

# On the Performance Evaluation of Quasi-Cyclic LDPC Codes with Arbitrary Puncturing

Ying Xu, and Yuejun Wei

Department of Wireless Research

Huawei Technologies Co., Ltd, Shanghai, 201206, China

Email: {eaglexu, weiyuejun}@huawei.com

Wen Chen

Department of Electronic Engineering

Shanghai Jiao Tong University, Shanghai, 200240, China

Email: wenchen@sjtu.edu.cn

**Abstract**—A novel algorithm, named Improved Protopraph-based Extrinsic Information Transfer (IP-EXIT), is proposed in this paper. With the proposed algorithm, quite accurate performance evaluations of Quasi-Cyclic LDPC (QC-LDPC) codes with arbitrary puncturing can be provided. LDPC codes combined with IR-HARQ technique have been a hot topic at present and puncturing is definitely one of the most efficient ways to achieve LDPC combined with HARQ. Unlike Turbo codes, different puncturing patterns for the same LDPC code may result in huge performance gap, so researchers are trying to design the puncturing pattern with better performance for LDPC codes. Inevitably, a large number of puncturing pattern with different performance should be compared in order to choose the best one from them, and usually this work is done by simulations with computer. In this paper, a novel algorithm is proposed to replace the work of simulation. It has been proved that the proposed algorithm can accurately evaluate the performance of arbitrary puncturing pattern, with very low complexity and very little time.

**Keywords:** LDPC; Protopraph; Arbitrary Puncturing; P-EXIT; IP-EXIT; Performance Evaluation

## I. INTRODUCTION

Recently, LDPC codes [1], especially QC-LDPC codes [2]-[4], have became more and more mature, with better performance and lower complexity. In the present days, LDPC codes combined with HARQ technique is one of the most important issue in the area of LDPC codes. HARQ is quite an efficient means to improve the throughput in a wireless system, and it has been widely used in wireless communication systems such as WCDMA, WiMAX and LTE. So, if LDPC code is going to be used in practical wireless communication systems, it must perfectly cooperate with HARQ technique, that's to say, rate-compatible LDPC codes with steady performance are essential.

Puncturing is quite an efficient way to provide rate-compatible LDPC codes. Unfortunately, as a kind of linear codes, the performance of LDPC codes may be greatly degraded because of puncturing, so many researchers have been working very hard trying to design better puncturing algorithm for LDPC codes [5]-[8], and wish to work out the optimal puncturing pattern for any LDPC codes. When some puncturing patterns are designed, how can we evaluate their performance and choose the really best one from them? Usually, simulation is the only way to achieve this goal, but

simulation always involves too many calculations and too much time. Is there a simpler way to do this work? Is there any algorithm to quickly evaluate the performance of LDPC codes with puncturing, just like what Density Evolution [9] or EXIT chart [10] can do for the LDPC codes with no puncturing?

Gianluigi has proposed an algorithm to evaluate the performance of QC-LDPC codes with puncturing [11], named as Protopraph-base EXIT (P-EXIT). P-EXIT algorithm is able to evaluate the puncturing performance accurately with low complexity. But there is one constrain of P-EXIT algorithm that it can only deal with the situation that the punctured bits are integer block matrix, which means when the parity matrix is punctured, the minimum punctured unit is the block matrix instead of the bit. This constrain badly reduce the area that P-EXIT algorithm can be available, because in many practical situations, we will handle these puncturing patterns which are not integer block matrix.

A more powerful algorithm, named IP-EXIT, is proposed in this paper, and it can totally solve the problems mentioned above without any accuracy or complexity sacrifice.

## II. PROTOGRAPH OF QC-LDPC CODES

### A. Protopraph-based LDPC codes

Protopraph-based LDPC [12] codes is a kind of special LDPC codes. A common protograph with 4 variable nodes and 3 check nodes is illustrated in Figure 1, and the related matrix is given in Figure 2.

