Generalised Quasi-Cyclic LDPC Codes Based on Progressive Edge Growth Techniques For Block Fading Channels

André G. D. Uchôa[1], Cornelius Healy[2] and Rodrigo C. de Lamare[3]
University of York,
Heslington York, YO10 5DD, UK
agdu@ieee.org[1], cth503@york.ac.uk[2], rcdl500@york.ac.uk[3]

Richard D. Souza
CPGEI, UTFPR,
Curitiba-PR, Brazil
richard@utfpr.edu.br

***Abstract*—Generalised Quasi-Cyclic Root-Check LDPC codes based on Progressive Edge Growth (PEG) techniques for block-fading channels are proposed. The proposed Root-Check LDPC codes are suitable for channels under $F = 3,4$ independent fadings per codeword which is a scenario that has not been previously considered. A generalised Quasi-Cyclic Root-Check structure is devised for $F = 3,4$ independent fadings. The proposed PEG-based algorithm for generalised Quasi-Cyclic Root-Check LDPC codes takes into account the specific structure, and designs a parity-check matrix with a reduced number of cycles. The performance of the new codes is investigated in terms of the Frame Error Rate (FER). Numerical results show that the codes constructed by the proposed algorithm perform about 0.5dB from the theoretical limit.**

***Index Terms*—LDPC, Root-Check, PEG, Outage Probability**

I. Introduction

Due to multi-path propagation and mobility, wireless systems are characterized by time-varying channels with fluctuating signal strength. In applications subject to delay constraints and slowly-varying channels, only limited independent fading realizations are experienced. In such non-ergodic scenarios, the channel capacity is zero since there is an irreducible probability, termed outage probability [1], that the transmitted data rate is not supported by the channel. The case of interest in this work is the block-fading type [2]. The block-fading channel is a simple and useful model that captures the essential characteristics of non-ergodic channels [2]. It is especially important in wireless communications with slow time-frequency hopping (e.g., cellular networks and wireless Ethernet) or multi-carrier modulation using Orthogonal Frequency Division Multiplexing (OFDM) [3].

In [3] the authors proposed a family of LDPC codes called Root-Check for block-fading channels. Root-check codes are able to achieve the maximum diversity of a block-fading channel and have a performance near the limit of outage when decoded using an iterative decoding Sum Product Algorithm (SPA). Root-check codes are always designed with code rate $R = 1/F$, since the Singleton bound determines that this is the highest code rate possible to obtain the maximum diversity order [3]. In representation of a bipartite graph of Root-Check LDPC codes, a sub-set of connections is deterministically selected to ensure maximum diversity for information bits and the others connections are randomly generated.

Li and Salehi [4] proposed the construction of structured Root-Check LDPC codes via circulating matrices, i.e., Quasi-Cyclic LPDC codes (QC-LDPC). In [4] the authors show that the QC-LDPC codes are as good as the Root-Check LDPC codes randomly generated on Block-Fading channels. It is known that the girth, the length of the shortest cycle in the graph of this code has a significant effect on the performance of the code. For regular codes, the number of independent messages passed in the SPA decoder is proportional to the girth of the code [5]. In [4] an algorithm to generate QC-LDPC codes with large girth is presented. This algorithm is based on random selection of the coefficients for a right cyclic shift of the square matrices of QC-LDPC code. The essence of this algorithm is to generate random coefficients until a code with the desired girth is found. Among the algorithms capable of producing high performance LDPC codes for short to medium lengths is the Progressive Edge Growth (PEG) algorithm [6].

In order to improve the girth of the Root-Check LDPC codes the PEG-Root-Check LDPC codes [7], [8] which are designed in a PEG based technique have been developed. In these works, we only considered the case of a block-fading channel with $F = 2$. The proposed PEG-Root-Check LDPC codes presented in [7] outperformed other LDPC codes based on Root-Check. The best result presented by the PEG-Root-Check LDPC code reinforces that a Root-Check LDPC code generated with an algorithm based on PEG produces better performance.

Currently there are few studies in the literature related to the design of LDPC codes for Block-Fading channels. The main works found are the following: [3], [4], [7], [8]. These works focused only on code designs of rate one-half capable of achieving full diversity on channels with $F = 2$ fadings per codeword. An important fact is that it has not been found in the seminal papers of LDPC codes for Block-Fading channels designs for other types of fading, e.g., $F > 2$.

