A Survey of Error Floor of LDPC Codes

Yejun He , Jie Yang and Jiawei Song
Department of Communication Engineering, College of Information Engineering
Shenzhen University
Shenzhen, China
Email: heyejun@ieee.org

***Abstract*—The research about error floor of finite-length LDPC code is a very hot topic recently. This survey briefly introduces the existing works in lowering and estimating error floor of LDPC codes, and sheds some light on some potential future research areas.**

Keywords-LDPC codes; error floor; girth; stopping set; trapping set

I. INTRODUCTION

Low-density parity-check (LDPC) codes were discovered by Gallager in 1962 [1]. It has been an attractive research area due to their near-capacity performance and low complexity. However, the error-floor problem is still an open question. In brief, error floor is that BER performance curve does not decreases obviously with the increase of the SNR. Some potential applications of LDPC codes such as data storage and deep-space communications require bit error rate (BER) as low as 10^{-15}. Therefore, the research of error floor of LDPC codes is of great importance.

There have been many developments in the past few years on the error floor of LDPC codes. In this paper, we will only focus on some aspects of LDPC codes. The rest of this paper is organized as follows. In Section Ⅱ, we overview the fundamental terminologies and principles related to the error floor of LDPC codes. In Section Ⅲ, we review the existing work in error floor, including theoretic analysis of error floor, algorithms for lowering the error floor and error floor of structured LDPC codes. Finally, conclusion and potential research areas are given in Section Ⅳ.

II. FUNDAMENTALS

There are two kinds of representation of LDPC codes: parity check matrix and Tanner graph. Girth is the smallest cycle length in the Tanner graph associated with a LDPC parity check matrix. Small girth will reduce the independence of the transmitted messages in the decoding process, and will result in failure of convergence to a valid codeword. It was observed that codes with low girth tend to have high error floors.

Generally speaking, error floor of LDPC codes is dominated by the bad sub-structures in the Tanner graph. For the binary erasure channel (BEC), the error floor problem is well understood. Di *et al.* [2] proposed an accurate formula to evaluate the BER and frame error rate (FER) of regular LDPC codes, and pointed out that the "stopping set" is the only factor affecting the performance of iterative decoding. A variable-node set is called a stopping set if all its neighbors are connected to this set at least twice. One example of stopping set is shown in Figure 1. The empty set is a stopping set. The support set of any codeword is a stopping set. A stopping set needs not be the support of a codeword. Since the union of stopping sets is also a stopping set, every set of variable nodes contains a largest stopping set. The message-passing decoder needs a check node with at most one edge connected to an erasure to proceed. So, if the remaining erasures form a stopping set, the decoder must stop. Let E be the initial set of erasures. When the message-passing decoder stops, the remaining set of erasures is the largest stopping set S in E. An optimal (MAP) decoder for a code C on the BEC fails if and only if the set of erased variables includes the support set of a codeword. The message-passing decoder fails if and only the set of erased variables includes a non-empty stopping set. The error floor of a LDPC code over BEC is dominated by the small stopping sets.

For an AWGN channel, MacKay *et al.* [3] discovered a weakness in the construction of the Margulis code which led to high error floors, and that the error floor is caused by a so called near-codeword. Richardson [4] found that error floor of LDPC codes is caused by trapping sets instead of low weight codewords. The so called trapping set (a,b) refers to a variable-node set of a variable nodes in the associated bipartite graph which are connected to b odd-degree and arbitrary number of even-degree parity-check nodes. In the induced subgrap of a trapping set, if the degree of the check nodes is 1 or 2, the trapping sets is called an elementary trapping set. The size of a trapping set is the number of variable nodes in the trapping set. Trapping set with small a and b is the dominant trapping set because when the variable nodes in the trapping set get trapped, there will be no sufficient message flowing into the trapping set to correct the trapped variable nodes.

Let the decoder be the maximum likelihood decoder in one step. Then the trapping sets are the non-zero codeword. Let the decoder be belief propagation over the BEC. Then the trapping sets are the stopping sets. Let the decoder be serial flipping over the binary symmetric channel (BSC). T is a trapping set if and only if in the subgraph induced by T, each node has more even than odd degree neighbors, and the same holds for the complement of T.