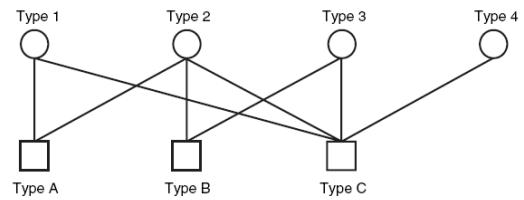


Figure 1. Protopraph with 4 variable nodes and 3 check nodes

1	1	0	0
0	1	1	0
1	1	1	1

Figure 2. Matrix of the protograph in Figure 1

Based on the protograph, a derived graph can be obtained with the “copy-and-permute” operation. For example, after 4 times copy, the resulted graph is shown in Figure 3, and finally the derived graph of the protograph in Figure 1 is shown in Figure 4.

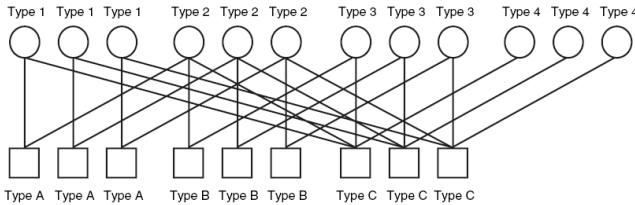


Figure 3. Protograph after 4 times copied

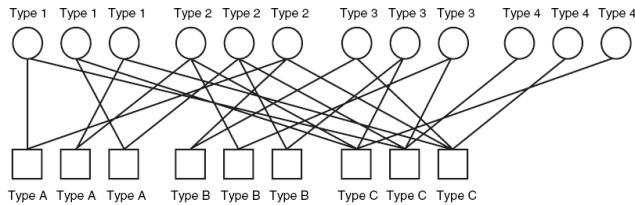


Figure 4. Derived graph after 4 times copy and permutation

1	0	0		0	0	1		0	0	0		0	0	0	Type A
0	0	1		1	0	0		0	0	0		0	0	0	Type A
0	1	0		0	1	0		0	0	0		0	0	0	Type A
0	0	0		0	0	1		1	0	0		0	0	0	Type B
0	0	0		1	0	0		0	0	1		0	0	0	Type B
0	0	0		0	1	0		0	1	0		0	0	0	Type B
0	1	0		1	0	0		0	1	0		0	0	1	Type C
1	0	0		0	1	0		0	0	1		1	0	0	Type C
0	0	1		0	0	1		1	0	0		0	1	0	Type C
Type 1	Type 1	Type 2		Type 2	Type 2	Type 3		Type 3	Type 3	Type 4		Type 4	Type 4	Type 4	

Figure 5. The parity check matrix of protograph LDPC codes

Taking the derived graph as the Tanner graph of LDPC codes, the resulting LDPC codes are called “Protograph-based LDPC codes”. The parity check matrix of the constructed protograph LDPC codes above is shown in Figure 5. More details about Protograph-based LDPC codes will not be introduced here and the interested readers are suggested to reference [12].

### B. Protograph of QC-LDPC codes

QC-LDPC codes are a kind of LDPC codes with block-circulant parity check matrix, as shown in Figure 6.

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{1,1} & \mathbf{H}_{1,2} & \cdots & \mathbf{H}_{1,n} \\ \mathbf{H}_{2,1} & \mathbf{H}_{2,2} & \cdots & \mathbf{H}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{m,1} & \mathbf{H}_{m,2} & \cdots & \mathbf{H}_{m,n} \end{bmatrix}$$

Figure 6. Parity check matrix for QC-LDPC codes

Usually,  $\mathbf{H}_{i,j}$  is the circular shift result of the identity matrix  $\mathbf{I}$  or null matrix  $\mathbf{0}$ .

*Lemma 1:* QC-LDPC codes can be regarded as a kind of special protograph-based LDPC codes, and their protographs and derived graphs can be deduced from their parity check matrices or base matrices.

