Novel LDPC code structures for the nonergodic block-fading channels

Jun Li, Jinhong Yuan, Marwan H. Azmi, and Robert Malaney

School of Electrical Engineering and Telecommunications,
University of New South Wales, Sydney, NSW 2052, AUSTRALIA
Email: jun.li@unsw.edu.au, j.yuan@unsw.edu.au, marwan@student.unsw.edu.au, r.malaney@unsw.edu.au

Abstract—**In this paper we study LDPC code design in block-fading (BF) channels. Our main purpose is to introduce a new Root Check (RC)-LDPC code design which obtains full diversity whilst delivering improved coding gain relative the other LDPC code designs. More specifically, we show that when our codes are combined with the check splitting method for the ARQ BF channels, we obtain the highest coding gain relative to conventional RC-LDPC codes.**

I. Introduction

In wireless communications, mobility and multipath propagation are the two prime factors which lead to time-varying channel conditions. It is known that the nonergodic block-fading (BF) channel model is a very useful model in this regard, in that it characterizes the main time-dependent features of the wireless channel [1]. In studying channel coding in such BF channels, the ability to obtain maximum (full) diversity is usually a constraint imposed on the adopted coding scheme. Parameterizing diversity in terms of the Singleton bound [2] is often convenient. In [3, 4] the Singleton bound was first utilized as an upper bound on the diversity using pairwise error probability arguments. More recently, in [5], it shown that the Singleton bound is indeed optimal in that it is the slope of the outage probability (OP) vs. SNR curve (the OP curve).

Achieving full diversity gain in the BF channel using a maximum likelihood (ML) decoder at the destination is shown in the works of [5, 6]. However, ML decoders are complex and difficult to implement for large block lengths. On the other hand, iterative belief propagation (BP) decoders are considered to possess less complexity and be implementable for large block length codes. But, the design methodologies for the ML decoder in [5, 6] cannot guarantee full diversity when applying the iterative belief propagation (BP) decoder at the destination due to the pseudo-codewords [7]. LDPC codes on the other hand, can achieve full diversity in BF channels whilst using a BP decoder, provided that the LDPC code design is based on root-check LDPC (RC-LDPC) codes [6, 8, 9]. For AWGN channels or fast fading channels, related work using BP decoding can be found elsewhere, *e.g.* [10–13].

RC-LDPC codes are intentionally constructed to ensure that each information digit has one edge connected to an RC node, so that all the information digits can obtain full diversity gain in the BF channels. In [14], the check splitting method of [15] is utilized with RC-LDPC codes to achieve full diversity in the automatic-repeat-request (ARQ) BF channels. However, the *connection constraint* in the RC-LDPC codes is that all the information digits must be connected to the RC nodes. This connection constraint reduces the randomness of the connections between the variable nodes and the check nodes in the code design. The structures of RC LDPC codes are designed to achieve full diversity, however, the coding gain of such codes is not optimal. It is the main purpose of this work to introduce a new RC-LDPC code design which obtains full diversity whilst delivering improved coding gain relative the above mentioned earlier works.

More specifically, we study RC-LDPC code design in the context of BF channels, where we introduce the notion of check splitting into the code design. Such codes, termed check-splitting RC-LDPC (CSRC-LDPC) codes, are actually the repetition codes in the ARQ stages [14]. Even if full diversity is achieved by the CSRC-LDPC codes in the ARQ BF channels, the use of repetition codes in the ARQ stages again limits the coding gain. Here we propose code structures based on $(3, 6)$-regular LDPC codes, which can achieve full diversity gain for both the BF channels, and the ARQ BF channels (with check splitting method). Our proposed code structures relax the connection constraints in the RC-LDPC codes in that not all the information digits need be connected to the RC nodes. Our proposed scheme provides more freedom in connecting the variable nodes and the check nodes. We demonstrate that our proposed codes, some of which utilize the check splitting method, can offer a higher coding gain relative to conventional RC-LDPC codes.

