

On Deriving Good LDPC Convolutional Codes from QC LDPC Block Codes

Ali Emre Pusane

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

Roxana Smarandache

Dept. of Math. and Stat.
San Diego State Univ.
San Diego, CA 92182
rsmarand@math.sdsu.edu

Pascal O. Vontobel

Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304
pascal.vontobel@ieee.org

Daniel J. Costello, Jr.

Dept. of EE
Univ. of Notre Dame
Notre Dame, IN 46556
costello.2@nd.edu

Abstract—In this paper we study the iterative decoding behavior of time-invariant and time-varying LDPC convolutional codes derived by unwrapping QC LDPC block codes. In particular, for a time-varying LDPC convolutional code, we show that the minimum pseudo-weight of the convolutional code is at least as large as the minimum pseudo-weight of the underlying QC code. We also prove that the unwrapped convolutional codes have fewer short cycles than the QC codes. These results taken together lead to improved BER performance in the low-to-moderate SNR region, where the decoding behavior is influenced by the complete pseudo-codeword spectra and by the Tanner graph cycle histogram, with the time-varying convolutional codes outperforming both the underlying QC block codes and their time-invariant convolutional counterparts.

Keywords — Low-density parity-check codes, convolutional codes, pseudo-codewords, cycles, iterative decoding.

I. INTRODUCTION

The idea of deriving a convolutional code from a quasi-cyclic (QC) block code was first introduced in a paper by Tanner [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].

The same idea was then applied to deriving LDPC convolutional codes based on QC LDPC block codes in [4], [5], and an iterative, sliding window, message-passing decoder was described. Even though the two LDPC codes had essentially the same graphical representations, it was observed that the convolutional code significantly outperformed its block code counterpart in the waterfall region of the bit error rate (BER) curve. In [7], we proposed a possible explanation for the improved performance based on the differences between the pseudo-codeword spectra of the LDPC convolutional codes and the underlying QC LDPC block codes. We showed that the minimum free pseudo-weight of the convolutional code is at least as large as the minimum pseudo-weight of the underlying QC code. Based on this relation, we conjectured that the pseudo-weight spectrum of the convolutional code was “thinner” than that of the block code, resulting in improved BER performance at low-to-moderate signal-to-noise ratios (SNRs).

In [7], we considered the method of obtaining convolutional codes from QC block codes presented in [1], which results in

a time-invariant convolutional code structure. In [8], Jiménez-Feltström and Zigangirov proposed a method of deriving periodically time-varying LDPC convolutional codes from randomly constructed LDPC block codes that used a matrix-based unwrapping process. Although their method specifically targeted randomly constructed block codes, it can be adapted to work for any block code. In this paper, we present several methods of deriving LDPC convolutional codes, both time-varying and time-invariant, from QC LDPC block codes. We also extend the pseudo-codeword analysis of [7] for time-invariant LDPC convolutional codes to the time-varying case.

II. PRELIMINARIES

In this section we introduce the background needed for the later development of the paper. Note that all codes are binary linear codes. Any length $n \triangleq r \cdot L$ quasi-cyclic (QC) code $C_{\text{QC}} \triangleq C_{\text{QC}}^{(r)}$ with period L can be represented by a scalar block parity-check matrix $\mathbf{H}_{\text{QC}}^{(r)} \in \mathcal{M}_{rJ \times rL}(\mathbb{F}_2)$ that consists of $J \cdot L$ circulant matrices of size $r \times r$. Using the isomorphism between the commutative ring of $r \times r$ binary circulant matrices and the ring of polynomials $\mathbb{F}_2[X]/(X^r - 1)$ of degree less than r , we can associate with the scalar parity-check matrix $\mathbf{H}_{\text{QC}}^{(r)}$ the polynomial parity-check matrix $\mathbf{H}_{\text{QC}}^{(r)}(X) \in (\mathbb{F}_2[X]/(X^r - 1))^{J \times L}$, with polynomial operations performed modulo $X^r - 1$. 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. For example, the following matrices are the scalar and polynomial representations of a length $n = 21$ QC code $C_{\text{QC}}^{(7)} \subset \mathbb{F}_2^{21}$ with period 3:¹

$$\mathbf{H}_{\text{QC}}^{(7)} = \begin{bmatrix} \mathbf{I}_1 & \mathbf{I}_2 & \mathbf{I}_4 \\ \mathbf{I}_6 & \mathbf{I}_5 & \mathbf{I}_3 \end{bmatrix}, \quad \mathbf{H}_{\text{QC}}^{(7)}(X) = \begin{bmatrix} X & X^2 & X^4 \\ X^6 & X^5 & X^3 \end{bmatrix}. \quad (1)$$

