Monte Carlo Simulations of Infinite Shortened Non-Binary LDPC Codes

Panyawat Techo, Virasit Imtawil, and Puripong Suthisopapan
Department of Electrical Engineering, Khon Kaen University, Thailand
Email: t.panyawat@kkumail.com; {virasit; purisu}@kku.ac.th

***Abstract*—**Shortening is a technique that can be used for constructing rate-compatible low-complexity LDPC codes. However, minimum SNR required to achieve error free decoding in the case of long block length, known as decoding threshold, for shortened LDPC code has not been reported. This motivates us to slightly modify the Monte Carlo simulation originally invented M. Davey to compute decoding threshold of the shortened LDPC codes. From the simulation results, we found that decoding threshold of shortened LDPC codes based on uniform shortening algorithm is identical to conventional LDPC codes designed for specific coding rate. Moreover, the number of iteration used to achieve successful decoding of shortened code and conventional code are slightly different.

***Index Term*—**Monte carlo simulation, uniform shortening algorithm, non-uniform shortening algorithm, decoding threshold

I. Introduction

Low density parity check (LDPC) code is a class of powerful channels coding, invented by Robert Gallager in 1962s. Nowadays, this code has already been applied to many modern digital communication systems since it can bring the system with very high performance [1], [2], i.e., low BER and low transmission power. Typically, performance of this code depends on many parameters such as code length, field order and code rate, ranging between 0 and 1. It is known that LDPC code with low code rate can achieve better performance comparing with that of high code rate [3], [4]. It is worth noting that LDPC code is normally designed to operate at specific code rate.

In fact, the channel condition for many realistic communication systems can be changed from time to time, known as time-varying channel. In order to deal with various channel conditions, code rate should be adaptable, i.e., low-rate code for bad channel condition or high-rate code for normal channel condition.

Unfortunately, the code rate adaptation have to utilize more than one pairs of encoder and decoder. This requires more space to install additional hardware and leads to higher cost. Rate-Compatible (RC) codes are the technique invented to overcome this problem [5]–[8]. This code can change its code rate according to channel condition by using a single pair of encoder and decoder.

Manuscript received October 10, 2018; revised June 2, 2019.
doi:10.12720/jcm.14.7.601-606

Shortening is one of the techniques used to construct rate- compatible codes. This technique can adapt code rate of mother code to lower code rates by inserting known symbols [9]–[12]. Typically, performance of shortened RC code strongly depends on selected shortening algorithm, i.e., how to insert known symbols and there are many shortening technique proposed in a literature [9], [11], [13], [14]. To the best of my knowledge, uniform shortening algorithm is by far the best technique which can provide identical BER performance to the code designed for specific rate. For this algorithm, each check nodes of LDPC code uniformly connects to shortened variable nodes of LDPC code [13]. However, performances of uniform shortening algorithm are investigated only for the case of short and medium code length [13], [15].

This work focuses on analyzing the performance of uniform shortening algorithm in the case of very long code length (more over 100,000 symbols) [4], [16]. The Monte Carlo simulation proposed by Matthew Davey is slightly modified to calculate decoding threshold, i.e., minimum SNR to achieve error free transmission, of non-binary LDPC code shortened by uniform shortening algorithm.

The rest of the paper is organized as follows. The basic background are described in Section II. The simulation results are reported and illustrated Section III. Finally, some discussion and conclusions are given in Section IV.

II. Basic and Background

The definition of LDCP codes and the structure of RC codes will be briefly described. Monte Carlo simulation which are the main tool for this work is also presented in this section.

A. Code Definition

Let N and K be code length and message length, in terms of symbol from finite field, i.e., $\mathrm{GF}(q)$, respectively. LDPC code is a type of error correcting codes. This code has been defined in terms of sparse parity check matrix $\mathbf{H}$ over $\mathrm{GF}(q)$ whose its dimension is $(N-K)$? N [1], [2]. Note that the LDPC code with $q>2$ is known as non-binary LDPC codes. The rate of code is given by the ratio between message length and code length $R=K/N$.

LDPC codes can be graphically represented by using Tanner graph. The structure of this graph can be drawn from parity check matrix. This Tanner graph consists of two main sets of nodes which are N variable nodes and $P = N - K$ check nodes. The edge in Tanner graph that connect check and variable nodes relates to the position of non-zero entry in **H.** Figure 1 shows the example of small size parity check matrix over $\mathrm{GF}(4)$ and its corresponding Tanner graph.

