Modified Algorithm for Hard Decision Decoding of Product Codes

A. A. Al-Shaikhi
electrical engineering department
king fahd university of petroleum and minerals
Dhahran, Saudi Arabia
shaikhi@kfupm.edu.sa

M. A. Kousa
electrical engineering department
king fahd university of petroleum and minerals
Dhahran, Saudi Arabia
makousa@kfupm.edu.sa

***Abstract*—Product coding produces powerful long codes from short constituent codes. The conventional row-column decoding algorithm of the product code does not exploit its full power of correcting random errors. We propose a modification to the conventional decoding algorithm, which makes it capable of reaching the theoretical error correction capability of the code. In addition to its theoretical significance, the modified algorithm is shown to provide a gain of 0.5 dB over the conventional algorithm for AWGN channels.**

Keywords—AWGN channel; decoding; Hamming code; product code.

I. INTRODUCTION

Product codes, a kind of serially concatenated coding schemes, were introduced as early as 1954 [1]. This scheme [2] combines two codes C_1 (n_1,k_1) and C_2 (n_2,k_2), where n_i is the code length and k_i is the number of information bits, to produce the code C_p (n_1n_2,k_1k_2). Encoding is achieved by arranging the information bits in an array of k_2xk_1, and then encoding each of the k_2 rows using C_1 and each of the resultant n_1 columns using C_2. Product coding is attractive because it provides a mechanism for constructing long error correction codes without increasing the complexity of the decoder [3]. If the minimum distances of the codes C_1 and C_2 are d_1 and d_2 respectively, then the minimum distance of the resultant product code C_p is the product d_1xd_2. For example, two Hamming codes which are single-error correcting codes form a product code, call it Hamming product code, having a minimum distance of nine, and hence should be able to correct all patterns of four errors or less.

The simplest and widely adopted strategy of decoding is the two-round row-column (or column-row) hard decision decoding algorithm [3]. In this algorithm, the received matrix is first decoded row-by-row using C_1 decoder. The resultant row-decoded matrix is then decoded column-by-column using C_2 decoder. We refer to this algorithm here as Conventional Hard Decision Decoding (CHDD). The flowchart of the CHDD algorithm is depicted in Fig. 1. This algorithm is not effective, as it does not recover all error patterns promised by the minimum distance of the code [3]. Let us demonstrate this limitation for the Hamming product code applied on a rectangular four-error pattern. It can be easily seen that the CHDD algorithm fails to correct this error pattern. In fact, the CHDD algorithm will make things worse. The decoding of the rows infected with two errors will add a third error in that row. Column decoding will do the same. As a result, the decoded matrix will contain nine errors!

In fact, the error correction capability of the CHDD algorithm is given by $(t_1+1)(t_2+1)-1$, where t_1 and t_2 are the error correction capability of the codes C_1 and C_2, respectively. For the Hamming product code, the CHDD can correct all patterns of three errors or less, although it can correct other higher error patterns (but not all). For constituting codes with large minimum distances, the error correction capability of CHDD would be roughly one half of that guaranteed by the minimum distance of the product code [4].

Applying the CHDD algorithm twice may handle some of the higher error patterns, but it still does not reach the correction capability of the code. Needless to say, such algorithm increases the delay excessively.