Proof: For QC-LDPC codes with parameters  $m_b, n_b, z$ , we can get a binary matrix  $\mathbf{H}_1$  from its parity check matrix  $\mathbf{H}$  (or base matrix  $\mathbf{H}_b$ ) as in (1):

$$\mathbf{H}_1(i, j) = \begin{cases} 0 & \text{if } \mathbf{H}_{i,j} \text{ is null matrix} \\ 1 & \text{else} \end{cases} \quad (1)$$

If we treat matrix  $\mathbf{H}_1$  as the protograph of a LDPC code, and define the expansion factor  $z$  as the “copy times”, we can find that the circular times of each edge in the protograph corresponding to the “permutation pattern” of each edge in the derived graph when the derived graph is created. So, the parity matrix  $\mathbf{H}$  can be regarded as the derived graph and the matrix  $\mathbf{H}_1$  is just the protograph. To conclude, a QC-LDPC code is in fact a kind of protograph-based LDPC codes, with circular permutation.

### III. IMPROVED PROTOGRAPH-BASED EXIT

EXIT Chart algorithm [10] can evaluate the performance of LDPC codes by tracing the variation of mutual information during the decoding process. It has been proved that EXIT Chart is a precise and effective algorithm for performance evaluation, but it can not work when there are degree-1 columns in the parity matrix, or some columns are punctured. P-EXIT algorithm [11] is proposed to deal with the situations of degree-1 column or punctured column for protograph-based LDPC codes, but when the columns are not punctured in the unit of expansion parameter (i.e. integer block matrix), P-EXIT still can not work.

So a more powerful algorithm named “IP-EXIT” algorithm is proposed in this paper. By introducing a puncturing factor  $\lambda_j$  to modify the mutual information transfer model, the performance of protograph-based codes with arbitrary puncturing patterns can be evaluated.

In fact, the proposed IP-EXIT algorithm can deal with all kinds of protograph codes, including multi-edge type LDPC codes [13], repeat-accumulate (RA) [14] and RA-like codes [15]. But in this paper, we mostly focused on the performance evaluation of QC-LDPC codes (the most popular LDPC codes

at the present). According to *Lemma 1*, the QC-LDPC codes are indeed single-edge type LDPC codes, so the following introduction will be simplified to the single-edge case.

#### A. Mutual Information Model

First of all, we will give some MI (Mutual Information) related definitions. It should be noticed that all the nodes below denote the nodes in protograph.

$I_{ch}(j)$ : the MI between the message of the  $j$ th node in protograph from the channel and the associated bits.

$I_{Ev}(i,j)$ : the MI between the extrinsic message sent by variable node  $V_j$  to check node  $C_i$  and the associated bits.

$I_{Ec}(i,j)$ : the MI between the extrinsic message sent by check node  $C_i$  to variable node  $V_j$  and the associated bits.

$I_{APP}(j)$ : the MI between the a posteriori probability LLR of the  $j$ th node and the associated bits.

#### B. Improved Protograph-based EXIT

The IP-EXIT algorithm can calculate the  $I_{APP}(j)$  for each column of the matrix  $\mathbf{H}_1$  by iterative process, and if all the  $I_{APP}(j)$  reaches 1 after a predetermined iteration times, we believe the decoding is successful. With this consideration, a threshold value of  $E_b/N_0$  (the minimum value to satisfy  $I_{APP}(j)=1, \forall j=1,2\dots N$ ) can be obtained, which can be regarded as the performance threshold of QC-LDPC codes.

The detailed IP-EXIT algorithm is as follows:

**Step 1:**  $\forall j=1,2\dots N$ , calculate:

$$I_{ch}(j)=J(\sigma_{ch,j}) \quad (2)$$

Where  $\sigma_{ch,j}=8R\frac{E_b}{N_0}$ ,  $R$  is the coding rate of QC-LDPC

codes after puncturing, and  $\frac{E_b}{N_0}$  is a predetermined signal-to-noise ration of the channel.

