Pseudo-Codewords in LDPC Convolutional Codes

Roxana Smarandache[1]
Dept. of Math. and Stat.
San Diego State Univ.
San Diego, CA 92182
rsmarand@nd.edu

Ali Emre Pusane
Dept. of EE
Univ. of Notre Dame
Notre Dame, IN 46556
apusane@nd.edu

Pascal O. Vontobel
Dept. of EECS
MIT
Cambridge, MA 02139
pascal.vontobel@ieee.org

Daniel J. Costello, Jr.
Dept. of EE
Univ. of Notre Dame
Notre Dame, IN 46556
dcostel1@nd.edu

***Abstract*— Iterative message-passing decoders for low-density parity-check (LDPC) block codes are known to be subject to decoding failures due to so-called pseudo-codewords. These failures can cause the large signal-to-noise ratio performance of message-passing decoding to be worse than that predicted by the maximum-likelihood decoding union bound.**

In this paper we study the pseudo-codeword problem for the class of LDPC convolutional codes decoded continuously using an iterative, sliding window, message-passing decoder. In particular, for an LDPC convolutional code derived by unwrapping a quasi-cyclic LDPC block code, we show that the free pseudo-weight of the convolutional code is at least as large as the minimum pseudo-weight of the underlying quasi-cyclic code. This result parallels the well-known relationship between the free Hamming distance of convolutional codes and the minimum Hamming distance of their quasi-cyclic counterparts.

Finally, simulation results are included that show improved performance for unwrapped LDPC convolutional codes compared to their underlying quasi-cyclic codes.

I. Introduction

In this paper we discuss a class of low-density parity-check (LDPC) convolutional codes derived by unwrapping quasi-cyclic LDPC block codes. The idea of unwrapping a quasi-cyclic (QC) block code to obtain a convolutional code was first introduced in a paper by Tanner in [1], where it was shown that the free distance of the unwrapped convolutional code is lower bounded by the minimum distance of the underlying QC code. This idea was later extended in [2], [3]. More recently, a construction for LDPC convolutional codes based on QC-LDPC block codes was introduced by Tanner et al. [4], [5], and an iterative, sliding window, message-passing decoder was described. In that paper it was noted that the convolutional versions of these codes significantly outperformed their block code counterparts in the waterfall region of the bit error rate (BER) curve, even though the graphical representations of the message-passing decoders were essentially equivalent. Extensions of this construction have been given in [6], [7]. In this paper, we provide a possible explanation for this performance difference. Based on the results of [8] that relate code performance to the existence of pseudo-codewords, we examine the iterative decoding related pseudo-codeword weight spectra of QC-LDPC block codes and their associated convolutional codes. We take the approach of [8]–[10] which connects the presence of pseudo-codewords in message-passing iterative decoding and linear programming (LP) decoding.

LP decoding was introduced by Feldman, Wainwright, and Karger [11], [12] (see also [13], [14]) and consists of relaxing the optimization over the set of codewords that describes the maximum likelihood decoding problem into a computationally easier optimization over an associated “fundamental polytope”.

In order to analyze the behavior of unwrapped LDPC convolutional codes under LP decoding, we therefore need to examine the fundamental polytope/cone [8], [9] of the underlying QC-LDPC block codes. Our goal is to formulate analytical results (or at least efficient procedures) that will allow us to bound the minimum pseudo-weight of the pseudo-codewords of the block and convolutional codes.

The paper aims at addressing this question and related issues. In the following sections, we will study the connections that exist between pseudo-codewords in QC codes and pseudo-codewords in the associated convolutional codes and show that this connection mimics the connection between the codewords in QC codes and their associated convolutional codes.

The paper is structured as follows. In Sec. II we briefly discuss the well-known connection between convolutional codes and their associated QC codes, especially how codewords in the former can be used to construct codewords in the latter. In Sec. III we define the fundamental polytope/cone and the various pseudo-weights of a binary linear code. The main part of the paper is Sec. V, where we show how pseudo-codewords in unwrapped LDPC convolutional codes yield pseudo-codewords in the associated QC-LDPC codes and how this can be used to bound the minimum pseudo-weight of an LDPC convolutional code in relation to the minimum pseudo-weight of the QC-LDPC code. In Sec. VI we present some simulation results comparing LDPC convolutional codes to their associated QC-LDPC codes, and in Sec. VII we offer some conclusions. Proofs of lemmas and theorems have been sketched in the Appendix.

II. Quasi-Cyclic and Convolutional Codes

In this section we introduce the background needed for the later development of the paper. Note that all codes will be binary linear codes.

It is well known that the set of binary circulant matrices of size $r \times r$ forms a ring isomorphic to the ring of polynomials of degree less than r, $\mathbb{F}_2[X]/\langle X^r - 1\rangle$: to each circulant matrix $\mathbf{M}$ we can associate uniquely a polynomial $M(X)$ with nonzero coefficients corresponding to the nonzero entries of the first row of $\mathbf{M}$. Adding and multiplying two circulant matrices is equivalent to adding and multiplying their associated polynomials modulo $X^r - 1$. Multiplying a circulant matrix $\mathbf{M}$ from the left with a vector $\mathbf{v} \triangleq (v_0, v_1, \ldots, v_{r-1})$ corresponds to multiplying the associated polynomial $v(X) \triangleq v_0 + v_1 X + \cdots + v_{r-1}X^{r-1}$ by the polynomial $M(X)$, i.e., $M(X) \cdot v(X) \bmod (X^r - 1)$. Multiplying $\mathbf{M}$ from the right with a vector $\mathbf{v}$ is equivalent to multiplying $v(X)$ by the reciprocal polynomial $(X^r M(1/X)) \bmod (X^r - 1)$.