The contribution of this paper is to present a PEG-based algorithm to design QC-LDPC codes with Root-Check properties for Block-Fading channel with $F = 3,4$ fadings per coded word. A strategy that imposes Quasi-Cyclic and Root-Check constraints on a PEG-based algorithm is devised. The codes generated by the proposed algorithm can achieve a significant performance in terms of FER with respect to the theoretical limit. We have also presented the design of QC-Root-Check-LDPC codes to compare with our proposed QC-PEG-Root-Check LDPC codes. The new design can save up to 3dB in terms of signal to noise ratio (SNR) to achieve the same FER when compared to other codes.

The rest of this paper is organized as follows. In Section II we define the system model, while in Section III we present the structure of a QC-Root-Check LDPC codes for $F = 3, 4$ fadings. In Section IV we introduce the proposed PEG-based algorithm. Section V presents numerical results, while Section V concludes the paper.

II. System Model

Consider a block fading channel, where F is the number of independent fading blocks per codeword of length N. Following [4], the t-th received symbol is given by:

$$r_t = h_f s_t + ng_t, \tag{1}$$

where $t = \{1, 2, \cdots, N\}$, $f = \{1, 2, \cdots, F\}$, f and t are related by $f = \lceil F \frac{t}{N} \rceil$, where $\lceil \phi \rceil$ returns the smallest integer not smaller than ϕ, h_f is the real Rayleigh fading coefficient of the f-th block, s_t is the transmitted signal, and ng_t is additive white Gaussian noise with zero mean and variance $N_0/2$. In this paper, we assume that the transmitted symbols s_t are binary phase shift keying (BPSK) modulated. We assume that the receiver has perfect channel state information, and that the SNR is defined as E_b/N_0, where E_b is the energy per information bit. The information transmission rate is $R = K/N$, where K is the number of information bits per codeword of length N. We consider $R = 1/F$, since then it is possible to design a practical diversity achieving code [4].

The performance of a communication system in a non-ergodic block fading channel can be investigated by means of the outage probability [4], which is defined as:

$$P_{out} = \mathcal{P}(I < R), \tag{2}$$

where $\mathcal{P}(\phi)$ is the probability of event ϕ. The mutual information I_G, supposing Gaussian channel inputs, is [4]:

$$I_G = \frac{1}{F} \sum_{f=1}^{F} \frac{1}{2} \log_2 \left(1 + 2R \frac{E_b}{N_0} h_f^2 \right), \tag{3}$$

so that an outage occurs when the average accumulated mutual information among blocks is smaller than the attempted information transmission rate.

III. QC-LDPC PCM Structure

The parity check matrix (PCM) $\mathbf{H}$ of a QC-LDPC code can be put in the form below [9]:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{0,0} & \mathbf{H}_{0,1} & \cdots & \mathbf{H}_{0,w-1} \\ \mathbf{H}_{1,0} & \mathbf{H}_{1,1} & \cdots & \mathbf{H}_{0,w-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{c-1,0} & \mathbf{H}_{c-1,1} & \cdots & \mathbf{H}_{c-1,w-1} \end{bmatrix}, \tag{4}$$

where $\mathbf{H}_{ij}$ is an $n \times n$ circulant or all-zeros matrix, and c and w are two positive integers with $c < w$. The null space of $\mathbf{H}$ gives a QC-LDPC code over $GF(2)$ of length $N = wn$. The rank of $\mathbf{H}$ is at most cn. Hence the code rate is at least $\frac{w-c}{w}$.

A. QC-PEG-Root-Check LDPC $R = \frac{1}{3}$

In the case of a Root-Check LDPC code, it is required to divide both variable and check nodes in F equal parts. Following the non-systematic Root-Check structure reported in [3], the PCM becomes:

$$\mathbf{H} = [\mathbf{S}_1\mathbf{P}_1, \cdots, \mathbf{S}_F\mathbf{P}_F], \tag{5}$$

where the subscripts represent the variable nodes (systematic and parity respectively) under an specific fading block. A systematic PCM can be defined as being $\mathbf{H} = [\mathbf{A}|\mathbf{B}]$, where $\mathbf{A}$ are connected to the systematic and $\mathbf{B}$ are connected to the parity bits, respectively. So, if we consider the structure in (5) in systematic form, the PCM becomes $\mathbf{H} = [\mathbf{S}_1, \cdots, \mathbf{S}_F\mathbf{P}_1, \cdots, \mathbf{P}_F]$. In order to obtain the generator matrix, the sub-matrix $\mathbf{B}$ formed by parity matrices $\mathbf{P}_1, \cdots, \mathbf{P}_F$ must be a non-singular matrix, which means it is invertible under $GF(2)$ [4].