By permuting the rows and columns of the scalar parity-check matrix $\mathbf{H}_{\text{QC}}^{(r)}$ (i.e., by taking the first row in the first block of r rows, the first row in the second block of r rows, etc., then the second row in the first block, the second row in the second block, etc., and similarly for the columns), we obtain

¹ \mathbf{I}_a denotes the a -times left-circularly-shifted identity matrix.

the parity-check matrix $\overline{\mathbf{H}}_{\text{QC}}^{(r)}$ of a code that is equivalent to $\mathbf{C}_{\text{QC}}^{(r)}$. The scalar parity-check matrix $\overline{\mathbf{H}}_{\text{QC}}^{(r)}$ has the form

$$\overline{\mathbf{H}}_{\text{QC}}^{(r)} \triangleq \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_{r-1} & \cdots & \mathbf{H}_1 \\ \mathbf{H}_1 & \mathbf{H}_0 & \cdots & \mathbf{H}_2 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{r-1} & \mathbf{H}_{r-2} & \cdots & \mathbf{H}_0 \end{bmatrix},$$

where the scalar $J \times L$ -matrices $\mathbf{H}_0, \mathbf{H}_1, \dots, \mathbf{H}_{r-1}$ satisfy $\mathbf{H}_{\text{QC}}^{(r)}(X) = \mathbf{H}_0 + \mathbf{H}_1 X + \cdots + \mathbf{H}_{r-1} X^{r-1}$.

Given the polynomial parity-check matrix of a QC code, it is easy to see the natural connection that exists between quasi-cyclic codes and time-invariant convolutional codes (see, e.g., [1]–[3], [5]). Briefly, with any QC block code $\mathbf{C}_{\text{QC}}^{(r)} \subseteq \mathbb{F}_2^{r \times L}$, given by the polynomial parity-check matrix $\mathbf{H}_{\text{QC}}^{(r)}(X) \in \mathcal{M}_{J \times L}(\mathbb{F}_2[X]/(X^r - 1))$, we can associate a rate $(L - J)/L$ time-invariant convolutional code \mathbf{C}_{conv} given by the same $J \times L$ polynomial parity-check matrix $\mathbf{H}_{\text{conv}}(D) = (h_{jl}(D))_{J \times L} = \mathbf{H}_{\text{QC}}^{(r)}(D) = \mathbf{H}_0 + \mathbf{H}_1 D + \cdots + \mathbf{H}_{r-1} D^{r-1} \in \mathcal{M}_{J \times L}(\mathbb{F}_2[D])$, where the change of variables indicates the lack of modulo $D^r - 1$ operations [5]. Let the *syndrome former memory* $m_s \leq r - 1$ of \mathbf{C}_{conv} be the largest integer in $\{0, 1, \dots, r - 1\}$ such that $\mathbf{H}_{m_s} \neq \mathbf{0}$. Then the polynomial parity-check matrix $\mathbf{H}_{\text{conv}}(D)$ has the following scalar description:²

$$\overline{\mathbf{H}}_{\text{conv}} \triangleq \begin{bmatrix} \mathbf{H}_0 & & & & & & \\ \mathbf{H}_1 & \mathbf{H}_0 & & & & & \\ \vdots & \vdots & \ddots & & & & \\ \mathbf{H}_{m_s} & \mathbf{H}_{m_s-1} & & & & & \\ & & \mathbf{H}_{m_s} & & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & \ddots & \\ & & & & & & \ddots \end{bmatrix}. \quad (2)$$