[1] On leave at Dept. of Math., Univ. of Notre Dame, Notre Dame, IN 46556, USA.

Let $\mathbf{H}$ be the parity-check matrix of a binary linear code C. A linear code C of length $n \triangleq r \cdot L$ is called a quasi-cyclic (QC) code with period L if the right-shift by L positions of any codeword is again a codeword. Such a code can be represented by an $rJ \times rL$ parity-check block matrix $\mathbf{H}_{\mathrm{QC}}^{(r)}$ that consists of circulant matrices of size $r \times r$. By the isomorphism mentioned above, we can associate to the matrix $\mathbf{H}_{\mathrm{QC}}^{(r)} \in \mathbb{F}_2^{rJ\times rL}$ the polynomial parity-check matrix $\mathbf{H}_{\mathrm{QC}}^{(r)}(X) \in \big(\mathbb{F}_2[X]/\langle X^r - 1\rangle\big)^{J\times L}$. The entry in the i-th row and j-th column of $\mathbf{H}_{\mathrm{QC}}^{(r)}(X)$ will be denoted by $h_{ij}(X)$. Due to the existence of this isomorphism, we can identify two descriptions, scalar and polynomial, and use either of the two depending on their usefulness.

Given the polynomial description of a QC-code, it is easy to see the natural connection that exists between quasi-cyclic codes and convolutional codes (see, e.g., [1]–[4], [6]). To any QC block code $\mathsf{C}_{\mathrm{QC}} \triangleq \mathsf{C}_{\mathrm{QC}}^{(r)}$ of length $r \cdot L$, given by a $J \times L$ polynomial matrix parity-check matrix $\mathbf{H}_{\mathrm{QC}}^{(r)}(X) = \big[h_{ij}(X)\big]_{ij}$, we can associate a (designed) rate $(L-J)/L$ convolutional code $\mathsf{C}_{\mathrm{conv}}$ given by the polynomial parity-check matrix $\mathbf{H}_{\mathrm{conv}}(D) = \big[h_{ij}(D)\big]_{ij}$. An important parameter that determines the encoding and decoding complexity of $\mathsf{C}_{\mathrm{conv}}$ is the syndrome former memory m_{s}. If we let $m_{\mathrm{s}}^{(i)}$ be the difference between the maximum degree and the minimum delay of the L polynomials in row i of $\mathbf{H}_{\mathrm{conv}}(D)$, then the syndrome former memory is given by $m_{\mathrm{s}} = \max_{1\leqslant i\leqslant J} m_{\mathrm{s}}^{(i)}$ (see [5] for details). Finally, the delay decomposition $\mathbf{H}_{\mathrm{conv}}(D) = \mathbf{H}_0 + \mathbf{H}_1 D + \mathbf{H}_2 D^2 + \ldots + \mathbf{H}_{m_{\mathrm{s}}} D^{m_{\mathrm{s}}} \in \mathbb{F}_2^{J\times L}[D]$ of the polynomial parity-check matrix $\mathbf{H}_{\mathrm{conv}}(D)$ leads to a scalar description of the convolutional code by a semi-infinite parity-check matrix that we denote $\mathbf{H}_\infty$ (see, e.g., [15]). We note that $\mathbf{H}_\infty$ can be obtained from $\mathbf{H}_{\mathrm{QC}}^{(r)}$ (and vice versa) by unwrapping, (respectively, wrapping), the last m_{s} columns, and repeating the shifted rows indefinitely of a scalar $rJ \times rL$ parity-check matrix $\mathbf{H}_{\mathrm{QC}}^{(r)}$ of the QC code $\mathrm{mod}(X^r - 1)$, $r \geqslant m_{\mathrm{s}} + 1$.

For any codeword $\mathbf{c}(D)$ with finite support in the convolutional code, its r *wrap-around*, defined as the vector $\mathbf{c}(X) \bmod (X^r - 1) \in \big(\mathbb{F}_2[X]/\langle X^r - 1\rangle\big)^L$, is a codeword in the associated QC-code, since $\mathbf{H}_{\mathrm{QC}}^{(r)}(X) \cdot \mathbf{c}(X)^{\mathsf{T}} = \mathbf{0}^{\mathsf{T}}$ in the algebra $\big(\mathbb{F}_2[X]/\langle X^r - 1\rangle\big)^L$. In addition, the Hamming weight of the two codewords is linked through the following inequality: $w_{\mathrm{H}}\big(\mathbf{c}(X) \bmod (X^r - 1)\big) \leqslant w_{\mathrm{H}}(\mathbf{c}(D))$, which gives the inequality [1], [2]

$$d_{\min}(\mathsf{C}_{\mathrm{QC}}^{(r)}) \leqslant d_{\mathrm{free}}(\mathsf{C}_{\mathrm{conv}}), \text{ for all } r \geqslant 1.$$

Moreover, $d_{\min}(\mathsf{C}_{\mathrm{QC}}^{(r)}) \leqslant d_{\min}(\mathsf{C}_{\mathrm{QC}}^{(2r)}) \leqslant d_{\min}(\mathsf{C}_{\mathrm{QC}}^{(4r)}) \leqslant \ldots$, for all $r \geqslant 1$, and $\lim_{r\to\infty} d_{\min}(\mathsf{C}_{\mathrm{QC}}^{(r)}) = d_{\mathrm{free}}(\mathsf{C}_{\mathrm{conv}})$. It is well known that to each matrix $\mathbf{H}$ we can associate a Tanner graph with $\mathbf{H}$ its incidence matrix. The Tanner graphs of this tower of QC codes are also related, as the larger graphs are finite covers of the smaller graphs. The above relationship between minimum distances of the above codes in the tower is easily verified in the graph language, since a codeword $\mathbf{c}(\mathbf{X})$ of a larger graph, say $\mathsf{C}_{\mathrm{QC}}^{(4r)}$, when projected to the graphs of $\mathsf{C}_{\mathrm{QC}}^{(r)}$ and $\mathsf{C}_{\mathrm{QC}}^{(2r)}$, by the formula $\mathbf{c}(X) \bmod (X^r - 1)$, resp. $\mathbf{c}(\mathbf{X}) \bmod (X^{2r} - 1)$, gives again codewords. Finally, the graph of the associated convolutional code is an infinite (but usually not universal[2]) cover of the graphs in the tower.

