A New Hard-decision Iterative Decoding Method for Hamming Product Codes

Xunhuan Ren, Jun Ma, Viktar Yurevich Tsviatkou, Valery Kanstantinavich Kanapelka

Abstract—**Iterative decoding is quite useful for decoding Hamming product codes. The idea behind the method is to divide the entire decoding process into sequence steps, such that later phases can use the output of previous stages to produce their results. Hard-decision iterative decoding is easy to implement, has low complexity and can correct many error patterns with quantities of error bits greater than half the minimum distance of the code. However, the existing hard-decision iterative decoding methods fail to correct the errors up to half the minimum distance of the code due to the existence of stall patterns. In this paper, a new hard-decision iterative decoding method is proposed to overcome this limitation. The proposed method is implemented and assessed based on error bit experiments and channel experiments (including the Gaussian channel and binary symmetric channel). The results show that the proposed method outperforms other methods.**

Index Terms—**Hamming Product Codes, Iterative Decoding, Hard Decision, Stall Pattern, Gaussian Channel, Binary Symmetric Channel.**

I. INTRODUCTION

Product coding techniques enable the constituting of long and powerful code using short component codes, which were first introduced in 1954 by Elias [1]. Product codes have many good characteristics [2]. One of them is that product codes can correct burst errors. Clearly, all burst error patterns that meet the number of row (column) errors less than half the minimum distance of the column (row) are correctable. In addition, product codes can handle random errors when the number of mistakes in each row (column) is less than or equal to half the minimum distance of the current row (column) codes. Another vital characteristic is that the covering radius of a product codes is much larger than half the minimum distance of the code [3,4]. This provides the possibility of correcting partial error patterns whose number is greater than half the minimum distance of the code using a proper decoder.

Decoding methods for product codes can roughly be divided into hard-decision decoding and soft-decision decoding. The hard-decision decoding method, also called the iterative decoding method, was initially suggested by Elias in [1] and then presented adequately in [5]. Decoders derived from this idea have high efficiency and can rectify most error patterns up to half the minimum distance. However, an iterative decoder's performance is limited because it fails to correct all error patterns whose weight is less than half the minimum distance [6]. These error patterns are named stall patterns. Many improved iterative decoding methods have been proposed [6-12] to overcome this disadvantage. Soft-decision decoding methods have better decoding results than hard-decision decoding methods, but they tend to have high complexity. Additional information indicates that the reliability of each input data point is required in a soft-decision decoding method. Methods such as those presented in [6,13-14] are derived from this idea.

A Hamming product code is one of the most attractive product codes since it comprises Hamming codes. Hamming codes provide easier coding schemes and simpler hardware to correct errors, yet only for single-bit errors [15-16]. The hard-decision decoding of Hamming product codes is a heated topic in this field; therefore, many different decoding methods used in Hamming product codes have been proposed in the past, such as those presented in [8-12,17]. However, these methods are based either on extended Hamming codes instead of the standard Hamming code or on methods that fail to correct all error patterns up to half the minimum distance.

The main contribution of this paper is that we have proposed a novel hard-decision decoding method for Hamming product codes whose component codes are all Hamming codes rather than extended Hamming codes. Furthermore, this decoding method can correct all error patterns that do not exceed half the minimum distance of the code.

The remainder of this paper is organized as follows: Some preliminary concepts are introduced, and some important related works are briefly reviewed in section 2. Then, the proposed method is described in section 3. Next, section 4 presents the experimental results. Finally, the conclusion and future work are given in section 5.

II. PRELIMINARY CONCEPTS AND RELATED WORKS

A. Basic Concepts

In coding theory, a linear block code C can be represented by (n,k,d), where n is the length of the code, k is the length of the message bits (also named the dimension in some literature) and d is the minimum Hamming distance, which

Manuscript received January 12, 2022; revised May 13, 2022.

Xunhuan Ren is a Postgraduate student of Belarusian State University of Informatics and Electronics, Minsk, Belarus. e-mail: rxh1549417024@gmail.com).

Jun Ma is a Postgraduate student of Belarusian State University of Informatics and Electronics, Minsk, Belarus. (e-mail: majun1313@hotmail.com).

Viktar Yurevich Tsviatkou is an Associate Professor at the Department of Info-communication Technology, Belarusian State University of Informatics and Electronics, Minsk, Belarus. (e-mail: vtsvet@bsuir.by).