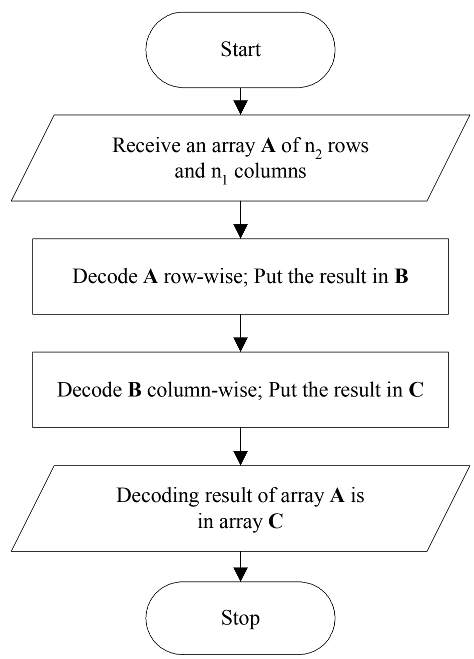

Figure 1. The CHDD algorithm

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

In this paper, we modify the CHDD algorithm to a one that achieves the theoretical error correction capability of the product code. The modified algorithm, referred to as Modified Hard Decision Decoding (MHDD) algorithm, tries to deduce the error pattern from the set of actions taken in the course of decoding. The details of the algorithm are presented in Section 2. Simulation results and comparison with the CHDD algorithm are presented in Section 3. The performance of the algorithm over AWGN channels is shown in Section 4. Conclusions and main findings are presented in Section 5.

II. THE MODIFIED ALGORITHM

The idea behind the MHDD algorithm evolved from realizing that an error pattern causes the decoder to do certain set of actions (corrections in certain locations). If it is possible to relate the actions of the decoder to a particular error pattern, the errors can then be corrected. In other words, certain actions of the decoder may be viewed as *symptoms* of a particular error pattern, and thus serve as the *syndrome* for error correction.

The MHDD algorithm, which is depicted in Fig. 2 is carried out in three steps:

Step1: **Learning**. In here, we monitor the actions of the decoder and record the number of corrections and their locations. The learning step is applied to each of the following:
(a) Row decoding of the received undecoded matrix.
(b) Column decoding of the received undecoded matrix.
(c) Column decoding of the row-decoded matrix (that is in a).

Step2: **Identification**. The information acquired in the learning step is examined against a pre-defined set of conditions in an attempt to identify the error pattern in the undecoded matrix. Those conditions are derived from analyzing the response of the decoder to a known error pattern. Consider again the example of the Hamming product code applied on the rectangular four-error pattern. This pattern causes the decoder to make two inversions in the same column during row decoding, and three inversions in the same row during column decoding, where one of those three inversions lies in the same column of the two inversions which took place earlier in row decoding. Such an observation is then translated to a set of conditions on the numbers and locations of errors as learned from step 1. Obviously, the conditions are designed in such a way that they correspond to a particular error pattern in an "if-and-only-if" relation. Based on this examination, the algorithm can tell if the symptoms:
(i) are those of a pattern correctable by the CHDD algorithm; if so, the resultant matrix of Step 1.c is assumed to be the correct codeword, and the algorithm is terminated.
(ii) are those of a pattern uncorrectable by the CHDD algorithm; the associate error pattern is then identified.

Step3: **Decoding**. Based on the identification concluded from step2, the erroneous bits are inverted.

