A Further Research On Bit-Rectification Strategy With Lower Error Floor

Hui Xue, Lijun Zhang

[1]*School of Electronics and Information Engineering*

Beijing Jiaotong University

Beijing, China

{17120194, ljzhang}@bjtu.edu.cn

Abstract — **This paper do a further on bit-rectification strategy to lower error floor of Low-density parity-check (LDPC) codes. The most unstable bits are located by bit-searching algorithm and then obtain more reliable message over parallel concatenated Gallager codes (PCGC) decoder. The negative effect of unstable bits along with corresponding trapping sets will be avoided.**

Keywords—error floor, bit-searching, parallel concatenated

I. INTRODUCTION

With the rapid development of wireless communication systems, Low-density parity-check (LDPC) codes [1] with the ability of approaching the Shannon limit have become the focus of the channel coding field. Due to the excellent bit error rate (BER) performance of the LDPC code under the iterative decoding algorithm based on belief propagation theory, it has been incorporated into the coding scheme of the fifth generation communication system. However, most LDPC codes have a serious error floor [2] phenomenon in the iterative decoder. That is, as the signal-to-noise ratio (SNR) increases, the BER reaches a saturated state and does not continue to decrease. This problem seriously limits its application of the field with higher BER requirement. In summary, reducing the error floor of LDPC codes is an important and meaningful research.

There has been many research aiming to optimize the error floor phenomenon, most of them aim at breaking the trapping set [3], which is the main cause of error floor. The existing research can be roughly divided into three categories: code construction [4][5], optimization of decoding scheme [6][7] and concatenated strategy [8][9]. The main motivation of code construction is to avoid generating harmful trapping sets in the Tanner graph. However, the construction strategy usually has limitations because it only apply to specific code. The optimization can also be considered on decoding schemes like the backtracking scheme and the two-stage algorithm, both of them need to get a priori information of the harmful trapping sets, which is extremely difficult to get. Concatenated strategy has also be implemented according the cooperation of component decoders, but the weakness is the increased redundancy may lead to rate loss. In general, it is necessary to obtain a efficient strategy with low complexity and widely available.

For the existed bit-rectification strategy, it committed to eliminate the negative effect of trapping sets by searching the most erroneous bits based on Monte Carlo simulation, which can store the number of decoding errors per bit. However, the most common trapping set is periodically oscillating, and the number of oscillations is more accurate in representing whether the bits exists in a harmful trapping sets.

Therefore, in this paper, we do a further on bit-rectification strategy over bit-searching algorithm and optimized parallel concatenated Gallager codes (PCGC) decoder [10]. The most unstable bits are determined by bit-searching algorithm, which based on the number of oscillation. And then the unstable bits will be re-encoded to obtain a targeted parallel concatenated Gallager codes (PCGC). The PCGC structure ensure that the most unstable can obtain more stable channel information from another component decoder. Finally, the negative effects of these unstable bits can be exhausted with high possibility. Simulation results show that it can effectively reduce the error floor while keeping the coding rate basically unchanged.

II. PRELIMINARIES

A. Monte Carlo simulation

In previous research, the bit-searching method is based on Monte Carlo simulation. For the LDPC codes, this simulation can locate the most unstable bits rather than locate the trapping sets directly. The Monte Carlo simulation formula can be mathematically expressed as:

$$\hat{P}_{MC} = \frac{1}{N}\sum_{i=1}^{N} E_j(i) \tag{1}$$

Where the $\hat{P}_{MC}$ denoted the Monte Carlo simulation value of bit j, $E_j(i)=1$ if the bit j do not correctly converge during the i th iteration. $E_j(i)=0$ if the bit j correctly converge. N is the number of Monte Carlo simulation runs.

But in order to get the stability of the bit node instead of the error level, we need to make some improvements based on the Monte Carlo simulation, which can be called oscillating simulation, it can be expressed mathematically as:

$$\hat{P}_{IMC} = \frac{1}{N}\sum_{i=1}^{N} O_j(i) \tag{2}$$

Unlike the Monte Carlo simulation, $\hat{P}_{MC}$ denoted the level of oscillation of bits, $O_j(i)=1$ if the state of bit j during the i th iteration is different from the state during the i-1 th iteration. $O_j(i)=0$ if the state of bit j is not change during the i-1th and i th iteration. N is the number of oscillating simulation runs.

