УДК 621.391(075.8)

AN ALGORITHM FOR DECODING PRODUCT CODES BASED ON SOFT-DECISION DECODING OF DUAL CODES OF THE COMPONENT CODES

PHAM KHAC HOAN[1], NGUYEN TRUNG THANH[1],
NGUYEN THI HONG NHUNG[2], NGUYEN ANH TUAN[3]

[1]*Le Quy Don Technical University, Ha Noi, Viet Nam*
[2]*University of Economic and Technical Industries, Ha Noi, Viet Nam*
[3]*Belarusian State University of informatics and radioelectronics, Republic of Belarus*

Submitted 20 March 2019

Abstract. Product codes have large minimum Hamming distance and their complexity might be compared with Turbo codes. Conventional algorithms for decoding product codes have low decoding performances and very high complexities. In order to ensure the applicability of product codes in practice, this article proposes a method for decoding product codes which provide good decoding performance and acceptable complexity. Simulation results of the proposed algorithm show that, a signal-to-noise ratio of about only 4,1 dB is required in order to produce a bit error ratio (BER) not to exceed10^{-6} with rate code at 0,7034.

Keywords: product code, concatenated code, block turbo code, iterative decoding, SDDDCA-PC.

Introduction

Recently, coding theory has made great progress and has found good codes that be capable of closely approaching the Shannon capacity limit [1]. The Turbo code based on concatenation of small length component codes is one of mostly used codes. Although structure of the first concatenated code introduced was the block turbo codes (BTC), in practice, researchers mainly focus on turbo codes under the convolutional turbo codes (CTC) [2]. The reason that not much attention is paid to the BTC is the decoding performance is poor. If the same decoding performance to both the CTC and BTC to be achieved, the BTC has greater complexity.

Product codes are serial concatenated codes, and constituted of component codes which are short length block codes, firstly introduced by Elias in 1954 [3]. Product codes have large error-correction capability due to large minimum Hamming distance, however, decoding algorithms are complicated.

The iterative (turbo) decoding can be applied to product codes, using the maximum a posterior probability (MAP) to component codes. If the turbo decoding is applied to product codes, in block codes, the maximum number of vertices at each time in the trellis will be equivalent (use the famous Wolf bound [4]) to the total code words of linear code or dual code of a linear code. This is a major obstacle to the use of good block codes instead of convolutional codes in the turbo decoding concatenated codes. Obviously, the reduction in the complexity of the MAP decoding of block codes really decreases the total complexity of turbo decoding for product codes. One of the possible solutions is to increase the number of iterations in the turbo decoding to compensate for the use of the non-optimum MAP decoder for the component codes. However, as the number of iterations increases, the decoding delay increases as well. Many proposals have been implemented in order to solve the complex problem of MAP decoding for component codes.

In 1996, Hagenauer studied the turbo decoding for product codes with the MAP decoder for component codes [5]. Hagenauer proposed an optimum decoding algorithm, named soft output viterbi algorithm (SOVA), which was similar to the MAP decoding using dual codes. Although this decoding method is optimal, but it is only suitable for codes with high coding rate. Meanwhile, for linear block codes with high coding rate, it is difficult to achieve a sufficient minimum Hamming distance for modern communication applications. Moreover, the use of nonlinear functions results in very high complexity, making this method is useful for the theoretical research but difficult for practical applications [5, 6].

Later, further researches tried to reduce the complexity of the turbo decoding method for product codes, by trading off decoding time against performance [7, 8]. In 1998, Pyndiah suggested another approximation for the MAP decoding of the component codes. In this algorithm decoding a Chase type II decoder was used to obtain a list of codewords, those are the closest to the received sequence, then the MAP decoder is implemented using only a subset of those codewords instead of the whole codewords [9, 10]. This algorithm produces a decoding performance equivalent to turbo decoding with Hagenauer's MAP decoding for component codes. However, Pyndiah's approximation cannot always be explained or presented with a theoretical basis. This makes his algorithm difficult to analyze, and therefore, it is not feasible to be neither improved nor applied to other codes.