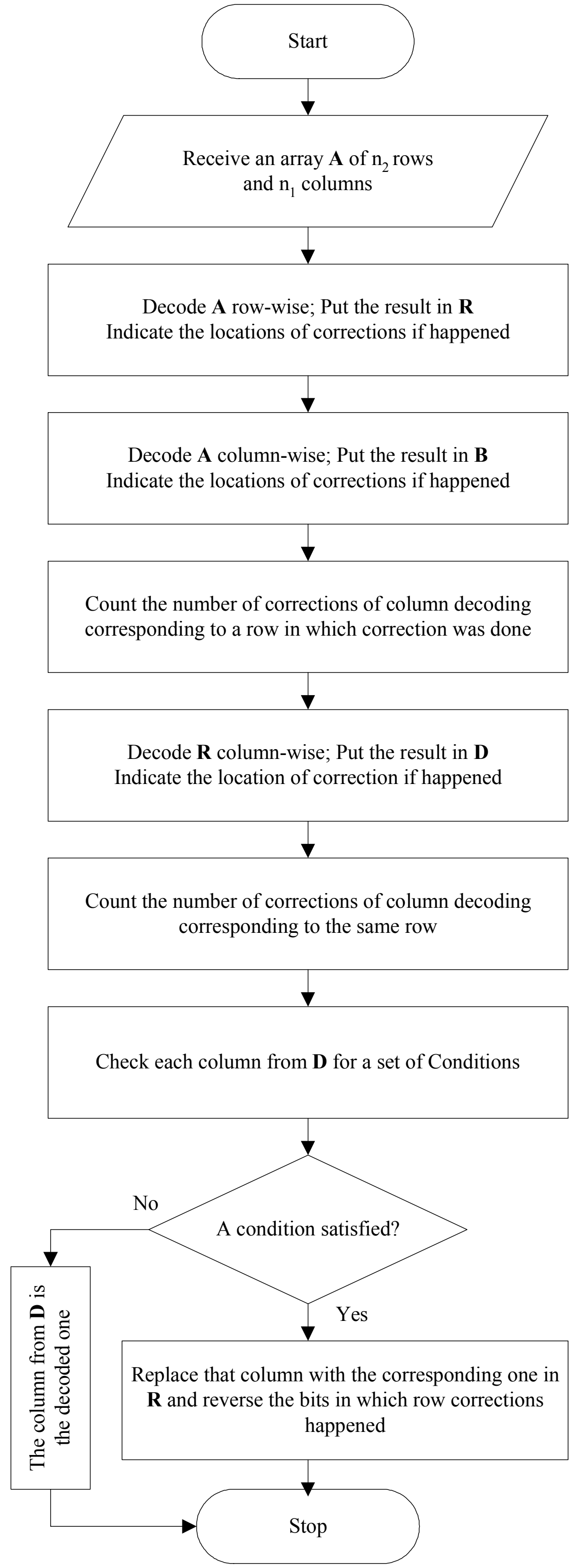

Figure 2. The MHDD algorithm

III. RESULTS AND DISCUSSION

The MHDD algorithm was simulated for a (49,16) product code formed from the same (7,4) Hamming code for row and column encoding. The minimum distance of this code is nine and is thus a 4-error correcting code. We generated all error patterns up to eight errors and applied both the CHDD and the MHDD algorithms to the corrupted codewords. The results of the decoding algorithms are summarized in Table 1. It is worth noting the following:

1) The MHDD algorithm corrected all 4-error patterns, while the CHDD algorithm fails to correct 4.37% of those patterns.
2) For higher error patterns, the capabilities of the two algorithms are comparable, with the MHDD algorithm being marginally better.
3) Except for 5-error patterns, all error patterns correctable by the CHDD algorithm are correctable by the MHDD. The inverse is not true; the CHDD algorithm fails to detect some of the patterns that the MHDD algorithm can correct.
4) Of significant concern to the performance of the product code is the post-decoding bit error rate. The table shows the average number of bits in error after decoding, using both algorithms. For example, out of a 5-error pattern, the MHDD algorithm produces 2.8 erroneous bits, while the CHDD algorithm produces 4 erroneous bits on the average. It indicates that even when the matrix is not correctable, the MHDD algorithm provides a better "repair" to the fault. For higher-error pattern, the two algorithms behave almost the same in this regard.

IV. PERFORMANCE OVER AWGN CHANNEL

We next examine the performance of the product code using the MHDD algorithm over AWGN channel and how it compares to the CHDD algorithm. The codeword error probability is given by:

$$P_C = \sum_{i=t+1}^{n} \binom{n}{i} \varepsilon^{i} (1-\varepsilon)^{n-i} \qquad (1)$$

Where ε = the channel bit error probability.

The post-decoding bit error probability can be approximated by:

$$P_{bit} \approx \frac{1}{n} \sum_{i=t+1}^{n} i\, B_i\, \varepsilon^{i} (1-\varepsilon)^{n-i} \qquad (2)$$

In deriving (2), it is assumed that all patterns of more than t errors are not correctable. It is further assumed that the number of bit errors after decoding is, on the average, equal to the number of bit errors before decoding. In our case, we removed these assumptions for $t \leq 8$ and used the figures for the number of uncorrectable patterns, B_i, and the average number of bit errors from Table 1. Higher error patterns are assumed completely uncorrectable.