From the above example, we obtain a time-invariant convolutional code with polynomial parity-check matrix $\mathbf{H}_{\text{conv}}(D)$ given by

$$\mathbf{H}_{\text{conv}}(D) = \begin{bmatrix} D & D^2 & D^4 \\ D^6 & D^5 & D^3 \end{bmatrix}, \quad (3)$$

and with syndrome former memory $m_s = r - 1 = 6$.³ This process is referred to as the unwrapping of the QC code to obtain a time-invariant convolutional code. In Figure 1, we use the above example to illustrate this process using the scalar parity-check matrices $\mathbf{H}_{\text{QC}}^{(7)}$ and $\overline{\mathbf{H}}_{\text{conv}}^{(7)}$ which describe the QC and the convolutional codes, respectively. Starting from the binary parity-check matrix $\mathbf{H}_{\text{QC}}^{(7)}$ (Fig. 1(a)), we reorder the rows and columns as described above to obtain $\overline{\mathbf{H}}_{\text{QC}}^{(7)}$ and cut it along the diagonal in steps of $J \times L$ (Fig. 1(b)). Then the upper-diagonal portion is patched to the bottom of the lower-diagonal portion, and the resulting diagonally-shaped matrix is repeated

²A similar convolutional-like parity-check matrix for a class of QC LDPC block codes was recently proposed in [6].

³In some cases, a smaller value of m_s may be obtained by removing common factors from the rows of the polynomial parity-check matrix $\mathbf{H}_{\text{conv}}(D)$. For example, in (2), an equivalent $\mathbf{H}_{\text{conv}}(D)$ with $m_s = 3$ can be obtained.

indefinitely to obtain the matrix $\overline{\mathbf{H}}_{\text{conv}}$ in (2) (Fig. 1(c)). This method of cutting and pasting parts of $\mathbf{H}_{\text{QC}}^{(r)}$ to obtain the matrix $\overline{\mathbf{H}}_{\text{conv}}$ is naturally referred to as unwrapping $\mathbf{H}_{\text{QC}}^{(r)}$ to $\overline{\mathbf{H}}_{\text{conv}}$.

The relation between the minimum Hamming weight and the minimum pseudo-weight of a QC code and those of the unwrapped time-invariant convolutional code obtained from it can be found in [1], [2] and [7]. For any codeword $c(D)$ with finite support in the time-invariant convolutional code, its r *wrap-around* vector $c(X) \bmod (X^r - 1) \in (\mathbb{F}_2[X]/(X^r - 1))^L$, is a codeword in the associated QC code, and their Hamming weights are linked by the inequality $w_H(c(X) \bmod (X^r - 1)) \leq w_H(c(D))$, which gives the inequality [1], [2] $d_{\min}(\mathbf{C}_{\text{QC}}^{(r)}) \leq d_{\text{free}}(\mathbf{C}_{\text{conv}})$, for all $r \geq 1$. Moreover, for the AWGNC, BEC, and BSC pseudo-weights, if $\omega(D) \in (\mathbb{R}[D])^L$ is a pseudo-codeword in the time-invariant convolutional code, then $\omega(X) \bmod (X^r - 1) \in (\mathbb{R}[X]/(X^r - 1))^L$ is a pseudo-codeword in the associated QC code and their pseudo-weights are linked by the inequality $w_p(\omega(X) \bmod (X^r - 1)) \leq w_p(\omega(D))$, which leads to [7]

$$w_p^{\min}(\mathbf{H}_{\text{QC}}^{(r)}(X)) \leq w_p^{\min}(\mathbf{H}_{\text{conv}}(D)).$$

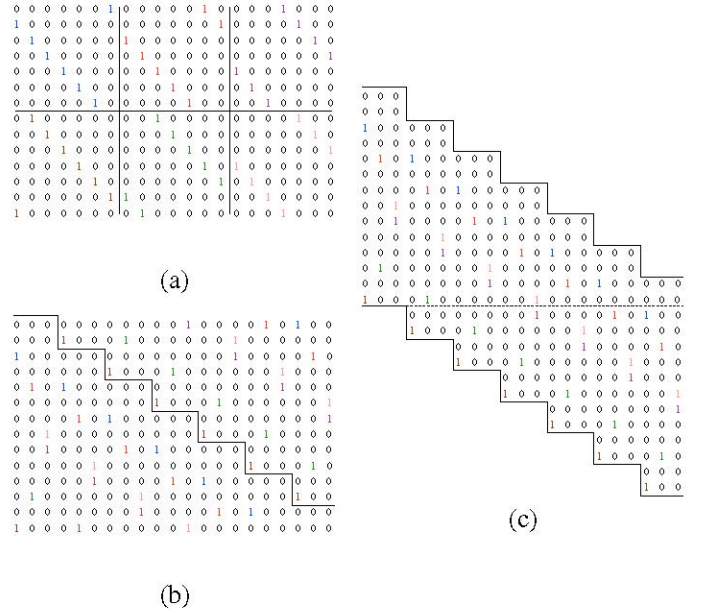


Fig. 1. Deriving a time-invariant LDPC convolutional code from a QC LDPC block code: (a) QC LDPC code, (b) after reordering of rows and columns, (c) after unwrapping.