Valery Kanstantinavich Kanapelka is a Professor at the Department of Info-communication Technology, Belarusian State University of Informatics and Electronics, Minsk, Belarus. (e-mail: volos@bsuir.by).

can be calculated using the following formula:

$$d = min\left\{d_H\left(C_i, C_j\right)i, j = 1,\ldots,k), i \neq j\right\} \quad (1)$$

where d_H is the Hamming distance, which is easy to compute using formula (2).

$$d_H\left(C_i, C_j\right) = sum\left(mod\left(C_{im} + C_{jm}, 2\right)\right) \quad (2)$$

The code rate, also named the information rate in some studies, is used to describe the redundancy degree of one given code.

$$R = k / N \quad (3)$$

The detection and error correction capability of a linear block code depend on the value of distance *d*. If a linear block code *C* is given and its distance is *d*, then this code can detect t_d bit errors and correct t_r bit errors, which can be calculated using the following formula.

$$t_d = d - 1 \quad (4)$$

$$t_r = floor\left(\left(d - 1\right) / 2\right) \quad (5)$$

Hamming codes are a type of binary linear block code. Its code length *n* is $2^r - 1$, length of message bit *k* is $2^r - 1 - r$ and distance *d* is 3, and *r* is an integral greater than 2. Therefore, Hamming codes can correct all patterns with a 1-bit error and can detect all patterns with two errors.

The product codes can be easily constructed using two codes. Given a linear block code *A* with parameters $\left(n_1, k_1, d_1\right)$ and another linear block code *B* with parameters $\left(n_2, k_2, d_2\right)$, their product codes $P = A \times B$ can be obtained after conducting the following steps.

Step 1: Lay the $k_1 \times k_2$ message bits on a matrix whose number of rows is k_2 and number of columns is k_1.

Step 2: Encode the k_2 rows with the encoding rules of code *A*.

Step 3: Encode all columns with the encoding rules of code *B*.

The whole process of the construction of product codes is shown in Fig. 1.

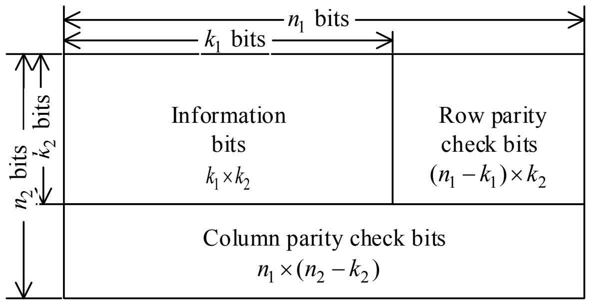

Fig. 1. Construction of product codes

The parameters of the final product code are $\left(n_1 n_2, k_1 k_2, d_1 d_2\right)$.

Since Hamming product codes are constructed using two Hamming codes whose code distance is three, the distance of Hamming product codes is nine. Therefore, a suitable hard-decision decoder for Hamming product codes should correct all errors less than or equal to 4 and can correct as many errors as possible greater than 4.

B. Related Works

The hard-decision decoding method for product codes is an iterative decoding method.

The two-step method is the most primitive decoding method proposed in [17], which is described as follows.

Step 1: Calculate the syndrome of all columns of the received code, and correct all possible errors according to the decoding method corresponding to the column encoding method.

Step 2: Calculate the syndrome of all rows of the code, and correct all possible errors according to the decoding method corresponding to the row encoding method. Then, we can obtain the final estimated code.

This decoding method can properly correct many error patterns beyond half the minimum distance of the code. However, it fails to fix all error patterns within half the minimum distance of the product code. As a result, this method may suffer decoding failure from stall patterns when applied to Hamming product codes.

Three-step decoding methods, such as those presented in [8-10,12], employ an erasure process to relieve the influences of stall patterns. However, to identify some stall patterns, they introduce component code with a larger code distance, detecting some uncorrectable errors. These uncorrectable errors can provide approximate reference information for conducting erasure later. These methods can correct more error patterns than two-step decoding, but they are still not able to amend all error patterns less than or equal to half the minimum distance of the product codes. This is because the minimum distance of the product codes increases with increasing component code. For Hamming product codes, some known methods introduce the extended Hamming code as component codes partly or wholly. In [8,9], the author constructed a Hamming product code with one extended Hamming code and one Hamming code so that the row code could detect all uncorrectable double-bit errors. Associated with their elaborate three-step iterative decoding, in which the row status vector and column status vector are used to provide the reference information for erasure, the method can properly decode some stall patterns with four-bit errors. However, their approach is still not able to correct the mistakes up to half the minimum distance of the product code. Moreover, all component codes used in [10,12] are extended Hamming product codes. Their capability of error correctness is further improved, but they still fail to correct errors up to half the minimum distance of the code.