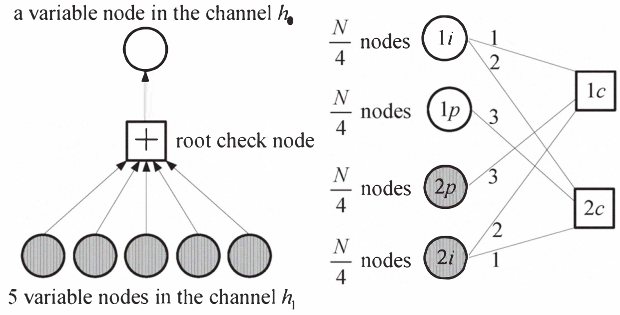
II. Channel model and RC-LDPC codes

Consider the codewords of N binary digits transmitted on a BF channel. During a codeword period, there are n independent fading intervals. The length N is a multiple of n, with $\lambda \triangleq N/n$ denoting the number of bits per fading block. The received signal is given by

$$y_i = h_j x_i + v_i, \tag{1}$$

where the subscripts $i = 0, \cdots, N-1$, $j = 0, \cdots, n-1$, and $j = \lfloor i/\lambda \rfloor$, and where $\lfloor r \rfloor$ denotes the integer part of a real number r. Each transmitted symbol x_i is a BPSK symbol with $x_i = \pm\sqrt{E_s}$. The noise samples are $i.i.d$

978-1-4244-xxxx-x/10/$26.00 ©2010 Crown



(a) Concept of root check node (b) Tanner graph

Fig. 1. A $(3,6)$-regular RC-LDPC code and its Tanner graph. The left figure illustrates the concept of RC nodes and the right figure is the corresponding Tanner graph.

with $v_i \sim \mathcal{N}(0,\sigma^2)$, and $\sigma^2 = N_0/2$. We assume that the channel gains $h_0, \cdots, h_{n-1}$ in a codeword are $i.i.d$ Rayleigh distributed from block to block. We denote the average signal-to-noise ratio (SNR) per symbol as $\rho = E_s/N_0$. When the code rate is R, the average SNR per bit is denoted as $E_b/N_0 = \rho/R$.

According to the Singleton bound, the relationship between the code rate R and the diversity gain d in the BF channels is

$$d \leq 1 + \lfloor n(1-R) \rfloor. \tag{2}$$

So to obtain the full diversity gain $d = n$, a necessary condition is that the code rate R should be no more than $1/n$.

In addition, for LDPC codes with a BP decoder, the root check nodes are utilized to design the parity check matrix of a full diversity achieving code [6, 8]. The concept of RC nodes is illustrated by a $(3,6)$-regular LDPC code in Fig. 1(a) with $n = 2$ and $R = 1/2$. Suppose that the two channel gains are h_0 and h_1. If a check node has one edge connected to a variable node (information digit) with channel gain h_0, and five edges connected to the variable nodes with channel gain h_1, then the check node is defined as an RC node for the variable node with a channel gain h_0. This variable node can obtain additional diversity from its RC node.

In order to achieve full diversity during a codeword transmission, each information digit must have one edge connected to its RC node. The $(3,6)$-regular RC-LDPC code presented in [6, 8] is designed to guarantee this connection for each information digit. Fig. 1(b) shows the corresponding Tanner graph. The idea behind the $(3,6)$-regular RC-LDPC code is straightforward: A codeword is split into four parts as shown in Fig. 1(b), in which, $1i$ and $2i$ represent the variable nodes related to the information digits transmitted in the two fading channels h_0 and h_1, respectively; and $1p$ and $2p$ represent the variable nodes related to the parity digits transmitted in the two fading channels h_0 and h_1, respectively. The check nodes in $1c$ and $2c$ are the RC nodes for the variable nodes in $1i$ and $2i$, respectively. Hence all the information digits obtain 2 orders of diversity gain.