III. DERIVING TIME-VARYING CONVOLUTIONAL CODES FROM QC BLOCK CODES

Motivated by the results of [8], we now consider an alternative version of unwrapping, without prior row and column reordering. In this approach, we directly unwrap the binary parity-check matrix $\mathbf{H}_{\text{QC}}^{(r)}$ in steps of $J \times L$ to obtain the parity-check matrix $\tilde{\mathbf{H}}_{\text{conv}}$. In general, $\tilde{\mathbf{H}}_{\text{conv}}$ describes a periodically time-varying rate $R = (L - J)/L$ convolutional code with syndrome former memory $m_s \leq r - 1$ (the same as for

the time-invariant codes) and period r . We demonstrate this approach in Figure 2, using the same QC block code as in Figure 1.

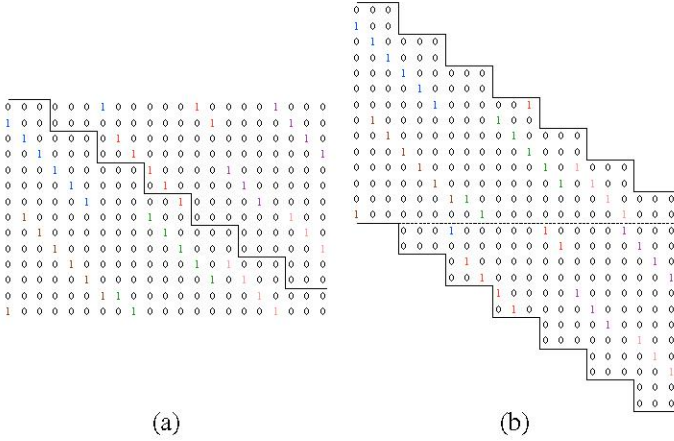


Fig. 2. Deriving a time-varying LDPC convolutional code from a QC LDPC block code: (a) QC LDPC code, (b) after unwrapping.

Although the unwrapping step size of $J \times L$ is fixed for the time-invariant code construction, there is no such constraint on the step size of the time-varying unwrapping. We can choose any step size $Jk \times Lk$, where $0 < k \leq r$ and $Jk, Lk \in \mathbb{Z}$. The special case of $k = r$, for example, corresponds to repeating the original QC block code indefinitely, and is therefore of no practical interest. On the other hand, this special case helps to show the connection between the block and convolutional codes. Another convenient feature of the time-varying unwrapping is that a family of time-varying convolutional codes can be derived by choosing different values for the circulant size r of the QC block code. The resulting time-varying codes have syndrome former memory $m_s \leq r - 1$ and period r for different values of r . This property is not shared by the time-invariant unwrapping, however, since for any circulant size r the resulting time-invariant code is the same.

IV. PSEUDO-CODEWORDS AND CYCLES IN TIME-VARYING CONVOLUTIONAL CODES

Similar to the comparison of the minimum Hamming weights and the minimum pseudo-weights of the original QC code and the unwrapped time-invariant convolutional code, we can also compare the original QC code to the time-varying convolutional code \tilde{C}_{conv} given by the parity-check matrix $\tilde{\mathbf{H}}_{\text{conv}}$. Let \mathbf{c} be a codeword with finite support in the time-varying convolutional code and let $\mathbf{c} \bmod (rL) \in \mathbb{F}_2^{rL}$ be its r wrap-around vector in the associated QC code.⁴ So, by definition, \mathbf{c} is in the null-space of the matrix $\tilde{\mathbf{H}}_{\text{conv}}$. Note that any row vector \mathbf{r} of the matrix $\mathbf{H}_{\text{QC}}^{(r)}$ can be obtained by patching together two row vectors \mathbf{r}' and \mathbf{r}'' of the matrix $\tilde{\mathbf{H}}_{\text{conv}}$ that have disjoint support (no overlapping ones). Since \mathbf{c} is in the null space of each of the vectors \mathbf{r}' and \mathbf{r}'' , and these vectors have disjoint support, we obtain that the vector $\mathbf{c} \bmod (rL)$ is in the null space of the vector \mathbf{r} . (For