Now, consider a Block-Fading channel with $F = 3$ fadings per codeword. To design a practical code which is able to achieve the channel diversity, the rate of such code must be $R = \frac{1}{F} = \frac{1}{3}$. We have figured out one way to design such Root-Check code in a structured Quasi-Cyclic LDPC code. As a result, the PCM for $R = \frac{1}{3}$ can be defined as in (6), where $\mathbf{H}_{ij}$ and $\mathbf{I}_{ij}$ are $n \times n$ circulant matrices, while $\mathbf{0}$ is an all-zeros matrix. The notation $\mathbf{I}_{ij}$ was used to reinforce that such connections are the Root-Check connections [3]. The restrictions that should be imposed are only the $\mathbf{I}_{ij}$ to be placed in the positions described in (6) and the upper and down triangular sub-matrices in the parity part, $\mathbf{B}$, of $\mathbf{H}$. The upper and down triangular sub-matrices in $\mathbf{H}$ are there to ensure that $\mathbf{B}$ is invertible under $GF(2)$. Once $\mathbf{B}$ is invertible we can generate the generator matrix $\mathbf{G}$ as discussed in [4]. Then, in order to perform our PEG-based design we just need the polynomials which describe the variable and check nodes degrees for the PCM defined in (6). In this work we consider:

$$\lambda(x) = 0.5x^3 + 0.25x^2 + 0.17x + 0.08 \tag{7}$$

for the degree distribution of the variable nodes, and

$$\rho(x) = 0.42x^4 + 0.33x^3 + 0.25x^2 \tag{8}$$

for the degree distribution of the check nodes.

B. QC-PEG-Root-Check LDPC $R = \frac{1}{4}$

Similarly to the case of $R = \frac{1}{3}$, in this section we describe the PCM structure for a $R = \frac{1}{4}$ QC-Root-Check LDPC code. The PCM can be defined as in (9), With respect to our proposed QC-PEG-Root-Check LDPC codes design, there are some restrictions that should be imposed which are the same as those described for $R = \frac{1}{3}$. Moreover, in the case of $R = \frac{1}{4}$ we consider the polynomials

$$\lambda(x) = 0.57x^7 + 0.21x^2 + 0.14x + 0.07, \tag{10}$$

for the degree distribution of the variable nodes, and

$$\rho(x) = 0.43x^5 + 0.27x^4 + 0.14x^3 + 0.07x^2, \tag{11}$$

for the degree distribution of the check nodes.

$$
\mathbf{H} = \begin{bmatrix}
\mathbf{I_{0,0}} & \mathbf{H_{0,1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{0,6}} & \mathbf{H_{0,7}} & \mathbf{H_{0,8}} \\
\mathbf{I_{1,0}} & \mathbf{0} & \mathbf{H_{1,2}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{1,7}} & \mathbf{H_{1,8}} \\
\mathbf{H_{2,0}} & \mathbf{I_{2,1}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{2,8}} \\
\mathbf{0} & \mathbf{I_{3,1}} & \mathbf{H_{3,2}} & \mathbf{H_{3,3}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{H_{4,0}} & \mathbf{0} & \mathbf{I_{4,2}} & \mathbf{H_{4,3}} & \mathbf{H_{4,4}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{H_{5,1}} & \mathbf{I_{5,2}} & \mathbf{H_{5,3}} & \mathbf{H_{5,4}} & \mathbf{H_{5,5}} & \mathbf{0} & \mathbf{0} & \mathbf{0}
\end{bmatrix}, \tag{6}
$$

