IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 11, 2017, accepted February 12, 2017, date of publication March 6, 2017, date of current version April 24, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2678515

An Efficient Construction Method for Quasi-Cyclic Low Density Parity Check Codes

YIMING LEI, (Member, IEEE), and MINGKE DONG, (Member, IEEE)
[1]School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China

Corresponding author: M. Dong (mingke.dong@pku.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61470005, in part by the China Postdoctoral Science Foundation under Grant 2015T80024, and in part by the GPINFOSYS Foundation under Grant 315040403.

ABSTRACT In this paper, we propose an optimized belief propagation (OBP) based progressive edge-growth (PEG) method for constructing quasi-cyclic low density parity check (QC-LDPC) codes. In this proposed method, Tanner graphs are built by progressively appending the check nodes rather than the variable nodes. Moreover, this OBP-PEG method considers three new constraint conditions to select the QC-LDPC code sets constructed by the PEG method. Compared with the PEG-based QC-LDPC decoders, the proposed OBP-PEG decoders decrease the number of the input ports of check-node processer by up to 25%, accelerates the convergence speed by up to 11.7% in layered decoding process, and improves the success probability by up to ten times in constructing fast-convergence QC-LDPC codes.

INDEX TERMS LDPC, PEG, code construction, convergence.

I. INTRODUCTION

Low density parity check (LDPC) codes are widely included in modern communication systems [1], e.g. 5G wireless communications and quantum communications [2], [3], and the convergence speed and hardware consumption are the fundamental challenges for the designers of LDPC decoders [4]. Besides decoding algorithms [5], the code construction methods can be also used to tackle these two challenges. Progressive edge-growth (PEG) is a typical random construction method of LDPC codes [6]. However, PEG method progressively appends the variable nodes, and the variations of the check-node degrees enlarge the hardware cost of check-node processer (CNP) (also known as CFU, i.e., check function unit) [7]. Moreover, PEG method leads to a high number of iterations in the decoding process because of the variable nodes having low belief refresh speed. This degrades both the convergence of LDPC decoders and the success rate in constructing fast-convergence codes.

In [8], the authors introduced two constraint conditions into PEG method, and this clearly improves the BER performance of the constructed LDPC codes. Inspired by above works, here we propose an optimized belief propagation (OBP) based PEG method for constructing quasi-cyclic (QC) LDPC codes, which applies to the efficient layered belief propagation (LBP) algorithms [9], [10]. Differently with PEG method, this OBP-PEG method builds the large-girth Tanner graphs by progressively appending the check nodes, but not the variable nodes. This ensures that the numbers of variable nodes connecting to every check node are identical, i.e., the number of input ports of each CNP can be well controlled. Simulation shows that, compared with PEG decoders, the number of input ports of each CNP in OBP-PEG decoders decreases by up to 25%. Also, we introduce three constraint conditions, i.e., relaxant constraint condition (RCC), belief prorogation condition (BPC), and weight constraint condition (WCC). RCC ensures the satisfactory variable-node degree ratio, BPC ensures the sufficient belief propagation, and WCC updates OBP-PEG method for the construction of QC-LDPC codes. Simulation shows that, compared with PEG method, the OBP-PEG method improves the convergence by 11.7% in decoding process, and improves the success probability by up to 10 times in constructing fast-convergence QC-LDPC codes.

II. THE PROPOSED OBP-PEG CONSTRUCTION METHOD

A. SYSTEM MODEL AND LBP DECODING ALGORITHM

A $(n,\ k)$ LDPC code is described by a $m \times n$ parity check matrix $\mathbf{H}$ or the Tanner graph [1], where n denotes the number of variable nodes; m denotes the number of check nodes; k denotes the original information bits ($k = n-m$). The check nodes in the Tanner graph correspond to the parity check functions, and the variable nodes correspond to the coded bits including information bits and parity bits. $\{c_i\}$ $(1 \leq i \leq m)$ denotes the check nodes, and $\{v_j\}$ $(1 \leq j \leq n)$ denotes the variable nodes. The posteriori probability (APP) message of v_j is denoted by Q_j. r_{ij} denotes the check-to-variable (CTV) extrinsic message from c_i to v_j. q_{ji} denotes the variable-to-check (VTC) extrinsic message from v_j to c_i. In this paper, all the messages are in the shape of the Log Likelihood Ratio (*LLR*) denoted by $L[\cdot]$.