⁴If $\mathbf{v} = (v_i)_{0 \leq i < N}$ is a length- N vector then $\mathbf{v} \bmod A = (\sum_{j: 0 \leq s+jA < N} v_{s+jA})_{0 \leq s < A}$.

example, note that the first and the fifteenth rows of the convolutional code matrix in Figure 2(b) can be patched together to form the first row of the block code matrix in Figure 2(a).) Hence any codeword with finite support \mathbf{c} in the time-varying convolutional code wraps to a codeword $\mathbf{c} \bmod (rL)$ in the associated QC code. Their Hamming weights are linked through $w_{\text{H}}(\mathbf{c} \bmod (rL)) \leq w_{\text{H}}(\mathbf{c})$, which, similar to the time-invariant case [1], [2], gives (under very mild conditions on the convolutional code) the inequality

$$d_{\min}(\mathbf{C}_{\text{QC}}^{(r)}) \leq d_{\text{free}}(\tilde{C}_{\text{conv}}), \text{ for all } r \geq 1.$$

In order to compare the minimum pseudo-weights of the two codes, we recall from [7] that we can describe the pseudo-codewords in a code $\mathbf{C} \subset \mathbb{F}_2^n$ given by a parity-check matrix \mathbf{H} as elements of the fundamental cone $\mathcal{K}(\mathbf{H})$ of \mathbf{H} . The fundamental cone can be described by a set of inequalities [9], [10]. Let $\mathcal{I} \triangleq \mathcal{I}(\mathbf{H})$ be the set of column indices of \mathbf{H} and $\mathcal{J} \triangleq \mathcal{J}(\mathbf{H})$ be the set of row indices of \mathbf{H} , respectively, and, for each $j \in \mathcal{J}$, let $\mathcal{I}_j \triangleq \mathcal{I}_j(\mathbf{H}) \triangleq \{i \in \mathcal{I} \mid h_{ji} = 1\}$ be the support of the j -th row of \mathbf{H} . Then a vector $\boldsymbol{\omega} = (\omega_1, \dots, \omega_n) \in \mathbb{R}^n$ is in the fundamental cone $\mathcal{K} \triangleq \mathcal{K}(\mathbf{H})$ if and only if

$$\omega_i \geq 0 \quad \text{for all } i \in \mathcal{I}(\mathbf{H}), \quad (4)$$

$$\sum_{i \in (\mathcal{I}_j \setminus \{i'\})} \omega_i \geq \omega_{i'} \quad \text{for all } j \in \mathcal{J}(\mathbf{H}) \text{ and for all } i' \in \mathcal{I}_j(\mathbf{H}). \quad (5)$$

Hence a pseudo-codeword satisfies certain inequalities associated with each row of the parity-check matrix. Now let $\boldsymbol{\omega}$ be a pseudo-codeword with finite support in the time-varying convolutional code and let $\boldsymbol{\omega} \bmod (rL) \in \mathbb{R}_2^{rL}$ be its r wrap-around vector in the associated QC code. As is shown next, we have that $\boldsymbol{\omega}$ satisfies the fundamental cone inequalities described above associated with each of the rows of the matrix $\tilde{\mathbf{H}}_{\text{conv}}$. Indeed, as explained above, any row vector \mathbf{r} of the matrix $\mathbf{H}_{\text{QC}}^{(r)}$ can be obtained by patching together two row vectors \mathbf{r}' and \mathbf{r}'' of the matrix $\tilde{\mathbf{H}}_{\text{conv}}$ that have disjoint support, we obtain that the vector $\boldsymbol{\omega} \bmod (rL)$ satisfies the inequalities corresponding to the row \mathbf{r} . Hence any pseudo-codeword with finite support $\boldsymbol{\omega}$ in the time-varying convolutional code wraps to a pseudo-codeword $\boldsymbol{\omega} \bmod (rL)$ in the associated QC code. Moreover, we showed in [7] that their AWGNC, BEC, and BSC pseudo-weights are then linked through the inequality $w_{\text{p}}(\boldsymbol{\omega} \bmod (rL)) \leq w_{\text{p}}(\boldsymbol{\omega})$, which leads to

$$w_{\text{p}}^{\min}(\mathbf{H}_{\text{QC}}^{(r)}) \leq w_{\text{p}}^{\min}(\tilde{\mathbf{H}}_{\text{conv}}).$$

