

IHDD-GRAND Algorithm for Decoding Symmetric Block-Wise Concatenated BCH Codes

Gabriel*, Jeongseok Ha*

ABSTRACT

Block-wise concatenated Bose-Chaudhuri-Hocquenghem (BC-BCH) codes are shown to have excellent error-correcting capability under hard-decision decoding. The variant of BC-BCH codes, namely symmetric BC-BCH (SBC-BCH) codes, is later introduced to improve the error-correcting performance of BC-BCH codes. In this work, we propose a new hard-decision algorithm, IHDD-GRAND, which aims to further enhance the error-rate of SBC-BCH codes. The proposed algorithm uses GRAND as an auxiliary decoder for the IHDD, and it can be efficiently implemented by utilizing the error-confinement property of SBC-BCH codes. It will be shown that the proposed algorithm significantly improves the error-correcting performance of SBC-BCH codes.

Key Words : BCH codes, symmetric block-wise concatenation, GRAND algorithm, hard-decision decoder

I. Introduction

Block-wise concatenation of Bose-Chaudhuri-Hocquenghem (BCH) [1] codes was first proposed in [2], [3] and is shown to have outstanding error-correcting capability under an iterative hard-decision decoding (IHDD) algorithm. Consequently, the block-wise concatenated BCH (BC-BCH) codes are particularly suitable for applications where obtaining soft information is

infeasible, such as solid-state drives (SSDs) and optical communications. In [4], symmetric BC-BCH (SBC-BCH) codes are devised to enhance the error-correcting performance of the conventional BC-BCH codes. Furthermore, it is shown that the SBC-BCH codes achieve significantly better error-rate performance than the conventional BC-BCH codes when the IHDD algorithm is used.

Guessing random additive noise decoding (GRAND) algorithm [5, 6] has recently been proposed to be a universal decoder, and its efficient hardware architecture is introduced in [7]. GRAND algorithm operates by iteratively testing putative noise vectors, starting from the most probable to the least probable noise vector. Each test requires subtracting the putative noise vector from the received word and querying if the result is a valid codeword. Generally, the parity check matrix of the code is used for the querying. The first instance where a codeword is found ends the iterative tests, and that codeword is the decoding output.

Unique to the SBC-BCH codes, the block-wise concatenation structure ensures that the erroneous bits on each constituent code are confined in blocks when the IHDD fails, which we call as error-confinement property. In other words, the erroneous and non-erroneous bits can be clearly distinguished on each constituent code. In this paper, we will show that the GRAND algorithm takes advantage of the error-confinement property and uses this property to simplify the iterative tests of putative noise vectors. As a result, the GRAND algorithm corrects errors more effectively, and we can enjoy the extra error-correction performance by integrating the GRAND algorithm into the IHDD.

II. The IHDD-GRAND Algorithm

The conventional GRAND algorithm [5] aims to

* 논문은 2023년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임.(NRF-2021R1A2B5B01002 204)

• First Author : (ORCID:0009-0008-7577-922X) Korea Advanced Institute of Science and Technology, gbdaniel97@kaist.ac.kr, 학생 (석사과정), 정회원

◦ Corresponding Author : (ORCID:0000-0003-1262-151X) Korea Advanced Institute of Science and Technology, jsha@ee.kaist.ac.kr, 정교수, 종신회원

논문번호 : 202311-151-B-LU, Received: November 22, 2023; Revised December 6, 2023; Accepted: December 6, 2023

correct any number of errors in the received word. Due to the high number of testing putative noise vectors, it is impractical to use the conventional GRAND algorithm. Instead, another variant of GRAND, called GRAND with abandonment (GRANDAB), is proposed in [5], and its decoding latency can be easily controlled by changing the abandonment threshold. The GRANDAB algorithm is described in Algorithm 1.