TABLE I. COMPARISON BETWEEN CHDD AND MHDD ALGORITHMS OF THE (49, 16) PRODUCT CODE

Number of Introduced Errors	Total Number of Error Patterns	Total Number of Bits	CHDD Algorithm		MHDD Algorithm		Percentage of Corrected Patterns by CHDD and MHDD	Percentage of Corrected Patterns by MHDD not CHDD	Percentage of Corrected Patterns by CHDD not MHDD	Percentage of Not Corrected Patterns by CHDD and MHDD
			Average # of bits in error after decoding per codeword	Percentage of uncorrected patterns	Average # of bits in error after decoding per codeword	Percentage of uncorrected patterns				
4	211876	10381924	3.857	4.371	0	0	4.371	4.371	0	0
5	1906884	93437316	3.986	18.432	2.826	17.576	0.925	0.925	0.0694	17.507
6	13983816	685206984	4.408	42.003	4.421	41.826	0.1766	0.1766	0	41.826
7	85900584	4209128616	5.223	67.322	5.195	67.318	0.004107	0.004107	0	67.318
8	450978066	22097925234	6.526	85.532	6.526	85.515	0.0172	0.0172	0	85.515

Simulation of the bit error rate of the two algorithms for (49,16) product code for BPSK over AWGN channel is shown in Fig. 3. Equation (2) is also plotted in Fig. 3. The figure shows that the MHDD algorithm provides about 0.5 dB gain over the CHDD algorithm at a BER of 10^{-5}. Also, it can be seen that (2) provides a very good estimate of the performance of the CHDD and MHDD algorithms.

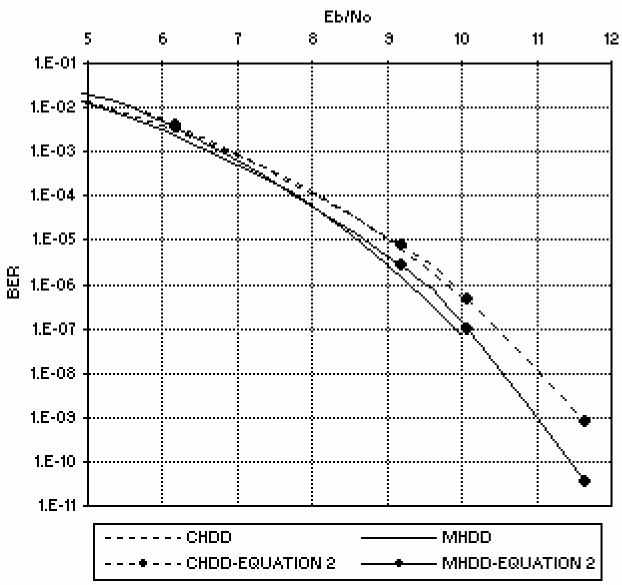

Figure 3. Hard decision decoding performance of the (49,16) product code over AWGN channel.

V. CONCLUSION

In this paper, we presented a modification to the conventional row-column hard decision decoding of product codes. The main feature of the modified algorithm lies in its capability to correct error patterns up to the theoretical limit guaranteed by the minimum distance of the code. It also reduces the average number of bit errors after decoding for the most dominant error patterns. The proposed algorithm goes through the steps of learning the actions of the decoder, identifying the error pattern and finally decoding. When tested over AWGN channel, the modified algorithm provided a gain of 0.5 dB at practical bit error rates.

Finally, it is to be mentioned that the modified algorithm consumes more processing time as compared to the conventional algorithm. Our simulation for the (49,16) product code indicates that the ratio reaches that of 5:1. With today's advancement in technology and processing speed, the added delay can be easily absorbed.

ACKNOWLEDGMENT

The authors acknowledge the support of KFUPM

REFERENCES

[1] Elias, P., "Error Free Coding," *IRE Trans. Inf. Theory,* 1954.

[2] Rhee, Man Young, **Error-Correcting Coding Theory**, USA, *McGraw-Hill Inc.* 1989.

[3] Sweeney, P., **Error Control Coding: an Introduction**, UK: *Prentice Hall International Ltd*. 1991.

[4] Wilson, Stephen, **Digital Modulation and Coding**, New Jersey: *Prentice Hall*. 1999.