It is well known that cycles in the Tanner graph representation of a sparse code affect the iterative decoding algorithm, with short cycles generally pushing its performance further away from optimum. (Indeed, an attempt to investigate and minimize these effects has been made in [11] and [12], where the authors propose LDPC code construction procedures to maximize the connectivity of short cycles to the rest of the graph, thus maximizing the independence of the messages

flowing through a cycle.) Hence it is common practice to design codes that do not contain short cycles, so as to obtain independent messages in at least the initial steps (iterations) of the decoding process.

We now show that unwrapped convolutional codes have better cycle properties than their QC block code counterparts. In particular, we show that any cycle in a convolutional code derived by unwrapping a QC block code always maps to a cycle of the same or smaller length in the underlying QC block code. This result holds for both time-invariant and time-varying convolutional codes, since the additional step of row/column reordering in the time-invariant code derivation does not affect the cycle structure in the Tanner graph of the underlying block code.

Let C_{conv} be the convolutional code derived by unwrapping a QC block code $C_{\text{QC}}^{(r)}$. Take a cycle $(v_{I(0)}, v_{I(1)}, \dots, v_{I(L-1)})$ of length $2L$ in the Tanner graph of the convolutional code, such that the sequence of variable nodes, in cyclic order, $v_{I(0)} \rightarrow v_{I(1)} \rightarrow \dots \rightarrow v_{I(L-1)} \rightarrow v_{I(0)}$ constitutes a cycle in the Tanner graph associated with the convolutional code. Here, $\mathbf{I} = (I(0), I(1), \dots, I(L-1))$ is an index vector of length L and all of its elements are unique by definition of a cycle. Consider the modulo index vector $\tilde{\mathbf{I}}$ obtained by letting $\tilde{I}(i) = I(i) \bmod (n)$, $i = 0, 1, \dots, L-1$. This operation corresponds to wrapping the convolutional code back to the underlying QC block code and yields the path $(v_{\tilde{I}(0)}, v_{\tilde{I}(1)}, \dots, v_{\tilde{I}(L-1)})$ in the Tanner graph of the QC code. If all elements of this path are unique, the path is a cycle of length $2L$ in the Tanner graph of the underlying QC code. Otherwise, there is at least one repetition in the vector $\tilde{\mathbf{I}}$, i.e., it cannot be a cycle itself, although it can be partitioned into shorter cycles. For any repetition in vector $\tilde{\mathbf{I}}$, let i and j be two positions such that $\tilde{I}(i) = \tilde{I}(j)$, with $j > i$. This repetition yields a cycle of length $2 \min(j-i, L+i-j)$. It follows that

$$2 \leq 2 \min(j-i, L+i-j) \leq 2 \left\lfloor \frac{L}{2} \right\rfloor \leq L < 2L.$$

The only case where the lower bound is achieved with equality is when $j = i + 1$. However, $v_{I(i)}$ and $v_{I(i+1)}$ cannot follow each other in a cycle in the convolutional code graph, since by construction two distinct copies of a variable node in the QC block code cannot participate in a common check equation of the unwrapped convolutional code. Therefore,

$$2 < 2 \min(j-i, L+i-j) \leq 2 \left\lfloor \frac{L}{2} \right\rfloor \leq L < 2L$$

and a cycle in the unwrapped convolutional code maps to a cycle of the same or smaller length in the QC block code.

V. SIMULATION RESULTS

In the previous section, we showed that better pseudo-weight properties result when we unwrap a QC block code using the time-varying unwrapping procedure, similar to the results of [7] for the time-invariant unwrapping. This suggests that a time-varying LDPC convolutional code constructed in this fashion will perform better than the underlying QC LDPC block code. In this section we use computer simulations

on an additive white Gaussian noise (AWGN) channel to demonstrate this improved BER performance. In addition, we investigate how the time-varying convolutional code performs compared to its time-invariant counterpart obtained from the same underlying QC block code.