$$
\mathbf{H} = \begin{bmatrix}
\mathbf{I_{0,0}} & \mathbf{H_{0,1}} & \mathbf{0} & \mathbf{H_{0,3}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{0,10}} & \mathbf{0} & \mathbf{H_{0,12}} & \mathbf{0} & \mathbf{H_{0,14}} & \mathbf{0} \\
\mathbf{I_{1,0}} & \mathbf{0} & \mathbf{H_{1,2}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{1,11}} & \mathbf{0} & \mathbf{H_{1,13}} & \mathbf{0} & \mathbf{H_{1,15}} \\
\mathbf{I_{2,0}} & \mathbf{H_{2,1}} & \mathbf{0} & \mathbf{H_{2,3}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{2,12}} & \mathbf{0} & \mathbf{H_{2,14}} & \mathbf{0} \\
\mathbf{H_{3,0}} & \mathbf{I_{3,1}} & \mathbf{H_{3,2}} & \mathbf{H_{3,3}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{3,13}} & \mathbf{0} & \mathbf{H_{3,15}} \\
\mathbf{0} & \mathbf{I_{4,1}} & \mathbf{H_{4,2}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{4,14}} & \mathbf{0} \\
\mathbf{H_{5,0}} & \mathbf{I_{5,1}} & \mathbf{0} & \mathbf{H_{5,3}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H_{5,15}} \\
\mathbf{H_{6,0}} & \mathbf{0} & \mathbf{I_{6,2}} & \mathbf{0} & \mathbf{H_{6,4}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{H_{7,1}} & \mathbf{I_{7,2}} & \mathbf{0} & \mathbf{0} & \mathbf{H_{7,5}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{I_{8,2}} & \mathbf{H_{8,3}} & \mathbf{H_{8,4}} & \mathbf{0} & \mathbf{H_{8,6}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{H_{9,0}} & \mathbf{0} & \mathbf{H_{9,2}} & \mathbf{I_{9,3}} & \mathbf{0} & \mathbf{H_{9,5}} & \mathbf{0} & \mathbf{H_{9,7}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{H_{10,1}} & \mathbf{H_{10,2}} & \mathbf{I_{10,3}} & \mathbf{H_{10,4}} & \mathbf{0} & \mathbf{H_{10,6}} & \mathbf{0} & \mathbf{H_{10,8}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{H_{11,0}} & \mathbf{H_{11,1}} & \mathbf{0} & \mathbf{I_{11,3}} & \mathbf{0} & \mathbf{H_{11,5}} & \mathbf{0} & \mathbf{H_{11,7}} & \mathbf{0} & \mathbf{H_{11,9}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}
\end{bmatrix}. \tag{9}
$$

IV. Proposed Design Algorithm

Here, we introduce some definitions and notations. Then, we present the pseudo-code of our proposed algorithm. In this work the case of a block-fading channel with $F = 3$ and $F = 4$ are considered. In extending to a greater number of fadings, $F > 4$, the general structure presented is maintained, with the systematic VNs for each fading possessing Root-Check identity matrices connecting to parity VNs in each of the other fading blocks only, ensuring the upper and lower triangular sections of parity bits observed in (6) and (9). The placement of the remaining cyclic sub-matrices is required to maintain this relationship and provide satisfactory final code degree distribution. The LDPC code in systematic form is specified by its sparse PCM $\mathbf{H} = [\mathbf{A}|\mathbf{B}]$, where $\mathbf{A}$ is a matrix of size M-by-K, and $\mathbf{B}$ is an M-by-M matrix. The generator matrix for the code is $\mathbf{G} = [(\mathbf{B}^{-1}\mathbf{A})^T|\mathbf{I}_K]$, $\mathbf{I}_K$ is an identity matrix of size K.

The variable node degree sequence D_s is defined as the set of column weights of the designed $\mathbf{H}$, and is prescribed by the variable node degree distribution $\lambda(x)$ as described in [10]. Moreover, D_s is arranged in non-decreasing order. The proposed algorithm, called QC-PEG Root-Check, constructs $\mathbf{H}$ by operating progressively on variable nodes to place the edges required by D_s. The Variable Node (VN) of interest is labelled v_j and the candidate check nodes are individually referred to as c_i. The PEG Root-Check algorithm chooses a check node c_i to connect to the variable node of interest v_j by expanding a constrained sub-graph from v_j up to maximum depth l. The set of check nodes found in this sub-graph is denoted $N_{v_j}^l$ while the set of check nodes of interest, those not currently found in the sub-graph, are denoted $\overline{N_{v_j}^l}$. For the QC-PEG Root-Check algorithm, a check node is chosen at random from the minimum weight check nodes of this set.

