The Design and Simulation of q-ary LDPC Codes Based on the PEG algorithm

Yong Li[1]
1. Dept.of Communication Engineering, Xiamen University
Xiamen,361005,China
Email: liyongbhj@126.com

Lin Wang[1.2]
2. ChongQing University Of Post and Telecommunication
ChongQing, 400065,China
Email: wanglin@xmu.edu.cn

Junbin Chen[1]
1. Dept.of Communication Engineering, Xiamen University
Xiamen,361005,China
Email: junbin@vip.sina.com

***Abstract*— In this paper the PEG (Progressive Edge-Growth) algorithm is introduced into the design of H of q-ary LDPC codes, as well the fourier transform(FT) decoding algorithm with a equivalent transform(ET) is proposed. Simulation results show that the performance of LDPC codes on GF(4) based on PEG Tanner graphs is significantly better than that based on random graphs at the short and medium block lengths. And LDPC codes on GF(4) outperform that on GF(2) slightly with analogous complexity. Obviously LDPC codes on GF(4) based on PEG principle are of importantly practical value in future digital communication systems.**

I. INTRODUCTION

Binary LDPC codes have displayed near Shannon limit performance when the BP (Belief Propagation) algorithm is used in decoding [2], [3]. Although random graphs have been used to construct LDPC codes with impressive performance [4], [5], there is no guarantee that any given random graph also defines a good code with a suitable shortest cycle (girth) to facilitate iterative decoding, especially for relatively short block lengths. To construct a Tanner graph with a relatively large girth a sub-optimal algorithm called PEG algorithm is proposed by X. -Y. Hu [1]. It is proved that the resulting LDPC codes of PEG Tanner graphs significantly outperform randomly constructed ones, and the corresponding check matrixes **H** are so steady that there is no need to search the best ones, which must have been done in random construction method.

Moreover, the research results of David.J.Mackay indicate that the non-binary LDPC codes excel the binary LDPC codes both based on random graphs in some low rates [6]. Enlightened by [1], the PEG algorithm is introduced into the design of **H** of non-binary LDPC codes different from the encoding algorithm in David.J.Mackay and others in this paper. And a new decoding algorithm based on FT algorithm with the equivalent transform is proposed different from that in [3]–[7]. It is much simpler than the BP algorithm although its performance decreases a little. Meanwhile the LDPC codes on GF(4) decoded by this algorithm still outperform binary LDPC codes. The paper is arranged as below. In section II we describe the principle of encoding using the PEG algorithm in q-ary LDPC codes. In section III the simpler decoding algorithm is provided. Some good simulation results are obtained In section IV. Our remarks are arranged in conclusions.

II. THE CONSTRUCTION OF CHECK MATRIX OF Q-ARY LDPC CODES

The bipartite graphs of q-ary LDPC codes are similar to those of binary LDPC codes, but variable nodes have q possible values, besides, the constraint restrict of check nodes is more complex. We let $q = 2^p$ because we use p bits in binary channel to transmit a non-binary symbol.

In this paper, the **H** is constructed by the PEG algorithm. Thereinto, for convenience the non-zero elements are chosen randomly from the ensemble $\{1, 2, \ldots, q-1\}$. Of course, the performance of LDPC Codes on GF(q) may be improved obviously if the non-zero elements are chosen based on maximizing the entropies of row. Namely, the non-zero elements $\{1, 2, \ldots, q-1\}$ are chosen according to a special distribution, rather than completely randomly. First we assume channel models known, e.g., BSC. Then we choose in each row to maximize the entropy of the corresponding bit of the syndrome vector **z**=**Hx**, where **x** is a sample from the assumed channel noise model. The bigger the entropies of syndrome, the better the performance of decoder.

A. The Principle of PEG Algorithm

Given a **H** having dimension $m \times n$, its Tanner graph is denoted as (*V*, *E*), with *V* the set of vertices (nodes), i.e. $V = V_c \cup V_s$ where, $V_c = \{c_0, c_1, \ldots, c_{m-1}\}$is the set of check nodes and $V_s = \{s_0, s_1, \ldots, s_{n-1}\}$ the set of symbol nodes. *E* is the set of edges such that $E = E_c \times E_s$, with edge $(c_i, s_j) \in E$ if and only if $h_{ij} \neq 0$, $h_{ij} \in \mathbf{H}$, $0 \leq i \leq m-1$, $0 \leq j \leq n-1$. Denote the degree of symbol node s_j by d_{s_j}. Let also the set of edges *E* be partitioned in terms of V_s as $E = E_{s_0} \cup E_{s_1} \cup \ldots \cup E_{s_{n-1}}$, with E_{s_j} containing all edges incident on symbol node s_j. Finally, denote the *k*-th edge incident on s_j by $E^k_{s_j}$, $0 \leq E^k_{s_j} \leq d_{s_j} - 1$.