III. The Fundamental Cone

In our pseudo-codeword analysis we mainly take the approach of [8]–[10] which connects the presence of pseudo-codewords in message-passing iterative decoding and linear programming (LP) decoding. In this section we repeat the main definitions concerning pseudo-codewords and pseudo-weights [8], [9], [12] in a linear programming setting. We let $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_{++}$ be the set of real numbers, the set of non-negative real numbers, and the set of positive real numbers, respectively.

Definition 3.1 ([8], [9], [11], [12]): Let $\mathbf{H}$ be a binary matrix of size $m \times n$, let $\mathcal{J} \triangleq \{0, \ldots, n-1\}$ be the set of column indices, and let $\mathcal{I} \triangleq \{0, \ldots, m-1\}$ be the set of row indices of $\mathbf{H}$. For each $i \in \mathcal{I}$, we let $\mathcal{J}_i \triangleq \big\{j \in \mathcal{J} \mid h_{ij} = 1\big\}$. The *fundamental polytope* $\mathcal{P} \triangleq \mathcal{P}(\mathbf{H})$ of $\mathbf{H}$ is defined as [8], [9]

$$\mathcal{P} \triangleq \bigcap_{i=1}^{m} \mathrm{conv}(\mathsf{C}_i) \text{ with } \mathsf{C}_i \triangleq \Big\{\mathbf{x} \in \{0,1\}^n \,\big|\, \mathbf{r}_i \mathbf{x}^{\mathsf{T}} = 0 \bmod 2\Big\},$$

where $\mathbf{r}_i$ is the i-th row of $\mathbf{H}$. The *fundamental cone* $\mathcal{K} \triangleq \mathcal{K}(\mathbf{H})$ of $\mathbf{H}$ is defined as the conic hull of the fundamental polytope, i.e., the part of the fundamental polytope $\mathcal{P}$ around the vertex $\mathbf{0}$ and stretched to infinity. Note that if $\boldsymbol{\omega} \in \mathcal{K}(\mathbf{H})$, then also $\alpha \cdot \boldsymbol{\omega} \in \mathcal{K}(\mathbf{H})$ for any real $\alpha > 0$. Moreover, for any $\boldsymbol{\omega} \in \mathcal{K}(\mathbf{H})$, there exists an $\alpha > 0$ (in fact, a whole interval of α's) such that $\alpha \cdot \boldsymbol{\omega} \in \mathcal{P}(\mathbf{H})$.

Vectors in $\mathcal{P}(\mathbf{H})$ are called *pseudo-codewords of* $\mathbf{H}$. Actually, we will call any vector in $\mathcal{K}(\mathbf{H})$ a pseudo-codeword and two pseudo-codewords that are equal up to a positive scaling constant will be considered to be equivalent. Clearly, pseudo-codewords are not codewords in general, but codewords are pseudo-codewords. □

A computationally useful description of the fundamental cone is given by the following sets of linear inequalities [9], [12]:

$$\mathcal{K} = \left\{ \boldsymbol{\omega} \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \forall j \in \mathcal{J} : 0 \leqslant \omega_j \text{ and} \\ \forall i \in \mathcal{I}, \forall j' \in \mathcal{J}_i : \omega_{j'} - \sum_{j \in (\mathcal{J}_i \setminus \{j'\})} \omega_j \leqslant 0 \end{array} \right\}.$$

In other words, there exists a matrix $\mathbf{K}$ such that $\boldsymbol{\omega} \in \mathcal{K}$ if and only if $\mathbf{K}\boldsymbol{\omega}^{\mathsf{T}} \geqslant \mathbf{0}^{\mathsf{T}}$.

For a convolutional code defined by $\mathbf{H}_{\mathrm{conv}}(D)$, this can be translated into polynomial terms: there is a matrix $\mathbf{K}_{\mathrm{conv}}(D)$ such that $\boldsymbol{\omega}(D) \in \mathcal{K}(\mathbf{H}_{\mathrm{conv}}(D))$ if and only if $\mathbf{K}_{\mathrm{conv}}(D)\boldsymbol{\omega}(D)^{\mathsf{T}} \geqslant \mathbf{0}^{\mathsf{T}}$.[3]

Example 3.2: Let

$$\mathbf{H}_{\mathrm{conv}}(D) \triangleq \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & D & D^2 & D^3 \\ 1 & D^4 & D^3 & D^2 \end{bmatrix}.$$

be a polynomial parity-check matrix of a rate-$1/4$ convolutional code. The matrix $\mathbf{K}_{\mathrm{conv}}(D)$ can be chosen to be a 16×4 polynomial matrix and can be easily derived from $\mathbf{H}_{\mathrm{conv}}(D)$. With this, one can e.g. check that the vector $\boldsymbol{\omega}(D) \triangleq (3D^2 + D^3, 4D + D^2, 3 + D + 4D^2 + D^3, 3 + 4D + D^2)$

[2] Note that the universal cover of a graph is a tree.