In efficient LBP algorithms, all the extrinsic messages are initialized to zeros, i.e., $L[r_{ij}] = 0$ and $L[q_{ji}] = 0$. Afterward, the CNP are operated node-by-node until all the check functions are satisfied or the number of iterations reaches the pre-given limit. CNP is a serial of operations that update the values of $L[r_{ij}]$ and $L[Q_j]$ for the v_j in $N(i)$, where $N(i)$ is the set of variable nodes neighbouring c_i. The decoded result is decided to be zero and one when $L[Q_j] > 0$ and $L[Q_j] \leq 0$, respectively.

B. BELIEF PROPAGATION CONDITION (BPC)

Based on the regulations of LBP algorithms [9], [11], we can approximately have that,

$$E\left(L[q_{ji}]\right) = E[L(Q_j')] - E\left(L[r_{ij}']\right) \quad (1)$$

$$1 - \varphi\left[E(L[r_{ij}])\right] = \prod_{k \in N(i)\backslash j} \left[1 - \varphi\left[E(L[q_{ki}])\right]\right] \quad (2)$$

$$E(L[Q_j]) = E\left(L[q_{ji}]\right) + E\left(L[r_{ij}]\right) \quad (3)$$

wherein $E(.)$ is the expectation operator; Q_j' and r_{ij}' respectively denotes the previous values of Q_j and r_{ij} before the current iteration; and $N\ (i)\ \backslash j$ denotes $N(i)$ excluding variable node $v(j)$; and

$$\varphi(x) = \begin{cases} 1 - \dfrac{1}{\sqrt{4\pi x}} \displaystyle\int_{-\infty}^{\infty} \tanh\frac{u}{2} e^{\frac{-(u-x)^2}{4x}} du & (x > 0) \\ 0 & (x = 0) \end{cases} \quad (4)$$

Here $\varphi(x)$ is a monotonically decreasing function. Equations (1) to (4) presents the iteration process for calculating $E(L[Q_j])$.

Based on above equations and Gaussian approximation [12], the error probability function of decoded symbols, denoted as P_e, can be written as,

$$P_e = \frac{\int_{-\infty}^{0} e^{-\frac{(x-E(L[Q_j]))^2}{4E(L[Q_j])}} dx}{2\sqrt{\pi E(L[Q_j])}} = \frac{1}{2}\mathrm{erfc}\left(\frac{\sqrt{E(L[Q_j])}}{2}\right) \quad (5)$$

Equation (5) indicates that P_e is inversely proportional to the value of $E(L[Q_j])$, and the variable nodes having low values of $E(L[Q_j])$ accordingly have low belief propagation. Thus, in OBP-PEG method, we select the v_j having low values of $E(L[Q_j])$ to improve their belief propagation, and this is the belief prorogation condition (BPC) for accelerating the decoding convergence of constructed LDPC codes.

C. THE PROCEDURE OF OBP-PEG METHOD

As mentioned above, we build the Tanner graph on a layer-to-layer basis, i.e., we progressively append the check nodes, but not the variable nodes in PEG method. Here we define the variable-node degree ratio γ_j as $\gamma_j = \tilde{d}_v^j / d_v^j$, where $\tilde{d}_v^j$ and d_v^j denotes the current number and the planned number of the edges connecting to v_j, respectively. In proposed OBP-PEG method, we define the relaxant constraint condition (RCC) to select $v_{\hat{j}}$ $(1 \leq \hat{j} \leq n)$ satifying,

$$\hat{j} = \arg\left[\gamma_{\hat{j}} < \alpha \times \min_{j\ (1 \leq j \leq n)} (\gamma_j)\right] \quad (6)$$

where α is relaxant factor typically ranging from 1.3 to 1.7, and here it is set as 1.5.

