Pseudocodeword Performance Analysis for LDPC Convolutional Codes

Roxana Smarandache, *Member, IEEE*, Ali E. Pusane, *Member, IEEE*, Pascal O. Vontobel, *Member, IEEE*, and Daniel J. Costello, Jr., *Life Fellow, IEEE*

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

In this paper, we address the pseudocodeword problem from the convolutional code perspective. In particular, we compare the performance of LDPC convolutional codes with that of their "wrapped" quasi-cyclic block versions and we show that the minimum pseudoweight of an LDPC convolutional code is at least as large as the minimum pseudoweight of an underlying quasi-cyclic code. This result, which parallels a well-known relationship between the minimum Hamming weight of convolutional codes and the minimum Hamming weight of their quasi-cyclic counterparts, is due to the fact that every pseudocodeword in the convolutional code induces a pseudocodeword in the block code with pseudoweight no larger than that of the convolutional code's pseudocodeword. This difference in the weight spectra leads to improved performance at low-to-moderate SNRs for the convolutional code, a conclusion supported by simulation results.

***Index Terms*—Convolutional codes, linear programming decoding, low-density parity-check (LDPC) codes, message-passing iterative decoding, pseudocodewords, pseudoweights, quasi-cyclic codes.**

I. Introduction

ALTHOUGH low-density parity-check (LDPC) block codes have very good performance under message-passing iterative (MPI) decoding [1]–[3], they are known to be subject to decoding failures due to so-called pseudocodewords. These are real-valued vectors that can be loosely described as error patterns that cause nonconvergence in MPI decoding due to the fact that MPI decoding algorithms work locally on a Tanner graph and can give priority to a vector that fulfills the equations of a Tanner graph cover rather than the Tanner graph itself. The reason is that locally operating decoding algorithms *cannot distinguish* if they are operating on the original Tanner graph or any finite cover of this graph. Therefore, MPI decoding algorithms will automatically take into account all possible codewords in all possible covers of the original graph.

Let us be more precise and consider a binary linear code C of length n that is described by some parity-check matrix $\boldsymbol{H} \in \mathbb{F}_2^{m \times n}$, i.e.,

$$\mathsf{C} = \left\{ \boldsymbol{c} \in \mathbb{F}_2^n \;\middle|\; \boldsymbol{H} \cdot \boldsymbol{c}^{\mathsf{T}} = \boldsymbol{0}^{\mathsf{T}} \pmod 2 \right\}.$$

To such a parity-check matrix $\boldsymbol{H}$ we can associate a Tanner graph [4]: it consists of n variable nodes, m check nodes, and an edge between the ith variable node and the jth check node if and only if the entry in the ith column and jth row of $\boldsymbol{H}$ equals 1.

Example 1: Consider the parity-check matrix

$$\boldsymbol{H} \triangleq \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{1}$$

that defines the code

$$\mathsf{C} \triangleq \{(0,0,0),\ (1,1,0)\}$$

of length 3 and dimension 1. The Tanner graph associated with the parity-check matrix $\boldsymbol{H}$ is depicted in Fig. 1 (left). □

The relevance of the Tanner graph comes from the fact that the operation of MPI decoding algorithms is tightly related to the structure of the Tanner graph. Namely, in one typical iteration, an MPI decoding algorithm sends messages from variable to check nodes along the edges, every check node processes the incoming messages and produces outgoing messages, these outgoing messages are then sent along the edges to the variable nodes, and finally every variable node processes the incoming messages and produces outgoing messages. What exactly is done when producing the outgoing messages from the incoming messages depends on the chosen MPI decoder. In any case, the crucial fact is that these algorithms operate locally, i.e., outgoing messages are computed based on locally available (at a check node or at a variable node) incoming messages. This locality, which is one of the reasons why MPI decoding algorithms are so efficient and popular, is also the main drawback of these types of algorithms.

Example 2: Consider again the code and its parity-check matrix that were introduced in Example 1. Assume that a codeword was transmitted over some memoryless channel and that the vector (y_1, y_2, y_3) was received. Decoding this vector with

Manuscript received September 27, 2006; revised August 19, 2008. Current version published May 20, 2009. The work of R. Smarandache was supported in part by the National Science Foundation (NSF) under Grants CCR-0205310 and DMS-0708033. The work of A. E. Pusane and D. J. Costello, Jr. was supported in part by the National Science Foundation (NSF) under Grants CCR-0205310 and CCF-0515012 and by NASA under Grant NNX07AK536. The work of P. O. Vontobel was supported in part by the National Science Foundation (NSF) under Grant CCF-0514801. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Seattle, WA, July 2006.The work for this paper was partially done while R. Smarandache was on leave at the Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556 USA.