One of the new effective decoding methods is to scan all the decoding message in the dual code set of the original code set. Derived from this research direction, the article proposes a new decoding method for product codes. This method is based on soft-decision decoding of dual codes of the component codes, which are block codes with small redundancy. For codes with small redundancy, the decoding will reduce the complexity while the decoding messageremains the same as the original code. This is because the number of codewords in the dual code of the high-speed codes is much less than that of the original code [11]. This suggestion can avoid the MAP decoding of component codes, which may contribute to exploitthe capacity of error control of product code in new generation communication systems. The remainder of this article is organized as follows: Section II reviews encoding of communication system using product codes and proposes a novel algorithm with sufficient theoretical basis. Section III presents a simulation results of the proposed algorithm on a binary input additive white Gaussian noise (AWGN) channel. Finally, the conclusion is given in section IV.

The soft-decision of dual codes decoding algorithm of the component codes of product codes

1. Encoding product codes. Let $C_1(n_1, d_1, k_1)$ and $C_2(n_2, d_2, k_2)$ are linear block codes with the generator matrices $\mathbf{G}_1$, $\mathbf{G}_2$, and with the parity check matrices $\mathbf{H}_1$, $\mathbf{H}_2$, respectively.

In the encoding process, $k_1 \times k_2$ bits of message are coded to codeword with $n_1 \times n_2$ bits, the code rate is (k_1 / n_1), (k_2 / n_2) and minimum Hamming distance is $(d_1 \times d_2)$, in which d_1 and d_2 are minimum Hamming distance of C_1 and C_2 respectively. Fig. 1 illustrates the structure of product code, $C = C_1 \otimes C_2$.

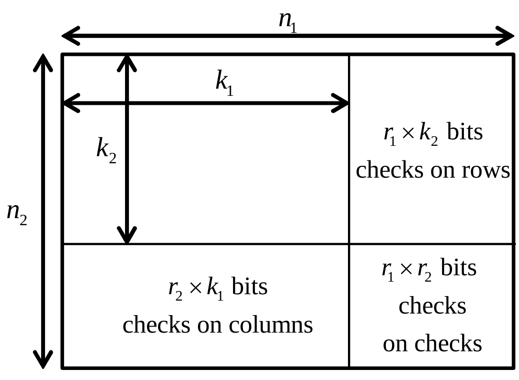

Fig. 1. Construction of product codes

At the input of the coder message u (size $k_1 \times k_2$) is coded by product code C with generator matrix $\mathbf{G}$ to produce codeword c (size $n_1 \times n_2$). A codeword c in the product code can either be generated by multiplying a $k_1 \times k_2$ long binary vector with generator matrix for C or by using the following equation:

$$c = \mathbf{G}_2^T \otimes u \otimes \mathbf{G}_1 \quad (1)$$

in which $\mathbf{G}_2^T$ is the transposed of the matrix $\mathbf{G}_2$; $\otimes$ is the Kronecker product.

Codeword c is then modulated BPSK. Suppose this codeword is modulated to binary signal ± 1 with rule $x = 1 - 2c$ and is transmitted over the Gaussian channel with zero mean and variance $2\sigma^2$. The received signal is:

$$y = x + w, \tag{2}$$

in which $w = (w_1, w_2, w_3, ..., w_n)$ is the noise vector and $y_m = x_m + w_m,\ 1 \le m \le n.$

2. Theoretical basis of the soft-decision decoding algorithm of dual codes. This section poposes the theoretical basis of the soft-decision decoding algorithm for dual codes of the component codes of product codes.