B. Decoding scheme of PCGC

The code construction after bit-searching is a special kind of parallel concatenation structure, so some improvements can be considered on the conventional decoder.

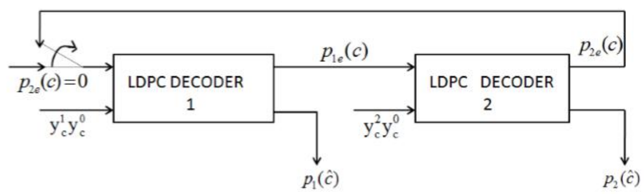

Fig.1. Conventional decoder of PCGC

The conventional decoder has already dedicated in [10]. As shown in Fig.1, y_c^0 represents the received bits which corresponding to the information bits x, y_c^1 and y_c^2 indicate the bits corresponding to the parity check bits of two component codes. A whole iterate decoding process is named a super iteration. At the first super iteration, the decoder 1 computes the extrinsic information $p_{1e}(\hat{c})$ by utilizing the received sequence y_c^0 and y_c^1. The decoder 2 will calculate the extrinsic information $p_{2e}(\hat{c})$ by $p_{1e}(\hat{c})$ obtaining from decoder1. In the next iterations, the decoder 1 will calculate the extrinsic information $p_{1e}(\hat{c})$ by $p_{2e}(\hat{c})$, which is product from the decoder 2. This process of super iteration will continue until all the component decoders get a valid word or reach the maximum number of super iterations. If all decoders do not successfully converge until the max number of iterations is reach, the final output is the decoding result of the decoder 2.

III. BIT-RECTIFICATION STRATEGY

For the error floor phenomenon of LDPC codes, we proposed an effective Bit-rectification strategy based on bit-searching method and modified PCGC decoder. Firstly, we obtain the stability-level of bits by oscillate simulation. The detail is shown in algorithm 1.

Algorithm 1 The Monte Carlo simulation algorithms

Intialization: Give the codeword *Y* ,code length *n* and run number *N*.
$O=(O_1,O_2,\ldots O_n)$ *and* $O_1=O_2=\ldots=O_n=0$
for *i*=1 to *N* **do**
  Decode the codeword *Y* under AWGN Channel
  for *j*=1 to *n* **do**
    if bit *j* is not same as the j during the i-1 th iteration
      $O_j=O_j+1$
    end if
  end for
end for
Obtain the reliable level vector $O=(O_1,O_2,\ldots O_n)$

Now consider the stability-level vector ***O***. as the value of O_j get bigger, the stability-level will become lower. So the unstable bits can be selected from the top of the vector ***O***.

As shown in Fig.2 (a), the variable nodes (*VN*s) with different gray-scales represent different stability-levels, the darker the bits, the less stable the bits. We assume that the number of unstable bits we selected by oscillating simulation is *a*. As shown in Fig.2 (b), the *a* bits will be consider as the message sequence to be re-encoded by the (*a*+*b*, *a*) LDPC code. In addition, for the different original code length and rate, the quantity of the unstable bits should be carefully considered. The purpose of this idea is to avoid serious rate loss while ensuring better performance. If we choose the (*a*+*b, a*) code for (*n, m*) original code in the process of re-encoding, the modified coding rate $r_m = m/(n+b)$. Since only the a little percentage of bits are involved in trapping sets, so $b \ll n$ and it will not lead much rate loss.