978-1-4577-0101-6/11/$26.00 ©2011 IEEE

 978-1-4577-0101-6/11/$26.00 © 2011 IEEE

As we all know, the sub-optimal belief propagation (BP) decoding algorithm for LDPC codes causes undetected errors and detected errors, which are the main contributors to error floor of LDPC codes. The undetected errors occur when the decoding process will converge to a valid codeword that satisfies all the parity-check nodes, but it is not the originally transmitted codeword. The detected errors occur when the decoding process cannot converge to a valid codeword after a certain number of iterations. The former is attributed to small $d_{\min}$. The small trapping sets and stopping sets are the main reason for the presence of the detected errors. It is proved that preventing small stopping sets can avoid small $d_{\min}$. Tian *et al.* [5] found that the graph connectivity also contribute to the error floor, and a metric called EMD/ACE is proposed to evaluate the effect to the error floor. The decoding algorithm also affects the bit error rate (BER) over high SNR. Therefore the factors affecting error floor of LDPC code are girth, stopping sets, trapping sets, EMD/ACE, and decoding algorithm. But the exact relationship between these attributes and the error floor is still unknown.

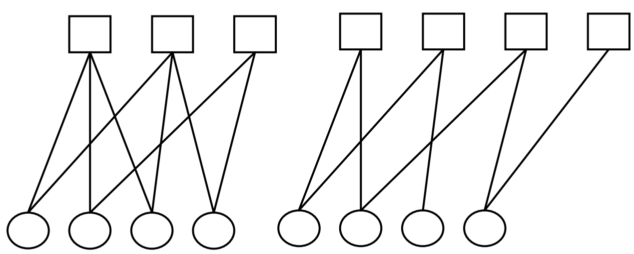

Figure 1. A stopping set (left) and a (4, 1) trapping set (right)

□ represent a check node, ○represent a variable node

III. ANALYSIS OF ERROR FLOOR

A. *Theoretic Estimation of Error Floor*

Now that we know the causes of error floor, it is necessary to detect and eliminate as many bad sub-structures as possible. It has been proved that finding small stopping set or trapping sets by means of combinational characteristic is a NP-complete problem. But there is still several approximate algorithms. For instance, Wang *et al.* [6] proved that it is NP-complete to exhaustively enumerate small stopping sets and trapping sets, and provided an exhaustively enumeration algorithm which can exhaustive enumerate all sub-structures with certain size. But when the code length is greater than 500, the complexity of this algorithm is intractable. Rosnes et al. [7] introduced an efficient algorithm to find all stopping sets with size less than some thresholds in a LDPC matrix. And some other research focus on enumerating trapping sets for specific LDPC codes.

Up to present, no theoretical tool is available to accurately predict the error floor of LDPC codes. Richardson presented a method to estimate error floors of LDPC codes. The first stage aims to find as many dominant trapping sets as possible. Then he classifies all the trapping sets and estimates the FER according to each classified trapping set. However, it shows only the lower bound of FER, because it is very hard to find all the trapping sets. Xia *et al.* [8] and Holzlohner *et al.* [9] proposed two independent importance sampling approaches to evaluate the performance of LDPC codes at very low BERs, respectively. In [8], an error set partitioning method is introduced to reduce the error boundary complexity and to improve the simulation efficiency.□ In [9], a two phase adaptive importance sampling method called Dual Adaptive Importance Sampling (DAIS) based on Multicanonical Monte Carlo (MMC) simulations is introduced. The results show the BER as low as 10^{-19} . Unfortunately, both techniques are based on finding dominant trapping sets and their complexity depends on the codeword length. In [10], Cavus *et al.* introduced an efficient Importance Sampling (IS) method for performance evaluation at very low BER. This method identifies the dominant trapping sets and effectively biases these dominant trapping sets. But the simulation time reduction by the classification of trapping sets cannot give unbiased estimators. In [11], the authors propose an instanton-based techniques for analyzing and reduction the error floor of LDPC codes.

B. *Methods to Reduce Error Floor*

One of the main research subjects on LDPC codes is to find methods for lowering their error floor. There are two kinds of approaches. The first one is to construct better parity-check matrix with low error floor, and the second is to modify the iterative decoding algorithm. Next we will respectively review the two methods.

- Construction of LDPC codes

As we have discussed above, increasing the size of the smallest trapping set or stopping set inevitably reduce the error floor of LDPC codes. However, the design method avoiding small trapping sets or stopping sets directly is yet to be known. It is one of the most challengeable problems in the LDPC code design.

It is quite difficult to construct LDPC codes by means of eliminating all the stopping sets and trapping sets. Now, many researchers design LDPC codes based on the girth or ACE/EMD, while others work with direct measures of error floor performance such as the distribution of stopping sets or trapping sets.