Therefore, we decided to develop a novel iterative decoding method for Hamming product codes. The proposed method can correct the errors up to half the minimum distance of the code and correct many mistakes beyond this range.

III. Proposed Method

The proposed method is a three-step hard-decision iterative decoding method in which an erasure operation and decision operations are embedded. The major difference between our method and others is that we conduct a decision operation.

Our method can be divided into three procedures: the decision procedure, decoding procedure and erasure-decoding procedure.

The proposed method first calculates in parallel the syndrome of the received code from the direction of the row and column. When errors have occurred in one row or one column, this row or column's syndrome will not equal a zero vector. Therefore, it is very convenient to record an error row or column in a row or column register using the characteristics of the syndrome. Both the row and column registers are binary registers, in which 1 denotes that some errors exist, and 0 indicates that there is no error. Comparing the total number of error rows and that of the error columns can provide a reference to decide how to conduct the subsequent decoding. This is called a decision procedure.

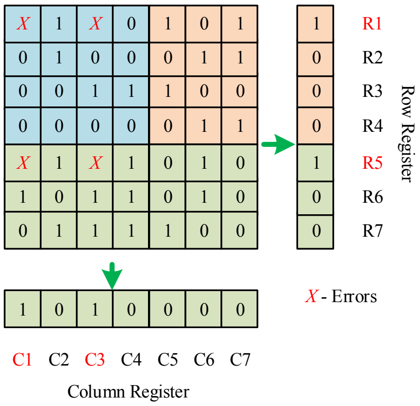

Fig. 2. Example of a 4-bit error pattern

The decoding process is performed when the total number of error rows is not equal to that of the error columns, or if they are equivalent, the total number is not less than the minimum distance d of the component code. Furthermore, suppose the total number of error rows is greater than that of the error columns, or they are equivalent but the total number is greater than or equivalent to the minimum distance *d*. In that case, it is necessary to conduct a continuous row-column-row three-step decoding. However, if the total number of error rows is less than that of the error columns, it is necessary to conduct a subsequent column-row-column three-step decoding. Here, it is noted that each decoding step is standard Hamming decoding.

The erasure-decoding procedure is applied when the total number of error rows is less than or equal to that of the error columns and this number is less than the component code distance d. We flip the bits where the row register and column register are all one and follow a row decoding process. To further illustrate this procedure, an example is given in the following.

Suppose there are rectangular 4-bit errors that have occurred in the received code, and we take the (49,16,9) Hamming product constructed from two (7,4,3) Hamming codes as an example.

In Fig. 2, '*X*' denotes an error bit. The row register and column register save the result obtained in the decision procedure. Clearly, the total number of ones in the row register is equal to that in the column register, and this number is 2, which is less than or equal to the minimum distance (which is three in this example) of the component code. Therefore, for this type of error, we can use the erasure-decoding procedure to correct it. The row register and column register provide the coordinates where we need to conduct the erasure. In this example, since the R_1, R_5 bit in the row register and C_1, C_3 bit in the column register are one, it is necessary to flip all bits whose coordinate values are (R_1, C_1), (R_1, C_3), (R_5, C_1) and (R_5, C_3). Then, one-time row decoding is performed to correct the remaining error if it exists.

A flowchart of the proposed method is shown in Fig. 3.