**Step 2:** for each  $j$ ,  $j=1,2\dots N$ , there will be a corresponding circular matrix in the parity check matrix  $\mathbf{H}$ . If  $z_j$  bits are punctured in the  $j$ th circular matrix, the puncturing factor  $\lambda_j$  will be:

$$\lambda_j=\frac{z_j}{z} \quad (3)$$

$\lambda_j$  is actually the proportion of punctured bits to all the bits in each circular matrix.

**Step 3:** for  $j=0\dots N-1$  and  $i=0\dots M-1$ , if  $\mathbf{H}_1(i,j)=0$ ,  $I_{Ev}(i,j)=0$ , else if  $\mathbf{H}_1(i,j)\neq 0$ , calculate:

$$I_{Ev}(i,j)=\left(1-\lambda_j\right)\times J\left(\sqrt{\sum_{s\neq i}\mathbf{H}_1(s,j)[J^{-1}(I_{Av}(s,j))]^2+[J^{-1}(I_{ch}^{(j)})]^2}\right) \\ +\lambda_j\times J\left(\sqrt{\sum_{s\neq i}\mathbf{H}_1(s,j)[J^{-1}(I_{Av}(s,j))]^2}\right) \quad (4)$$

The initial value of  $I_{Ev}(i,j)$  and  $I_{Av}(i,j)$  are 0.

For  $j=0\dots N-1$  and  $i=0\dots M-1$ , calculate:

$$I_{Ac}(i,j)=I_{Ev}(i,j) \quad (5)$$

**Step 4:** for  $j=0\dots N-1$  and  $i=0\dots M-1$ , if  $\mathbf{H}_1(i,j)=0$ ,  $I_{Ec}(i,j)=0$ , else if  $\mathbf{H}_1(i,j)\neq 0$ , calculate:

$$I_{Ec}(i,j)=1-J\left(\sqrt{\sum_{s\neq j}\mathbf{H}_1(i,s)[J^{-1}(1-I_{Ac}(i,s))]^2}\right) \quad (6)$$

For  $j=0\dots N-1$  and  $i=0\dots M-1$ , calculate:

$$I_{Av}(i,j)=I_{Ec}(i,j) \quad (7)$$

After a predetermined iterative times between step 3 and step 4, go to step 5. The larger the iterative times is, the more accurate threshold can be obtained.

**Step 5:** for  $j=0\dots N-1$ , calculate:

$$I_{APP}(j)=\left(1-\lambda_j\right)\times J\left(\sqrt{\sum_s\mathbf{H}_1(s,j)[J^{-1}(I_{Av}(s,j))]^2+[J^{-1}(I_{ch}^{(j)})]^2}\right) \\ +\lambda_j\times J\left(\sqrt{\sum_s\mathbf{H}_1(s,j)[J^{-1}(I_{Av}(s,j))]^2}\right) \quad (8)$$

According to the definition of mutual information, when  $I_{APP}(j)=1 \forall j=1,2\dots N$ , the codeword bits can be totally known from the a posteriori probability LLR of the decoder, that's to say, the decoding is successful under the current  $E_b/N_0$ .

When IP-EXIT algorithm is used for performance evaluation, the puncturing pattern and the protograph of mother code is essential, then a max iteration times and  $E_b/N_0$  are predetermined. When the  $I_{APP}(j)$  equals to 1 for  $j=0\dots N-1$ , the  $E_b/N_0$  should be decreased and the whole process is repeated; on the contrary, if any  $I_{APP}(j)$  can not reach 1 when the iteration times reaches the maximum value, the  $E_b/N_0$  should be increased and the whole process is repeated, finally the minimum value of  $E_b/N_0$  which satisfy  $I_{APP}(j)=1 \forall j=1,2\dots N$ , is exactly the performance evaluation result of puncturing.

