Available online at www.sciencedirect.com
ScienceDirect
Journal of the Franklin Institute 347 (2010) 1368–1373
Journal of The Franklin Institute
www.elsevier.com/locate/jfranklin

Short communication

A fast BER evaluation method for LDGM codes ☆

Cheng-Chun Chang[a], Zhi-Hong Mao[b], Heung-No Lee[c,*]

[a]*Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan, ROC*
[b]*Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA, USA*
[c]*Gwangju Institute of Science and Technology (GIST), Republic of Korea*

Received 27 June 2008; received in revised form 26 May 2009; accepted 14 April 2010

Abstract

Low-density generator matrix (LDGM) codes have recently drawn researchers' attention thanks to their satisfying performance while requiring only moderate encoding/decoding complexities as well as to their applicability to network codes. In this paper, we aim to propose a fast simulation method useful to investigate the performance of LDGM code. Supported by the *confidence interval analysis*, the presented method is, for example, 10^8 times quicker than the Monte-Carlo computer simulation for bit-error-rate (BER) in 10^{-10} region.

Keywords: Density evolution; Sum–product iterative decoding; LDGM codes

1. Introduction

Systematic binary linear block codes generated by simple parity-check operations have been of interest in the past due to the properties that the codes can provide good performance while the encoding and decoding complexities can be managed to be low [1–3].

We study the class of low-density generator matrix (LDGM) codes. LDGM codes are linear block codes with sparse generator matrix in systematic form. Codes of sparse

☆The work of C.-C. Chang and Z.-H. Mao was supported in part by the National Science Foundation, U.S.A., under Grant CMMI-0727256. The work of H.-N. Lee was in part supported by the Korea National Research Foundation (NRF) under Grant K20901000004-09E0100-00410, and in part by the National Science Foundation, U.S.A., under Grant CMMI-0727256.

*Corresponding author. Tel.: +82 62 970 2237; fax: +82 62 970 2204.
E-mail addresses: ccchang@ntut.edu.tw (C.-C. Chang), maozh@engr.pitt.edu (Z.-H. Mao), heungno@gist.ac.kr, heungno@gmail.com (H.-N. Lee).

0016-0032/$32.00 © 2010 The Franklin Institute. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.jfranklin.2010.04.011

generator matrices have been reported to be asymptotically bad due to the fact that the minimum distance of the codes does not grow with code length [4]. Even with these disadvantages, LDGM codes are still interesting not only because they can deliver satisfying error-performance at moderate block length (e.g. a thousand) while the encoding and decoding complexities are low, but also because there exist some new applications such as joint source channel encoding systems [5] and network coding on the cooperative wireless multiple access relay networks [6]; in all these systems the inherent *systematic* form of LDGM codes make the codes very useful.

In this paper, we investigate the error-performance of regular LDGM code with the simple binary phase shift keying (BPSK) modulation system. For a given code length, the number of ones in columns and the number of ones in rows of the generator matrix are the key design parameters for the encoder of LDGM codes, as these parameters determine code rates and the error-performance. In order to select an LDGM code that delivers the best performance, an extensive search on design parameter space is essential; thus a fast simulation method for LDGM codes would be of great help.

We aim in this paper to propose a fast simulation method for LDGM codes which is constructed using the general idea of *density evolution* [7,8]. On the one hand, the fast method tracks the mean of *log-likelihood ratio* (LLR) samples and thus enables faster evaluation. On the other hand, it exploits the inherent structure of LDGM codes. The density evolution originally developed for LDPC codes [7] is computation intensive. While it was used to determine the capacity of LDPC codes, as was done in [7], it is generally not a good idea to use density evolution for bit-error-rate (BER) evaluation. This is true for LDPC codes, but with a little surprise not for LDGM codes. There is a significant structural difference between the parity-check matrices of two codes; as the result, the idea of density evolution can be useful as a BER evaluation tool for LDGM codes, removing the need for time-consuming Monte-Carlo simulation. To systematize this approach, we include a *confidence interval analysis* which allows us to determine the number of samples required for a targeted accuracy.

The rest of the paper is organized as follows. Section 2 briefly introduces LDGM codes. The fast simulation method is presented in Section 3. Simulation results and discussions are provided in Section 4. Section 5 gives the conclusion.