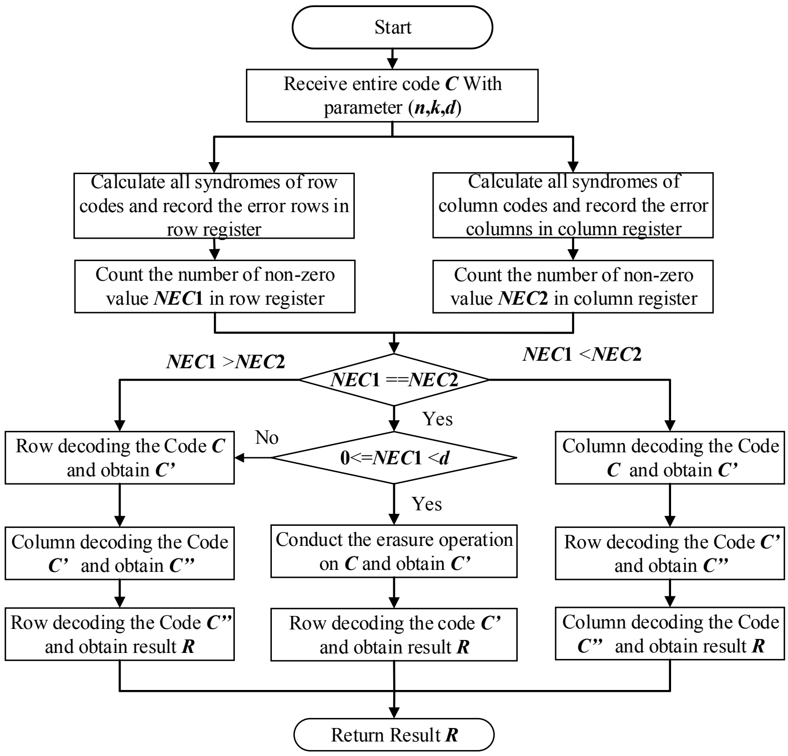

Fig. 3. Flowchart of the proposed method

IV. Experiment And Results

A. Experimental Platform

1. Hardware

The computations were conducted on a standard PC laptop with a Core i7-4720Q CPU (2.6 GHz) and 16 GB of memory.

2. Software

All experiments were performed in MATLAB R2017b, and partial data were processed in Microsoft Excel 2010. The operating system of the PC was Windows 8.1.

B. Overall structure of the experiments

To better analyze the performance of the proposed method, three different experiments were performed. The first experiment was conducted on the Gaussian channel, where additive white Gaussian noise (AWGN) was introduced. The second experiment was performed on the binary symmetric channel (BSC), where we manually set the error probabilities. The third experiment explored the error correction capabilities of different decoders with a given number of errors. We constructed Hamming product codes with parameters (49,16,9) and implemented the proposed method, two-step decoding method [17], and three-step decoding method [12].

C. Experiment on the Gaussian Channel

This experiment aimed to explore the performance of all three decoding methods on the Gaussian channel. The bit error rate (BER) [18,19] and word error rate (WER) were measured under different signal-to-noise ratios. During the experiment, we repeated 100,000 encoding and decoding processes. A flowchart of the experiment for each algorithm is shown in Fig. 4.

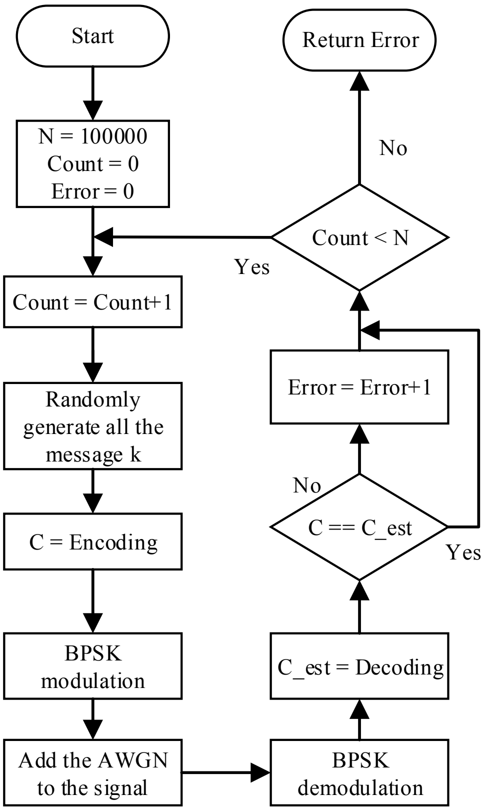

Fig. 4. Flowchart of the experiment on the Gaussian channel

Fig. 4 shows that all message bits were generated randomly, and then the entire code was obtained in the encoding process, which was performed according to the method mentioned in section 2. Instead of directly sending the code to the channel, the code was first changed into a signal that could be transferred in the Gaussian channel using binary phase-shift keying (BPSK) modulation [20]. In the channel, AWGN was added to the signal, which could cause some errors to occur. The degree of noise was influenced by the signal-to-noise ratio, which was set manually. The signal was first demodulated, and then each of the three decoding methods was implemented to correct the error and estimate the original code on the receiving end. By comparing the estimated code with the original code, we could count the instances of a decoding error made by the different methods. In Fig. 5 and Fig. 6, we present the results of this experiment.