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 0 & \alpha^2 & 0 & 1 & 0 \\ \alpha & 0 & \alpha & 0 & 0 & 1 & 0 & \alpha^2 \\ 0 & 1 & 1 & \alpha & 0 & 0 & \alpha^2 & 0 \\ 0 & 0 & 0 & \alpha^2 & 1 & 1 & 0 & \alpha \end{bmatrix}$$

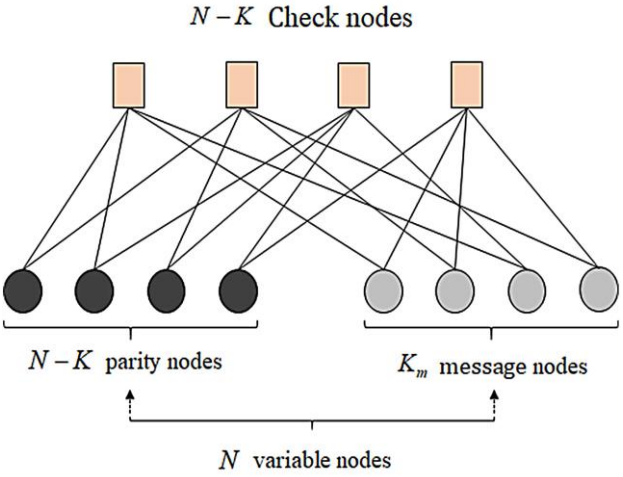

Fig. 1. Parity check matrix and corresponding Tanner graph of LDPC codes over $\mathrm{GF}(4)$ with $R = 1/2$.

Let W_r and W_c be the number of non-zero elements in each row and each column, respectively, of parity check matrix. From Fig. 1 each row has $W_r = 4$ and each column has $W_c = 2$. This means that each variable node connects to exactly W_c check nodes and each check nodes connects to exactly W_r variable nodes. The LDPC code that has constant W_r and W_c for each row and each column is called regular LDPC code.

The common decoding algorithm for non-binary LDPC codes is FFT based belief propagation algorithm [17]. During decoding process, the information (in terms of probability) between variable and check nodes of Tanner graph are exchanged. This will be done until the codeword is found or maximum iteration is reached.

B. Shortening Technique

Shortening is one of the techniques for generating rate compatible codes. The method to perform shortening is illustrated in Fig. 2

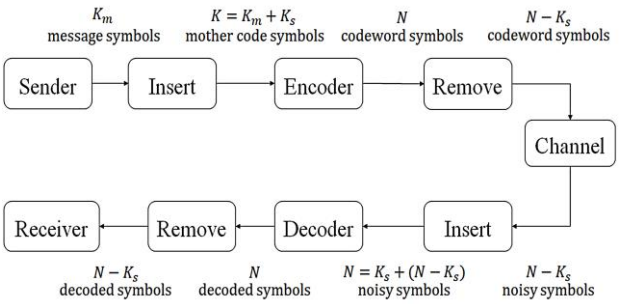

Fig. 2. Block diagram of coded systems with shortening technique.

The concept of shortening can be described as follows. At transmitter side, the original K_m message symbols are inserted with K_s known symbols to form $K = K_m + K_s$ mother code symbols. After encoding, the codeword of length $N = K + P$ symbols is produced. Before transmitting through the channel, the K_s known symbols are removed from the codeword symbols. So, the length of transmitted symbols $N - K_s$ symbols and the code rate of shortened code is given by

$$R_s = \frac{K - K_s}{N - K_s} \tag{1}$$

At receiver side, K_s known symbols are inserted back to the $N - K_s$ noisy received symbols. So, the length of input of decoder is $K_s + (N - K_s) = N$ symbols. Note that these known symbols are inserted back into the old position. For LDPC decoding, known symbols contribute perfect a priori information that helps the decoding process. After decoding, K_s symbols are removed from N decoded symbols. Finally, the estimation has length $K_m = N - K_s$ symbols.