A. Pseudo-code for the QC-PEG-Root-Check Algorithm

Initialization: A matrix of size $M \times N$ is created with the circulant identity matrices $I_{i,j}$ in the positions shown in (6) and (9) and zeros in all other positions. We define the indicator vectors $\mathbf{z}_1, \cdots, \mathbf{z}_{F^2}$ for the cases $R = \frac{1}{3}$, $R = \frac{1}{4}$ respectively as:

$$
\begin{aligned}
\mathbf{z}_1 &= [\mathbf{0}_{1\times\frac{2N}{9}}, \mathbf{1}_{1\times\frac{N}{9}}, \mathbf{0}_{1\times\frac{N}{9}}, \mathbf{1}_{1\times\frac{N}{9}}, \mathbf{0}_{1\times\frac{N}{9}}]^T, \\
\mathbf{z}_2 &= [\mathbf{1}_{1\times\frac{N}{9}}, \mathbf{0}_{1\times\frac{4N}{9}}, \mathbf{1}_{1\times\frac{N}{9}}]^T, \\
\mathbf{z}_3 &= [\mathbf{0}_{1\times\frac{N}{9}}, \mathbf{1}_{1\times\frac{N}{9}}, \mathbf{0}_{1\times\frac{N}{9}}, \mathbf{1}_{1\times\frac{N}{9}}, \mathbf{0}_{1\times\frac{2N}{9}}]^T, \\
\mathbf{z}_\chi &= [\mathbf{0}_{1\times\frac{(i-1)\cdot N}{9}}, \mathbf{1}_{1\times\frac{(7-i)N}{9}}]^T \quad \text{for } \chi = 4, 5, 6, \\
\mathbf{z}_\gamma &= [\mathbf{1}_{1\times\frac{(i-6)\cdot N}{9}}, \mathbf{0}_{1\times\frac{(12-i)N}{9}}]^T \quad \text{for } \gamma = 7, 8, 9,
\end{aligned} \tag{12}
$$

$$
\begin{aligned}
\mathbf{z}_1 &= [\mathbf{0}_{1\times\frac{3N}{16}}, \mathbf{1}_{1\times\frac{N}{16}}, \mathbf{0}_{1\times\frac{N}{16}}, \mathbf{1}_{1\times\frac{2N}{16}}, \mathbf{0}_{1\times\frac{2N}{16}}, \\
&\qquad \mathbf{1}_{1\times\frac{N}{16}}, \mathbf{0}_{1\times\frac{N}{16}}, \mathbf{1}_{1\times\frac{N}{16}}]^T, \\
\mathbf{z}_2 &= [\mathbf{1}_{1\times\frac{N}{16}}, \mathbf{0}_{1\times\frac{N}{16}}, \mathbf{1}_{1\times\frac{N}{16}}, \mathbf{0}_{1\times\frac{4N}{16}}, \mathbf{1}_{1\times\frac{N}{16}}, \\
&\qquad \mathbf{0}_{1\times\frac{2N}{16}}, \mathbf{1}_{1\times\frac{2N}{16}},]^T, \\
\mathbf{z}_3 &= [\mathbf{0}_{1\times\frac{N}{16}}, \mathbf{1}_{1\times\frac{N}{16}}, \mathbf{0}_{1\times\frac{N}{16}}, \mathbf{1}_{1\times\frac{2N}{16}}, \mathbf{0}_{1\times\frac{4N}{16}}, \\
&\qquad \mathbf{1}_{1\times\frac{2N}{16}}, \mathbf{0}_{1\times\frac{N}{16}}]^T, \\
\mathbf{z}_4 &= [\mathbf{1}_{1\times\frac{N}{16}}, \mathbf{0}_{1\times\frac{N}{16}}, \mathbf{1}_{1\times\frac{2N}{16}}, \mathbf{0}_{1\times\frac{N}{16}}, \mathbf{1}_{1\times\frac{N}{16}}, \\
&\qquad \mathbf{0}_{1\times\frac{2N}{16}}, \mathbf{1}_{1\times\frac{N}{16}}, \mathbf{0}_{1\times\frac{3N}{16}},]^T, \\
\mathbf{z}_\chi &= [\mathbf{0}_{1\times\frac{(i+1)N}{16}}, \mathbf{v}_{ALT}(0{:}\tfrac{(11-i)N}{16}-1)]^T \\
&\qquad \text{for } \chi = 5, \cdots, 10, \\
\mathbf{z}_\gamma &= [\mathbf{v}_{ALT}(\tfrac{(17-i)N}{16}{:}\tfrac{7N}{16}-1), \mathbf{0}_{1\times\frac{(22-i)N}{16}}]^T \\
&\qquad \text{for } \gamma = 11, \cdots, 16,
\end{aligned} \tag{13}
$$

$$
\mathbf{v}_{ALT} = [\mathbf{1}_{1\times\frac{N}{16}}, \mathbf{0}_{1\times\frac{N}{16}}, \mathbf{1}_{1\times\frac{N}{16}}, \mathbf{0}_{1\times\frac{N}{16}}, \mathbf{1}_{1\times\frac{N}{16}}, \mathbf{0}_{1\times\frac{N}{16}}, \mathbf{1}_{1\times\frac{N}{16}}] \tag{14}
$$