[3] An inequality of the form $\mathbf{a}(D) \geqslant \mathbf{0}$ is to be understood as follows: every coefficient of each polynomial component of $\mathbf{a}(D)$ is non-negative.

is a pseudo-codeword for the convolutional code because $\mathbf{K}_{\mathrm{conv}}(D)\boldsymbol{\omega}(D)^{\mathsf{T}} \geqslant \mathbf{0}^{\mathsf{T}}$ holds.

Similarly, for a QC code defined by $\mathbf{H}_{\mathrm{QC}}^{(r)}(X)$, there is a polynomial matrix $\mathbf{K}_{\mathrm{QC}}^{(r)}(X)$ such that $\boldsymbol{\omega}(X) \in \mathcal{K}\big(\mathbf{H}_{\mathrm{QC}}^{(r)}(X)\big)$ if and only if $\mathbf{K}_{\mathrm{QC}}^{(r)}(X)\boldsymbol{\omega}(X)^{\mathsf{T}} \bmod (X^r - 1) \geqslant \mathbf{0}^{\mathsf{T}}$.[4]

Lemma 3.3: Let $\boldsymbol{\omega}(D)$ be a pseudo-codeword in the convolutional code defined by $\mathbf{H}_{\mathrm{conv}}(D)$, i.e., $\boldsymbol{\omega}(D) \in \mathcal{K}(\mathbf{H}_{\mathrm{conv}}(D))$. Assuming that the largest degree in $\mathbf{H}_{\mathrm{conv}}(D)$ is smaller than some non-negative integer r, then the r wrap-around polynomial vector of $\boldsymbol{\omega}(D)$ is a pseudo-codeword in the associated QC-code defined by $\mathbf{H}_{\mathrm{QC}}^{(r)}(X)$, i.e., $\boldsymbol{\omega}(X) \bmod (X^r - 1) \in \mathcal{K}\big(\mathbf{H}_{\mathrm{QC}}^{(r)}(X)\big)$.

PROOF: For any r that fulfills the assumption we have $\mathbf{H}_{\mathrm{QC}}^{(r)}(X) = \mathbf{H}_{\mathrm{conv}}(X)$. Moreover, it can be verified that $\mathbf{K}_{\mathrm{QC}}^{(r)}(X) = \mathbf{K}_{\mathrm{conv}}(X)$ holds. We know that $\boldsymbol{\omega}(D)$ fulfills $\mathbf{K}_{\mathrm{conv}}(D)\boldsymbol{\omega}(D)^{\mathsf{T}} \geqslant \mathbf{0}^{\mathsf{T}}$ and therefore

$$\begin{aligned}
&\overset{(*)}{\Rightarrow} \mathbf{K}_{\mathrm{conv}}(D)\boldsymbol{\omega}(D)^{\mathsf{T}} \bmod (D^r - 1) \geqslant \mathbf{0}^{\mathsf{T}},\\
&\Rightarrow \mathbf{K}_{\mathrm{conv}}(X)\boldsymbol{\omega}(X)^{\mathsf{T}} \bmod (X^r - 1) \geqslant \mathbf{0}^{\mathsf{T}},\\
&\Rightarrow \mathbf{K}_{\mathrm{QC}}^{(r)}(X)\boldsymbol{\omega}(X)^{\mathsf{T}} \bmod (X^r - 1) \geqslant \mathbf{0}^{\mathsf{T}},\\
&\Rightarrow \mathbf{K}_{\mathrm{QC}}^{(r)}(X)\left(\boldsymbol{\omega}(X)^{\mathsf{T}} \bmod (X^r - 1)\right) \bmod (X^r - 1) \geqslant \mathbf{0}^{\mathsf{T}},
\end{aligned}$$

which is the desired result.[5] □

Example 3.4: Let $r \triangleq 5$ and let $\mathbf{H}_{\mathrm{QC}}^{(5)}(X)$ be obtained from $\mathbf{H}_{\mathrm{conv}}(D)$ in Ex. 3.2: $\mathbf{H}_{\mathrm{QC}}^{(5)}(X) = \mathbf{H}_{\mathrm{conv}}(X)$. It can be verified that the QC block code of length $n = 20$ defined by $\mathbf{H}_{\mathrm{QC}}^{(5)}(X)$ has $\boldsymbol{\omega}(X) \triangleq (3X^2 + X^3, 4X + X^2, 3 + X + 4X^2 + X^3, 3 + 4X + X^2)$ as a pseudo-codeword.

IV. Minimum Pseudo-weights

The fundamental cone is *independent* of the specific memoryless binary-input channel through which we are transmitting; however, the influence of a pseudo-codeword on LP or iterative decoding behavior is measured by a *channel-dependent* pseudo-weight defined in the following.

Definition 4.1 ([8], [9], [11], [12], [16], [17]): Let $\boldsymbol{\omega} = (\omega_0, \ldots, \omega_{n-1})$ be a nonzero vector in $\mathbb{R}_+^n$. The AWGNC-pseudo-weight and the BEC-pseudo-weight of the vector $\boldsymbol{\omega}$ are defined to be, respectively,

$$w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega}) \triangleq \frac{\|\boldsymbol{\omega}\|_1^2}{\|\boldsymbol{\omega}\|_2^2}, \qquad w_{\mathrm{p}}^{\mathrm{BEC}}(\boldsymbol{\omega}) = |\operatorname{supp}(\boldsymbol{\omega})|,$$

where $\|\boldsymbol{\omega}\|_1$ and $\|\boldsymbol{\omega}\|_2$ are the 1-norm, resp. 2-norm, of $\boldsymbol{\omega}$. In order to define the BSC-pseudo-weight $w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega})$, we let $\boldsymbol{\omega}'$ be the vector of length n with the same components as $\boldsymbol{\omega}$ but in decreasing order. Now let

$$\begin{aligned}
&f(\xi) \triangleq \omega_i' \quad (i < \xi \leqslant i+1,\ 0 < \xi \leqslant n),\\
&F(\xi) \triangleq \int_0^{\xi} f(\xi')\, \mathrm{d}\,\xi', \quad \text{and} \quad e \triangleq F^{-1}\left(\frac{F(n)}{2}\right).
\end{aligned}$$