One of the most successful approaches for the construction of finite-length LDPC codes is the progressive-edge-growth (PEG) algorithm proposed by Hu *et al.* [12]. PEG algorithm aims to increase girth by maximizing the local girth at variable nodes in a greedy fashion. The performance comparison of (1024, 512) regular LDPC code generated at random and (1024, 512) irregular LDPC code constructed by PEG algorithm are show in Figure 2. With the use of "approximate cycle extrinsic message degree", Tian *et al.* [5] proposed an algorithm, called ACE algorithm, to produce the check matrices with low error floor by avoiding short cycles with ACE below a given threshold, which indirectly increases the minimum stopping set. The codes constructed by ACE algorithm however perform worse than the codes constructed by standard PEG algorithm in the waterfall region. Therefore, in [13], a PEG-ACE algorithm has been proposed by the combination of PEG and ACE algorithm. PEG-ACE algorithm has good performance at both the waterfall region and the high SNR region. Figure 3 presents the BER performance of

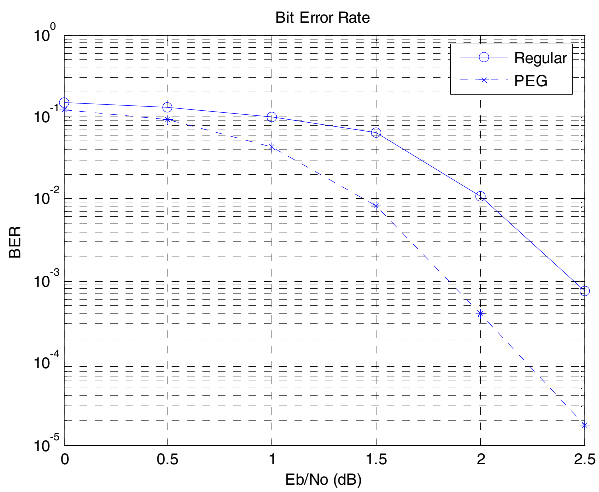

Figure 2. Performance of (1024, 512) LDPC code constructed by PEG algorithm and regular LDPC code

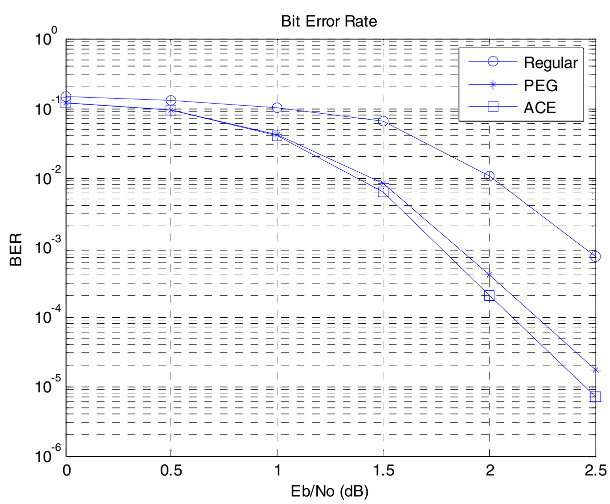

Figure 3. Performance of (1024,512) LDPC code constructed by PEG and ACE algorithm and regular LDPC code. The maximum number of iterations is 30 and the maximum number of codeword errors is 50.

regular LDPC code generated randomly, irregular LDPC code constructed by PEG algorithm, and irregular LDPC code constructed by ACE algorithm with parameter (9, 4). In [14], the authors tried to avoid all the small-size stopping sets with degree no larger than three in the PEG construction method. These algorithms cannot totally avoid stopping set and trapping set, so the error floor is still very high. Wang *et al.* [15] presented a CA (code annealing) algorithm to construct LDPC codes. It is the first paper on code modification with maximizing the size of stopping set as the objective. Based on Tanner graph covers, Ivkovic *et al.* [16] discussed a method which uses the edge swapping between two copies of the same code to avoid most dominant trapping sets. The advantage of this method is that it can be applied to any code/channel/ decoding algorithm, but it obviously increases the code length.

- Modification of Decoding Algorithm

Because of the difficulty in the construction LDPC codes based on trapping sets or stopping sets, the modification of decoder strategies attracts great attention. It can be classified to three categories. The first category modifies iterative decoding algorithm according to the change of extrinsic message during the decoding process. But this method corrects part of the errors caused by trapping sets. The second one improves the decoding algorithm based on finding the dominant trapping sets. The third one does not depend on the trapping sets. On the contrary, it adjust the message transmission according to the check message accumulated during the decoding process.