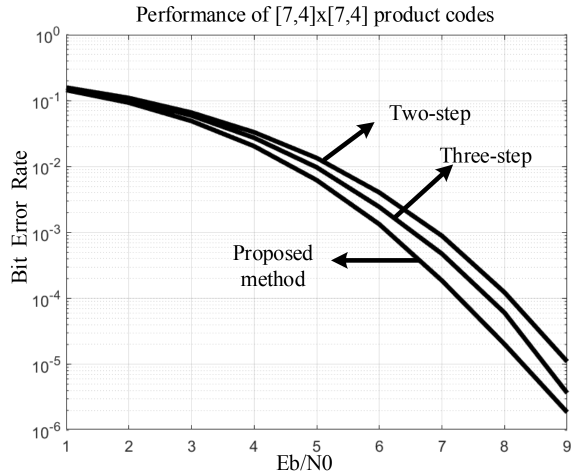

Fig. 5. Bit error rate for the (49,16,9) Hamming product codes on the Gaussian channel

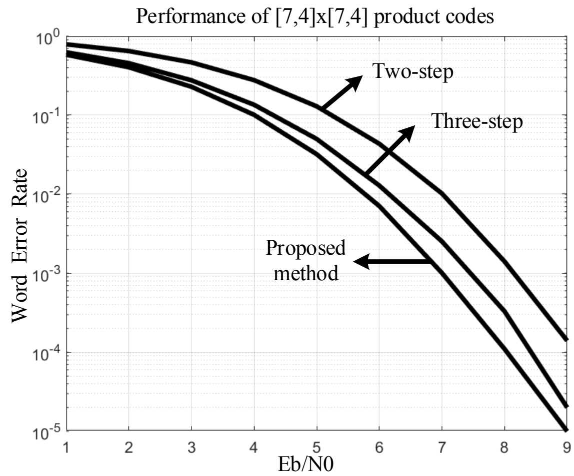

Fig. 6. Word error rate for (49,16,9) hamming product codes on Gaussian channel

Fig. 5 and Fig. 6 support the view that our method has better performance in resisting additive Gaussian white noise compared with the other two decoding methods under a certain signal-to-noise ratio, as demonstrated by the lower bit error rate. For example, to reach a 0.001 bit error rate, the signal-to-noise ratio of the proposed method is nearly 6.2 dB, which is 0.3 dB smaller than that of the three-step decoding method and 0.7 dB smaller than that of the two-step decoding method. Similarly, to reach a 0.001 word error rate, the signal-to-noise ratio of the proposed method is nearly 7 dB, which is 0.4 dB smaller than that of the three-step decoding method and 1.4 dB smaller than that of the two-step decoding method. As a result, the proposed method outperforms the other two methods on the Gaussian channel.

D. Experiment on the Binary Symmetric Channel

Similar to the former experiment, the BER and WER were used to evaluate the performance of the decoding methods under various noises. The difference between this test and the former is that we used the error probability to reflect the degree of noise rather than the signal-to-noise ratio. The number of encoding and decoding processes was set to 500,000. Fig. 7 is a flowchart of this experiment.

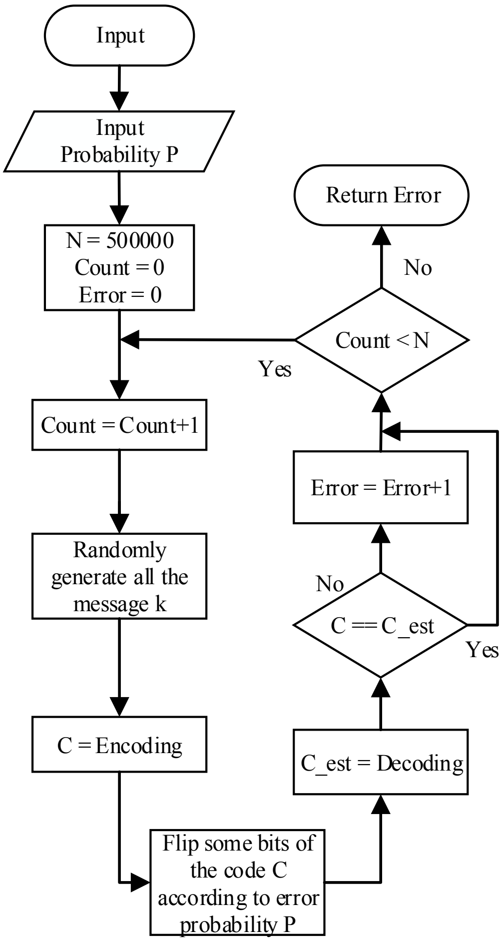

Fig. 7. Flowchart of the experiment on the binary symmetric channel

Based on Fig. 7, the whole procedure can be divided into the encoding process, noise introduction process, decoding process, and error analysis process, whose structures are similar to those in Fig. 6. Since the channel was a binary symmetric channel, there was no need to implement a modulation and demodulation process. Additionally, the noise was added to the original code, which could randomly flip some bits according to the given error probability P. The BER and WER were obtained using the three decoding methods under different error probabilities, which increased from 0 to 0.1. The results are shown in Fig. 8 and Fig. 9.