For a given symbol node s_j, define its *neighbor within depth l* , $\mathcal{N}^l_{s_j}$, as the set consisting of all check nodes reached by a tree spreading from symbol node s_j within depth *l*, as shown in the example in Figure. 1. Its complementary set, $\bar{\mathcal{N}}^l_{s_j}$, is defined as $V_c \backslash \mathcal{N}^l_{s_j}$. In graph theory, *girth* g refers to the length of the shortest cycle in a graph. For each symbol node s_j, we define a local girth g_{s_j} as the length of the shortest cycle passing through that symbol node. The set of local girths $\{g_{s_j}\}$ is referred to as girth histogram. It follows, by definition, that $g = min_j\{g_{s_j}\}$.

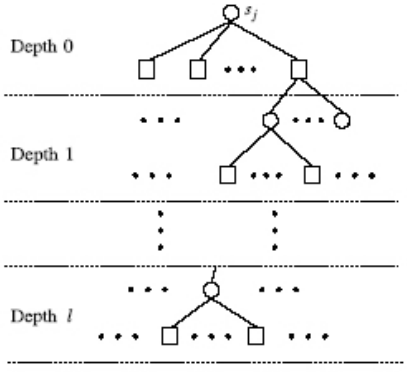

Fig. 1: Neighbor $\mathcal{N}^l_j$ within depth *l* of symbol node s_j.

Constructing a Tanner graph with the largest possible girth is combinatorially difficult. Nevertheless, a sub-optimum algorithm to construct a Tanner graph with a relatively large girth is possible. One such algorithm is the PEG algorithm, in which the local girth of a symbol node is maximized whenever a new edge is added to the node. Suppose we have finished constructing edges of the first *j* symbol nodes on a Tanner graph, i.e., $E = E_{s_0} \cup E_{s_1} \cup \ldots \cup E_{s_{j-1}}$ have been established. Let g^t be the temporary girth under the current graph, i.e., g^t=min$\{g_{s_0}, g_{s_1}, \ldots, g_{s_{j-1}}\}$. The problem now lies in selecting the edge set E_{s_j} such that adding these new edges to the current graph setting does not excessively impair the current g^t. In PEG algorithm, d_{s_j} edges of E_{s_j} are added to the current graph on an edge-by-edge basis, and the length of the shortest cycle passing through symbol node s_j is maximized whenever a new edge originating in s_j is being added. First the tree originating in symbol node s_j is expanded up to depth *l* each time a new edge of s_j is being determined, such that $\bar{\mathcal{N}}^l_{s_j} \neq \varnothing$ but $\bar{\mathcal{N}}^{l+1}_{s_j} = \varnothing$ or the cardinality of $\mathcal{N}^l_{s_j}$ stops increasing but is smaller than *m*, and then placing an edge between s_j and a check node selected from $\bar{\mathcal{N}}^l_{s_j}$. The shortest cycle passing through this new edge is guaranteed to be no shorter than 2(*l*+2).

A subtle point in the PEG algorithm needs further comment. Whenever we encounter multiple choices for connecting to s_j, i.e., multiple check nodes exist in $\bar{\mathcal{N}}^l_{s_j}$, we select the one having the smallest number of incidence edges under the current graph setting. This renders the resulting PEG Tanner graph as check-node-degree uniform as possible. Multiple choices may still exist because multiple check nodes in $\bar{\mathcal{N}}^l_{s_j}$ might have the same lowest degree, particularly at the initial period of construction. Here we randomly select one of these check nodes.

B. Sum and Product Operation on GF(4)

Sum and product operations on GF(*q*) refer to [8], and that on GF(4) are shown in Table I.

TABLE I: Sum and Product operation on GF(4)

$\oplus$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

$\otimes$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

When **H** is a large-scale sparse matrix, we use Gaussian elimination to derive the generator matrix **G** to generate codewords.