2. LDGM codes

LDGM codes are linear block codes with parity-check matrix $\mathbf{H}=[\mathbf{P}\ ;\ \mathbf{I}]$, where $\mathbf{P}$ is an $(n-k)$ by k sparse matrix and $\mathbf{I}$ is the $(n-k)$ by $(n-k)$ identity matrix. The index k denotes the number of input bits and n denotes the length of the code. An LDGM code is called *regular* if both the number of 1's in a column in the $\mathbf{P}$ matrix and that in a row stay fixed for all columns and rows. Though irregularity can provide performance improvement, regularity could lead to simplified modular hardware implementation. The *degree* of a variable node, denoted as d_v, which is the number of ones in a column in the $\mathbf{P}$ matrix. Similarly, the *degree* of a check node, d_c, represents the number of ones in a row in the $\mathbf{H}$ matrix. Based on the structure of $\mathbf{H}$ matrix, the code rate R of (d_v, d_c)-*regular* LDGM code is given by

$$R = 1/(d_v/(d_c-1)+1). \quad (1)$$

The degree of a variable node d_v and the degree of a check node d_c are the main design parameters for LDGM codes. For a given code length n and design parameters (d_v, d_c), a $\mathbf{P}$

matrix can be obtained through random generation. The encoding complexity can be maintained to be low due to the sparseness of the **P** matrix.

3. The fast simulation method

The simulation flow diagram of the fast simulation method is depicted in Fig. 1. As shown in Fig. 1, we first obtain the initial *LLRs* of all-zero codeword from a discrete-time Gaussian channel. We then feed the channel *LLRs* of the all-zero codeword to the standard iterative decoder. After running the sum–product iterative decoding algorithm, we obtain the posterior *LLRs*,$LLR^P s$. Instead of determining the decoded bits from these $LLR^P s$ as used for conventional decoding, we calculate the mean $-\alpha$, $\alpha \geq 0$, of the set of $LLR^P s$. In the method, we use (3) to determine the required number of LLR^P samples for reliable estimation of the mean value. If the number of samples is insufficient, we then feed another all-zero codeword, and compute the mean $-\alpha$ cumulatively. When reaching sufficient number of samples, we substitute the cumulative mean value into (2), and the error performance of the LDGM coded BPSK modulation system is extracted. The discussion and working principle of the fast simulation method is provided in the following.

We note that the value α evolves only up to a certain limit for LDGM codes after which it stays at the same level no matter how many iteration are carried out further. This is one of the distinct features of LDGM codes: The information provided by the parity-check variable nodes always remains the same during the iterations. The reason for this is that all those variable nodes which participate in a single parity-check cannot pass along any extrinsic message to the check node. Extrinsic messages are generated by excluding the old message. Thus, to those variable nodes that only have a *single* connection to the check nodes, this is not possible. This causes obstruction to continued evolution of density and hence results in error-floors. We show that this behavior can be well evaluated by our fast evaluation system.

With the LDGM coded BPSK modulation system over Gaussian channels, we have the output-symmetric channels. Then, we note that the distribution for channel *LLRs* is symmetric and the symmetry is preserved throughout the message-passing decoding algorithm [7]. Thus, we may treat $LLR^P s$ as Gaussian distributed samples with mean $-\alpha$, $\alpha \geq 0$, and variance $\beta^2 = 2\alpha$, see Ref. [8]. Then, the error-probability which is the Gaussian tail probability can be obtained by

$$P_e = \int_0^\infty \frac{1}{\sqrt{2\pi \cdot 2\alpha}} e^{-(\lambda+\alpha)^2/2 \cdot 2\alpha} d\lambda, \tag{2}$$

or by $0.5erfc(\sqrt{\alpha/4})$ in the form of complementary error function.

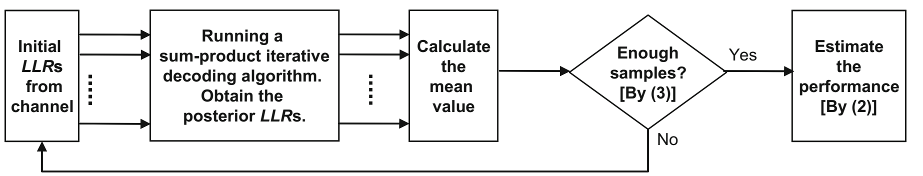

Fig. 1. The flow diagram of the fast simulation method.