Let C^* is the dual code of C, and $c_j^* = (c_{j1}^*, c_{j2}^*, ..., c_{jn}^*)$ is its j-th codeword. Denote $P(y_m \mid i),\ i \in \{0,1\}$ as the conditional probability of the event obtaining y_m when bit node $c_m = i$ is sent and denote $\varphi_m = P(y_m \mid 1) / P(y_m \mid 0)$ as the likelihood ratio of m-th bit. It is possible to deduce $\varphi_m = \exp(-2y_m / \sigma^2).$ Let following equations:

$$A_m(0) = \sum_{t=0}^{1} \sum_{j=1}^{2^{n-k}} \prod_{l=1}^{n} \sum_{i=0}^{1} (-1)^{i(c_{jl}^* + t\delta_{ml})} P(y_m \mid i), \tag{3}$$

$$A_m(1) = \sum_{t=0}^{1} (-1)^t \sum_{j=1}^{2^{n-k}} \prod_{l=1}^{n} \sum_{i=0}^{1} (-1)^{i(c_{jl}^* + t\delta_{ml})} P(y_m \mid i), \tag{4}$$

In [5] it was proved that $A_m(0) = \lambda P(0 \mid y)$ and $A_m(1) = \lambda P(1 \mid y),$ with λ is a positive constant. It's possible to obtain:

$$\frac{A_m(1)}{A_m(0)} = \frac{P(1 \mid y)}{P(0 \mid y)} = \frac{P(1 \mid y_m) P(y_m \mid y)}{P(0 \mid y_m) P(y_m \mid y)} = \frac{P(y_m \mid 1) P(1)}{P(y_m \mid 0) P(0)} = \frac{P(y_m \mid 1)}{P(y_m \mid 0)} = \varphi_m, \tag{5}$$

with an assumption that bit 0 and 1 are sent with a same probability.

Therefore, according to equation (5), proposed decoder can decide output message based on the value of the likelihood ratio of posterior probability

$$\begin{cases} \varphi_m \ge 1 \to c_m = 1 \\ \varphi_m < 1 \to c_m = 0 \end{cases}$$

and the value of φ_m can be the decoding message for the next decoder.

On the other hand:

$$\frac{A_m(1)}{A_m(0)} = \frac{[A_m(0) + A_m(1)] - [A_m(0) - A_m(1)]}{[A_m(0) + A_m(1)] + [A_m(0) - A_m(1)]}$$

And from [5]:

$$A_m(0) - A_m(1) = 2 \sum_{j=1}^{2^{n-k}} \prod_{l=1}^{n} \sum_{i=0}^{1} (-1)^{i(c_{jl}^* + \delta_{ml})} P(y_m \mid i) = \lambda \sum_{j=1}^{2^{n-k}} \prod_{l=1}^{n} \left(\frac{1 - \varphi_l}{1 + \varphi_l} \right)^{c_{jl}^* \oplus \delta_{ml}}. \tag{6}$$

From equation (3) and (4) it's possible to deduce:

$$
\begin{aligned}
A_m(0)+A_m(1) &= \sum_{j=1}^{2^{n-k}}\prod_{l=1}^{n}\sum_{i=0}^{1}(-1)^{i(c^*_{jl})}P(y_m\,|\,i)+\sum_{j=1}^{2^{n-k}}\prod_{l=1}^{n}\sum_{i=0}^{1}(-1)^{i(c^*_{jl}+\delta_{ml})}P(y_m\,|\,i)+\\
&+\sum_{j=1}^{2^{n-k}}\prod_{l=1}^{n}\sum_{i=0}^{1}(-1)^{i(c^*_{jl})}P(y_m\,|\,i)-\sum_{j=1}^{2^{n-k}}\prod_{l=1}^{n}\sum_{i=0}^{1}(-1)^{i(c^*_{jl}+\delta_{ml})}P(y_m\,|\,i)=\\
&=2\sum_{j=1}^{2^{n-k}}\prod_{l=1}^{n}\sum_{i=0}^{1}(-1)^{i(c^*_{jl})}P(y_m\,|\,i)=\lambda\sum_{j=1}^{2^{n-k}}\prod_{l=1}^{n}\left(\frac{1-\varphi_l}{1+\varphi_l}\right)^{c^*_{jl}}
\end{aligned}
\qquad (7)
$$