R. Smarandache is with the Department of Mathematics and Statistics, San Diego State University, San Diego, CA 92182 USA (e-mail: rsmarand@sciences.sdsu.edu).

A. E. Pusane and D. J. Costello, Jr. are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: apusane@nd.edu; costello.2@nd.edu).

P. O. Vontobel is with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: pascal.vontobel@ieee.org).

Communicated by T. Richardson, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2009.2018347

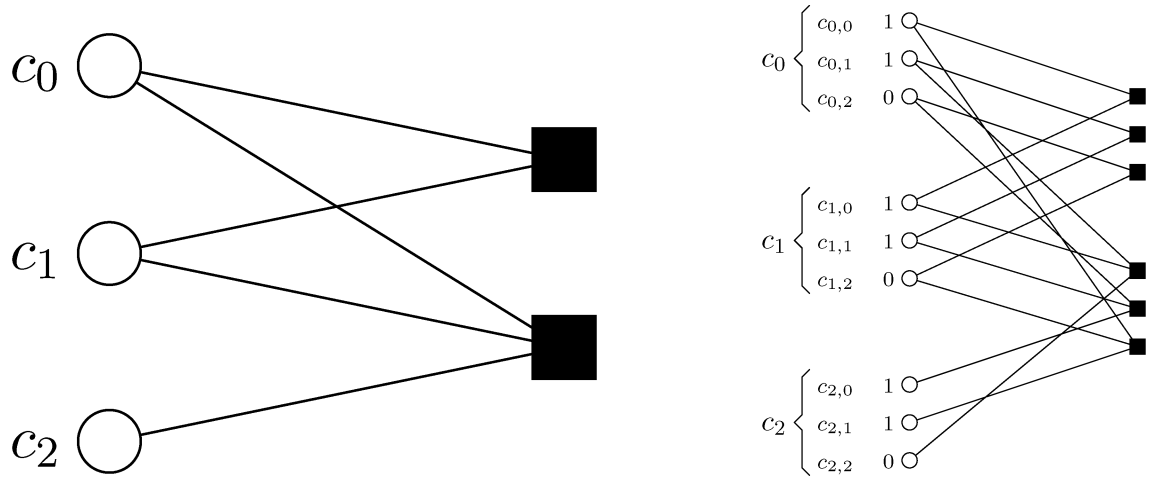

Fig. 1. Left: Tanner graph associated with the parity-check matrix $\boldsymbol{H}$ that was defined in (1). Right: a possible triple cover of the Tanner graph on the left. Also shown is the vector $(1,1,0,\ 1,1,0,\ 1,1,0)$, which is a valid codeword in this cubic cover.

an MPI decoding algorithm means that we are sending some iteration-dependent messages back and forth along the edges of the Tanner graph in Fig. 1 (left).

Consider now the Tanner graph in Fig. 1 (right). It is a "threefold copy" of the Tanner graph in Fig. 1 (left) in the sense that for every variable node in Fig. 1 (left) there are three variable nodes in Fig. 1 (right) and for every check node in Fig. 1 (left) there are three check nodes in Fig. 1 (right). Moreover, for every edge in Fig. 1 (left) there are three edges in Fig. 1 (right) such that there is an edge between a copy of a variable and a copy of a check node only if there was an edge between the original variable and the original check node, and such that the degree of a copy of a variable (check) node equals the degree of the original variable (check) node. A possible codeword in this Tanner graph is, e.g., the vector $(1,1,0,\ 1,1,0,\ 1,1,0)$.

Now assume that we have received the vector $(y_1,y_1,y_1,\quad y_2,y_2,y_2,\quad y_3,y_3,y_3)$ and that we are applying the same MPI decoding algorithm as above, simply this time on Fig. 1 (right) and not on Fig. 1 (left). Because the Tanner graph in Fig. 1 (right) looks locally like the Tanner graph in Fig. 1 (left), and because of the way that the received vector was defined, it is rather straightforward to see that at any iteration the messages that are sent along the three copies of an edge equal the message that is sent along the original edge.

A similar observation can also be made about computation trees [5]. Namely, for a given number of iterations, the computation tree rooted at a variable node in Fig. 1 (right), together with the messages on it, equals the computation tree rooted at the corresponding original variable node in Fig. 1 (left), together with the messages on it.

Based on these observations, [6], [7] concluded that an MPI decoding algorithm cannot distinguish if it is decoding the code defined by the Tanner graph in Fig. 1 (left) or the code defined by the Tanner graph in Fig. 1 (right). □

Of course, there is nothing special about the "threefold copy" above, i.e., for any "M-fold copy" the same argument can be made. Such an "M-fold copy" is in graph theory better known as an M-*cover*, as specified in the following definition.