Reference [17] and [18] attempt to modify the scheduling of the decoders in order to lower the floors of LDPC code. In [19], an averaged decoding algorithm was proposed, which decreases the frequency of trapping set error event by averaging the messages over several iterations. In [20], Chertkov *et al.* uses linear programming-based decoding to reduce the floors. In [21], a post-processor is proposed which exploits the knowledge of the deleterious subgraph, and lowers the floor by inverting the bits of a known trapping set after the decoder becomes trapped in that trapping set. Yang Han *et al.* [22] proposed three kinds of decoder: Bit-mode decoder, Bit-pinning decoder, and three generalized-LDPC decoder. Kang *et al.* [23] presented an iterative decoding algorithm with backtracking to lower the error floors of LDPC codes.

In addition, research on stopping redundancy and trapping redundancy is beneficial to reduce the error floor of LDPC codes. Adding some redundant check nodes can suppress the influence of trapping sets or stopping sets on iterative decoding.

C. Error Floor of Structured LDPC codes

The error floors of structured LDPC codes have also been investigated recently. Reference [24] built a hardware emulation platform to decode structured LDPC codes. Based on Reed-Solomon LDPC codes, it showed that when the implementation precision is high, error floors are attributed to special configurations defined as absorbing sets. Absorbing set is a special subclass of a near-codeword or a trapping set. In the subgraph induced by a trapping set, when each variable node is connected to more check nodes with even degrees than those with odd degrees, the trapping set is an absorbing set. A post-processing approach is formulated to exploit the structure of the absorbing set by biasing the reliabilities of selected messages in a message-passing decoder in [25]. In [26], an adaptive quantization schemes was proposed to alleviate the effects of weak absorbing sets. As a result, only the strong ones dominate the error floor of an optimized decoder implementation. Reference [27] develops a deterministic method of predicting error floors of structured LDPC codes due to absorbing sets based on high (SNR) asymptote.

IV. CONCLUSION

In this paper, we reviewed the concept of trapping set, stopping set, and their related properties, and introduced the reasons leading to error floor problem. Then we introduced the problem of find trapping sets and stopping sets, and the methods that approximately predicted the error floor. After this, we briefly overviewed some aspects of the development of the error floor of LDPC codes, which included construction of LDPC codes and modifications of decoding algorithm. There are still some potential research directions about the error floor of LDPC codes. The relationship between these attributes mentioned above and the error floor performance need to be figured out. The construction of short-length LDPC codes with low error floor and efficient decoding algorithm need further research. No analytical tool is available to evaluate performance of LDPC codes at very low BER. The error floors of most structured LDPC codes such as QC-LDPC codes and array LDPC codes remain to be found. The algorithm enumerated small trapping set or stopping set as much as possible is also anticipated. The research on stopping redundancy and trapping redundancy is also expected.

ACKNOWLEDGMENT

This paper was supported by the National Natural Science Foundations of China (No.60972037 and No.61001182), the Fundamental Research Program of Shenzhen City (No.JC200903120101A and No.JC201005250067A) and the Science and Technology Research Program of Nanshan District (No.2009045).

REFERENCES

[1] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory,* vol. 8, pp. 21-28, 1962.

[2] D. Changyan, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1570-1579, 2002.

[3] D. J. C. MacKay, and M. Postol, "Weaknesses of margulis and ramanujan-margulis low-density parity-check codes," *Electronic Notes in Theoretical Computer Science*, vol. 74, 2003.

[4] T.J.Richardson,"Error floors of LDPC codes," in *Proc. 41st Annu. Allerton Conf. Commun., Control, Comput., Urbana-Champaign*, 2003, pp.1426-1435.

[5] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. Commun.*, vol. 52, pp. 1242-1247, 2004.

[6] C.-C .Wang, S. R. Kulkarni, and H. V. Poor, "Finding All Small Error-Prone Substructures in LDPC Codes,", *IEEE Trans. Inform. Theory*, vol. 55, pp. 1976-1999, 2009.

[7] E. Rosnes and O. Ytrehus, "An Efficient Algorithm to Find All Small-Size Stopping Sets of Low-Density Parity-Check Matrices," *IEEE Trans. Inform. Theory,* vol. 55, pp. 4167-4178, 2009.