The code construction procedure of the proposed OBP-PEG method can be shown as follows,

---

Add n variable nodes into Tanner Graph without any edges;
Initialize d_v^j by Density Evolution [13], or Extrinsic Information Transfer (EXIT) chart analysis [14];
Initialize $E\left(L[r_{ij}]\right)$ and $E(L[Q_j])$ as 0 and $2/\sigma^2$, respectively.
for $i = 1$ to m **do**
for $k = 0$ to d_c^i -1 **do** (d_c^i denotes the number of the edges connecting to c_i)
if $i = 1$
Connect c_1 to v_j, (as an example, here we let $j = k + 1$)
else
Expand a sub-graph from check node c_i up to expansion depth l under the current graph setting such that the cardinality of N_i^l (where N_i^l denotes the set of all variable nodes reached by the sub-graph spreading from the root node c_i within depth l) stops increasing, or $\bar{N}_i^l \neq \phi$ but $\bar{N}_i^{l+1} = \phi$ (where $\bar{N}_i^l$ is the complementary set of N_i^l).
Select the elements in $\bar{N}_i^l$ satisfied in RCC, to build a candidate set S_c^1.
Select the elements in S_c^1 satisfied in BPC, to build a candidate set S_c^2. Then, connect c_i to any v_j in S_c^2.
end
Update $E(L[Q_j])$ for $j \in N(i)$ based on BPC.
end
end

As shown above, the proposed OBP-PEG method not only select the variable nodes having the small-value degree ratio γ_j (as in eq.(6)), but also enhances the belief propagation of the variable nodes having minimum $E(L[Q_j])$.

III. THE UPDATE OF OBP-PEG METHOD FOR QC-LDPC

QC-LDPC codes are a popular member of LDPC codes due to their trade-off between complexity and throughput. In QC-LDPC codes, the parity check matrix $\mathbf{H}$ is equivalent to a prototype matrix $\mathbf{H}^P$. Each element of $\mathbf{H}^P$ (i.e., $[\mathbf{H}^P]_{i,j}$), the circulant value, corresponds to either a $p \times p$ circulant permutation matrix or a $p \times p$ all-zero matrix, where $[\mathbf{H}^P]_{i,j}$ corresponding to CPM ranges from 0 to $p - 1$.

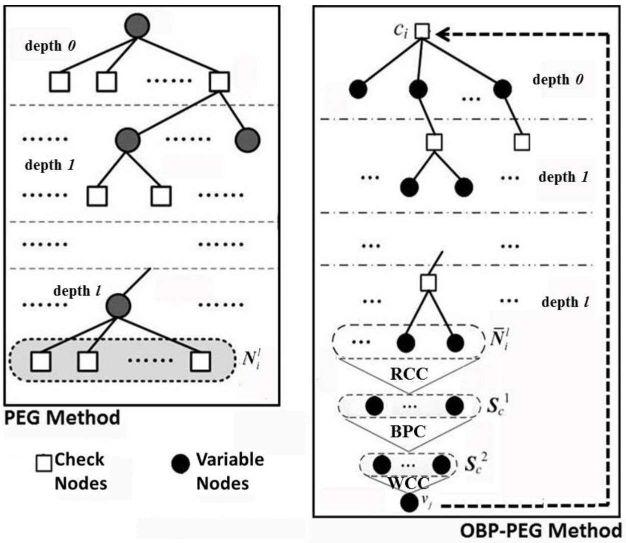


FIGURE 1. This figure compares the procedure of PEG method and the proposed OBP-PEG method, where the latter is illustrated in QC-LDPC applications.