Then the BSC-pseudo-weight $w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega})$ is $w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega}) \triangleq 2e$ if $\boldsymbol{\omega} \neq \mathbf{0}$. Finally, the fractional and max-fractional weight of a vector $\boldsymbol{\omega} \in \mathbb{R}_+^n$ are defined to be, respectively,

$$w_{\mathrm{frac}}(\boldsymbol{\omega}) = \|\boldsymbol{\omega}\|_1, \qquad w_{\mathrm{max-frac}}(\boldsymbol{\omega}) \triangleq \frac{\|\boldsymbol{\omega}\|_1}{\|\boldsymbol{\omega}\|_\infty}.$$

For $\boldsymbol{\omega} = \mathbf{0}$ we define all of the above pseudo-weights, fractional weights, and max-fractional weights to be zero. (For a motivation of these definitions, see the above references.) □

Definition 4.2 ([8], [9], [11], [12]): Important quantities in characterizing the LP decoding performance are the minimum AWGNC, BSC, and BEC pseudo-weights, and the minimum fractional and max-fractional weights, which are, respectively,

$$w_{\mathrm{p}}^{\min}(\mathbf{H}) \triangleq \min_{\boldsymbol{\omega} \in \mathcal{V}(\mathcal{P}(\mathbf{H}))\setminus\{\mathbf{0}\}} w_{\mathrm{p}}(\boldsymbol{\omega})$$

where $\mathcal{V}(\mathcal{P}(\mathbf{H}))\setminus\{\mathbf{0}\}$ is the set of all non-zero vertices of the fundamental polytope $\mathcal{P}(\mathbf{H})$, and the pseudo-weights are the appropriate ones for each channel. □

Computing these values can be quite challenging, since the task of finding the set of vertices of $\mathcal{P}(\mathbf{H})$ can be very complex. However in the case of four of the above pseudo-weights, the minimum AWGNC, BSC, and BEC pseudo-weights and minimum max-fractional weights, there is a computationally easier description as:

$$w_{\mathrm{p}}^{\min}(\mathbf{H}) = \min_{\boldsymbol{\omega} \in \mathcal{K}(\mathbf{H})\setminus\{\mathbf{0}\}} w_{\mathrm{p}}(\boldsymbol{\omega}),$$

with the appropriate pseudo-weight of each of the above channels, see, e.g. [9]. (Note that there is no such statement for the minimum fractional weight.)

A complete characterization of LP decoding is given by so-called minimal pseudo-codewords. However, for longer codes it is usually computationally too demanding to obtain the whole list of minimal pseudo-codewords; therefore, one often restricts oneself to looking for minimal pseudo-codewords with low pseudo-weight [18]–[20].

In what follows, we compare the minimum pseudo-weight and minimum max-fractional weight of a QC block code to those of its corresponding convolutional code.

V. Pseudo-Weights in LDPC-QC and LDPC Convolutional Codes

We saw in the last section that in order to analyze the minimum pseudo-weight and minimum max-fractional weight, it is sufficient to analyze the weights of the non-zero vectors in the fundamental cone. Throughout this section w.l.o.g. all pseudo-codewords $\boldsymbol{\omega}(D)$ under investigation are assumed to have finite support.

Theorem 5.1: For the AWGNC, BEC, and BSC pseudo-weights, if $\boldsymbol{\omega}(D) \in \mathcal{K}(\mathbf{H}_{\mathrm{conv}}(D))$, then

$$w_{\mathrm{p}}\big(\boldsymbol{\omega}(X) \bmod (X^r - 1)\big) \leqslant w_{\mathrm{p}}(\boldsymbol{\omega}(D)).$$

Therefore,

$$w_{\mathrm{p}}^{\min}(\mathbf{H}_{\mathrm{QC}}^{(r)}(X)) \leqslant w_{\mathrm{p}}^{\min}(\mathbf{H}_{\mathrm{conv}}(D)).$$

PROOF: See App. A. □

Th. 5.1 implies that low pseudo-weight vectors in the block code may correspond to higher pseudo-weight vectors in the convolutional code, but the opposite is not possible. This suggests that the pseudo-codewords in the block code that

[4] In the following, an expression of the form $\mathbf{a}(X) \bmod (X^r - 1)$ will be understood as follows: $\mathbf{a}(X) \bmod (X^r - 1)$ is a reduced vector such that all components are polynomials where only the coefficients of $X^0, X^1, \ldots X^{r-1}$ are allowed to be non-zero.

[5] Note that step $(*)$ depends on special properties of $D^r - 1$, i.e. this step does in general not work when the modulo-operation is with respect to an arbitrary polynomial.

result in decoding failures may not cause such failures in the convolutional code, thereby resulting in improved performance for the convolutional code at low-to-moderate signal-to-noise ratios (SNRs). Further, it is not difficult to adapt Th. 5.1 such that similar conclusions can be drawn with respect to a QC block code with the same polynomial parity-check matrix, but with larger circulant size r' multiple of r. In fact, most QC block codes with the same structure but a larger circulant size r', even if not a multiple of r, behave according to Th. 5.1.