III. Decoding of q-ary LDPC Codes

We will refer to elements of **x** as noise symbols and elements of **z** as checks. Let $\mathcal{N}(m) := \{n : H_{mn} \neq 0\}$ be the set of noise symbols that participate in check *m*. The decoding problem is to find the most probable vector **x** such that **Hx=z**, with the likelihood of **x** determined by the channel model. Let $\mathcal{M}(n) := \{m : H_{mn} \neq 0\}$ be the set of checks that depend on noise symbol *n*.

With each nonzero entry in the parity check matrix H_{mn}, we associate quantities q^a_{mn}, r^a_{mn} for $a \in GF(q)$. The quantity q^a_{mn} is meant to be the probability that symbol *n* of **x** is *a*, given the information obtained via checks other than check *m*. The quantity r^a_{mn} is meant to be the probability of check *m* being satisfied if symbol *n* of **x** is considered fixed at *a* and the other noise symbols have a separable distribution given by the probabilities $\{q^a_{mn'} : n' \in \mathcal{N}(m)\backslash n,\ a \in GF(q)\}$. The value of r^a_{mn} is:

$$r^a_{mn} = \sum_{x:x_n=a} \delta\left(\sum_{n'\in\mathcal{N}(m)} H_{mn'}x_{n'} = z_m\right) \prod_{j\in\mathcal{N}(m)\backslash n} q^{x_j}_{m_j} \tag{1}$$

A. Fourier Transform Decoding

If LDPC codes are decoded by the classic BP algorithm, the complexity of decoding scales as q^2 per iteration, but it can be reduced using a Fourier transform of the probabilities [6], [10]. Because (1) represents a convolution of the quantities q^a_{mj}, the summation can be replaced by a product of the Fourier transforms (taken over the additive group of GF(q)) of q^a_{mj} for $j \in \mathcal{N}(m)\backslash n$, followed by an inverse Fourier transform. The Fourier transform *F* of a function *f* over GF(2) is given by $F^0 = f^0 + f^1$, $F^1 = f^0 - f^1$. Transforms over $GF(2^p)$ can be viewed as a sequence of binary transforms in each of *p* dimensions. Hence for GF(4) we have

$$\begin{aligned} F^0 &= [f^0 + f^1] + [f^2 + f^3] \\ F^1 &= [f^0 - f^1] + [f^2 - f^3] \\ F^2 &= [f^0 + f^1] - [f^2 + f^3] \\ F^3 &= [f^0 - f^1] - [f^2 - f^3] \end{aligned} \tag{2}$$

The inverse transform is the same, followed by division by 2^p.

Let (Q^0_{mj}, ... , Q^{q-1}_{mj}) represent the Fourier transform of the vector (q^0_{mj}, ... , q^{q-1}_{mj}), now r^a_{mn} is the *a*'th coordinate of the inverse transform of

$$\left(\left(\prod_{j\in\mathcal{N}(m)\backslash n} Q^0_{mj}\right), \ldots, \left(\prod_{j\in\mathcal{N}(m)\backslash n} Q^{q-1}_{mj}\right)\right)$$

B. The Equivalent Transform

Assume **C** to be a codeword, then $\sum_j h_{ij}c_j = 0$(for *i*'th check node). On GF(2), we can have $\sum_j c_j = 0$(*j* satisfies the inequation $h_{ij} \neq 0$) for non-zero elements are all '1'. However, on GF(*q*) it is not the case for the nonzero elements are selected from the ensemble $\{1, 2, \ldots, q-1\}$, so we can not copy the flow of decoding on GF(2) simply before modifying the computing of Q^a_{mj} and r^a_{mn}. For decoding accurately, the equivalent transform(ET) is proposed.

Now we analyse the equation $\sum_j h_{ij}c_j = 0$ on GF(*q*). Here, h_{ij} can not be omitted for $h_{ij} \in \{1, 2, \ldots, q-1\}$. Let $c'_j = h_{ij}c_j$, then $\sum_j c'_j = 0$(*j* satisfies the inequation $h_{ij} \neq 0$). on this condition, LDPC codes can be decoded by copying the decoding process on GF(2). We obtain the equivalent transform vector (q'^0_{mj}, ... , q'^{q-1}_{mj}) of the vector (q^0_{mj}, ... , q^{q-1}_{mj}) from (3), (Q'^0_{mj}, ... , Q'^{q-1}_{mj}) from (2), r'_{mn} by inverse Fourier transform(IFT) of (($\prod_{j\in\mathcal{N}(m)\backslash n} Q'^0_{mj}$), ... , ($\prod_{j\in\mathcal{N}(m)\backslash n} Q'^{q-1}_{mj}$)), and r_{mn} by the inverse equivalent transform(IET) of r'_{mn}. The complexity of decoding isn't added because just a mathematical equivalent transform is done, which scales as $Ntq\log^q$(N is the blocklength and t is the average column weight). The equivalent transform is given by