From the Tanner graph in Fig. 1(b), we can see that the connections between the variable nodes and the check nodes are fixed in RC-LDPC codes. It is well known in the AWGN channels [10] and the ergodic fading channels [11], for a given LDPC code, *e.g.* a regular code or an irregular code with fixed degree distribution, the connections between the variable nodes and the check nodes do not change code performance (if we do not consider the cycles in the code). However, this is not the case in the non-ergodic BF channels due to their asymmetry.

To address this issue, we investigate the extrinsic mutual information transfer (EXIT) of the Tanner graph in Fig. 1(b). For a channel realization h_j, we utilize the "soft bit" to estimate a transmitted (modulated) bit x_i given a channel observation y_i, *i.e.* $\xi = Pr(x_i = 1|y_i, h_j) - Pr(x_i = -1|y_i, h_j)$. Also, we represent the bit's log likelihood ratio (LLR) as $\mathcal{L} = \log\left(Pr(x_i = 1|y_i, h_j)/Pr(x_i = -1|y_i, h_j)\right)$. If $x_i = 1$ is transmitted, then the mean of $\mathcal{L}$ is $m_{\mathcal{L}} = 2|h_j|^2/\sigma^2$. According to [13], the conditional channel mutual information $\mathrm{I}(x_i; y_i|h_j)$ is giveb by the function $\mathrm{J}(m_{\mathcal{L}})$ where,

$$\begin{aligned} \mathrm{J}(m_{\mathcal{L}}) &= \sum_{k=1}^{\infty} \Gamma_k \Phi_k(m_{\mathcal{L}}), \text{ where } \Gamma_k = \frac{\ln 2}{2k(2k-1)}, \\ \Phi_k(m_{\mathcal{L}}) &= \int_{-1}^{1} \frac{\sigma \xi^{2k}}{(1-\xi^2)|h_j|\sqrt{2\pi}} e^{-\frac{\left(\ln \frac{1+\xi}{1-\xi} - \frac{2|h_j|^2}{\sigma^2}\right)^2}{8|h_j|^2/\sigma^2}} \mathrm{d}\xi. \end{aligned} \tag{3}$$

Note that, in Fig. 1(b) there are six types of edges. We denote $\mathcal{E}_{1i1c}$ as the type of edges between $1i$ and $1c$. Similarly, we define the other five types of edges as $\mathcal{E}_{1i2c}$, $\mathcal{E}_{1p2c}$, $\mathcal{E}_{2p1c}$, $\mathcal{E}_{2i1c}$, and $\mathcal{E}_{2i1c}$. Because of the asymmetry of the BF channel, the EXIT, as well as the LLR values between the variable nodes and the check nodes, can only be averaged inside each edge type. We denote the LLR mean values observed in different channels as m with the corresponding subscripts (*e.g.* the LLR mean value of the channel h_0 is m_{h_0}). We also denote the LLR mean values transmitted along different edge types as m with the corresponding subscripts (*e.g.* the mean of the check-to-variable LLR value along the edge type $\mathcal{E}_{1i1c}$ is $m_{1c\to 1i}$, and the variable-to-check LLR mean value along the edge type $\mathcal{E}_{1i1c}$ is $m_{1i\to 1c}$). Without loss of generality, we focus on a variable node in $1i$ and the three edges connected to it (one edge in $\mathcal{E}_{1i1c}$ and the other two edges in $\mathcal{E}_{1i2c}$). By applying the Gaussian approximation (GA) method, we obtain the variable-to-check update rules in the l-th iteration as

$$\begin{aligned} m_{1i\to 1c}^{(l)} &= m_{h_0} + 2m_{2c\to 1i}^{(l-1)}, \\ m_{1i\to 2c}^{(l)} &= m_{h_0} + m_{1c\to 1i}^{(l-1)} + m_{2c\to 1i}^{(l-1)}. \end{aligned} \tag{4}$$