Algorithm 1: The GRANDAB Algorithm

Input : received word \mathbf{r} , parity check matrix \mathbf{H} , abandonment threshold AB

Output: decoded codeword \mathbf{c}

```

1 for  $j \leftarrow 0$  to  $AB$  do
2   Generate the weight- $j$  test noise vectors  $\tilde{\mathbf{e}}_j$ 
3    $\hat{\mathbf{r}} = \mathbf{r} - \tilde{\mathbf{e}}_j$ 
4   Compute the syndrome vector  $\hat{\mathbf{s}} = \mathbf{H}\hat{\mathbf{r}}^T$ 
5   if  $\hat{\mathbf{s}} = \mathbf{0}$  then
6      $\mathbf{c} = \hat{\mathbf{r}}$  (decoding successful)
7   end if
8 end for
9  $\mathbf{c} = \mathbf{r}$  (decoding failure)
    
```

In this paper, we integrate the IHDD and GRANDAB algorithms to further improve the error-correcting performance of the SBC-BCH code and reduce the number of iterative tests associated with the GRANDAB algorithm. Once the received SBC-BCH code experiences decoding failure in the IHDD, there are several unrecovered constituent codes (which we will call failed constituent codes), and we can take advantage of the knowledge of non-erroneous bits to simplify the generation of test noise vectors in the GRANDAB algorithm. Figure 1 shows an example of SBC-BCH code with three failed constituent codes. Each failed constituent code has exactly three erroneous blocks where the bit errors are confined, and the bits outside these blocks are correctly decoded. The GRANDAB needs only to guess the noise vector corresponding to the location of the erroneous bits. Hence, the number of iterative tests of putative noise vectors is dramatically reduced, thus reducing the decoding latency of the GRANDAB algorithm.

Further simplification to the GRANDAB algorithm can be made by noting that we are dealing with binary

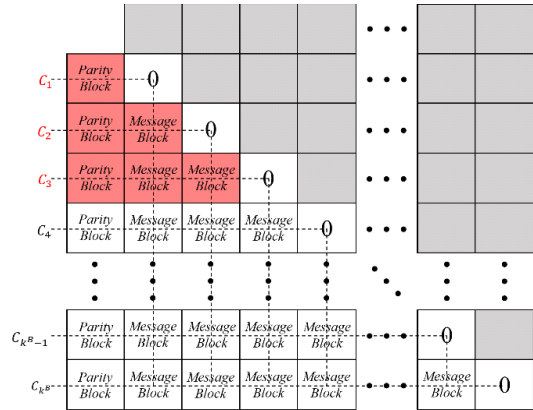


Fig. 1. The SBC-BCH code with three failed constituent codes, k^B is the total number of constituent codes.

BCH code in this work. The binary BCH code has a unique property that even-numbered syndromes can be easily computed by knowing the odd-numbered syndromes [8]. Specifically,

$$S_{2j} = r(\alpha^{2j}) = [r(\alpha^j)]^2 = S_j^2 \quad j = 1, \dots, 2t, \quad (1)$$

where S_j is the j -th syndrome value, $r(x)$ is the received word in polynomial form, α^j is the j -th root of the BCH code, and j starts from one because we consider narrow-sense BCH code. Therefore, we can avoid storing parity check matrix \mathbf{H} and matrix-vector multiplication when checking for valid codeword in the iterative tests of the GRANDAB algorithm.

The complete diagram block of our proposed decoder is shown in Fig. 2. First, the received SBC-BCH word is decoded by using the IHDD. If the IHDD encounters a decoding failure, the GRANDAB algorithm is activated, and it tries to correct errors on each constituent code. Once the GRANDAB successfully decodes a failed constituent code, the IHDD is resumed to continue the decoding process for the rest of the failed constituent codes. This

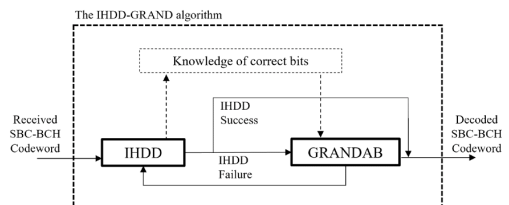


Fig. 2. Diagram block of IHDD-GRAND algorithm.

mechanism is more efficient than letting the GRANDAB decode all failed constituent codes since the IHDD is computationally cheaper than the GRANDAB algorithm. We declare decoding failure in the proposed algorithm when all failed constituent codes cannot be recovered by the GRANDAB algorithm.

III. Simulation Results

In this section, we show the error-rate improvement of the SBC-BCH code when the IHDD-GRAND algorithm is used. It is worth mentioning that we ignore the detailed analysis of computational complexity because it highly depends on hardware architecture. Hence, we put more focus on error-rate improvement instead. We use a codeword error rate (CER) over raw bit error rate (RBER) as the metric for measuring the error-correction performance. We design an SBC-BCH code having code rate $R = 0.74$ and message length of 4096 bits. The code consists of 40 constituent codes, each of which is a (247, 211, 4) shortened BCH code.