These indicator vectors are modelled on that of the original PEG algorithm [6], indicating submatrices for which placement is permitted, thus imposing the form of (6) or (9). The degree sequence as defined for LDPC codes must be altered to take into account the structure imposed by Root-Check codes, namely the circulant identity matrices, $I_{i,j}$, of (6) and (9). The pseudo-code for our proposed QC-PEG Root-Check algorithm is detailed in Algorithm 1, where the indicator vector, $\mathbf{z}_i$, is taken from (12), (13) for constructing codes of rate $R = \frac{1}{3}$, $R = \frac{1}{4}$ respectively.

Algorithm 1 QC-PEG Root-Check Algorithm

1. **for** $j = 1 : F^2$ **do**
2. **for** $k = 0 : D_s(j) - 1$ **do**
3. **if** $j \geq \frac{N}{F}$ & $k == 0$ **then**
4. Place edge at random among minimum weight submatrices permitted by the indicator $\mathbf{z}_j$, with a random first edge placement within the chosen submatrix, in column $\frac{(j-1)\cdot N}{F^2}$*-th*.
5. Place remaining edges in the submatrix by circulant shift of the first placement.
6. Null the entry in the indicator vector $\mathbf{z}_j$ in the position of the chosen submatrix, preventing further placements in that submatrix.
7. **else**
8. Expand the PEG subtree from the $\frac{(j-1)\cdot N}{F^2}$*-th* VN to depth l such that the tree contains all CNs allowed by the indicator vector **or** the number of nodes in the tree does not increase with an expansion to the *(l+1)-th* level.
9. Place edge connecting the $\frac{(j-1)\cdot N}{F^2}$*-th* VN to a CN chosen randomly from the set of minimum weight nodes which were added to the subtree at the last tree expansion.
10. Place remaining edges in the submatrix by circulant shift of the first placement.
11. Null the entry in the indicator vector $\mathbf{z}_j$ in the position of the chosen submatrix, preventing further placements in that submatrix.
12. **end if**
13. **end for**
14. **end for**

V. Simulations Results

The performance of the proposed QC-PEG-Root-Check LDPC codes when used in a Rayleigh block-fading channel with $F = 3$ and $F = 4$ independent fading blocks is analysed. All LDPC codes simulated have the same polynomials (7) and (8) for $R = \frac{1}{3}$ and (10) and (11) for $R = \frac{1}{4}$. Standard SPA algorithm is employed at the decoder with a maximum of 20 iterations. Following [3], [4], a maximum of 20 iterations are enough to obtain a good performance in terms of FER for block-fading channels. The Gaussian outage limit in (3) is drawn in dashed line in each figure for reference. Our proposed QC-PEG-Root-Check codes have a minimum girth of 12.

A. *Performance analysis for rate* $R = \frac{1}{3}$

In Fig. 1 it is compared the FER performance among the proposed QC-PEG-Root-Check LDPC, QC-Root-Check LDPC, and QC-PEG based LDPC [6] codes, all for $R = \frac{1}{3}$. The codeword length is $L = 900$ bits. From the results, it can be noted that the proposed QC-PEG-Root-Check LDPC code outperforms the other Root-Check based LDPC codes and it can save up to 3dB for the same FER performance. The new QC-PEG-Root-Check code performs very close to the theoretical limit. Moreover, note that all Root-Check-based codes are able to achieve the full diversity order of the channel, while (non root-check) QC-PEG LDPC codes fail to achieve full diversity.

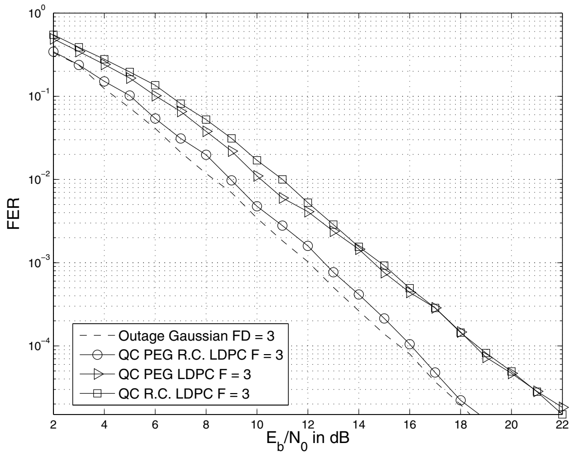

Figure 1. FER performance for the QC-Root-Check LDPC, QC-PEG-Root-Check LDPC and QC-PEG LDPC codes over a block-fading channel with $F = 3$ and $L = 900$ bits. The maximum number of iterations is 20.