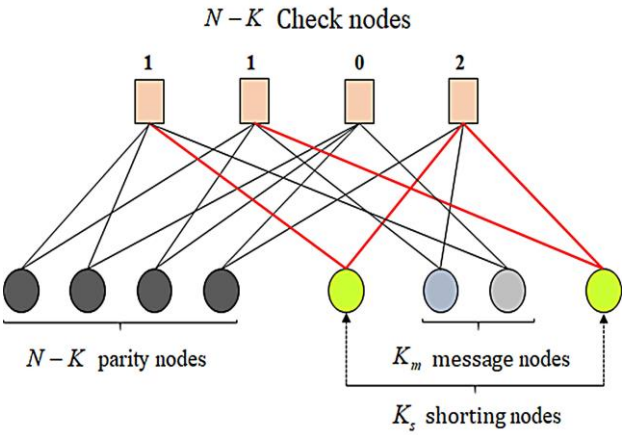

Fig. 3. The structure of Tanner graph for shortened LDPC codes of $R = 1/3$ constructed from $R = 1/2$.

To show an example, the structure of small size Tanner graph for $R = 1/3$ shortened LDPC codes constructed from $R = 1/2$ is depicted in Fig. 3. For this example, $(N = 8, K = 4)$ code of $R = 1/2$ is utilized as mother code. In order to construct $R = 1/3$ shortening code, message nodes of size $K_s = 2$ must be selected as shortened nodes. The first and the fourth are selected in this example and the rate of shortening code is equal to $R_s = \frac{4-2}{8-2} = \frac{1}{3}$. For this case, it is seen from the figure that each check node does not connect to the same amount of shortening node.

As mentioned earlier, the position of shortened nodes affects the performance shortened code [13], [14]. One way to obtain excellent performance is to select the position of shortened nodes in according to the concept of uniform shortening [13], [15]. By using the same code as shown in Fig. 3, the appropriate shortened position based on this concept is shown in Fig. 4. It is clearly seen that each check node connect to the same amount of shortened nodes (1 for this example), i.e., uniform shortening.

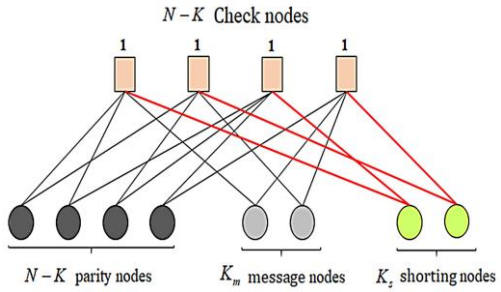

Fig. 4. Example of tanner graph with uniform shortening

C. Monte Carlo Simulation

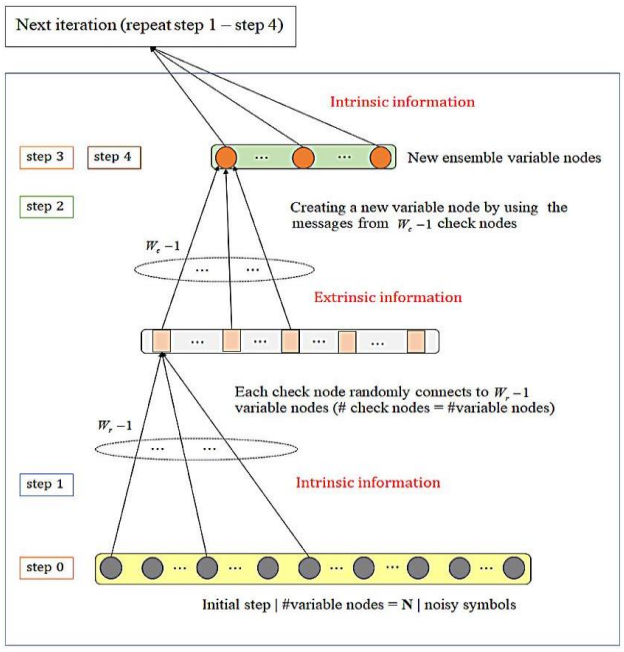

Fig. 5. The step to use Monte Carlo simulation for calculated decoding threshold of regular LDPC codes.

Monte Carlo simulation associates with belief propagation decoding over girth free Tanner graph (tree structure) with very large nodes. Therefore, this technique is used to calculate asymptotic decoding performance in the case of very long code length under ideal decoding assumption. The steps to use Monte Carlo simulation for obtaining decoding threshold of regular LDPC codes are depicted in Fig. 5.