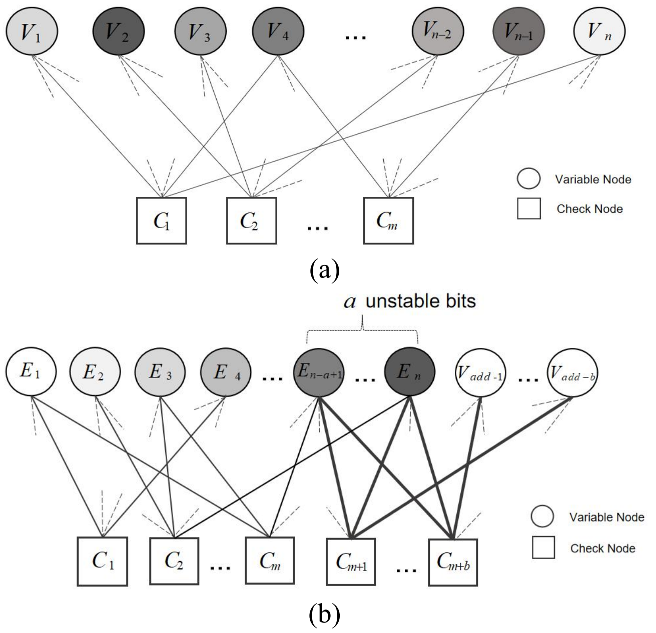

Fig.2. (a) Original Tanner graph (b) Modified Tanner graph before bit-searching

As we know, the trapping sets usually causes the bits oscillate all the time. Therefore, the bits we select through oscillating simulation will have a high probability of being in the trapping sets. After re-encoding, these bits will exist in two different Tanner graph, the probability that the different structures have a trapping set on the same bit is very low. For the unstable bits, more reliable information can be obtained from another structure, so the erroneous information of the original code can be rectified. The final result is that the original code may jump out of the trapping set with a high probability.

For the complexity and latency, the bit-searching algorithm locate the bits involving in the trapping sets with high probability rather than determine the exact location of the harmful trapping sets by complex analysis of Tanner graph, so that it can eliminate the negative effects of trapping sets while keeping the complexity and latency basically unchanged.

In order to make better use of this structural feature and ensure the full interaction of extrinsic information, we do an improvement on the conventional PCGC decoder. Two independent maximum number *k* of iteration is set for the component decoders respectively to ensure sufficient local iterations. The global iteration number is set as W.

As shown in Fig.3, a whole iterative process is called a super iteration. In the first super iteration, decoder 1 computes the $p_1(\hat{c})$ with the received sequence under SPA, this process continue until correctly converging or the maximum number of iterations is reached. The decoder 2 is activated in the case where the decoder 1 fail to converge, its only output is the extrinsic information $p_{1e}(\hat{c})$ transmitting to decoder 1 as a priori information for next super iteration. This process of decoding will continue until correctly converge or the maximum number of the maximum number of super iteration is reach.

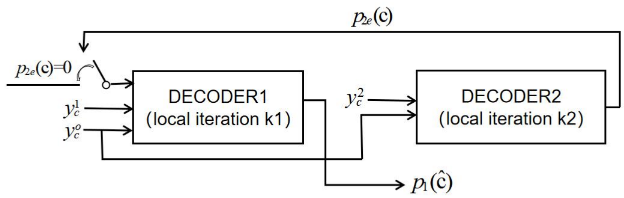

Fig.3. Modified PCGC decoder

It is noted that the (Log-likelihood Ratio) LLR held by the unstable bits is wrong with high probability, so we set the decoder 1 do not transmit the extrinsic information to the decoder 2. Since these unstable bits are in different Tanner graph structures denoted by different parity check matrix, the decoder 2 is set for the purpose of ensuring the unstable bits can get correct LLR information from another structure. In the case where the decoder 1 cannot converge correctly, the correct LLR information can be obtained from the decoder 2 with a high probability. These unstable bits transmit the correct LLR information to the decoder 1, effectively avoiding the trapping sets. Finally, the output of the decoder 1 is taken as the final output.

We find that in the improved PCGC decoder, the transmission of extrinsic information is equivalent to the LLR information rectification or flipping of the unstable bits in the trapping sets, and then it can effectively reducing the error floor.

IV. SIMULATION RESULT

A. *Simulation of (1057, 813) MacKay code*

In this section, we will give the example to show the effectiveness of the bit-rectification strategy. We assume that all the simulation experiments are under BPSK modulation and AWGN channel.

The LDPC codes we use for the simulation is the (1057, 813) MacKay codes. Because the bit-searching algorithm based on oscillating simulation is performed at a certain SNR, we choose a medium SNR=4.15dB to obtain a stability-level sequence. $\boldsymbol{O}$. Set the running number of oscillating simulation as 5000 and 10000 respectively. Considering the rate loss, we need to carefully select the number of unstable bits. We sort the sequences $\boldsymbol{O}$ to observe its characteristic. As shown in Figure 4, the two curves represent the sorted $\boldsymbol{O}$ with running number of 5000 and 10000, we found that there is always a waterfall area in $\boldsymbol{O}$. This waterfall area divides all the bits into two levels of reliability.