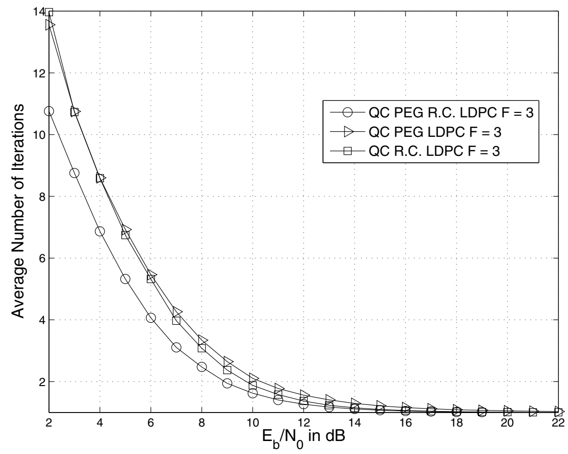

Figure 2. Average number of required iterations for the QC-PEG-Root-Check LDPC, the QC-Root-Check LDPC and the QC-PEG LDPC codes with codeword length $L = 900$ bits over a block-fading channel with $F = 3$.

Fig. 2 shows the average number of iterations required by the QC-PEG-Root-Check LDPC, QC-Root-Check LDPC and the QC-PEG LDPC codes. For the entire SNR region, in average, we can observe that the proposed QC-PEG-Root-Check LDPC code requires less decoding iterations than all other LDPC code designs. It must be mentioned that for medium to high SNR the average required number of iterations is less than 2 iterations. The average number of iterations, less than 2 at medium to high SNR, corroborates with hardware friendly capabilities of Quasi-Cyclic LDPC codes [4].

B. Performance analysis for rate $R = \frac{1}{4}$

In Fig. 3 it is compared the FER performance among the proposed QC-PEG-Root-Check LDPC, QC-Root-Check LDPC and QC-PEG LDPC codes, all for rate $R = \frac{1}{4}$. The codeword length is $L = 1600$ bits. Similar to the results for $R = \frac{1}{3}$, it can be noted that the proposed QC-PEG-Root-Check LDPC code outperforms the other designs and it can save up to 1.5dB for the same FER performance.

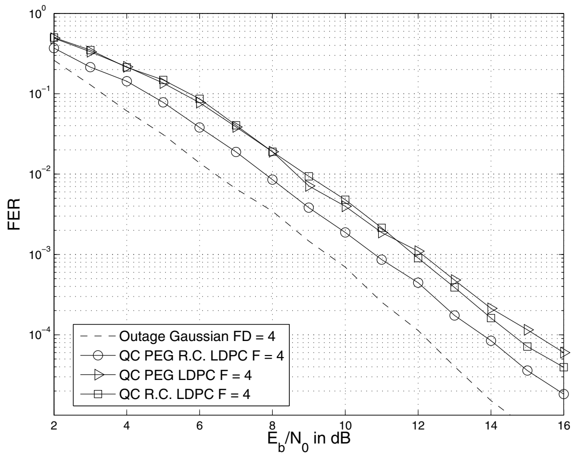

Figure 3. FER performance for the QC-Root-Check LDPC, QC-PEG-Root-Check LDPC and QC-PEG LDPC codes over a block-fading channel with $F = 4$ and $L = 1600$ bits. The maximum number of iterations is 20.

Fig. 4 shows the average number of required iterations for the QC-PEG-Root-Check LDPC, QC-Root-Check LDPC and QC-PEG LDPC codes for rate $R = \frac{1}{4}$. Note that, differently from the results obtained for $R = \frac{1}{3}$, here our proposed QC-PEG-Root-Check LDPC code requires less iterations than the other LDPC codes in the low to medium SNR regions. Moreover, note that, similar to the results for $R = \frac{1}{3}$, for medium to high SNR the average number of required iterations is less than 2.