It is useful to determine the number of required LLR^P samples such that the estimated mean is accurate. For this, we resort to the confidence interval analysis, and obtain that

$$N = 8664 \times [erfc^{-1}(2P_e)]^2, \tag{3}$$

where N is the required number of samples, $erfc^{-1}(\cdot)$ the inverse of the complementary error function, and P_e the target error probability in simulation. For example, having the number of samples around $N=10^5$ is sufficient to provide a BER result of $P_e=10^{-10}$ region. We note that for the conventional Monte-Carlo computer simulation, if one wants to collect one thousand bit errors to have a smooth BER curve in simulation, the required number of LLR^P samples is $N' \simeq 10^3/P_e$, which requires $N'=10^{13}$ for $P_e=10^{-10}$. Therefore, the presented method is about 10^8 times quicker in this example. In general, for a target P_e the method is F times quicker, where F is defined as

$$F \triangleq \frac{N'}{N} = \frac{10^3}{8664 \times P_e \times [erfc^{-1}(2P_e)]^2} \tag{4}$$

For more examples, we note $F \simeq 4.2$ when $P_e=10^{-2}$, $F \simeq 167$ when $P_e=10^{-4}$, $F \simeq 1.0 \times 10^4$ when $P_e=10^{-6}$, and $F \simeq 7.3 \times 10^5$ when $P_e=10^{-8}$.

The derivation of (3) can be elaborated as follows. The confidence interval for mean α of a Gaussian distribution with known variance β^2 is given by $CONF\{\overline{x}-\varepsilon \leq \alpha \leq \overline{x}+\varepsilon\}$ [9], where $\overline{x}$ is the experimental mean of samples. The positive constant ε is a tolerant error and can be described by $\varepsilon = c\beta/\sqrt{N}$, where N is the number of samples and c is a value depending on the confidence level γ. In particular, given a confidence interval, c is the value satisfying $P(-c \leq Z \leq c)=\gamma$ where Z is the Gaussian distributed random variable with mean 0 and variance 1. For example, $c=3.291$ when $\gamma=0.999$. In the analysis, we choose ε to be $\varepsilon=0.1$. The estimated mean within this chosen error tolerance results in a negligible difference in the error probability (2) in the region of interest. In fact, $\varepsilon=0.1$ results in the same order of P_e when P_e is above 10^{-14}. For a given target error probability P_e, we note that it is equivalent to investigate the Gaussian distribution with mean $\alpha=4 \times erfc^{-1}(2P_e)$ (from (2)) and variance $\beta^2=8 \times erfc^{-1}(2P_e)$. Since $\varepsilon = c\beta/\sqrt{N}$, for $\varepsilon=0.1$, we obtain $N=(c\beta/\varepsilon)^2=8664 \times [erfc^{-1}(2P_e)]^2$.

We found the performance obtained from the fast simulation method shows good match to the performance evaluated from Monte-Carlo simulation. In Section 4, we illustrate this point and investigate the relationship between the key design parameters (d_v,d_c) and the performance of LDGM codes.

4. Simulation results and discussion

Fig. 2 shows the results of Monte-Carlo computer simulation and the numerical results for rate half, $R=0.5$ LDGM codes of length 4080. The parameters (d_v,d_c) used are (7, 8), (8, 9), (9, 10), (10, 11), and (11, 12), respectively. Ten iterations are used in the standard log ratio based message-passing decoding algorithm. The *numerical* results, we say in comparison to *simulation*, are obtained from the fast evaluation method. The *simulation* and the *numerical* results match almost perfectly at the error-floor regions, while for the waterfall regions the simulation results are off a fraction of dB to the numerical results. We note that this gap can be closed by increasing the number of iterations in Monte-Carlo simulation.

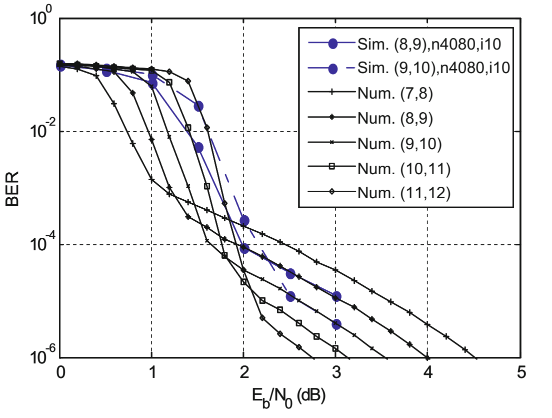

Fig. 2. Performance of rate $\frac{1}{2}$ LDGM codes with code length 4080 under AWGN channels. The Sim curves are obtained from Monte-Carlo computer simulation with 10 iterations. The Num curves are obtained from the fast simulation method. A higher density code has a later waterfall region but a lower error-floor level.