$$q'^a_{mj} = q^{a\div h_{mj}}_{mj}, r^a_{mj} = r'^{(a\otimes h_{mj})}_{mj} \tag{3}$$

where, the division operator $'\div'$ is added artificially for convenient for comprehension, deduced from the product rule. e.g., the division operation on GF(4) refers to Table II(row :dividend, column:divisor).

TABLE II: Division operation on GF(4)

÷	0	1	2	3
0	×	×	×	×
1	0	1	2	3
2	0	3	1	2
3	0	2	3	1

The detailed-step:

A.*Initialization*

We initialize the values of q^a_{mn} to f^a_n, the likelihood that $x_n = a$ according to the channel model.

B.*The ET and the FT*

$$\begin{aligned} q'^a_{mn} &= q^{a\div h_{mn}}_{mn} \\ Q'^a_{mn} &= FT[q'^0_{mn}, \ldots, q'^{q-1}_{mn}] \end{aligned} \tag{4}$$

C.*The IFT and IET(updating r^a_{mn})*

$$r'^a_{mn} = IFT[(\prod_{j\in\mathcal{N}(m)\backslash n} Q'^0_{mj}), \ldots, (\prod_{j\in\mathcal{N}(m)\backslash n} Q'^{q-1}_{mj})]$$

$$r^a_{mn} = r'^{(a\otimes h_{mn})}_{mn} \tag{5}$$

D. *Updating q^a_{mn}*

$$q^a_{mn} = \alpha_{mn} f^a_n \prod_{j\in\mathcal{M}(n)\backslash m} r^a_{jn} \tag{6}$$

where α_{mn} is chosen such that $\sum_{a=0}^{q-1} q^a_{mn} = 1$.

E. calculating q^a_n

$$q^a_n = \alpha_n f^a_n \prod_{j\in\mathcal{M}(n)} r^a_{jn} \tag{7}$$

where α_n is chosen such that $\sum_{a=0}^{q-1} q^a_n = 1$.

We then make a tentative decoding $\hat{x}$ such that

$$\hat{x}_n = k(q^k_n = max\{q^0_n, \ldots, q^{q-1}_n\}) \tag{8}$$

If $\mathbf{H}\hat{\mathbf{x}} = \mathbf{z}$ then the decoding algorithm halts having identified a valid decoding of the syndrome, otherwise the algorithm repeats(from (4) to (8)). A failure is declared if some maximum number of iterations (e.g., 200) occurs without a valid decoding.

IV. Simulation Results

In this section we give the performance comparison between the regular (3, 6) LDPC code based on the PEG Tanner graph and that based on the random graph, on GF(4), over AWGN channel in Fig. 2, and compare the LDPC code on GF(4) with the binary LDPC code both based on PEG Tanner graphs in Fig. 3. In the two figures all data, such as 1000, 4000, 20000, represent block lengths, and all code rates are 1/2.

Figure. 2 shows that when the blocklength is 1000bits, the LDPC code on GF(4) based on th PEG algorithm outperforms randomly constructed one by 0.6 dB at a BER of 10^{-6}, when 4000bits, the former outperforms the latter by 0.4dB at a BER of 10^{-6}, when 20000bits, the both performance is nearly the same. It is proved that the former has advantages over the latter at the short and medium block lengths.

Figure. 3 is the performance comparison between LDPC codes on GF(2) and GF(4) whose **H** are both based on the PEG algorithm. Through the figure we find that at rate R=1/2 the performance of the LDPC code on GF(4) is slightly better than that on GF(2) over AWGN channel, and the both computing complexity is analogous.

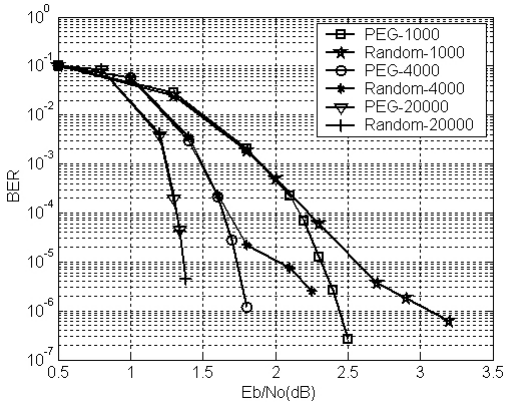

Fig. 2: Bit error rate(BER) of LDPC codes on GF(4) based on PEG Tanner graph and random graph, respectively.