The bit error rate and word (frame) error rate increase with increasing error probability. However, the error of the proposed method is always less than that of the other two methods. For example, when the error probability is 0.05, the bit error rate and frame error rate of the proposed method are approximately 0.001 and 0.005, whereas those of the three-step decoding are approximately 0.002 and 0.01, respectively. It is also noticed that the two-step decoding produces the most errors in both the BER and FER. Therefore, it is convincing that the proposed decoding method can correct more errors on the binary symmetrical channel.

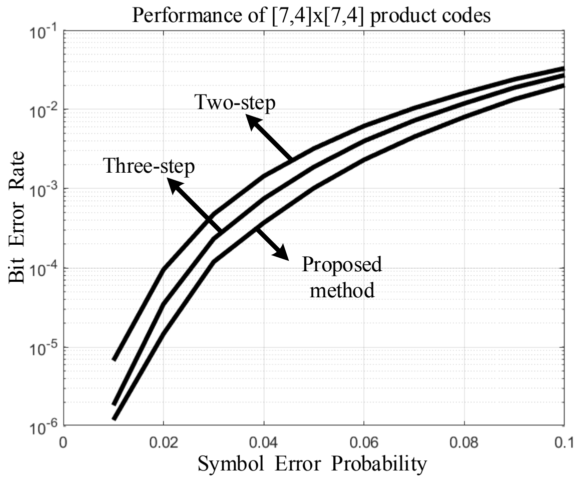

Fig. 8. Bit error rate for a (49,16,9) Hamming product codes on the binary symmetric channel

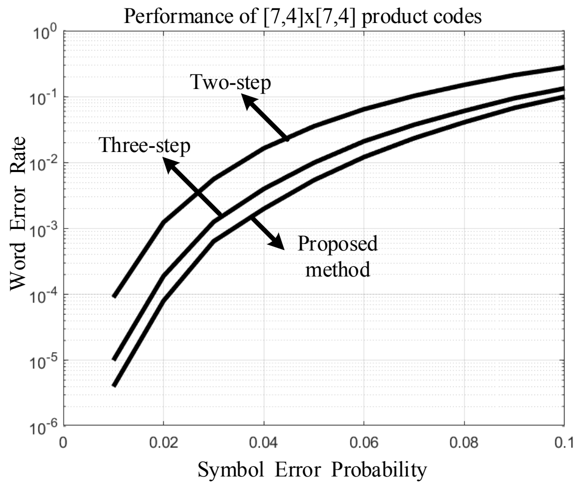

Fig. 9. Word error rate for a (49,16,9) Hamming product codes on the binary symmetric channel

E. Experiment with a Given Number of Error Bits

To thoroughly explore the correction capability of the three decoding methods, the number of code error bits was previously set. Since the current code is a Hamming product code with parameter (49,16,9), whose minimum distance is nine, it can correct all error patterns that are not greater than four. However, as mentioned in section 2, the existing hard-decision decoding methods fail to adequately correct all stall patterns. Therefore, we enumerated all possibilities of the error pattern whose number of errors did not surpass four and added it to the code, which was encoded using a random message generator by adding the corresponding check bits.

For the error patterns with the number of errors above half the minimum distance of the code, it was not feasible to enumerate all error patterns because the total number of error patterns was considerable. Therefore, we randomly extracted a limited number of error patterns from the set whose number of error bits was given and added them into the codeword. Here, this limited number was set to 300,000.

TABLE I
NUMBER OF ERROR PATTERNS AND NUMBER OF ERROR BITS

Number of Error Bits	Number of Error Patterns
1	49 (Full)
2	1176 (Full)
3	18424 (Full)
4	211876 (Full)
5~10	300000 (Not Full)

Table I presents the number of error patterns under different given numbers of error bits for (49,16,9) Hamming product codes.

All three decoding methods were used to correct those error patterns. Table II summarizes the number of decoding mistakes made in different ways under various numbers of error bits.

For easy observation and comparison, we normalized the data in Table II to obtain Table III. We then divided the number of decoding mistakes by the number of error patterns we introduced. The number of error patterns can be seen in Table I.