For the check-to-variable update rules, we have

$$\begin{aligned} m_{1c\to 1i}^{(l)} &= \mathrm{J}^{-1}\left(\sum_{k=1}^{\infty} \Gamma_k \Phi_k(m_{2i\to 1c}^{(l)})^2 \Phi_k(m_{2p\to 1c}^{(l)})^3\right), \\ m_{2c\to 1i}^{(l)} &= \\ \mathrm{J}^{-1}&\left(\sum_{k=1}^{\infty} \Gamma_k \Phi_k(m_{2i\to 2c}^{(l)}) \Phi_k(m_{1i\to 2c}^{(l)}) \Phi_k(m_{1p\to 2c}^{(l)})^3\right). \end{aligned} \tag{5}$$

Suppose the hard decision is made at the end of the l-th iteration, the output LLR mean value, denoted as $m_{1i}^{(l)}$, is given by

$$m_{1i}^{(l)} = m_{h_0} + m_{1c\to 1i}^{(l)} + 2m_{2c\to 1i}^{(l)}. \quad (6)$$

An outage event happens if $\mathrm{J}(m_{1i}^{(l)}) < 1$.

Due to the fact that the variable nodes in a codeword go through two different channels, a change of the connection in Fig. 1(b) may lead to the change of the instantaneous EXIT from both the variable nodes and the check nodes. Thus, the OP will be different for different connections. For example, the output mutual information by a node in $1i$ after the first iteration is

$$\sum_{k=1}^{\infty}\Gamma_k\Big(\Phi_k(m_{h_0})+\Phi_k(m_{h_1})^5+2\Phi_k(m_{h_1})\Phi_k(m_{h_0})^4\Big), \quad (7)$$

and the OP of the nodes in $1i$ after the l-th iteration is

$$P_{out} = Pr\Bigg(\sum_{k=1}^{\infty}\Gamma_k\Big(\Phi_k(m_{h_0})+\Phi_k(m_{2i\to 1c}^{(l)})^2\Phi_k(m_{2p\to 1c}^{(l)})^3 + 2\Phi_k(m_{2i\to 2c}^{(l)})\Phi_k(m_{1i\to 2c}^{(l)})\Phi_k(m_{1p\to 2c}^{(l)})^3\Big) < 1\Bigg). \quad (8)$$

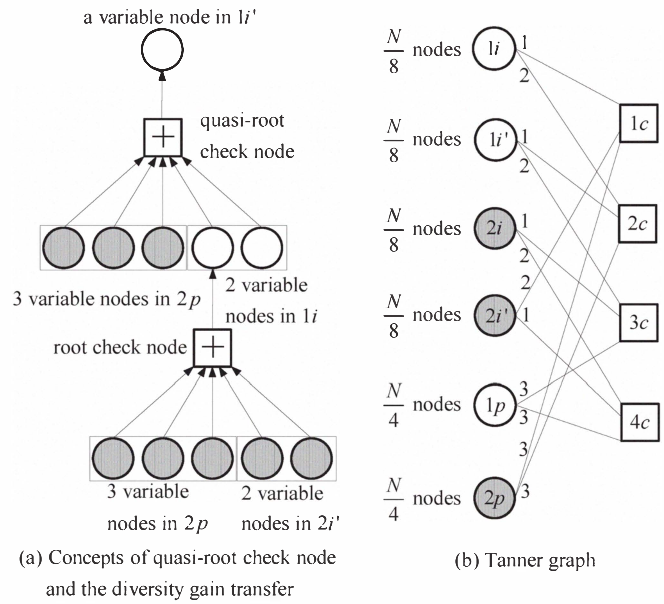
We can minimize the OP in (8) by finding the optimal connection at the check nodes, therefore obtaining a higher coding gain.

However, the connections in the RC-LDPC codes is fixed in order to guarantee the full diversity of a transmitted codeword. Therefore it is hard for the RC-LDPC codes to improve the coding gain. In the following, we present the design methodology of modified code structures which have more freedom in the connection without reducing the diversity gain.