Therefore, the value of φ_m can be calculated as:

$$
\frac{A_m(1)}{A_m(0)}=\varphi_m=\frac{\sum\limits_{j=1}^{2^{n-k}}\prod\limits_{l=1}^{n}\left(\frac{1-\varphi_l}{1+\varphi_l}\right)^{c^*_{jl}}-\sum\limits_{j=1}^{2^{n-k}}\prod\limits_{l=1}^{n}\left(\frac{1-\varphi_l}{1+\varphi_l}\right)^{c^*_{jl}\oplus\delta_{ml}}}{\sum\limits_{j=1}^{2^{n-k}}\prod\limits_{l=1}^{n}\left(\frac{1-\varphi_l}{1+\varphi_l}\right)^{c^*_{jl}}+\sum\limits_{j=1}^{2^{n-k}}\prod\limits_{l=1}^{n}\left(\frac{1-\varphi_l}{1+\varphi_l}\right)^{c^*_{jl}\oplus\delta_{ml}}}. \qquad (8)
$$

3. Decoding product codes based on the soft-decision decoding of dual codesof the component codes. Fig. 2 illustrates the decoding process of the soft-decision decoding of dual codes of the component codes for product codes.

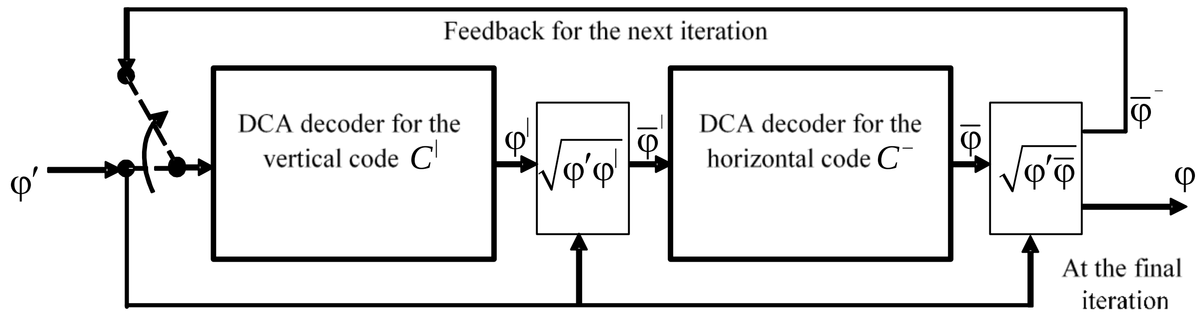

Fig. 2. Soft-decision decoding of dual codes for product codes

The decoder receives the signal matrix $y=\left[y_{uv}, 1\le u\le n_1, 1\le v\le n_2\right]$ and calculate a matrix of the likelihood ratio for each corresponding bit node $\varphi'=\left[\varphi'_{uv}, 1\le u\le n_1, 1\le v\le n_2\right]$, $\varphi'_{uv}=\exp(-2y_{uv}/\sigma^2)$. Set thematrix φ' as input of the proposed decoding algorithm (the Soft Decision Decoding of Dual Codes iterative Algorithm for Product Codes (SDDDCA-PC)). A decoder for product codes consists of two decoders: vertical $C^{|}$ and horizontal C^{-} in series. The SDDDCA-PC works as follows:

First, vertical (horizontal) decoder receives input message which is a matrix φ' and implement first iteration.

Step 1. Recalculate all values in each vertical (horizontal) in a matrix φ' to produce matrix $\varphi^{|}(n_2,n_1)$, *w*ith the corresponding value of bit node number *m* in any vertical (horizontal):

$$
\varphi^{|}_m=\frac{\sum\limits_{j=1}^{2^{n_2-k_2}}\prod\limits_{l=1}^{n_2}\left(\frac{1-\varphi'_l}{1+\varphi'_l}\right)^{c^*_{2\,jl}}-\sum\limits_{j=1}^{2^{n_2-k_2}}\prod\limits_{l=1}^{n_2}\left(\frac{1-\varphi'_l}{1+\varphi'_l}\right)^{c^*_{2\,jl}\oplus\delta_{ml}}}{\sum\limits_{j=1}^{2^{n_2-k_2}}\prod\limits_{l=1}^{n_2}\left(\frac{1-\varphi'_l}{1+\varphi'_l}\right)^{c^*_{2\,jl}}+\sum\limits_{j=1}^{2^{n_2-k_2}}\prod\limits_{l=1}^{n_2}\left(\frac{1-\varphi'_l}{1+\varphi'_l}\right)^{c^*_{2\,jl}\oplus\delta_{ml}}}. \qquad (9)
$$