Note that i-th codeword can be represented by i-th variable node and the length of codeword is equal to total number of variable nodes. Thus, we will use the terms codeword and variable nodes interchangeably. Let h be the non-zero element of $\mathrm{GF}(q)$, $i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, W_r - 1$. The Monte Carlo simulation can be described as follows:

Step 1: Assuming that the zero codeword is transmitted. Given a specific level of SNR, a large number of variable node is firstly generated. Each variable node associated with intrinsic information. We use $\mathbf{Q}_i = \left[Q_{i1}, Q_{i2}, \cdots, Q_{iQ}\right]'$ to denote the intrinsic information (in terms of vector of probabilities) to be sent from i-th variable node. Note that $Q_{ix} = \Pr(c_i = x)$ where c_i is i-th codeword symbol and $x \in \mathrm{GF}(q)$. Typically, the size of these symbols must be at least $N = 100{,}000$ symbols [4].

Step 2: $W_r - 1$ variable nodes are randomly selected to be connected to i-th check nodes. It is worth mentioning that random $h \in \mathrm{GF}(q)$ is assigned to each connection.

Step 3: According to FFT-based belief propagation algorithm [17], [18] each check node calculates an extrinsic information from incoming messages of $W_r - 1$ variable nodes defined in previous step as follows

$$\mathbf{R}_i = \mathcal{F}^{-1}\left(\prod_{j=1}^{W_r-1}\mathcal{F}\left(\mathbf{Q}_j\right)\right) \tag{2}$$

where $\mathbf{R}_i = \left[R_{i1}, R_{i2}, \cdots, R_{iq}\right]'$ is the extrinsic information from i-th check node. At the end of this step, N extrinsic messages associated with N check nodes are obtained.

Step 4: We will create a new ensemble of N variable nodes as follows. Each new variable node is randomly connected to $W_c - 1$ check nodes (each has its own extrinsic information). The intrinsic message of each new variable node is given by

$$\mathbf{Q}_i = \beta_i \gamma_i \prod_{l=1}^{W_c-1}\mathbf{R}_l \tag{3}$$

where β_i is normalizing factor and $\gamma_i = \Pr(c_i = 0)$ generated from the level of SNR defined in Step 0.

Step 5: The average entropy is used to measure the ambiguity among N variable nodes [4]. This average entropy can be computed by using this equation

$$\bar{H} = \frac{1}{N}\left(\sum_{i=1}^{N}\left(-\sum_{x=1}^{q} Q_{ix}\log_q Q_{ix}\right)\right) \quad (4)$$

Since the zero codeword is assumed, the average entropy approaches zero if the decoding process is done successfully, i.e., complete removing the effect of noise or no ambiguity.

Following [4], we will repeat step 1 – step 4 until average Shannon entropy of a new ensemble of variable node is less than 10^{-5} If average Shannon entropy is less than 10^{-5}, this means that the successful decoding can be accomplished at this SNR. We will utilize this Monte Carlo simulation to obtain the lowest level of SNR that the successful decoding can be achieved. This SNR level is called in this paper as decoding threshold.

Next, we present a method to use Monte Carlo simulation for computing decoding threshold of shortened LDPC codes based on uniform shortening. According to the concept of uniform shortening, each check node must be connected to the same amount of shortened variable nodes. For regular LDPC codes, this amount of shortened nodes is given by

$$K_s = \frac{(W_r - W_c) - R_s W_r}{1 - R_s} \quad (5)$$

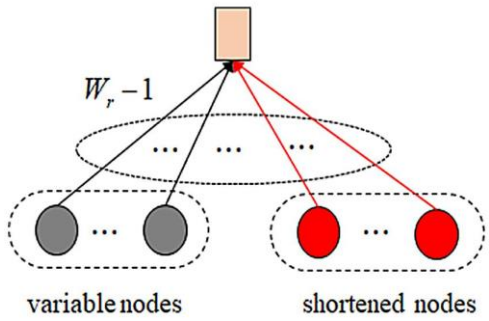

Fig. 6. The structure of i-th check node of Monte Carlo simulation for the case of shortened LDPC codes.

Fig. 6 shows the structure of i-th check node at step 1 of Monte Carlo simulation for the case of shortened LDPC codes.

