectively. All results consider receiver knowledge of fading amplitudes, as was described in Section 3. A loss in performance of about 1 dB was observed, if no knowledge of fading amplitudes is assumed. Unlike the AWGN channel, going from 2 to 3 dimensions with the (12,6,4) component code results in less additional improvement in performance than that obtained with the (8,4,4) component code. Moreover, the performance for the $(12,6,4)^2$ code is slightly better than that for the $(8,4,4)^2$ one. At a bit error rate of 10^{-5}, the performance is about 6 dB and 7 dB away from capacity for the $(12,6,4)^3$ code and the $(8,4,4)^3$ code, respectively. In spite of this fact, coding gains of about 38 dB are achieved.

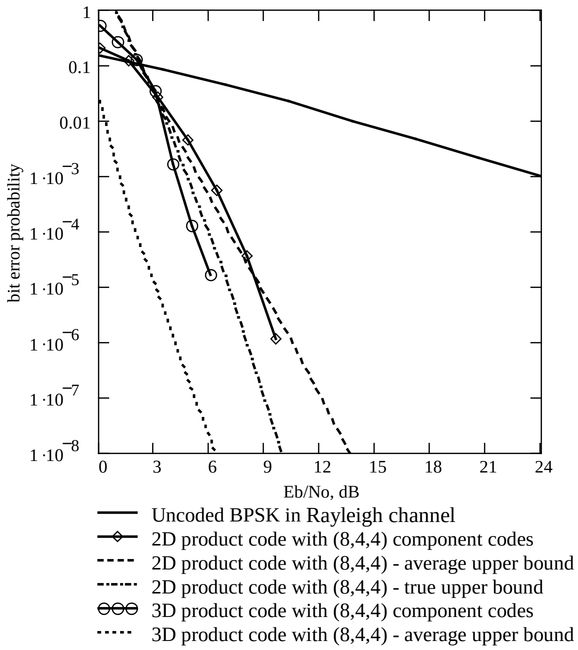

Figure 6: Performance results for the $(8,4,4)^2$ and $(8,4,4)^3$ product codes on Rayleigh channel.

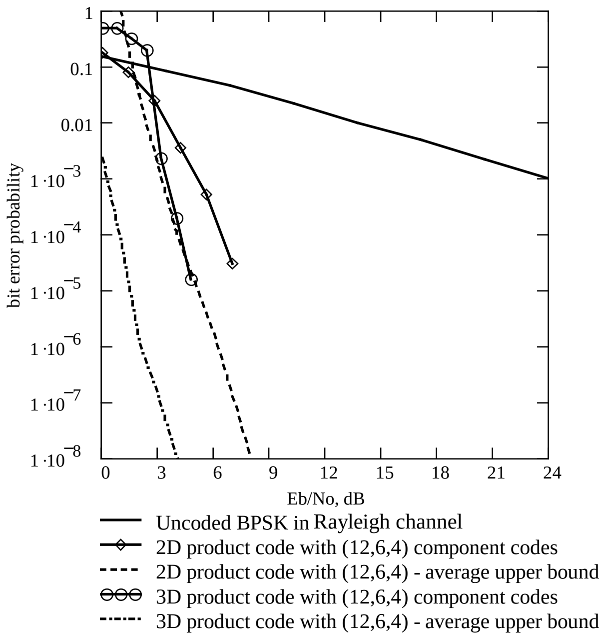

Figure 7: Performance results for the $(12,6,4)^2$ and $(12,6,4)^3$ product codes on Rayleigh channel.

5. CONCLUSIONS

A class of low rate multidimensional product codes and its iterative decoding algorithm were analysed. Its decoding algorithm is a modified form of Pyndiah´s algorithm that is based on a low complexity MD decoding procedure and does not generate concurrent codewords. Simulation results were obtained for both AWGN and Rayleigh fading channels and compared with MD performance bounds. They show that performance improves at each iteration and additional improvements can be obtained by augmenting the code dimension. Moreover, by including generation of concurrent codewords, significant performance improvements are obtained for low bit error probabilities. It remains an open question to characterize the complexity of generating concurrent codewords.

Since the suggested class of codes yields a low rate product code, and since the performance of the coding/decoding scheme unveiled good results for the fading channel, it is natural to conclude that this class should be appropriate for communication systems based on direct-sequence spread-spectrum techniques operating in Rayleigh fading channels. Performance results for this class of codes on orthogonal Multicarrier CDMA systems shall be reported in a separate work.

REFERENCES

[1] M. Bossert, *Channel-Coding for Telecommunications*, London, England: John Wiley & Sons, 2000.

[2] J. Portugheis and G. H. Kachatrian, “Linear Generalized Concatenated Codes over Ring Zq”, *Electron. Lett.*, vol. 28-3, pp. 243-244, Jan. 1992.

[3] R. A. Silverman and M. Balser, “Coding for Constant Data-Rate Systems”, *IRE Trans. Inform. Theory*, PGIT-4, pp. 50-63, 1954.

[4] D. M. Rankin and T. A. Gulliver, “Single Parity Check Product Codes”, *IEEE Trans. Commun.*, vol. 49, no. 8, pp. 1354-1362, Aug. 1998.

[5] R. M. Pyndiah, “Near-Optimum Decoding of Product Codes: Block Turbo Codes”, *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003-1010, Aug. 1998.

[6] D. A. Guimarães, *A Class of Product Codes and its Turbo Decoding Applied to a Multicarrier CDMA System*, Ph.D. thesis (in Portuguese), Unicamp, Campinas/SP, Brazil, June 2003.

[7] A. Picart and R. Pyndiah, “Adapted Iterative Decoding of Product Codes”, Globecom’99, pp. 2357-2362, 1999.

[8] S. Benedetto and E. Biglieri, *Principles of Digital Transmission With Wireless Applications*, New York: Kluwer/Plenum, 1999.

[9] E. K. Hall and S. G. Wilson “Design and Analysis of Turbo Codes on Rayleigh Fading Channels”, *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 160-174, February 1998.