A similar bound also holds for the max-fractional weight, as shown in the next theorem:

Theorem 5.2: If $\boldsymbol{\omega}(D) \in \mathcal{K}(\mathbf{H}_{\text{conv}}(D))$, then

$$w_{\max-\text{frac}}\big(\boldsymbol{\omega}(X) \bmod (X^r - 1)\big) \leqslant w_{\max-\text{frac}}(\boldsymbol{\omega}(D)).$$

Therefore,

$$w^{\min}_{\max-\text{frac}}(\mathbf{H}^{(r)}_{\text{QC}}(X)) \leqslant w^{\min}_{\max-\text{frac}}(\mathbf{H}_{\text{conv}}(D)).$$

PROOF: See App. B. □

Before comparing the minimum fractional weight of the convolutional and QC codes, we emphasize that these values must be computed over the set of nonzero pseudo-codewords that are vertices of the fundamental polytope. Indeed, although for any $\boldsymbol{\omega}(D) \in \mathcal{V}(\mathcal{P}(\mathbf{H}_{\text{conv}}(D))) \setminus \{\mathbf{0}\}$, we have $\|\boldsymbol{\omega}(D)\|_1 = \|\boldsymbol{\omega}(X) \bmod (X^r - 1)\|_1$, and hence $w_{\text{frac}}(\boldsymbol{\omega}(X) \bmod (X^r - 1)) = w_{\text{frac}}(\boldsymbol{\omega}(D))$, comparing the minimum fractional weight of the convolutional and the QC code is not an easy task, because a vertex pseudo-codeword in the convolutional code might not map into a vertex pseudo-codeword in the QC code. The following result, however, can be established.

Theorem 5.3: Assume that we transmit data over a BSC using the convolutional code and that bit flips happen at positions $\mathcal{E}_{\text{conv}} \subseteq \{\mathcal{I}(\mathbf{H}_{\text{conv}}(D))\}$. If $|\mathcal{E}_{\text{conv}}| < \frac{1}{2} w^{\min}_{\text{frac}}(\mathbf{H}^{(r)}_{\text{QC}}(X))$, then LP decoding decides for the correct codeword.

PROOF: Omitted. □

VI. SIMULATION RESULTS

In the previous sections, we showed that better pseudo-weight properties result when we unwrap a QC block code to form a convolutional code. This suggests that an LDPC convolutional code constructed in this fashion will perform better than the underlying QC-LDPC block code when decoded by local message-passing algorithms. In this section we use computer simulations on an AWGN channel to examine the performance of these codes.

We consider a rate $R = 2/5 = 0.40$ LDPC convolutional code with syndrome former memory $m_{\text{s}} = 21$, constructed by unwrapping a [155,64] (3,5)-regular QC-LDPC block code, i.e., a [155,64] code whose parity-check matrix contains 3 ones per column and 5 ones per row with circulant size $r = 31$. We also consider two larger QC-LDPC block codes, a [200,82] code and a [240,98] code, with parity-check matrices of increasing circulant sizes ($r = 40$ and $r = 48$), while keeping the same structure within each $r \times r$ circulant. (Note that each of the three block codes has rate slightly greater than 0.40.)

A sliding window message-passing decoder was used to decode the convolutional code (see, e.g., [5]). Conventional LDPC block code decoders were employed to decode the QC-LDPC block codes. All decoders were allowed a maximum of 50 iterations, where the block code decoders employed a syndrome-check based stopping rule. The resulting BER performance of these codes is shown in Fig. 1.

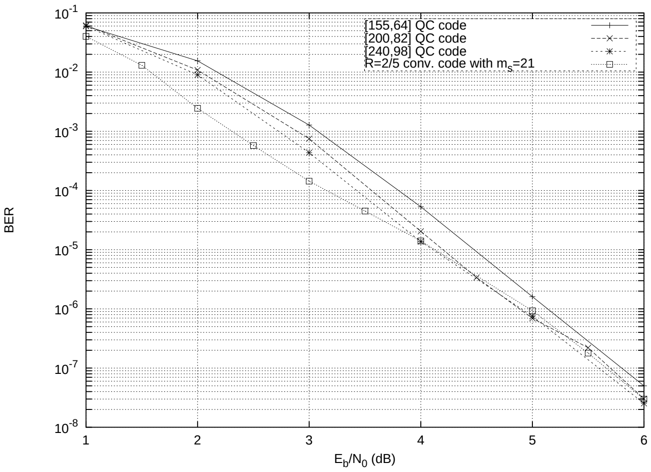

Fig. 1. The performance of a convolutional LDPC code and three associated (3,5)-regular QC-LDPC block codes.

We note that in the low-to-moderate SNR region, where the complete pseudo-weight spectrum plays an important role, the unwrapped LDPC convolutional code performs between 0.5dB and 1.0dB better than the associated QC-LDPC block codes. Also, as the circulant size increases, the BER performance of the block codes approaches that of the convolutional code. (It has been shown in [5] that similar performance differences are also observed for much larger QC-LDPC block codes and their corresponding LDPC convolutional codes.) The BER performance curves suggest that the pseudo-weight spectrum of the LDPC convolutional code is “thinner” than that of the associated QC-LDPC block codes, and that, as the circulant size becomes larger, the spectra of the block codes approaches that of the convolutional code. These results are consistent with the implications drawn from Theorem 5.1 in the previous section.

VII. CONCLUSIONS

For an LDPC convolutional code derived by unwrapping an LDPC-QC block code, we have shown that the free pseudo-weight of the convolutional code is at least as large as the minimum pseudo-weight of the underlying QC code. This result suggests that the pseudo-weight spectrum of the convolutional code is ”thinner” than that of the block code. This difference in the weight spectra leads to improved BER performance at low-to-moderate SNRs for the convolutional code, a conclusion supported by the simulation results presented in Figure 1.