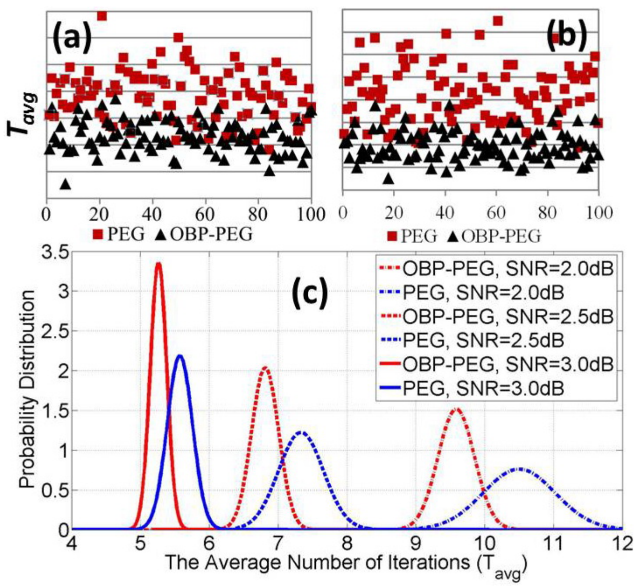


FIGURE 2. The simulated T_{avg} of 100 (64800, 32400) QC-LDPC code sets constructed by PEG and OBP-PEG methods; (a). SNR=2.0dB; (b). SNR=3.0dB; (c). the probability distribution of T_{avg}.

As shown in above construction procedure of OBP-PEG method, the expanded sub-graph has multiple paths from c_i to selected v_j, and the path weight $w_p(i,j)$ can be generally calculated as,

$$w_p(i,j) = \sum_{s=1}^{2l+1} (-1)^s [\mathbf{H}^P]_{i_s,j_s} \tag{7}$$

where the edges (c_{i_s}, v_{j_s}) are located in the path, and l denotes the expansion depth. For the cases of $i = 0$, the values of $[\mathbf{H}^P]_{0,j}$ can be randomly selected from 0 to $p-1$. For the cases of $i \neq 0$, the weight constraint condition (WCC) is included to determine the values of $[\mathbf{H}^P]_{i,j}$, i.e.,

$$\left[(w_p(i,j) + [\mathbf{H}^P]_{i,j}) \mod p\right] \neq 0 \tag{8}$$

WCC guarantees that there is no loop in the Tanner graph has a length of 4.

If $[\mathbf{H}^P]_{i,j}$ is unable to satisfy WCC, a new v_j in candidate set $S_c{}^2$ can be selected, and above procedure is repeated. Finally, OBP-PEG method will select the v_j to establish an edge with c_i. Figure 1 shows the procedure of PEG method and OBP-PEG method for QC-LDPC codes, including that how the latter expands the sub-graph and picks the optimal v_j. If WCC is not satisfied during the process, the proper circulant value (i.e., $[\mathbf{H}^P]_{i,j}$) can be obtained by re-selecting the variable node. The number of CNP input ports must equal to the maximum value of d_c^i, so that CNP is valid for all check node processing. The CNP complexity is determined by d_c^i,

TABLE 1. The decoding convergence of (960, 480) QC-LDPC codes, by PEG and OBP-PEG methods, SNR = 3.0dB.

Size of circulant permutation matrix, p	40	20	10	5
Convergence speed (T_{qef}), PEG	8.44	7.97	7.97	7.80
Convergence speed (T_{qef}), OBP-PEG	7.45	7.51	7.35	7.32
Reduction of T_{qef} by OBP-PEG	11.7%	7.5%	7.9%	6.1%
R_{fc}^{obp}, (Here R_{fc}^{peg} is set as 10%)	11%	13.5%	17.3%	17.5%
Number of CNP input ports, (PEG vs. OBP-PEG)	8 vs. 6	8 vs. 6	8 vs. 6	7 vs. 6

TABLE 2. The convergence of QC-LDPC codes when various code length applied, SNR = 3.0dB.