For example, we will demonstrate the structure of shortened LDPC codes of $R = 1/2$ constructed from mother code of $R = 3/4$ with $(2, 8)$-regular structure. With this structure, each check node connects to exactly 8 variable nodes. By using (5) the amount of shortened nodes is equal to

$$K_s = \frac{(8-2) - \frac{1}{2}\cdot 8}{1 - \frac{1}{2}} = \frac{2}{0.5} = 4$$

The structure of i-th check node for shortened case is shown in Fig. 7

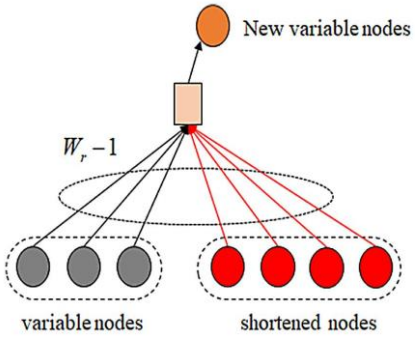

Fig. 7. The structure of i-th check node of shortened LDPC codes with $(2, 8)$-regular structure.

III. Simulation Result

Decoding thresholds of shortened LDPC codes obtained from Monte Carlo simulation are presented. Following M. Davey [4], 100,000 initial noisy symbols are used to represent infinite code lengths. FFT-based belief propagation algorithm with maximum 100 iterations is used as decoding algorithm for all Monte Carlo simulations.

For short and medium code lengths, it has been shown in [13], [15] that the BER performance of shortened LDPC code based on uniform shortening algorithm (referred to as shortened code) is identical to that of independently designed code (shortly called conventional code), i.e., code designed for specific rate.

As mentioned earlier, in this study, we would like to further investigate the performance of shortened in the case of long code length.

TABLE I: Comparison between Decoding Threshold of Shortened Codeand Conventional Code.

ield order	Code rate		Decoding threshold	
	Shortened	*Conventional*	*Shortened*	*Conventional*
GF(4)	1/2	1/2	1.10	1.10
	2/3	2/3	1.70	1.70
GF(8)	1/2	1/2	1.16	1.16
	2/3	2/3	1.73	1.73
GF(16)	1/2	1/2	1.27	1.27
	2/3	2/3	1.80	1.80
GF(64)	1/2	1/2	0.61	0.61
	2/3	2/3	1.41	1.41

Table I shows the performance comparison (in terms of decoding threshold) between shortened code and conventional code in the case of infinite code length The mother code of $R = 3/4$ is used in computing decoding threshold. The shortened code have $R = 2/3$ and $R = 1/2$ the structure of $(3, 12)$-regular LDPC code is utilized for the case of GF(4) - GF(16) whereas the results for GF(64) is done over the structure of $(2, 8)$-regular LDPC code. It is clearly seen from the table that decoding threshold of shortened code is the same as that

of conventional code for all cases. This means that the uniform shortening algorithm can provide excellent decoding performance for shortened LDPC code. This analysis can be used to confirm the correctness of the results previously reported in the case of short and medium code length.

As shown in Table I, it is implied that, at the same code rate BER performance of rate-compatible LDPC code shortened by uniform shortening algorithm is the same as that of independently designed LDPC code. Note that other shortening algorithms such as [14], [19] cannot provide identical performance to independently designed LDPC code. Therefore, uniform shortening algorithm is the best known algorithm so far.

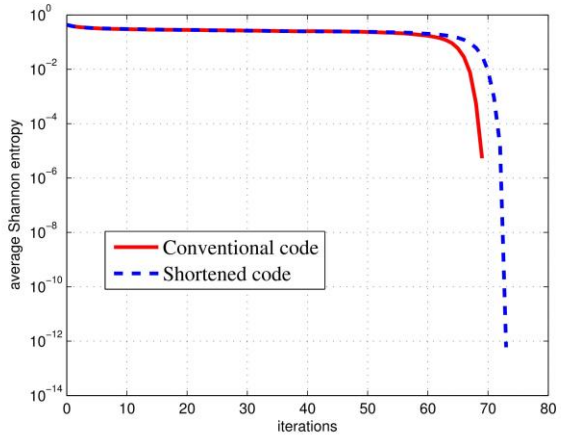

Fig. 8. The number of average iteration of shortened and conventional code over $\mathrm{GF}(4)$ with $R = 1/2$ used to achieve successful decoding.