APPENDIX

A. Proof of Theorem 5.1

In the following, we have to to analyze separately the AWGNC, BEC, and BSC pseudo-weights of $\boldsymbol{\omega}(D)$ and of its r wrap-around $\boldsymbol{\omega}(X) \bmod (X^r - 1)$. Let t be the length of the scalar vector $\boldsymbol{\omega}$ associated to $\boldsymbol{\omega}(D)$.

Case 1 (AWGNC): Since $\|\boldsymbol{\omega}(D)\|_1 = \|\boldsymbol{\omega}(X) \bmod (X^r - 1)\|_1$ and $\|\boldsymbol{\omega}(X) \bmod (X^r - 1)\|_2^2 =$

$$\sum_{i=0}^{r-1} \sum_{k=0}^{L-1} \left(\sum_{j=0}^{\lfloor t/r \rfloor} \omega_{i+jr,k} \right)^2 \geqslant \sum_{i=0}^{r-1} \sum_{k=0}^{L-1} \sum_{j=0}^{\lfloor t/r \rfloor} \omega^2_{i+jr,k} = \|\boldsymbol{\omega}(D)\|_2^2,$$

we obtain $w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega}(D)) \geqslant w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega}(X) \bmod (X^r - 1))$.

Case 2 (BEC): Since the components of the vector $\boldsymbol{\omega}(X) \bmod (X^r - 1)$ are obtained by adding in $\mathbb{R}_+$ certain non-negative components of $\boldsymbol{\omega}(D)$, it follows that $|\operatorname{supp} \boldsymbol{\omega}(D)| \geqslant |\operatorname{supp}(\boldsymbol{\omega}(X) \bmod (X^r - 1))|$. We obtain

$$w_{\mathrm{p}}^{\mathrm{BEC}}(\boldsymbol{\omega}(D)) \geqslant w_{\mathrm{p}}^{\mathrm{BEC}}(\boldsymbol{\omega}(X) \bmod (X^r - 1)).$$

Case 3 (BSC): In order to compare the BSC-pseudo-weight of the two vectors, we first need to arrange them in decreasing order. Let $M_0 \geqslant M_1 \geqslant \ldots \geqslant M_{t-1}$, and $m_0 \geqslant m_1 \geqslant \ldots \geqslant m_{rL-1}$ be a listing of the components of $\boldsymbol{\omega}(D)$ and $\boldsymbol{\omega}(X) \bmod (X^r - 1)$ respectively, in decreasing order. Since $\|\boldsymbol{\omega}(D)\|_1 = \|\boldsymbol{\omega}(X) \bmod (X^r - 1)\|_1$, we obtain that $\frac{\|\boldsymbol{\omega}(D)\|_1}{2} = \frac{\|\boldsymbol{\omega}(X) \bmod (X^r-1)\|_1}{2}$. Let M be this quantity and so $\sum_{i=0}^{t-1} M_i = \sum_{i=0}^{rL-1} m_i = 2M$. Hence the two sequences of positive integers form two partitions, λ and μ, respectively, of $2M$. We fill the shorter partition μ with $t-rL$ zeros in order to have both partitions of the same length t. It is enough to show that $\sum_{i=0}^{l-1} M_i \leqslant \sum_{i=0}^{l-1} m_i$, for all $l = 1, 2, \ldots, t$, i.e., that μ majorizes λ [21].

We show first that $m_0 \geqslant M_0$. Suppose the contrary, $m_0 < M_0$. Since $m_i \leqslant m_0$ for all $i = \overline{0, t-1}$, we obtain that $m_i < M_0$ for all $i = \overline{0, t-1}$. But $m_i, i = \overline{0, rL-1}$ was obtained by adding over $\mathbb{R}_+$ a certain subset of the set $\{M_j, j = \overline{0, t-1}\}$. So there should be at least one m_l that has M_0 in its composition, and hence $m_l \geqslant M_0$. This is a contradiction, from which we obtain $m_0 \geqslant M_0$.

We finish the proof by induction. Namely, we want to shown that from $\sum_{i=0}^{j-1} M_i \leqslant \sum_{i=0}^{j-1} m_i$ for some $j \in \{1, t-1\}$ it follows that $\sum_{i=0}^{j} M_i \leqslant \sum_{i=0}^{j} m_i$. If $M_j \leqslant m_j$ then this induction step clearly holds. So, assume that $M_j > m_j$. Since $m_{t-1} \leqslant \ldots \leqslant m_j < M_j \leqslant M_{j-1} \leqslant \ldots \leqslant M_0$, we can deduce that m_j, and in fact all m_i with $j \leqslant i \leqslant t-1$, cannot contain any M_i with $0 \leqslant i \leqslant j$ in its composition. Hence all possible M_i, $0 \leqslant i \leqslant j$, have occurred in the composition of m_i, for $0 \leqslant i \leqslant j-1$, which gives $\sum_{i=0}^{j} m_i \geqslant \sum_{i=0}^{j-1} m_i \geqslant \sum_{i=0}^{j} M_i$. This proves that μ majorizes λ and we obtain that

$$w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega}(D)) \geqslant w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega}(X) \bmod (X^r - 1)).$$

B. Proof of Theorem 5.2

We have $\|\boldsymbol{\omega}(D)\|_1 = \|\boldsymbol{\omega}(X) \bmod (X^r - 1)\|_1$ and

$$\|\boldsymbol{\omega}(X) \bmod (X^r - 1)\|_\infty = \max_{i=0}^{r-1} \max_{k=0}^{L-1} \sum_{j=0}^{\lfloor t/r \rfloor} \omega_{i+jr,k}$$

$$\geqslant \max_{i=0}^{r-1} \max_{k=0}^{L-1} \max_{j=0}^{\lfloor t/r \rfloor} \omega_{i+jr,k} \geqslant \|\boldsymbol{\omega}(D)\|_\infty,$$

which leads to $w_{\mathrm{max-frac}}(\boldsymbol{\omega}(X) \bmod (X^r - 1)) \leqslant w_{\mathrm{max-frac}}(\boldsymbol{\omega}(D))$. It now follows that

$$w_{\mathrm{max-frac}}^{\mathrm{min}}(\mathbf{H}_{\mathrm{QC}}^{(\mathrm{r})}) \leqslant w_{\mathrm{max-frac}}^{\mathrm{min}}(\mathbf{H}_{\mathrm{conv}}).$$

Acknowledgments

This work was partially supported by NSF Grant CCR-0205310 (R. S., A. E. P., and D. J. C.), NASA Grant NNG-05GH73G (A. E. P. and D. J. C.), and NSF Grants CCF-0514801 and CCF-0515109 and by HP through the MIT/HP Alliance (P. O. V).