[8] B. Xia and W. E. Ryan, "Estimating LDPC codeword error rates via importance sampling," in *Proc. International Symposium Information Theory (ISIT)*, 2004.pp. 473.

[9] R. Holzlohner, A. Mahadevan, C. R. Menyuk, J. M. Morris, and J. Zweck, "Evaluation of the very low BER of FEC codes using dual adaptive importance sampling," *IEEE Commun. Lett.*, vol. 9, pp. 163-165, 2005.

[10] E. Cavus, C. L. Haymes, and B. Daneshrad, "Low BER performance estimation of LDPC codes via application of importance sampling to trapping sets," *IEEE Trans. Commun.*, vol. 57, pp. 1886-1888, 2009.

[11] S. Chilappagari, M. Chertkov, M. Stepanov, and B. Vasic, "Instanton-based techniques for analysis and reduction of error floors of LDPC codes," *IEEE J. Select. Areas in Commun.*,vol. 27, pp. 855-865, 2009.

[12] X.-Y H, E. Eleftheriou, and D. M. Arnold, "Progressive edge-growth Tanner graphs," in Proc. *IEEE GLOBECOM '01*, 2001, vol. 2.pp. 995-1001

[13] H.Xiao and A. H. Banihashemi, "Improved progressive-edge-growth (PEG) construction of irregular LDPC codes," *IEEE Commun. Lett.*, vol. 8, pp. 715-717, 2004.

[14] G. Richter and A. Hof, "On a Construction Method of Irregular LDPC Codes Without Small Stopping Sets," in *Proc. IEEE ICC '06*. 2006, pp. 1119-1124.

[15] C. C.Wang,, "Code Annealing and the Suppressing Effect of the Cyclically Lifted LDPC Code Ensemble," in *Proc. IEEE Information Theory Workshop (ITW '06)*, 2006, pp. 86-90.

[16] M. Ivkovic, S. K. Chilappagari, and B. Vasic, "Eliminating Trapping Sets in Low-Density Parity-Check Codes by Using Tanner Graph Covers," *IEEE Trans. Inform. Theory,* vol. 54, pp. 3763-3768, 2008.

[17] X. Hua and A. H. Banihashemi, "Graph-based message-passing schedules for decoding LDPC codes," *IEEE Trans. Commun.*, vol. 52, pp. 2098-2105, 2004.

[18] A. I. Vila Casado, M. Griot, and R. D. Wesel, "LDPC Decoders with Informed Dynamic Scheduling," *IEEE Trans. Commun.,* vol. 58, pp. 3470-3479.

[19] S. Landner and O. Milenkovic, "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes," in *Proc. International Conference on Wireless Networks, Communications and Mobile Computing* , 2005, pp. 630-635 vol.1.

[20] M. Chertkov, "Reducing the Error Floor," in *Proc. IEEE Information Theory Workshop (ITW '07)*, 2007, pp. 230-235.

[21] E. Cavus and B. Daneshrad, "A performance improvement and error floor avoidance technique for belief propagation decoding of LDPC codes," in *Proc. IEEE PIMRC 2005*,vol. 4, pp. 2386-2390.

[22] H. Yang and W. Ryan, "Low-floor decoders for LDPC codes," *IEEE Trans. Commun.*, vol. 57, pp. 1663-1673, 2009.

[23] J. Kang, H. Qin, L. Shu, and K. Abdel-Ghaffar, "An Iterative Decoding Algorithm with Backtracking to Lower the Error-Floors of LDPC Codes," *IEEE Trans. Commun.*, vol. 59, pp. 64-73.

[24] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. Wainwright,, "GEN03-6: Investigation of Error Floors of Structured Low-Density Parity-Check Codes by Hardware Emulation," in *Proc. IEEE GLOBECOM '06.*, 2006, pp. 1-6.

[25] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. J. Wainwright, "Lowering LDPC Error Floors by Postprocessing," in *Proc.* IEEE *GLOBECOM 2008,* pp. 1-6.

[26] Z. Zhang, L. Dolecek, M. Wainwright, V. Anantharam, and B. Nikolic,, "Quantization Effects in Low-Density Parity-Check Decoders," in *Proc. ICC 2007*, pp. 6231-6237.

[27] L. Dolecek, P. Lee, Z. Zhang, V. Anantharam, B. Nikolic, and M. Wainwright, "Predicting error floors of structured LDPC codes: deterministic bounds and estimates," *IEEE J. Select. Areas in Commun.*, vol. 27, pp. 908-917, 2009.