### C. Comparison between P-EXIT and IP-EXIT

The most important improvement of IP-EXIT algorithm is the introduction of puncturing factor  $\lambda_j$ , as shown in formula (3), which means the proportion of the punctured bits to all the bits in a circular matrix. When the block matrix is partially punctured, the extrinsic message sent by the corresponding variable node to check node is actually from two kinds of bits: punctured bits and non-punctured bits. For simplicity consideration, we suggest to calculate the average value of the two kinds of messages with  $\lambda_j$ , just as shown by the formula (4) and (8), and the performance of arbitrary puncturing patterns can be evaluated. The simulation results show that our “average” method is quite simple and precise.

Further studying the relation and difference of IP-EXIT and P-EXIT, we can find that P-EXIT is in fact a special case of IP-EXIT, corresponding to  $\lambda_j = 1$  or  $\lambda_j = 0$ .

## IV. SIMULATION AND ANALYSIS

### A. Simulation Results

In order to verify the accuracy of the proposed algorithm, we choose the Quasi-Cyclic LDPC codes in 802.16e [16] for simulation, with  $m_b = 12$ ,  $n_b = 24$ ,  $z = 96$  and 1/2 code rate. The base matrix of the Quasi-Cyclic LDPC codes is shown in Figure 7:

```

-1 94 73 -1 -1 -1 -1 -1 55 83 -1 -1 7 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 27 -1 -1 -1 22 79 9 -1 -1 -1 12 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 24 22 81 -1 33 -1 -1 -1 0 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1
61 -1 47 -1 -1 -1 -1 -1 65 25 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 39 -1 -1 84 -1 -1 41 72 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 -1 -1 -1 46 40 -1 82 -1 -1 -1 79 0 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1
-1 -1 95 53 -1 -1 -1 -1 14 18 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1 -1 -1 -1 -1 -1
-1 11 73 -1 -1 -1 2 -1 -1 47 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1 -1
12 -1 -1 -1 83 24 -1 43 -1 -1 -1 51 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1
-1 -1 -1 -1 94 -1 59 -1 -1 70 72 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 -1 -1
-1 -1 7 65 -1 -1 -1 39 49 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0
43 -1 -1 -1 -1 66 -1 41 -1 -1 -1 26 7 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0

```

Figure 7. Base matrix of the Quasi-Cyclic LDPC codes in 802.16e

Some puncturing patterns are designed for the LDPC code, and IP-EXIT algorithm is used to evaluate their performance threshold. At the same time, simulation is also done to get their actual performance. If the evaluated threshold perfectly coincides with the actual result, the accuracy of the proposed algorithm is confirmed.

Three groups of puncturing patterns are designed for simulation, and they refers to three typical instances: (a) different number of bits punctured in the same circular matrix; (b) same number of bits punctured in different circular matrix; (c) same number of bits punctured in/not in the unit of circular matrix.

Since we try to evaluate the accuracy of the proposed algorithm, different patterns with the same (or similar) number of punctured bits are chosen (so the performance gap will be small). These patterns and corresponding thresholds are shown in Table 1, Table 2 and Table 3. Since these patterns are all “partial puncturing”, not “block puncturing”, so their performance can not be evaluated by P-EXIT algorithm.

Table 1. Puncturing pattern and threshold for group 1

Punctured pattern	Threshold (dB)
3(30): 30 bits punctured in the 3rd column	0.8412
3(60): 60 bits punctured in the 3rd column	0.8576
3(96): 96 bits punctured in the 3rd column	0.8793

Table 2. Puncturing pattern and threshold for group 2

Punctured pattern	Threshold (dB)
10(48): 48 bits punctured in the 10th column	0.8509
13(48): 48 bits punctured in the 13th column	0.8786

Table 3. Puncturing pattern and threshold for group 2

Punctured pattern	Threshold (dB)
3(96): 96 bits punctured in the 3rd column	0.8793
24(96): 96 bits punctured in the 24th column	0.9441
3(48)24(48): 48 bits punctured in both the 3rd and the 24th columns	0.9079

The simulation results of the three group puncturing pattern are shown in Figure 8, Figure 9 and Figure 10. It is very encouraging to find that the simulation results perfectly coincide with the threshold: punctured pattern with better threshold always provide better performance, which means the proposed IP-EXIT algorithm can accurately evaluate the performance gap between different puncturing patterns.