References

[1] R. M. Tanner, "Convolutional codes from quasi-cyclic codes: a link between the theories of block and convolutional codes," *University of California, Santa Cruz, Tech Report UCSC-CRL-87-21*, Nov. 1987.

[2] Y. Levy and D. J. Costello, Jr., "An algebraic approach to constructing convolutional codes from quasi-cyclic codes," in *Coding and Quantization (Piscataway, NJ, 1992)*, vol. 14 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pp. 189–198, Providence, RI: Amer. Math. Soc., 1993.

[3] M. Esmaeili, T. A. Gulliver, N. P. Secord, and S. A. Mahmoud, "A link between quasi-cyclic codes and convolutional codes," *IEEE Trans. on Inform. Theory*, vol. IT–44, no. 1, pp. 431–435, 1998.

[4] A. Sridharan, D. J. Costello, Jr., D. Sridhara, T. E. Fuja, and R. M. Tanner, "A construction for low density parity check convolutional codes based on quasi-cyclic block codes," in *Proc. IEEE Intern. Symp. on Inform. Theory*, (Lausanne, Switzerland), p. 481, June 30–July 5 2002.

[5] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, Jr., "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. on Inform. Theory*, vol. IT–50, pp. 2966–2984, Dec. 2004.

[6] A. Sridharan and D. J. Costello, Jr., "A new construction for low density parity check convolutional codes," in *Proc. IEEE Inform. Theory Workshop*, (Bangalore, India), p. 212, Oct. 20-25 2002.

[7] R. Smarandache and P. O. Vontobel, "On regular quasi-cyclic LDPC codes from binomials," in *Proc. IEEE Intern. Symp. on Inform. Theory*, (Chicago, IL, USA), p. 274, June 27–July 2 2004.

[8] R. Koetter and P. O. Vontobel, "Graph covers and iterative decoding of finite-length codes," in *Proc. 3rd Intern. Symp. on Turbo Codes and Related Topics*, (Brest, France), pp. 75–82, Sept. 1–5 2003.

[9] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," *submitted to IEEE Trans. Inform. Theory, available online under* `http://www.arxiv.org/abs/cs.IT/0512078`, Dec. 2005.

[10] P. O. Vontobel and R. Koetter, "On the relationship between linear programming decoding and min-sum algorithm decoding," in *Proc. Intern. Symp. on Inform. Theory and its Applications (ISITA)*, (Parma, Italy), pp. 991–996, Oct. 10–13 2004.

[11] J. Feldman, *Decoding Error-Correcting Codes via Linear Programming*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2003.

[12] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. on Inform. Theory*, vol. IT–51, no. 3, pp. 954–972, 2005.

[13] J. Feldman, D. R. Karger, and M. J. Wainwright, "Linear programming-based decoding of turbo-like codes and its relation to iterative approaches," in *Proc. 40th Allerton Conf. on Communications, Control, and Computing*, (Allerton House, Monticello, Illinois, USA), October 2–4 2002.

[14] M. Wainwright, T. Jaakkola, and A. Willsky, "MAP estimation via agreement on trees: Message-passing and linear programming approaches," in *Proc. 40th Allerton Conf. on Communications, Control, and Computing*, (Allerton House, Monticello, Illinois, USA), October 2–4 2002.

[15] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Codes*. New York, NY: IEEE Press, 1999.

[16] N. Wiberg, *Codes and Decoding on General Graphs*. PhD thesis, Linköping University, Sweden, 1996.

[17] G. D. Forney, Jr., R. Koetter, F. R. Kschischang, and A. Reznik, "On the effective weights of pseudo-codewords for codes defined on graphs with cycles," in *Codes, Systems, and Graphical Models (Minneapolis, MN, 1999)* (B. Marcus and J. Rosenthal, eds.), vol. 123 of *IMA Vol. Math. Appl.*, pp. 101–112, Springer Verlag, New York, Inc., 2001.

[18] P. O. Vontobel, R. Smarandache, N. Kiyavash, J. Teutsch, and D. Vukobratovic, "On the minimal pseudo-codewords of codes from finite geometries," in *Proc. IEEE Intern. Symp. on Inform. Theory*, (Adelaide, Australia), pp. 980–984, Sep. 4–9 2005. Available online under `http://www.arxiv.org/abs/cs.IT/0508019`.

[19] R. Smarandache and P. O. Vontobel, "Pseudo-codeword analysis of Tanner graphs from projective and Euclidean planes," *submitted to IEEE Trans. Inform. Theory, available online under* `http://www.arxiv.org/abs/cs.IT/0602089`, Feb. 2006.

[20] P. O. Vontobel and R. Koetter, "Lower bounds on the minimum pseudo-weight of linear codes," in *Proc. IEEE Intern. Symp. on Inform. Theory*, (Chicago, IL, USA), p. 70, June 27–July 2 2004.

[21] A. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications*. San Diego, CA: Academic Press, 1979.