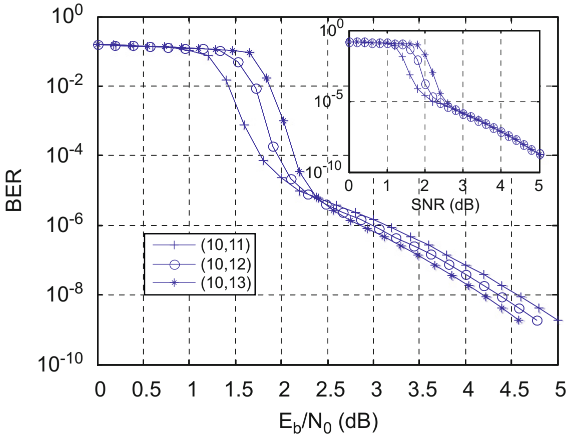

Fig. 3. Performance of rate 0.500 (10,11), rate 0.5238 (10,12), and rate 0.5455 (10,13) LDGM codes obtained from the fast simulation method. The curves in the sub-figure are drawn with respective to SNR (E_s/N_0), while the curves in the main figure are calibrated to E_b/N_0. A higher rate code shows a later waterfall region but a slightly lower error-floor level.

Fig. 3 shows the numerical results for rate $R=0.500$, (10, 11), rate $R=0.5238$, (10, 12), and rate $R=0.5455$, (10, 13), LDGM codes. We note that the higher the rate is, the larger E_b/N_0 gets for the waterfall region and the lower the error-floor gets. In fact, the error-floors remain at the same level in terms of SNR (E_s/N_0), and are determined by the minimum distance. From Ref. [10], the minimum distance of this code is determined by d_v. Since d_v is 10 for all these codes, they all have the same minimum distance. As the rate changes slightly with the variation of the parameter d_c, the waterfall region is affected. We note that the number of 1's in the row of generator matrix d_c determines the waterfall region – the larger the d_c, the later the waterfall region. The number of 1's in the column d_v determines the error-floor level – the larger the d_v, the lower the error-floor level.

Therefore, in Fig. 2, it is also noted that a higher density code has a later waterfall region but a lower error-floor level.

One major drawback of LDGM codes, as evidenced in the above examples, is relatively high error floors. Recently, it has been shown that the error-floors of LDGM codes can be lowered significantly through parallel or serial concatenation of LDGM codes, while maintaining low encoding and decoding computational complexity [11,12]. We envision that the presented fast simulation method can be extended to the performance analysis of parallel or serial concatenation LDGM codes.

5. Conclusion

The presented performance evaluation framework provides the capability to quickly assess the performance of LDGM codes. It is, for example, about 10^8 times quicker than the Monte-Carlo computer simulation for bit-error-rate in 10^{-10} region. The new methods shed lights on the error floor characteristics of LDGM codes. For rate around half codes, we have shown that the number of 1's in the columns of generator matrix determines the error-floor level, whereas the number of 1's in the rows determines the waterfall region.

References

[1] L. Ping, S. Chan, K.L. Yeung, Iterative decoding of multi-dimensional concatenated single parity check codes, IEEE Int. Conf. Commun. (1998) 131–135 June.

[2] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, D.A. Spielman, Efficient erasure correcting codes, IEEE Trans. Inform. Theory (2001) 569–584 February.

[3] T.R. Oenning, J. Moon, A low-density generator matrix interpretation of parallel concatenated single bit parity codes, IEEE Trans. Magnetics (2001) 737–741 March.

[4] D.J.C. MacKay, Good error-correcting codes based on very sparse matrices, IEEE Trans. Inform. Theory (1999) 399–431 March.

[5] W. Zhong, J. Garcia-Frias, LDGM codes for channel coding and joint source-channel coding of correlated sources, EURASIP J. Appl. Signal Process. (2005) 942–953 June.

[6] X. Bao, and J. Li, Matching code-on-graph with network-on-graph: adaptive network coding for wireless relay networks, in: Proceedings of the 43rd Annual Allerton Conference on Communication, Control, and Computing, Champaign IL, September, 2005.

[7] T.J. Richardson, M.A. Shokrollahi, R.L. Urbanke, Design of capacity-approaching irregular low-density parity-check codes, IEEE Trans. Inform. Theory (2001) 619–637 February.

[8] S.-Y. Chung, T.J. Richardson, R.L. Urbanke, Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation, IEEE Trans. Inform. Theory (2001) 657–670 February.

[9] E. Kreyszig, Advanced Engineering Mathematics, 9th edition, Wiley.

[10] Heung-No Lee, Exact distance spectrum for low density generator matrix codes, unpublished.

[11] M. Gonzalez-Lopez, F.J. Vazquez-Araujo, L. Castedo, J. Garcia-Frias, Serially-concatenated low-density generator matrix (SCLDGM) codes for transmission over AWGN and Rayleigh fading channels, IEEE Trans. Wireless Commun. (2007) 2753–2758 August.

[12] W. Zhong, H. Chai, and J. Garcia-Frias, Approaching the Shannon limit through parallel concatenation of regular LDGM codes, in: Proceedings of the International Symposium on Information Theory, September,2005, pp. 1753–1757.