Code Length (n, m)	(8064,4032)	(4320,2160)	(2016,1008)	(1008,504)
Size of circulant permutation matrix, p	112	60	84	42
Convergence (T_{qef}), PEG	7.39	8.43	6.73	7.33
Convergence (T_{qef}), OBP-PEG	6.9	7.62	6.39	7.02
Reduction of (T_{qef}) by OBP-PEG	6.6%	9.6%	5.0%	4.2%
R_{fc}^{obp}, (Here R_{fc}^{peg} is set as 10%)	100%	97.8%	63.0%	33.1%
Number of CNP input ports, (PEG vs. OBP-PEG)	9 vs. 7	8 vs. 7	8 vs. 6	7 vs. 6

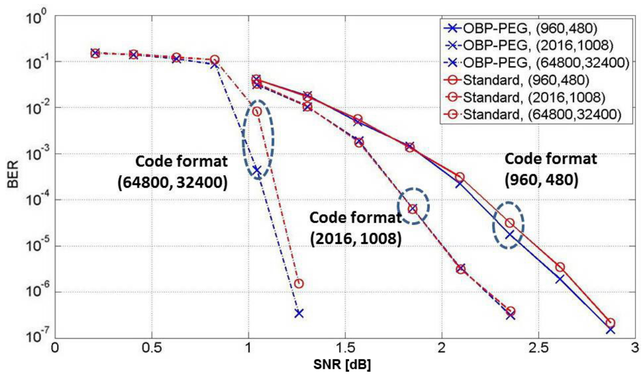


FIGURE 3. Comparison of BER performance of QC-LDPC codes constructed by proposed OBP-PEG method and the standard codes used in WiMax and DVB-S2 specifications, where the decoder uses LBP decoding algorithms. The selected code formats are (960, 480), (2016, 1008), and (64800, 32400), of which the sizes of circulant permutation matrix is 40, 84, and 360, respectively. The upper limit of the iteration numbers are 10, 12, and 20, for (960, 480), (2016, 1008), and (64800, 32400) codes, respectively.

$$R_{fc}^{obp}\left(T_{thr}\right)=\frac{1}{2}\left\{1+\text{erf}\left[\frac{\sigma\left(T_{avg}^{peg}\right)\times\text{erfinv}\left(2\cdot R_{fc}^{peg}\left(T_{thr}\right)-1\right)+E\left(T_{avg}^{peg}\right)-E\left(T_{avg}^{obp}\right)}{\sigma\left(T_{avg}^{obp}\right)}\right]\right\} \tag{12}$$

i.e., the number of the edges connecting c_i. In the proposed OBP-PEG method, d_c^i is set as a fixed value, this decreases its maximum value, and thus, the CNP complexity can be decreased as well.

IV. SIMULATION RESULTS AND DISCUSSIONS

This section compares the performance of PEG and OBP-PEG methods in QC-LDPC applications. Here we independently construct 100 QC-LDPC code sets, and we define T_{avg} to denote the average number of the iterations in the LBP decoding tests of each code set. As shown in Figure 2, the simulated T_{avg} of the QC-LDPC code sets, which are respectively constructed by PEG and OBP-PEG methods, follow Gaussian distribution. Here we define T_{qef} to denote the lower limit of T_{avg} ensuring the quasi-error-free decoding performance, and

$$T_{qef}=E(T_{avg})+2\sigma(T_{avg}) \tag{9}$$

where $E(T_{avg})$ and $\sigma(T_{avg})$ denote the mean and variance of the Gaussian distribution of T_{avg}, respectively. Considering the properties of Gaussian distribution [15], T_{qef} expressed in (9) guarantees the probability of error free decoding reaches more than 99%. In this paper, T_{qef} is used to denote the convergence speed in LBP decoding process.