We take the same example code used in [7], a [155,64] (3,5)-regular QC LDPC block code whose parity-check matrix contains 3 ones per column and 5 ones per row with circulant size $r = 31$. We apply the methods described in Section III to construct a variety of time-invariant and time-varying convolutional codes. The block code has a polynomial parity-check matrix given by

$$\mathbf{H}_{\text{QC}}^{(r)}(X) = \begin{bmatrix} X & X^2 & X^4 & X^8 & X^{16} \\ X^5 & X^{10} & X^{20} & X^9 & X^{18} \\ X^{25} & X^{19} & X^7 & X^{14} & X^{28} \end{bmatrix}.$$

The rate $R = b/c = 2/5$ time-invariant convolutional code obtained from this block code has syndrome former memory $m_s = 28$, corresponding to an overall constraint length of $\nu = (m_s + 1) \cdot c = 145$. On the other hand, we can obtain rate $R = 2/5$ time-varying convolutional codes with syndrome former memories $m_s = 30, 47$, and 79 by performing the time-varying unwrapping (cutting along the diagonal of $\mathbf{H}_{\text{QC}}^{(r)}$ in steps of 3×5) on QC block codes with the same polynomial parity-check matrix and circulant sizes of 31, 48, and 80, respectively. (The time-invariant unwrapping for each of the QC codes with different circulant sizes gives the same $m_s = 28$ convolutional code since the time-invariant unwrapping utilizes the polynomial parity-check matrix structure only and not the circulant size.)

A sliding window message-passing decoder was used to decode the convolutional codes (see, e.g., [5]). Conventional LDPC block code decoders were employed to decode the QC LDPC block code. All decoders were allowed a maximum of 50 iterations, and the block code decoders employed a syndrome-check based stopping rule. The resulting BER performance of these codes is shown in Figure 3.

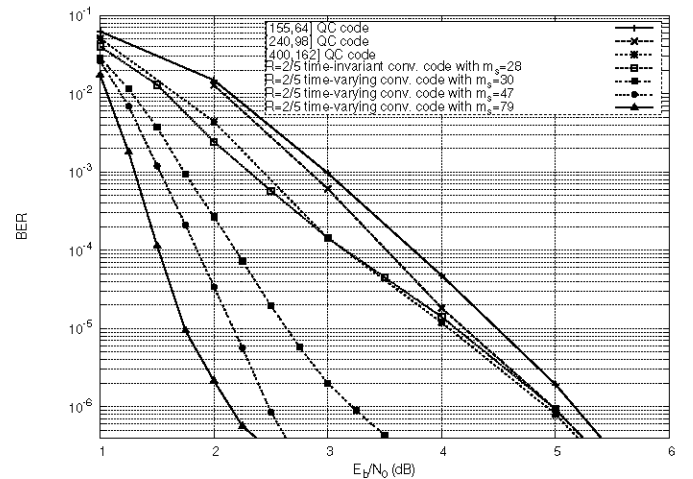


Fig. 3. Performance of three (3,5)-regular QC LDPC block codes and their associated LDPC convolutional codes.

We note that the time-invariant LDPC convolutional code

performs between 0.5dB and 1.0dB better than the QC LDPC block code in the low-to-moderate SNR region and that the gain drops to only about 0.2dB in the high SNR region. On the other hand, the time-varying LDPC convolutional codes achieve significantly better performance, with gains ranging between 1.9dB and 2.8dB at a BER of 10^{-6} compared to their corresponding QC LDPC block codes. We also note that the gains increase with increasing circulant size.