In which, $\oplus$ is the modul 2 addition; $\delta_{ml}=1$ if $m=l$ and $\delta_{ml}=0$ for other cases; $c'_{a\,ij}$ is the *l*-th bit of *j*-th codeword in the dual code $C'_a(n_a,r_a)$ of the original code $C_a(n_a,k_a)$, $a\in\{1,2\}$; $\varphi'_m=P\left(y_m|1\right)/P\left(y_m|0\right)$.

At the beginning of the vertical (horizontal) decoder, set the value of geometric average of matrix $\varphi^{|}$ and the initial input a message matrix φ', as the input of next horizontal (vertical) decoding step.

Step 2. The horizontal (vertical) directly receives the decoding message calculated in step 1, which is thematrix $\overline{\varphi}^{|}$:

$$\overline{\varphi}^{|} = \sqrt{\varphi^{|}\varphi'} \tag{10}$$

Similar to step 1, recalculae all values in each vertical (horizontal) in a matrix $\overline{\varphi}^{|}$ to produce the matrix φ^{-}, with value of bit node number m in each horizontal (vertical):

$$\varphi^{-}{}_{m} = \frac{\sum_{j=1}^{2^{n_1-k_1}} \prod_{l=1}^{n_1} \left(\frac{1-\varphi'_l}{1+\varphi'_l}\right)^{c^*_{1jl}} - \sum_{j=1}^{2^{n1-k_1}} \prod_{l=1}^{n_1} \left(\frac{1-\varphi'_l}{1+\varphi'_l}\right)^{c^*_{1jl}\oplus\delta_{ml}}}{\sum_{j=1}^{2^{n_1-k_1}} \prod_{l=1}^{n_1} \left(\frac{1-\varphi'_l}{1+\varphi'_l}\right)^{c^*_{1jl}} + \sum_{j=1}^{2^{n_1-k_1}} \prod_{l=1}^{n_1} \left(\frac{1-\varphi'_l}{1+\varphi'_l}\right)^{c^*_{1jl}\oplus\delta_{ml}}}. \tag{11}$$

in which, c^*_{1jl} is the *l*-th bit of *j*-th codeword in the dual code $C^*_1(n_1, r_1)$ of the original code $C_1(n_1, r_1)$. At the output of the second decoder, recalculatethe value of geometric average of matrix φ^{-}_{m} and the initial input message matrix φ' using equation (10) to get the matrix φ^{-}_{m} as the input for the next iteration. The decoding process is carried on until the final iteration.

Step 3. Decide the output codeword based on a matrix φ taken from the final iteration.

$$\begin{cases} C_{ij} = 1 \ \text{ when } \varphi_{ij} \geq 1 \\ C_{ij} = 0 \ \text{ in other cases,} \end{cases}$$

where C_{ij} is corresponding bit node in the product codeword $\left(1 \leq j \leq n_1, 1 \leq i \leq n_2\right)$.

Evaluation of performance of proposed decoding algorithm

The performance of the proposed algorithm for AWGN channel using Monte-Carlo simulation is conducted. The proposed algorithmis applied toproduct code, consistuted of Hamming codes (7,4); (15,11); (31,26), results are received after only two iteration as in Fig. 3.

Simulation results show that when using the SDDDCA-PC for product code, the signal to noise ratio required is only about 4,1 dB to achieve $BER = 10^{-6}$ with high coding ratio (0,7034). It is not necessary to increase number of iteration because this is optimal algorithms, so two iteration can provide reasonable decoding performance. The results also show that the product code with the longer distance of component codes and the higher code rate produces better the error control capacity. In order to evaluate the practical application for product codes when using the SDDDCA-PC decoding, we shall estimate number of calculations to be used in this algorithm. Table shows the number of calculations required for decoding one bit node through 2 iteration of the proposed algorithm. Evidently, the SDDDCA-PC has acceptable complexity which is linear function. Thus, this algorithm is very suitable with product codes whose component codes have small redundancy.