Table II and Table III show that the proposed method can entirely correct errors up to half the minimum distance of the Hamming product code. In contrast, the two-step and three-step methods fail to correct 9261 (4%) and 1323 (1%) error patterns, respectively, when the given number of error bits is four. However, when the given number of error bits increases from four to ten, all three methods experience decoding failure to different degrees. However, the proposed method has the best performance because its decoding failure rate is the lowest.

Fig. 10 presents a histogram, which may provide an intuitive sense of the superiority of the proposed method over the others.

TABLE II
NUMBER OF DECODING MISTAKES UNDER A GIVEN NUMBER OF ERROR BITS.

Given Number of Error Bits	Number of Decoding Mistakes		
	Two-step Method	Three-step Method	Proposed Method
1	0	0	0
2	0	0	0
3	0	0	0
4	9261	1323	0
5	55497	10949	5584
6	126283	37416	18847
7	201927	90915	54419
8	256916	165576	133169
9	283679	232975	221631
10	294158	272692	271361

TABLE III
NORMALIZATION OF THE DECODING MISTAKES UNDER A GIVEN NUMBER OF ERROR BITS.

Given Number of Error Bits	Normalization of Decoding Mistakes		
	Two-step Method	Three-step Method	Proposed Method
1	0	0	0
2	0	0	0
3	0	0	0
4	0.04	0.01	0
5	0.18	0.04	0.02
6	0.42	0.12	0.06
7	0.67	0.30	0.18
8	0.86	0.55	0.44
9	0.95	0.78	0.74
10	0.98	0.91	0.90

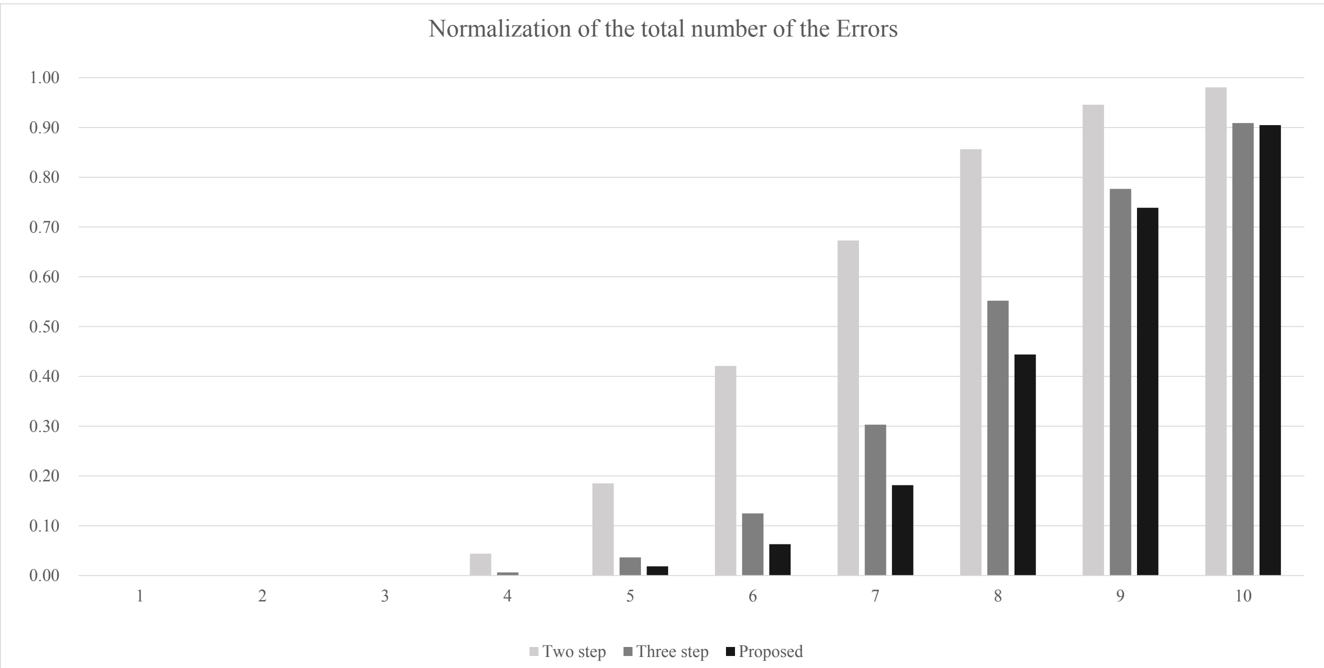

Fig. 10. Histogram of the normalization of the decoding mistakes under different given numbers of error bits

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel iterative hard-decision decoding method for Hamming product codes. By adding a decision process before directly conducting the decoding operation and introducing an erasure process, the proposed method can correct all error patterns within half the minimum distance of the Hamming product codes. In addition, it can correct many errors when the number of error bits is above half the minimum distance. Three experiments-Gaussian channel tests, binary symmetric channel tests and experiments with a given number of error bits-were conducted, and two other methods were implemented for comparison. The result proves that the proposed method can correct more errors than existing hard-decision decoding methods.