III. Structures of Diversity Achieving Codes

Recall from the RC-LDPC codes shown in Fig. 1 that the extra diversity of a digit is obtained through its RC node. Without loss of generality let us focus on the RC node ($1c$) and a digit in $1i$. We connect $1i$ digit with one edge to $1c$, and connect the digits in $2i$ to $1c$ with the remaining edges. Such connections represent one way to achieve full diversity for the digits in the codeword. The main idea of our new proposed code structures is that we allow for a different range of connections splitting the digits in $1i$ into two parts as shown in Fig. 2. Each digit in the first part of $1i$ has one edge connected to the RC nodes in order to obtain the extra diversity. Each digit in the other part of $1i$ has one edge connected to the so-called quasi-root check (QRC) nodes. The remaining edges of a QRC node can be connected to either the digits in $2i$, or the digits in $1i$ that have already obtained full diversity gain. Our codes structures relax the connection constraints in the RC-LDPC codes and hence there are more code ensembles satisfying the condition.

Fig. 2 illustrates a full diversity achieving $(3,6)$-regular LDPC code. In the Tanner graph, the check nodes in $1c$ and $3c$ are the RC nodes for $1i$ and $2i$, respectively. Likewise, the check nodes in $2c$ and $4c$ are the QRC nodes for $1i'$ and $2i'$, respectively. Note that since the variable nodes in $1i$ obtain the extra diversity in the first iteration, the variable nodes in $1i'$ achieve the full diversity in the second iteration. The principle that a variable node can obtain the extra diversity from its QRC node is termed *diversity population evolution* [9]. We focus on a variable node in $1i'$ to explain the process of diversity population evolution. The node in $1i'$ has one edge connected to its QRC node, with the other five edges of the QRC node connected to $1i$ and $2p$. According to [6, 8], a sub-optimal decoder (*i.e.* min-sum decoder) is utilized to investigate the output LLR value of a variable node. Suppose that an all-zero codeword has been transmitted, and $\epsilon^{(l)}$ is the average negative-probability of the input LLR value to a check node in the l-th iteration. Then in the first iteration, we have



(a) Concepts of quasi-root check node and the diversity gain transfer (b) Tanner graph

Fig. 2. A full diversity achieving $(3,6)$-regular LDPC code.

$$\begin{aligned} m_{1c\to 1i}^{(1)} &= (1-2\epsilon^{(1)})^4 \min\{m_{2i'\to 1c}^{(1)}, m_{2p\to 1c}^{(1)}\} \\ &= (1-2\epsilon^{(1)})^4 \frac{2|h_1|^2}{\sigma^2}, \end{aligned} \quad (9)$$

In the second iteration, we have

$$\begin{aligned} m_{1i\to 2c}^{(2)} &= m_{h_0} + m_{1c\to 1i}^{(1)} + m_{2c\to 1i}^{(1)} \\ &= \frac{2|h_0|^2}{\sigma^2} + (1-2\epsilon^{(1)})^4\frac{2|h_1|^2}{\sigma^2} + m_{2c\to 1i}^{(1)}. \end{aligned} \quad (10)$$

Then the average output LLR value of the nodes in $1i'$ is

$$\begin{aligned} m_{1i'}^{(2)} &= m_{h_0} + m_{2c\to 1i'}^{(2)} + 2m_{3c\to 1i'}^{(2)} \\ &= \frac{2|h_0|^2}{\sigma^2} + (1-2\epsilon^{(2)})^4 \min\{m_{1i\to 2c}^{(2)}, m_{2p\to 2c}^{(2)}\} + 2m_{3c\to 1i'}^{(2)} \end{aligned} \quad (11)$$

In fact, we have $\epsilon^{(1)} > \epsilon^{(2)} > \cdots > \epsilon^{(l)}$, as the variable nodes obtain more EXIT after each iteration, and $\epsilon^{(1)} \to 0$ when SNR is large enough. From (10) and (11), the diversity gain obtained by the nodes in $1i$ can be transferred to $1i'$ and thus the nodes in $1i'$ also achieve full diversity. In the same manner, the nodes in $2i'$ also obtain the full diversity gain. Note that,