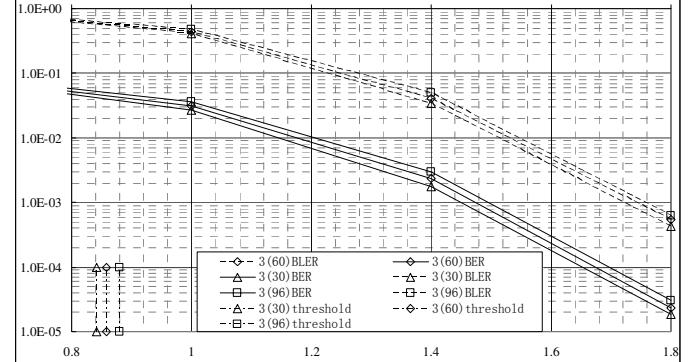


Figure 8. Simulation results and thresholds for group 1

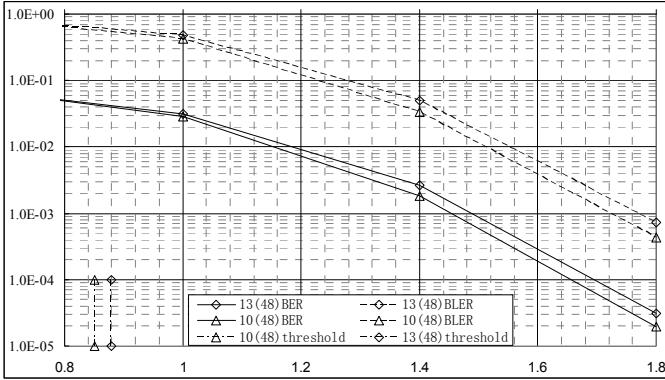


Figure 9. Simulation results and thresholds for group 2

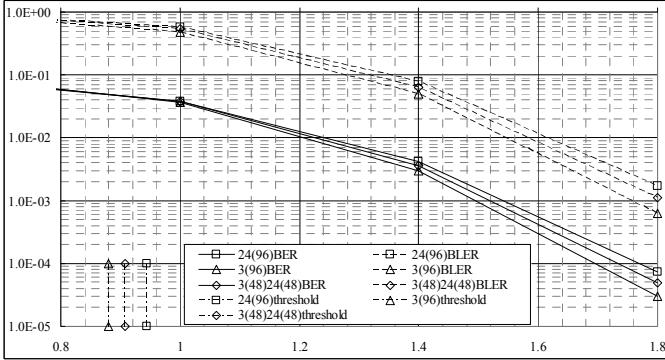


Figure 10. Simulation results and thresholds for group 3

More importantly, according to our statistic results, the time needed to calculate the threshold for one puncturing pattern is only a few seconds. So it will be much more efficient to compare hundreds of puncturing patterns with the proposed IP-EXIT algorithm, instead of simulations.

### B. Robustness Analysis

Since some ideal assumptions are made for the mutual information transfer model in EXIT algorithm, such as the code length is infinite, no cycle in the parity matrix etc. So there is always a gap between the evaluation result and the practical performance. There is the same instance for IP-EXIT algorithm. So it is not recommend taking the evaluation result of IP-EXIT algorithm as the real performance of the QC-LDPC codes. It is much more attractive to take IP-EXIT algorithm as the performance comparison tool between different puncturing patterns. For the same parity matrix, the performance gap resulting from the ideal assumptions are nearly the same for different puncturing patterns, the puncturing pattern with better evaluation result will always provide better performance. The performance evaluation of different puncturing patterns of different parity matrix is still an open question, and will not be included in this paper.