Besides T_{qef}, we also investigate the success rate in constructing fast-convergence codes, i.e., the probability that T_{avg} is smaller than a threshold of the iteration number. Considering the properties of Gaussian distribution [15], this success rate $R_{fc}(.)$ of fast convergence codes can be easily obtained as,

$$\begin{aligned} R_{fc}(T_{thr}) &\triangleq \text{Prob}\left|_{T_{avg}<T_{thr}}\right. \\ &= \frac{1}{2}\left\{1+\text{erf}\left(\frac{T_{thr}-E\left(T_{avg}\right)}{\sigma\left(T_{avg}\right)}\right)\right\} \end{aligned} \tag{10}$$

wherein T_{thr} denotes the threshold of the iteration number. The above equation can be further transferred to,

$$T_{thr}=\sigma\left(T_{avg}\right)\times\text{erfinv}\left(2\cdot R_{fc}\left(T_{thre}\right)-1\right)+E\left(T_{avg}\right) \tag{11}$$

Based on (10) and (11), we can have that, (12), as shown at the top of this page, wherein $R_{fc}^{obp}(T_{thr})$ and $R_{fc}^{peg}(T_{thr})$ denote the success rate that T_{avg} of OBP-PEG method and PEG method is smaller than T_{thr}, respectively. T_{avg}^{obp} and T_{avg}^{peg} denote T_{avg} of OBP-PEG method and PEG method, respectively. This equation shows the quantitative link between $R_{fc}^{obp}(T_{thr})$ and $R_{fc}^{peg}(T_{thr})$.

Tables I compares the convergence (T_{qef}), the number of the input ports of CNP, and $R_{fc}^{obp}(T_{thr})$ of (960, 480) QC-LDPC codes constructed by PEG and OBP-PEG methods. Here, $R_{fc}^{peg}(T_{thr}) = 10\%$, SNR is 3.0dB, and code rate is 0.5. It can be seen that, compared with PEG method, OBP-PEG method reduces T_{qef} by up to 11.7%, and increases $R_{fc}(T_{thr})$ by up to 1.75 times. Also, it can be seen that, OBP-PEG successfully decrease the number of CNP input ports by up to 25%. It can be concluded that the advantages of OBP-PEG method in constructing QC-LDPC codes are significant when p is relatively small.

Tables II compares the convergence speed (T_{qef}), the number of the input ports of CNP, and $R_{fc}^{obp}(T_{thre})$ of QC-LDPC codes as various code length applied. The code rate is 0.5, and SNR is 3.0dB. It can be seen that, compared with conventional PEG method, OBP-PEG method reduces T_{qef} by up to 9.6% and increases $R_{fc}(T_{thr})$ by up to 10 times.

Figure 3 compares the bit error rate (BER) performance of the QC-LDPC codes constructed by proposed OBP-PEG method, where the standard QC-LDPC codes of Second Generation Digital Video Broadcasting (DVB-S2) and Worldwide Interoperability for Microwave Access (WiMAX) specifications are also used as benchmark. The LBP decoding algorithm is applied, and BER results are obtained when various SNR and code lengths applied. It can be seen that, the QC-LDPC codes of proposed OBP-PEG method show similar (even slightly better) BER performance that standard QC-LDPC codes of WiMax and DVB-S2 do. As expected, the increase of iteration number decreases required SNR, and the change of SNR ranges only around 0.2 to 0.5 dB (BER around 1E-5) when iteration number increases up to 100.

V. CONCLUSION

This paper proposed a novel OBP-PEG method for constructing QC-LDPC codes. This OBP-PEG method appends the check nodes and considers three new conditions. Compared to PEG method, this proposed method improves the decoding convergence by up to 11.7% and increases the success rate by 10 times in constructing fast-convergence QC-LDPC codes. Moreover, this OBP-PEG method decreases the number of input ports of CNP by up to 25%.

REFERENCES

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

[2] *3GPP specification draft*, document Draft Rep. 3GPP TSG RAN WG1 87, V0.1.0, 2016.