Although decoding thresholds of both shortened and conventional code but we demonstrate in Fig. 8-11 that the convergent rates of Monte Carlo simulation of both codes are not the same. Noting that the convergent rate of Monte Carlo is the number of iteration used to achieve successful decoding, i.e., $H \le 10^{-5}$. Figure 8-11 shows that, the convergent rates of shortened code and conventional code are slightly different. For example, at $R = 2/3$ and $\mathrm{GF}(64)$ as shown in Fig. 10 shortened code uses about 80 iteration to achieve $H = 10^{-5}$ while conventional code uses about 75 iteration.

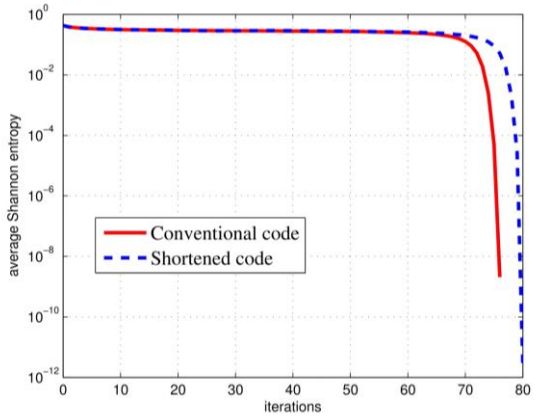

Fig. 9. The number of average iteration of shortened and conventional code over $\mathrm{GF}(8)$ with $R = 1/2$ used to achieve successful decoding.

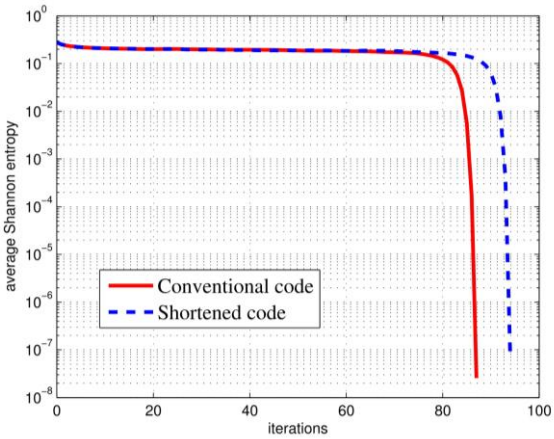

Fig. 10. The number of average iteration of shortened and conventional code over $\mathrm{GF}(16)$ with $R = 2/3$ used to achieve successful decoding.

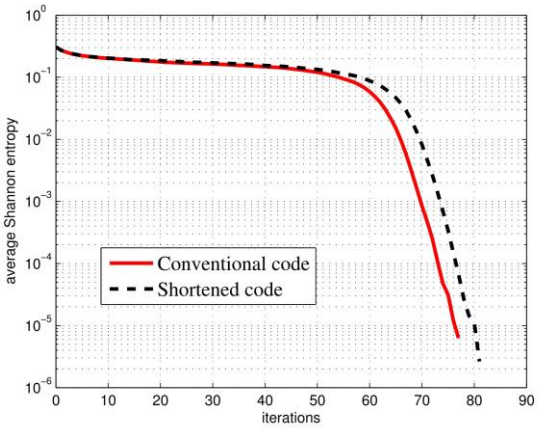

Fig. 11. The number of average iteration of shortened and conventional code over $\mathrm{GF}(64)$ with $R = 2/3$ used to achieve successful decoding.

IV. Concludsion

The ultimate goal of this work is to analyze the performance of uniform shortening algorithm in the case very long code length. By using modify Monte Carlo simulation, it is found that uniform shortening algorithm is the best known shortening algorithm since the shortened LDPC code based on this algorithm can give the same decoding threshold comparing with independently designed LDPC code. Therefore, we can utilize uniform shortening algorithm to construct rate-compatible LDPC codes which have excellent decoding performance at any coding rate.

Acknowledgment

The authors wish to thank the signal processing and coding laboratory, faculty of engineering, Khon Kaen University, which helped to support on the paper.