## V. CONCLUSION

Designing puncturing patterns for Quasi-Cyclic LDPC codes have been the one of the most important and critical

topics at present. So we usually have to evaluate the performance of a large number of puncturing patterns, in order to choose the best one out of them. In comparison with simulations, the proposed IP-EXIT algorithm is obviously a much better way to evaluate the performance gaps of different puncturing patterns. IP-EXIT algorithm takes the punctured proportion of the block matrix into account and modifies the mutual information transfer model, so as to predict the decoder behavior. As a result, IP-EXIT algorithm can accurately evaluate the performance of QC-LDPC codes with arbitrary puncturing, which is greatly helpful for the design and study of QC-LDPC codes puncturing.

## REFERENCES

- [1] G. R. Gallager, "Low density parity check codes," IRE. Trans. Information Theory, Vol. 8, pp. 21-28, 1962.
- [2] L. Lan, L. Zeng, Y. Tai, et al., "Construction of Quasi-Cyclic LDPC Codes for AWGN and Binary Erasure Channels: A Finite Field Approach," IEEE Trans. Information Theory, Vol.53, No. 7, pp. 2429-2458, 2007.
- [3] L. Lan, Y. Tai, S. Lin, et al., "New constructions of quasi-cyclic LDPC codes based on special classes of BIDB for the AWGN and binary erasure channels," IEEE Trans. Communications, Vol. 56, No. 1, pp. 39-48, 2008.
- [4] S. Kim, J. No, H. Chung, et al., "Quasi-Cyclic Low-Density Parity-Check Codes With Girth Larger Than 12," IEEE Trans. Information Theory, Vol. 53, No. 8, pp. 2885-2891, 2007.
- [5] J. Ha, J. Kim, and S. W. McLaughlin, "Optimal puncturing distribution for rate compatible low density parity check code," IEEE Trans. Inf. Theory, vol. 50, no. 11, pp. 2824-2836, Nov. 2004.
- [6] B. N. Vellambi and F. Fekri, "Rate-Compatible Puncturing of Finite-Length Low-Density Parity-Check Codes", in Proc. of 2006 IEEE International Symposium on Information Theory, pp. 1129-1133, Seattle, Washington, July 2006.
- [7] J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin, "Rate-compatible punctured low-density parity-check codes with short block lengths," IEEE Trans. Inf. Theory, vol. 52, no. 2, pp. 728-738, Feb. 2006.
- [8] Hyo Yol Park, Jae Won Kang, Kwang Soon Kim, Keum Chan Whang, "Efficient Puncturing Method for Rate-Compatible Low-Density Parity-Check Codes", IEEE Transactions on Wireless Communications, Vol. 6, no. 11, pp. 3914-3919, November 2007.
- [9] T. J. Richardson, and R. L. Urbanke, "The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding" IEEE Trans. Information Theory, Vol. 47, No. 2, pp. 599-618, 2001.
- [10] S. T. Brink, G. Kramer, and A. Ashikhmin, "Design of Low-Density Parity-Check Codes for Modulation and Detection," IEEE Trans. Communications, Vol. 52, No. 4, pp. 670-678, 2004.
- [11] G. Liva, M. Chiani, Protograph LDPC codes design based on EXIT analysis, in: Proc. of IEEE Global Telecommunications Conference (GLOBECOM), PISCATAWAY, NJ, IEEE, 2007, pp. 3250 - 3254
- [12] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," JPL INP, Tech. Rep., Aug. 2003, 42-154.
- [13] IEEE, IEEE P802.16Rev2/D7, Oct 2008
- [14] T. J. Richardson and R. L. Urbanke, "Multi-edge type LDPC codes," IEEE Trans. Inform. Theory, to be published. [Online]. Available: <http://lthcwww.epfl.ch/>
- [15] A. Abbasfar, K. Yao, and D. Disvalar, "Accumulate repeat accumulate codes," in Proc. IEEE Globecom, Dallas, Texas, Nov. 2004.
- [16] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in Proc. International Symposium on Turbo codes and Related Topics, Sept. 2000, pp. 1-8.