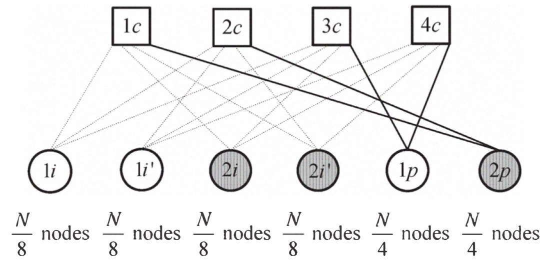


Fig. 3. Tanner graph of another full diversity achieving $(3, 6)$-regular LDPC code designed according to the principle of *diversity population evolution.* The dotted lines represent one edge and the solid lines represent three edges.

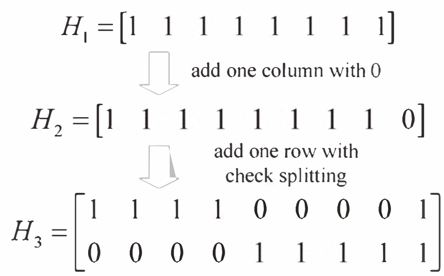


Fig. 4. Increasing one digit redundancy by splitting a check node in to two.

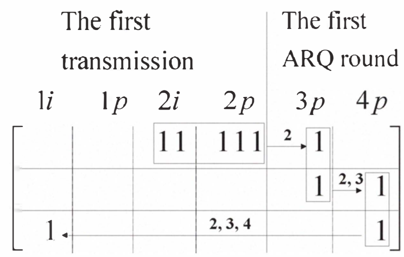


Fig. 5. Parity check matrix relative to a digit of $1i$ and its root check nodes after the first ARQ round. There are four $i.i.d$ channels indexed as $1, 2, 3, 4$. The arrows represent the process of diversity population evolution. The number over the arrow represents the index of the channel that provides the diversity gain.

due to the delay of diversity population evolution, the variable nodes in $1i'$ and $2i'$ obtain the extra diversity after the second iteration. While in the RC-LDPC codes, all the information digits achieve full diversity after the first iteration.

Due to the principle of diversity population evolution, we can design a group of code structures that achieve full diversity gain. For example, Fig. 3 is another structure of a full diversity achieving $(3, 6)$-regular LDPC code. We can also adapt the percentages of the variable nodes in $1i'$ and $2i'$ without reducing the diversity order. Among all the full diversity achieving $(3, 6)$-regular LDPC codes, there must be a code structure that minimizes the OP in (8). Moreover, we can jointly design the degree distribution and the connection of the Tanner graph by minimizing the OP in (8) to obtain a coding gain. In this paper we do not go into detail on the design of full diversity achieving codes with a higher coding gain. Instead, we will present the application of the proposed code structure for the ARQ BF channels with check splitting method.

IV. The application of full diversity achieving codes to ARQ BF channels

In the ARQ BF channels, puncturing/extending was often used for LDPC codes to obtain incremental redundancy. However, in the puncturing method, every punctured variable node disables several check nodes. The extending method may create many cycles in the code graphs or create parity check nodes with insufficient weight. In [15], the check splitting method was proposed in order to obtain incremental redundancy. Check splitting operates by replacing a row r in the parity check matrix with two new rows r_1 and r_2, of approximately equal weight and with $r_1 \oplus r_2 = r$. This causes the check nodes degree distribution to remain fairly concentrated and prevents the cycles in the Tanner graph as the rate decreases. For example, if we want to reduce the code rate by adding parity check digits to the codewords, we need to add one row and one column to the original parity check matrix. Ignoring the columns containing zeros in both the original and the new checks, Fig. 4 illustrates how to split a single check node of degree eight into two check nodes of degree five.