[3] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, “Fifteen years of quantum LDPC coding and improved decoding strategies,” *IEEE Access*, vol. 3, pp. 2492–2519, Nov. 2015.

[4] J. Andrade, G. Falcao, V. Silva, and L. Sousa, "A survey on programmable LDPC decoders," *IEEE Access*, vol. 4, pp. 6704–6718, Jul. 2016.

[5] J. Zhang, Y. Lei, M. Dong, and Y. Jin, "A serial layered scheduling algorithm for factor graph equalization," *IEEE Commun. Lett.*, vol. 18, no. 6, pp. 925–927, Jun. 2014.

[6] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

[7] M. M. Mansour and N. R. Shanbhag, "High-throughput LDPC decoders," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 6, pp. 976–996, Dec. 2003.

[8] C. T. Healy and R. C. D. Lamare, "Decoder-optimised progressive edge growth algorithms for the design of LDPC codes with low error floors," *IEEE Commun. Lett.*, vol. 16, no. 6, pp. 889–892, Jun. 2012.

[9] J. Zhang, Y. Lei, and Y. Jin, "Check-node lazy scheduling approach for layered belief propagation decoding algorithm," *Electron. Lett.*, vol. 50, no. 4, pp. 278–279, Feb. 2014.

[10] J. Zhang, M. Dong, Y. Lei, and Y. Jin, "Effective lazy schedule for the layered belief propagation algorithm," *IET Commun.*, vol. 7, no. 16, pp. 1871–1879, Nov. 2013.

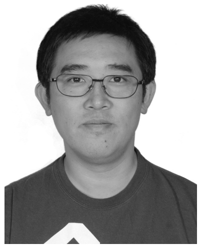
[11] D. E. Hocevar, "A reduced complexity decoder architecture via layered decoding of LDPC codes," in *Proc. IEEE Workshop Signal Process. Syst.*, Austin, TX, USA, Oct. 2004, pp. 107–112.

[12] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[13] X. Zheng, F. C. M. Lau, and C. K. Tse, "Performance evaluation of irregular low-density parity-check codes at high signal-to-noise ratio," *IET Commun.*, vol. 5, no. 11, pp. 1587–1596, Jul. 2011.

[14] E. Sharon, A. Ashikhmin, and S. Litsyn, "Analysis of low-density parity-check codes based on EXIT functions," *IEEE Trans. Commun.*, vol. 54, no. 8, pp. 1407–1414, Aug. 2006.

[15] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products*, 7th ed. San Francisco, CA, USA: Academic, 2007.



YIMING LEI received the B.Sc. degree in physics from Peking University, China, in 2001, and the Ph.D. degree in electronics from the University of Limerick, Ireland, in 2011. In 2005, he joined the Telecommunications Research Center, University of Limerick, Ireland, as a Research Engineer, where he was involved in the European Union Framework Project Six. He is currently a Lecturer with the State Key Laboratory of Advanced Optical Communication Systems and Networks, and a Principal Investigator with the Engineering Research Center, Digital Hospital System of State, Ministry of Education, Peking University, China. His research interests include nonlinearity analysis of wireless communications systems, digital predistortion techniques, coding of wireless channel, biophysical modeling, medical imaging, and brain science.

MINGKE DONG received the B.Sc. degree in communications engineering from the Beijing Institute of Technology, Beijing, China, in 1995, the M.S. degree in wireless communication from Peking University, Beijing, China, in 1998, and the Ph.D. degree in signal processing from Peking University. He is currently an Assistant Professor with the State Key Laboratory of Advanced Optical Communication Systems and Networks, School of Electronics Engineering and Computer Science, Peking University. He also takes an active part in the many key projects with the Institute of Advanced Communications, Peking University. He holds six patents on error control coding, especially LDPC codes design and decoding. He designed codes for several communication systems. He also implements codes such as LDPC, polar, turbo codes, and RS codes. His research interests include the construction and analysis of error control coding and other signal processing items.