Since all the simulated codes are (3,5)-regular, their computational complexity is equivalent. Thus we adopt the notion of *processor complexity* (see [13]) to compare the block and convolutional codes. A decoder's processor complexity is proportional to the maximum number of variable nodes that can participate in a common check equation. This is the block length n for a block code, since any two variable nodes that are $n - 1$ positions apart can participate in the same check equation. For a convolutional code, this is equal to the overall constraint length ν , since no two variable nodes that are more than ν positions apart can participate in the same check equation. Thus, for a comparison based on processor complexity, we require that $n = \nu$. The time-varying convolutional code derived from the QC code with circulant size $r = 31$ has overall constraint length $\nu = (m_s + 1)c = 150$, and hence approximately the same processor complexity as the QC block code of length $n = 155$ and the time-invariant convolutional code with $\nu = 140$, but it achieves large BER gains compared to both of these codes. We note, in addition, that the performance of the time-varying convolutional code with syndrome former memory $m_s = 80$ and overall constraint length $\nu = 400$ is quite remarkable, since, at a BER of 10^{-5} , it performs within 1dB of the iterative decoding threshold of 0.965dB, while having the same processor complexity as a block code of length only $n = 400$.

VI. DISCUSSION

The simulation results presented in Section V demonstrate substantial BER performance gains for LDPC convolutional codes obtained by unwrapping QC LDPC block codes. In addition, they demonstrate that larger gains are available with time-varying unwrappings than with time-invariant unwrappings. The results of [7] explain the superiority of the time-invariant convolutional codes compared to the QC block codes. Similarly, the results of Section IV provide a plausible explanation for the performance difference between the time-varying and time-invariant convolutional codes. For example, we have examined the cycle histograms of the convolutional codes obtained from the QC block code of length $n = 155$ for each type of unwrapping. The girth⁵ of the QC code is 8 and it has an average of 7 cycles of length 8 and 10.65 cycles of length 10. The corresponding time-invariant convolutional code has 5.8 cycles of length 8 and 9.6 cycles of length 10, and the time-varying convolutional code has 3.57 cycles of length 8 and 10.32 cycles of length 10. Hence, we see that many of the short cycles in the QC block code are broken to yield cycles of

larger length by the time-invariant unwrapping and that even more cycles are broken by the time-varying unwrapping.

VII. CONCLUSIONS

In this paper, we showed that significant BER performance gains can be achieved by unwrapping QC LDPC block codes to obtain time-invariant and time-varying LDPC convolutional codes. The resulting convolutional codes are shown to have better pseudo-codeword properties and better cycle histograms, allowing them to outperform their block code counterparts. For example, the time-varying convolutional codes derived from the QC LDPC block codes described in Section V have been shown to achieve gains ranging from 1.9dB to 2.8dB at a BER of 10^{-6} while operating at SNR's close to the iterative decoding threshold.

ACKNOWLEDGMENTS

This work was partially supported by NSF Grants CCR-0205310 and TF-0514801, NASA Grant NNG05GH73G, and a Graduate Fellowship from the Center for Applied Mathematics, University of Notre Dame.

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, A. Sridharan, D. Sridhara, 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, no. 12, pp. 2966–2984, December 2004.
- [6] Y.Y. Tai, L. Lan, L. Zeng, S. Lin, and K.A.S. Abdel-Ghaffar, "Algebraic construction of quasi-cyclic LDPC codes for the AWGN and Erasure Channels," *IEEE Trans. on Commun.*, vol. 54, no. 10, pp. 1765–1774, October 2006.
- [7] R. Smarandache, A. E. Pusane, P. O. Vontobel, D. J. Costello, Jr. "Pseudo-codeword Performance Analysis of LDPC convolutional codes", *IEEE Trans. Inform. Theory*, to appear, available online under <http://arxiv.org/abs/cs.IT/0609148>.
- [8] A. Jiménez-Feltström and K. Sh. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. on Inform. Theory*, vol. IT-45, pp. 2181–2191, September 1999.
- [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] 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, March 2005.
- [11] T. Tian, C. R. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. on Commun.*, vol. COM-52, no. 8, pp. 1242–1247, August 2004.
- [12] A. Ramamoorthy and R. D. Wesel, "Analysis of an algorithm for irregular LDPC code construction," in *Proc. IEEE Intern. Symp. on Inform. Theory*, (Chicago, IL), p. 69, June 27–July 2, 2004.
- [13] D. J. Costello, Jr., A. E. Pusane, S. Bates, and K. Sh. Zigangirov, "A comparison between LDPC block and convolutional codes," in *Proc. Information Theory and Applications Workshop*, (San Diego, CA), February 6–10, 2006.

⁵The class of QC LDPC codes constructed in [4], [5] is known to have good girth properties.