In [14], the check splitting method is applied to the RC-LDPC codes for the ARQ BF channels. Consider a RC-LDPC code with rate $R = \frac{1}{n}$ and code length λ. In each ARQ round, the source transmits λ redundant parity check digits to reduce the code rate. So after the l-th ARQ round, the code rate becomes $R_l = R/l$. To make sure that the overall codeword with the rate R_l can achieve full diversity after the l-th ARQ round, each check node should be retained as an RC node after check splitting. Without loss of generality, we pick out a digit of $1i$ in Fig. 1 to show the check splitting and the full-diversity-achieving process by the diversity population evolution.

Fig. 5 shows the check splitting after the first ARQ round, and the process of diversity population evolution. This illustrates how the digit of $1i$ can eventually achieve full (4 order) diversity. The digits in $2i$ also achieve full diversity in the same manner. However, because of the code structure of the RC-LDPC codes, the check splitting is mainly adding repetition check nodes in each ARQ round. Note that in the last row shown in Fig. 5, $1i$ and $4p$ form a repetition code. Similarly, $3p$ and $4p$ form a repetition code. These repetition codes ensure the diversity but do not improve coding gain. We can also see that the degree of a check node will be either 2 or 6. So the degree distribution of the check nodes is not concentrated.

We next apply check splitting to the LDPC code in Fig. 2. The check splitting and the diversity population evolution are shown in Fig. 6. We can see that all the digits $1i, 1i', 2i$ and $2i'$ achieve full diversity by the diversity population evolution. In addition, the check nodes produced by the check splitting method are no longer the repetition codes. Most of the check nodes (75%) have the concentrated degrees of 3 and 4, and only a small portion of the check nodes (25%) have the degree 2 and 6. So the code structure in Fig. 2 with the check splitting can lead to a higher coding gain than relative to an RC-LDPC

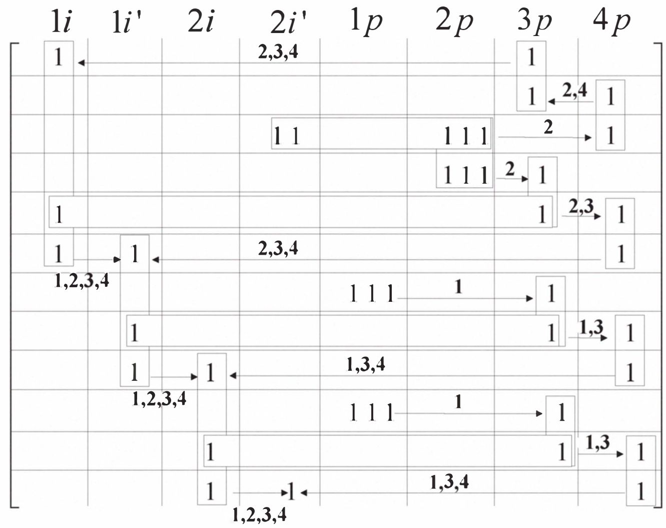


Fig. 6. Parity check matrix relative to the digits of $1i, 1i', 2i, 2i'$ and their root check nodes after the first ARQ round.

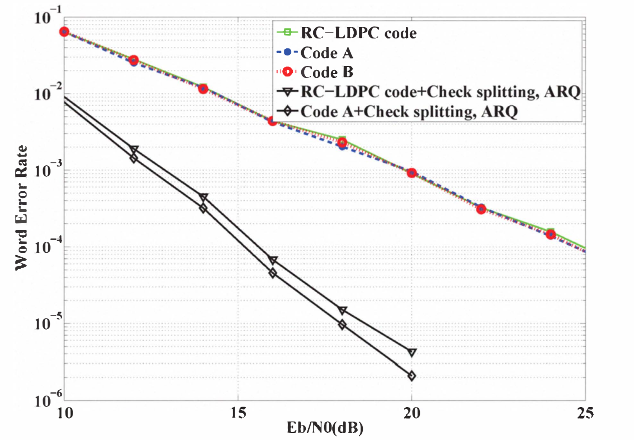


Fig. 7. Word error rate (WER) curves for various codes. First three curves are for the BF channels and the last two curves are for the ARQ BF channels. All the code lengths are 800.

code.