References

[1] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[2] M. C. Davey and D. MacKay, "Low-density parity check codes over gf (q)," *IEEE Communications Letters*, vol. 2, no. 6, pp. 165–167, 1998.
[3] D. J. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.
[4] M. C. Davey, "Error-correction using low-density parity-check codes," Ph.D. dissertation, University of Cambridge, 2000.
[5] J. Hagenauer, "Rate-compatible punctured convolutional codes (rcpc codes) and their applications," *IEEE transactions on communications*, vol. 36, no. 4, pp. 389–400, 1988.
[6] J. Ha, J. Kim, and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Transactions on information Theory*, vol. 50, no. 11, pp. 2824–2836, 2004.
[7] J. Li and K. R. Narayanan, "Rate-compatible low density parity check codes for capacity-approaching arq schemes in packet data communications." in *Proc. Communications, Internet, and Information Technology*, 2002, pp. 201–206.
[8] T. Okamura, "A hybrid arq scheme based on shortened low-density parity-check codes," in *Proc. Wireless Communications and Networking Conference*, 2008, pp. 82–87.
[9] T. Tian and C. R. Jones, "Construction of rate-compatible ldpc codes utilizing information shortening and parity puncturing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, no. 5, p. 692121, 2005.
[10] M. Yazdani and A. H. Banihashemi, "On construction of rate-compatible low-density parity-check codes," in *Proc. IEEE International Conference on Communications*, 2004, pp. 430–434.
[11] M. Beermann, T. Breddermann, and P. Vary, "Rate-compatible ldpc codes using optimized dummy bit insertion," in *Proc. ISWCS*, 2011, pp. 447–451.
[12] Y. Wei, Y. Yang, M. Jiang, W. Chen, and L. Wei, "Joint shortening and puncturing optimization for structured ldpc codes," *IEEE Communications Letters*, vol. 16, no. 12, pp. 2060–2063, 2012.
[13] P. Suthisopapan, M. Kupimai, V. Imtawil, and A. Meesomboon, "A novel structure of variable rate non-binary ldpc codes for MIMO channels," in *Proc. 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2014, pp. 1–6.
[14] X. Liu, X. Wu, and C. Zhao, "Shortening for irregular qc-ldpc codes," *IEEE Communications Letters*, vol. 13, no. 8, 2009.
[15] A. Wongsriwor, V. Imtawil, and P. Suttisopapan, "Design of rate- compatible ldpc codes based on uniform shortening distribution," *Engineering and Applied Science Research*, vol. 45, no. 2, pp. 140–146, 2017.
[16] M. C. Davey and D. J. MacKay, "Monte carlo simulations of infinite low density parity check codes over gf (q)," in *Proc. Int. Workshop on Optimal Codes and related Topics*. Citeseer, 1998, pp. 9–15.
[17] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary ldpc codes over gf (*q*)," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 633–643, 2007.
[18] G. Sarkis, S. Mannor, and W. J. Gross, "Stochastic decoding of ldpc codes over gf (q)," in *Proc. IEEE International Conference on Communications*, 2009, pp. 1–5.
[19] L. Zhou, B. Bai, and M. Xu, "Design of nonbinary rate-compatible ldpc codes utilizing bit-wise shortening method," *IEEE Communications Letters*, vol. 14, no. 10, pp. 963–965, 2010.

Panyawat Techo received the B.Eng degree in Computer Science from Khon Kaen University, Thailand in 2016. He is now a M.Eng student at the Electri- cal Engineering Department, Khon Kaen University, Thailand. His research interest includes error coreecting codes for digital communications.

Virasit Imtawil received the B.Eng in Electrical Engineering from Khon Kaen University, Thai- land in 1991 and Ph.D. degree in Electrical Engineering from the University of Manchester, United Kingdom in 1999. He is now an Associate Professor at the Electrical Engineering Department, Khon Kaen University, Thailand. His research interests cover error control coding for digital data transmis-sion and storage, wireless and optical communica-tions.

Puripong Suthisopapan received the B.Eng degree in Electrical Engineering from Khon Kaen University, Thailand in 2007. He received the M.Eng degree in Electrical Engineering from Khon Kaen University, Thailand in 2009 and Ph.D. degree in Electrical Engineering from Khon Kaen University, Thailand in 2013. He is now Instructor at the Electrical Engineering Department, Khon Kaen University, Thailand. His research interest includes error correcting codes for digital communications.