In the future, we would like to modify our method, apply it to the decoding of extended Hamming product codes and explore its performance. The minimum distance of an extended Hamming product codes is enlarged, which allows for correcting more error patterns compared with an original Hamming product codes. Moreover, implementing our method on a hardware device is an exciting prospect.

REFERENCES

[1] P. Elias, "Error-free Coding," *IEEE Transactions on Information Theory*, vol. 4, no. 4, pp. 29-37, Sep. 1954.

[2] O. Al-Askary, "Iterative decoding of product codes," in *Signaler*, sensorer och system, 2003.

[3] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein, "Covering codes," in *North-Holland Mathmatical Library Elsevier*, 1997.

[4] G. Cohen, M. Karpovsky, H. Mattson Jr., and J. Schatz, "Covering radius–survey and recent results," *IEEE Transactions on Information Theory*, vol. 31, May. 1985.

[5] N. Abramson, "Cascade Decoding of Cyclic Product Codes," *IEEE Transactions on Communication Technology*, vol. 16, no. 3, pp. 398-402, Jun. 1968.

[6] F. Blomqvist, "On hard–decision decoding of product codes," *Applicable Algebra in Engineering, Communication and Computing*, pp. 1-18, 2021.

[7] K. Alexey, Z. Victor, and R. Eygene, "A new iterative decoder for product codes," *Proceedings of the International Workshop on Algebraic and Combinational Coding Theory*, pp. 211-214, 2014.

[8] Fu Bo and P. Ampadu, "A multi-wire error correction scheme for reliable and energy efficient SoC links using Hamming product codes," *in SOC Conference, IEEE*, pp. 59-62, 2008.

[9] Fu Bo and P. Ampadu, "An energy-efficient multiwire error control scheme for reliable on-chip interconnects using Hamming product codes," *VLSI Design*, pp. 1-14, Dec. 2008.

[10] Fu Bo and P. Ampadu, "On hamming product codes with type-II hybrid ARQ for on-chip interconnects," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 9, pp. 2042-2054, Sep. 2009.

[11] J. Kim and Y. Jee, "Hamming product code with iterative process for NAND flash memory controller," in *2010 2nd International Conference on Computer Technology and Development*, *IEEE*, pp. 611-615, Nov. 2010.

[12] A. K. Chlaab, W. N. Flayyih, and F. Z. Rokhani, "Lightweight hamming product code based multiple bit error correction coding scheme using shared resources for on chip interconnects," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp.1979-1989, Oct. 2020.

[13] G. Forney, "Generalized Minimum Distance decoding," *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 125-131, Apr. 1966.

[14] S. Reddy and J. Robinson, "Random error and burst correction by iterated codes," *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 182-185, Jan. 1972.

[15] R. M. Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *IEEE Transactions on Communications*, vol. 46, no. 8, pp. 1003-1010, Aug. 1998.

[16] C. Wickman, D. G. Elliott, and B. F. Cockburn, "Cost models for large file memory DRAMs with ECC and bad block marking," *Proceedings 1999 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (EFT'99)*, pp. 319-327, Nov. 1999.

[17] S. Lin and D. J. Costello, "Error control coding," in *Scarborough: Prentice hall*, 2001.

[18] M. M. Karbassian and Ghafouri-Shiraz. Hooshang, "Phase-Modulations Analyses in Coherent Homodyne Optical CDMA Network Using a Novel Prime Code Family," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2007, WCE 2007, 2-4 July, 2007, London, U.K., pp 358-362.

[19] A. Idris, D. Kaharudin, and S. Y. SK, Idris, A. K. Dimyati, and S. S. Yusof, "Performance of Linear Maximum Likelihood Alamouti Decoder with Diversity Techniques," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2011, WCE 2011, 6-8 July, 2011, London, U.K., pp 1728-1731.

[20] H. Sandhu and D. Chadha, "Terrestrial free space LDPC coded MIMO optical link," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2009, WCECS 2009, 20-22 October, 2009, San Francisco, USA, pp 372-375.