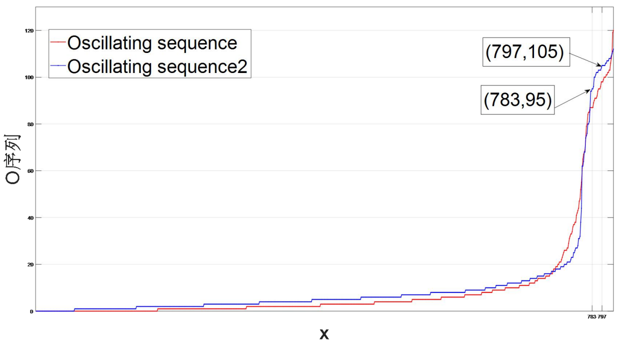

Fig.4. The curve of sorted $\boldsymbol{O}$

We randomly select the critical points (783, 95) and (797, 105) at the waterfall area, so the number of bits with lower stability-level is 30 and 16 respectively. The LDPC codes used for re-encoding all have a coding rate of 1/2. The effect caused by the number of unstable bits can be observed from the Fig.5.

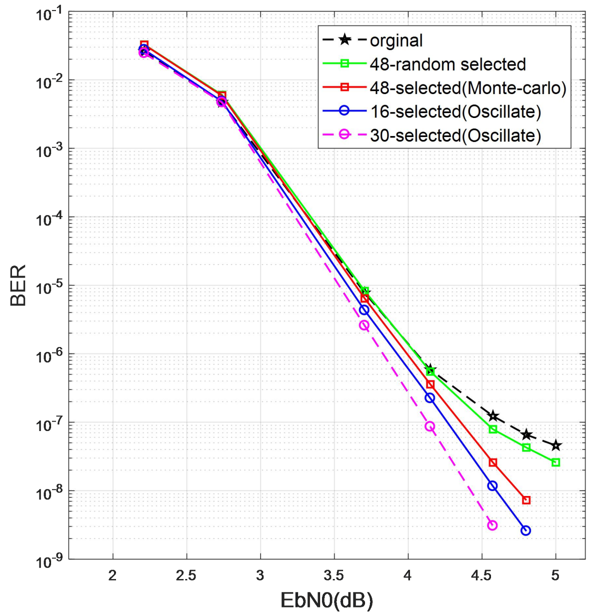

Fig.5. The performance of the bit-rectification strategy with different bit-selected

Taking the 30-selected as an example, after re-encoding, the coding rate is reduced from 0.769 to 0.748. The number of iterations is large enough to ensure reliability. It can be seen from Fig. 5 that the bit-rectification strategy based on the oscillating simulation and modified PCGC decoder can effectively improve the error floor phenomenon. Without considering the adding redundancy, the more unstable bits are selected, the better the performance will be. The random-selected scheme do not have efficient improvement of error floor, but the bit-rectification strategy with 30-selected bits reduce the error floor from 10^{-7} to 10^{-9}. There is no error floor phenomenon appear until the BER of 10^{-9}. It strongly confirms that the BER performance is improved better in the high E_b / N_0 area with the bit-searching algorithm and modified PCGC decoder, despite a negligible loss of coding rate.

Consider the trade-off may be made between the BER performance and rate loss to get the optimal solution. We found that when the selected number of unstable bits is 16, there still have little improvement of performance when compared to that of the Monte Carlo simulation with 48-selected bits, the advantage of more redundancy is almost negligible. Therefore, the bit-searching algorithm have better performance when it is based on oscillating simulation.