We set the abandonment threshold of GRANDAB algorithm by $t + 1$, that is the GRANDAB algorithm will attempt to correct $t + 1$ errors on each constituent code. A higher value of abandonment threshold results in better error-correcting capability at the expense of longer decoding latency. Figure 3 depicts the comparison of error-correction performance between

the conventional IHDD and our proposed decoding algorithm. At RBER of 0.01, the IHDD algorithm achieves a CER around 8×10^{-5} while our proposed algorithm obtains a CER value about 7×10^{-7} . Considering that our proposed algorithm only utilizes hard outputs from the channel, we believe this significant improvement on CER performance is particularly beneficial for applications where soft information is difficult to obtain, such as solid-state drives (SSDs) and optical communications.

IV. Conclusion

In this paper, we propose a new IHDD-GRAND algorithm to improve the error-correcting performance of the SBC-BCH code. The error-confinement property allows us to employ the GRANDAB algorithm with reduced complexity. Further simplification in the GRANDAB algorithm is made by using the even-syndrome property of a binary BCH code. We observed a notable CER performance improvement with the IHDD-GRAND algorithm compared to the conventional IHDD algorithm when decoding the SBC-BCH code.

References

- [1] J. Lee, S. Kim, and S. Song, "A study on efficient implementation of SW-based BCH encoding and decoding," *J. KICS*, vol. 45, no. 5, pp. 903-913, 2020. (<http://doi.org/10.7840/kics.2020.45.5.903>)
- [2] S.-G. Cho, D. Kim, J. Choi, and J. Ha, "Block-wise concatenated BCH codes for NAND flash memories," *IEEE Trans. Commun.*, vol. 62, no. 4, pp. 1164-1177, Apr. 2014. (<http://doi.org/10.1109/TCOMM.2014.021514.130287>)
- [3] S.-G. Cho and J. Ha, "Concatenated BCH codes for NAND flash memories," *2012 IEEE ICC*, pp. 2611-2616, Ottawa, ON, Canada, 2012. (<http://doi.org/10.1109/ICC.2012.6363960>)
- [4] D. Kim, K. R. Narayanan, and J. Ha,

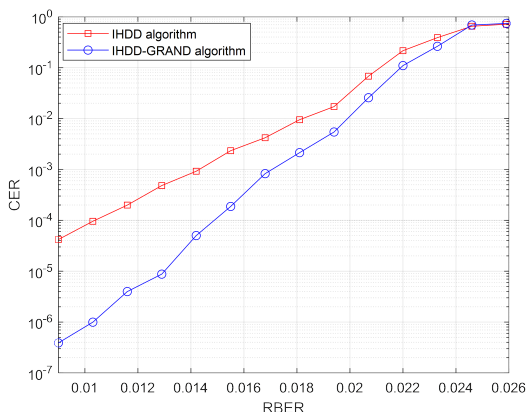


Fig. 3. CER performance comparison of IHDD and IHDD-GRAND algorithm.

- “Symmetric block-wise concatenated BCH codes for NAND flash memories,” *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4365-4380, Oct. 2018.
(<http://doi.org/10.1109/TCOMM.2018.2839606>)
- [5] K. R. Duffy, J. Li, and M. Médard, “Guessing noise, not code-words,” *IEEE Int. Symp. Inf. Theory*, 2018.
(<http://doi.org/10.1109/ISIT.2018.8437648>)
- [6] K. R. Duffy, J. Li, and M. Médard, “Capacity-achieving guessing random additive noise decoding,” *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4023-4040, Jul. 2019.
(<http://doi.org/10.1109/TIT.2019.2896110>)
- [7] S. M. Abbas, T. Tonnelier, F. Ercan, and W. J. Gross, “High-throughput VLSI architecture for GRAND,” *2020 IEEE Wkshp. SiPS*, pp. 1-6, Coimbra, Portugal, 2020.
(<http://doi.org/10.1109/SiPS50750.2020.9195254>)
- [8] R. E. Blahut, *Algebraic Codes for Data Transmission*, Cambridge, U.K. Cambridge University Press, 2003.
(<http://doi.org/10.1017/CBO9780511800467>)