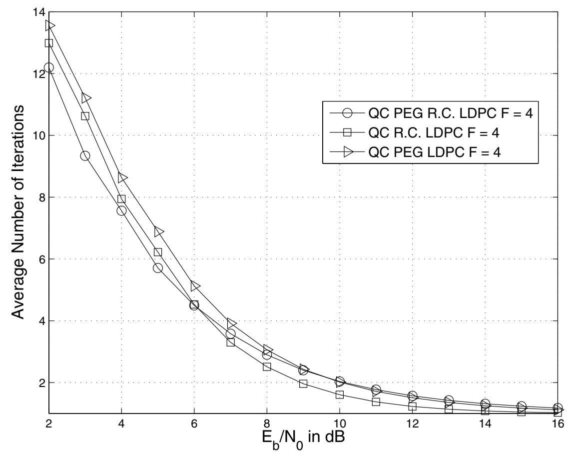

Figure 4. Average number of required iterations for QC-PEG-Root-Check LDPC, QC-Root-Check LDPC and QC-PEG LDPC codes with codeword length $L = 1600$ bits over a block-fading channel with $F = 4$.

VI. Conclusion

A novel PEG-based algorithm has been proposed to design Root-Check LDPC codes for $F > 2$ fading blocks. Based on simulations, the proposed method was compared to QC-Root-Check-LDPC codes and non Root-Check LDPC codes. The results demonstrate that the QC-PEG-Root-Check LDPC codes generated by our proposed algorithm outperform the QC-Root-Check-LDPC codes [4] in a wide range of SNR values and provide a gain of up to 3dB. In addition, as stated before, all designed Root-Check LDPC codes presented in this paper are new in the essence that no works were found for the considered channel conditions. There are some ongoing works to be done: first, is how to fire out one way to improve the QC-PEG-Root-Check LDPC codes for $R = \frac{1}{4}$ by a factor of 1dB in terms of SNR; second, after a better improvement on rate $\frac{1}{4}$ we want to extend this improvement for the other cases $F > 4$. Another aspect that we are considering is the use of iterative design methods [12], [13] to produce a new class of Root-Check LDPC codes.

Acknowledgment

This work was partially supported by Degraf & Uchôa Accounting, contabil@degrafuchoa.com.br, (Brazil).

References

[1] S. K. Kambhampati, G. Lechner, T. Chan, and L. K. Rasmussen, "Check splitting of root-check LDPC codes over ARQ block-fading channels," in *Proc. Australian Communications Theory Workshop'10*, Canberra, Australia, Feb. 2010, pp. 123–127.

[2] T. S. Rappaport, *Wireless Communications Principles and Practice*, 1st ed. New Jersey: Prentice-Hall, July 1999.

[3] J. Boutros, A. Guillen i Fabregas, E. Biglieri, and G. Zemor, "Design and analysis of low-density parity-check codes for block-fading channels," in *Proc. IEEE ITA'07*, Oct. 2007, pp. 54 – 62.

[4] Y. Li and M. Salehi, "Quasi-cyclic ldpc code design for block-fading channels," in *Proc. IEEE CISS'10*, Mar. 2010, pp. 1–5.

[5] R. G. Gallager, "Low-Density Parity-Check Codes," *IRE Trans. Inform. Theory*, no. IT-8, pp. 21–28, Jan. 1962.

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

[7] A. G. D. Uchôa, C. Healy, R. C. de Lamare, and R. D. Souza, "LDPC codes based on progressive edge growth techniques for block fading channels," in *Proc. IEEE ISWCS'11*, Aachen, Germany, Nov 2011, pp. 1–5.

[8] ——, "Design of ldpc codes based on progressive edge growth techniques for block fading channels," *IEEE Comms. Letters*, vol. 15, pp. 1221–1223, Nov. 2011.

[9] Z. Li and B. V. K. V. Kumar, "A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph," in *IEEE ASILOMARSSC'04*, Nov. 2004, pp. 1990–1994.

[10] T. Richardson and R. Urbanke, "The capacity of low density parity check codes under message passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[11] Y.-K. Lin, C.-L. Chen, Y.-C. Liao, and H.-C. Chang, "Structured ldpc codes with low error floor based on peg tanner graphs," in *IEEE ISCAS'08*, May 2008, pp. 1846 –1849.

[12] C. Healy and R. C. de Lamare, "Decoder optimised progressive edge growth algorithm," in *Proc. IEEE 73rd VTC Spring'11*, May 2011, pp. 1–5.

[13] C. T. Healy and R. C. de Lamare, "Decoder-optimised progressive edge growth algorithms for the design of ldpc codes with low error floors," *Accepted in: IEEE Comms. Letters*, 2012.