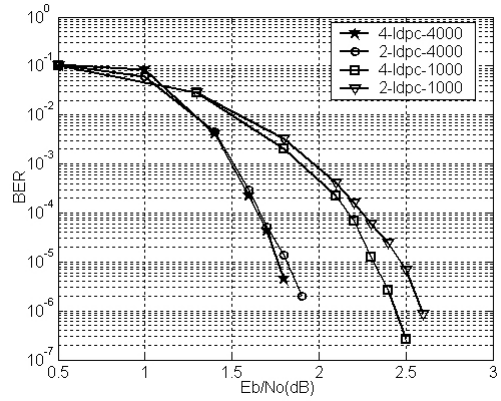

Fig. 3: BER of LDPC codes on GF (2) and on GF (4) both based on PEG Tanner graph.

V. Conclusions

Using the FT decoding algorithm with the equivalent transform is proposed, which achieves the decoding of LDPC codes on GF(q) efficiently, $q = 2^p$. And the PEG algorithm is introduced into the construction of check matrixes of q-ary LDPC codes. Simulation results show that the performance of LDPC codes on GF(4) based on PEG Tanner graphs is better than that on GF(4) based on random graphs, especially at the short block lengths. Whereas the optimization of **H** of LDPC codes is difficult and practical at the short block lengths in comparison with long block lengths, so this work is of important value in the design of **H** on GF(q). On the other hand, the performance comparison between LDPC codes on GF(4) and those on GF(2) both based on PEG Tanner graphs can be done more easily than that both based on random graphs where the comparison only can be done by expectation. Simulation results show that the

former outperforms the latter slightly at the rate R=1/2, and the both have analogous computing complexity.

In a word, it is believed that the non-binary LDPC code on GF(4) based on the PEG principle and decoded by the FT decoding algorithm with the equivalent transform is a preferable candidate in future digital communication systems, especially in transmission using short and medium block lengths. Next we will investigate the applications of q-ary LDPC codes to ADSL systems and satellite communication systems [9].

ACKNOWLEDGMENT

The authors would like to thank Mr. Weikai Xu for his useful help about the simulation sincerely. This work is sponsored by Chinese national nature science fund project(60272005) and Program for NCET in Universities of China(NCET-04-0601). The corresponder is wanglin@xmu.edu.cn.

REFERENCES

[1] X . -Y. Hu,E. Eleftheriou, and D.-M . Arnold, ”Progressive edge-growth Tanner graphs,” IEEE Proc. Globecom’2001,San Antorio, TX,Nov. 2001.

[2] R. G. Gallager. Low-Density Parity Chech Codes[J]. IRE Trans. Info. Theory1962IT-8(1):1-28

[3] D. J. C. Mackay, R. M. Neal,. Near, Shannon limit performance of low densiy parity check codes[J]. Electronics letters, 1996, 32(8)1645-1646

[4] D. J. C. MacKay, Good error-correcting codes based on very sparse matrices, IEEE Trans. Inform. Theory, vol. 45, pp. 399-431, Mar. 1999.

[5] S.-Y. Chung, G. D. Forney Jr., T. J. Richardson, and R. Urbanke, ”On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit,” IEEE Commun. Lett., vol. 5, no. 2, pp. 58C60, Feb. 2001.

[6] M. C. Davey, D. J. C. Mackay. Low-Density Parity Check Codes over GF(q)[J]. IEEE Commun. Lett., 1998,2(6): 165167.

[7] HongxinSong, J. R. Cruz. Reduced-complexity Decoding of q-ary LDPC Codes for Magnetic Recording[J]. IEEE Trans.Mang., 2003,39(3):1081 1087.

[8] XinMei Wang, GuoZhen Xiao. Error-correcting codes–principle and method Publishing House of Xi’Dian university. 2002.

[9] Anitkumar Mahadevan (2002) On LDPC codes for ADSL. Spring2002 UMBC(University of Maryland Baltimor County) Electrical Engineering Graduate Seminar Series[Z].

[10] T. Richardson and R. Urbanke. The capacity of low-density parity check codes under message-passing decoding. Submitted to IEEE Transactions on Information Theory, 1998.