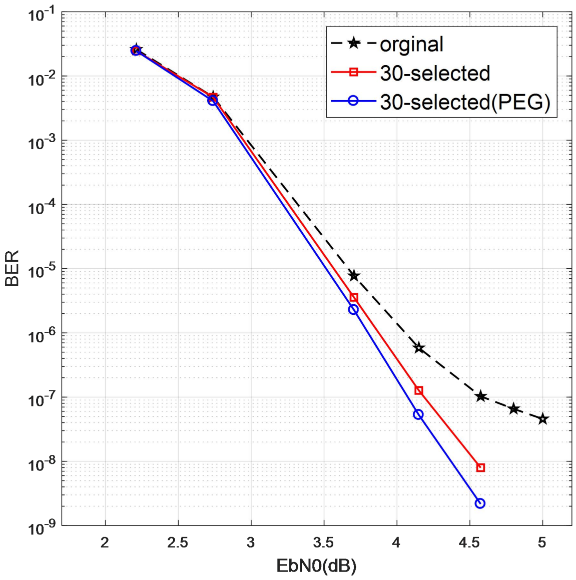

Fig.6. The performance of bit-rectification strategy with (60, 30) PEG code

In order to further optimize the performance of the bit-rectification strategy, we replaced the (60, 30) LDPC code with the (60, 30) PEG code, which has a better structure. Through many experiments, the performance has been further improved. As shown in Figure 6, the error floor is also further reduced.

B. Complexity Analysis

As we know, Conventional SPA decoder has about $6n_v d_{ave}$ floating-point multiplications per iteration, where d_{ave} is the average column-weight and n_v is the number of variable nodes. Hence, the total multiplication operations $c = 6n_v d_{ave} N_s$, where N_s is the average number of iteration. The complexity comparison between original code and the code under bit-rectification strategy can be seen from this aspect. For the original (1057, 813) MacKay code, $n_v = 1057$ and $d_{ave} = 3$, and for the modified code with 30-selected unstable bits, $n_v = 1105$ and $d_{ave} = d_v + (d_v * n_{add}) / n_v = 3.09$, where the n_{add} is the number of unstable bits being selected. In the high SNR=4.15 dB, the average numbers of original code is N_s =2.869 and the average numbers of modified code is N_s =2.897. Hence, the multiplication operations is approximate $1 - c_{original} / c_{modified} = 6.51\%$ reduction. Therefore, with the significantly reduction of error floor, the bit-rectification strategy only has a increased complexity of 6.51%.

V. CONCLUSION

In this paper, we proposed a bit-rectification strategy based on oscillating simulation and modified PCGC decoder. This research shows that the bit-searching simulation can efficiently reduce the error floor without getting concrete structure of the harmful trapping sets. In the further research, we are committed to find more precise bit selection criteria and other selection features.

ACKNOWLEDGMENTS

This work is supported by NSFC under grant number U1534201, 61872031.

REFERENCE

[1] R. G. Gallager, “Low-density parity-check codes,” IRE Trans. Inf. Theory, vol. 8, no. 1, pp. 21-28, 1962.

[2] T. Richardson, “Error floors of LDPC codes,” in Proceedings of the annual Allerton conference on communication control and computing, vol. 41. The University; 1998, 2003, pp. 1426-1435.

[3] S. Kang, J. Moon, J. Ha, and J. Shin, “Breaking the trapping sets in LDPC codes: check node removal and collaborative decoding,” IEEE Trans. Commun., vol. 64, no. 1, pp. 15-26, Jan. 2016

[4] Le Dong, Ziyu Zhao, Jing Lei, Er-bao Li, " Lower error floor of LDPC codes based on trapping sets elimination", in Proc. International Conference on Wireless Communications and Signal Processing (WCSP) 2014

[5] J. Li, S. Lin, K. Abdel-Ghaffar, W. E. Ryan, and D. J. Costello, Jr, "Globally Coupled LDPC Codes." in Proc. Inf. Theory Applic. Workshop, San Diego, CA., Jan. 31 - Feb. 5, 2016.

[6] J. Kang, L. Zhang, Z. Ding, and S. Lin, “An Iterative Decoding Algorithm with Backtracking to Lower the Error-Floors of LDPC Codes," in Proc. IEEE Globecom. Conf., New Orleans, LA, Nov. 2008

[7] J. Kang, L. Zhang, Z. Ding, and S. Lin, “A two-stage iterative decoding of LDPC codes for lowering error floors," in Proc.IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 59, NO. 1, JANUARY 2011

[8] X. Xiao, M. Nasseri, B. Vasić, and S. Lin, “Serial concatenation of ReedMüller and LDPC codes with low error floor,” in Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton), Oct. 2017, pp. 688-693

[9] M. Nasseri, T. Wang, S. Lin, "Concatenated finite geometry and finite field ldpc codes", International Conference on Signal Processing and Communication Systems (ICSPCS), 2017

[10] H. Behairy and S. Chang, "Parallel concatenated Gallager codes," Electronics Letters, vol. 36, no. 24, pp. 2025-2026, 2000