Fig. 7 shows the word error rate for various codes. We use the code in Fig. 1 as the benchmark, which is a $(3,6)$-regular RC-LDPC code. In addition, we denote the code in Fig. 2 as Code A, and the code in Fig. 3 as Code B. We can see from Fig. 7 that both Code A and Code B achieve full diversity under the principle of diversity population evolution. Note that the three codes almost have the same performance in the BF channel. For the ARQ BF channel, we assume there is one ARQ round. So the full diversity of the scheme is 4. The simulation results show that the combination of check splitting and RC-LDPC code can achieve full diversity gain, but that its coding gain is smaller relative to a check splitting version of Code A.

V. Conclusion

In this paper, we have focussed on the design of full diversity achieving LDPC codes for nonergodic BF channels. The codes we have proposed provide more freedom in connecting the variable nodes and the check nodes while keeping the same diversity as the RC-LDPC codes. Due to randomness in the parity check matrix, our code structures may be used to search for full diversity achieving codes with higher coding gains. We have shown, when combined with the check splitting method in the ARQ BF channels, our code can provider better error performance than the RC-LDPC codes.

VI. Acknowledgement

This work has been supported by the Australian Research Council under ARC Grant DP087940.

References

[1] L. H. Ozarow, S. Shamai (Shitz), and A. .D. Wyner "Information theotic considerations for cellular mobile radio," *IEEE Trans. Vehicular Tech.*, vol. 43, no. 2, pp. 359-378, May 1994.

[2] R.C. Singleton, "Maximum Distance qCNary Codes," *IEEE Trans. Inf. Theory*, 10:116C118, 1964.

[3] R. Knopp and P. A. Humblet, "On coding for block fading channels," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 189-205, Jan. 2000.

[4] E. Malkamaki and H. Leib, "Evaluating the performance of convolutional codes over block fading channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1643-1646, July 1999.

[5] A. Guillén i Fàbregas and G. Caire, "Coded modulation in the block-fading channel: Coding theorems and code construction," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 91-114, Jan. 2006.

[6] J. J. Boutros, A. Guillén i Fàbregas, E. Biglieri, G. Zemor, "Low-density parity-check codes for nonergodic block-fading channels," *submitted to IEEE Trans. Inf. Theory*, arXiv:0710.1182v1.

[7] C. A. Kelley and D. Sridhara, "Pseudocodewords of Tanner graphs," *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 4013-4038, Nov. 2007.

[8] J. J. Boutros, A. Guillén i Fàbregas, E. Biglieri, G. Zemor, "Design and Analysis of low-density parity-check codes for block-fading channels," in *Proc. Information Theory and Applications Workshop (ITA)*, pp. 54-62, Jan. 2007.

[9] J. J. Boutros, "Diversity and coding gain evolution in graph codes," *Proc. Information Theory and Applications Workshop (ITA)*, pp. 34-43, 2009.

[10] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619-637, Feb. 2001.

[11] J. Hou, P. H. Siegel, and L. B. Milstein, "Performance analysis and code optimization of low density parity-check codes on rayleigh fading channels," *IEEE J. of Sel. Areas Commun.*, vol. 19, no. 5, pp. 924-934, May 2001.

[12] S. Brink, G. Kramer, and A.Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670-678, Apr. 2004.

[13] 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.

[14] S. Kambhampati, G. Lechner, T. Chan, and L. K. Rasmussen, "Check splitting of root-check LDPC codes over ARQ block-fading channels," *Australian Communications Theory Workshop (AusCTW)*, Canberra, Australia, Feb. 2-6, 2010.

[15] M. Good, F. Kschischang, "Incremental redundancy via check splitting," *23rd Biennial Symp. on Comm.*, pp. 55-58, May-Jun. 2006.