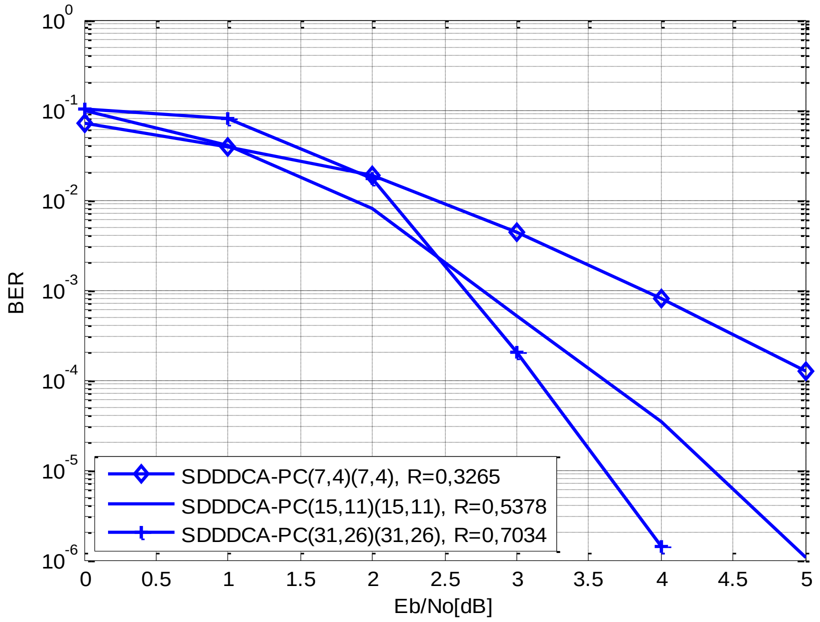

Fig. 3. Decoding performance of SDDDCA-PC for product code, consistuted of Hamming codes

Complexity of SDDDCA-PC

Algorithm	Number of multiplication	Number of addition
SDDDCA-PC	$2((n_1-1)2^{(r_1)}+(n_2-1)2^{(r_2)}+2)$	$2(2^{(r_1)}+2^{(r_2)}+2)$

Conclusion

In this article an algorithm for decoding product codes based on soft-decision decoding of component codes using dual code of the original code is proposed. The proposed decoding algorithm is based on strictly theoretical analysis and the Monte-Carlo simulation. Simulation results show that the algorithm is effective for product codes with component codes are codes with high coding rate. The proposed algorithm is applicable to modern communication systemswhereminimum processing latency is required.

References

1. Shannon C.E. // Bell Syst. Tech. J. 1948. Vol. 27. P. 379–423.
2. Morelos-Zaragoza R.H. The art of error correcting coding. John Wiley and Sons Inc., 2006.
3. Elias P. // IEEE Transactions on Information Theory. 1954. Vol. 4. P. 29–37.
4. Wolf J.K. // IEEE Transactions on Information Theory. 1978. Vol. 24. P. 76–80.
5. Hagenauer J., Offer E., Papke L. // IEEE Transactions on Information Theory. 1996. Vol. 42. P. 429–445.
6. Nickl H., Hagenauer J., Burkert F. // Proceedings of IEEE International Symposium on Information Theory. 1997.
7. Lim J.H., Lee J.H., Shin M.S., Han Cho G., Song Y.J. // Materials of International Conference on Control and Automation (CA). 2015. P. 13–16.
8. Son J., Kong J.J., Yang K. // Journal of Communications and Networks. 2018. Vol. 20. P. 345–353.
9. Pyndiah R.M. // IEEE Transactions on Communications. 1994. Vol. 46. P. 1003–1010.
10. MacWilliams F.J., Sloane N.J.A. The Theory of Error-Correcting Codes. New York (USA), 1981.