Definition 3 (see, e.g., [8], [9]): Let $\mathcal{G}$ be a graph with vertex set $\mathcal{V}(\mathcal{G})$ and edge set $\mathcal{E}(\mathcal{G})$, and let $\partial(v)$ denote the set of adjacent vertices of a vertex $v \in \mathcal{V}(\mathcal{G})$. An unramified, finite cover, or, simply, a cover of a (base) graph $\mathcal{G}$ is a graph $\widetilde{\mathcal{G}}$ along with a surjective map $\phi : \widetilde{\mathcal{G}} \to \mathcal{G}$, which is a graph homomorphism, i.e., which takes adjacent vertices of $\widetilde{\mathcal{G}}$ to adjacent vertices of $\mathcal{G}$ such that, for each vertex $v \in \mathcal{V}(\mathcal{G})$ and each $\widetilde{v} \in \phi^{-1}(v)$, the neighborhood $\partial(\widetilde{v})$ of $\widetilde{v}$ is mapped bijectively to $\partial(v)$. For a positive integer M, an M-cover of $\mathcal{G}$ is an unramified finite cover $\phi : \widetilde{\mathcal{G}} \to \mathcal{G}$ such that, for each vertex $v \in \mathcal{V}(\mathcal{G})$ of $\mathcal{G}$, $\phi^{-1}(v)$ contains exactly M vertices of $\widetilde{\mathcal{G}}$. An M-cover of $\mathcal{G}$ is sometimes also called an M-sheeted covering of $\mathcal{G}$ or a cover of $\mathcal{G}$ of degree M.[1]

It was then argued in [6], [7] that in order to understand MPI decoders one has to study the set of all codewords in all finite covers. In fact, it turns out to be sufficient to study the set of all pseudocodewords associated with all the codewords in all the finite covers.[2] At first, this sounds like a formidable task, but it turns out that the closure of this latter set is a polytope, called the *fundamental polytope* $\mathcal{P}(\boldsymbol{H})$, and has a relatively simple description in terms of the parity-check matrix $\boldsymbol{H}$.[3]

In the following we will only consider binary linear codes and binary-input output-symmetric memoryless channels. For such codes and channels it is sufficient to consider the case where the all-zero codeword was sent. This follows from the symmetries of the fundamental polytope. Therefore, understanding MPI decoding algorithms has a lot to do with understanding the set of nonzero vectors in $\mathcal{P}(\boldsymbol{H})$. This is in contrast to maximum-likelihood (ML) decoding that is characterized by the nonzero vec-

[1]It is important not to confuse the degree of a covering and the degree of a vertex.

[2]To every codeword in any finite cover we associate a pseudocodeword by projecting the codeword onto a length-n vector, e.g., with the codeword $(1,1,0,\ 1,1,0,\ 1,1,0)$ in Fig. 1 (right) we associate the pseudocodeword $(2/3, 2/3, 2/3)$. The exact definition of a pseudocodeword will be given in Section II.

[3]Note that the fundamental polytope $\mathcal{P}(\boldsymbol{H})$ is usually strictly larger than $\mathrm{ConvHull}(\mathsf{C})$, the convex hull of the set of codewords embedded in $\mathbb{R}^n$. For example, the above-mentioned pseudocodeword $(2/3, 2/3, 2/3)$ cannot be written as a convex combination of the codewords in C.

tors in C, especially the so-called *minimal codewords*.[4] Under MPI decoding, the role of minimal codewords is taken over by so-called *minimal pseudocodewords*; these are pseudocodewords that correspond to (points on) origin-adjacent edges of the fundamental polytope $\mathcal{P}(\boldsymbol{H})$ [7], [10].

In the same way as the Hamming weight $w_{\mathrm{H}}(\boldsymbol{c})$ is a very important characteristic of a nonzero codeword $\boldsymbol{c}$ under ML decoding, the pseudoweight $w_{\mathrm{p}}(\boldsymbol{\omega})$ is a very important characteristic of a nonzero pseudocodeword $\boldsymbol{\omega}$ under MPI decoding.[5] In particular, in the same way as the minimum Hamming weight

$$w_{\mathrm{H}}^{\min}(\mathsf{C}) \triangleq \min_{\substack{\boldsymbol{c}\in\mathsf{C}\\ \boldsymbol{c}\neq\boldsymbol{0}}} w_{\mathrm{H}}(\boldsymbol{c})$$

is a very important characteristic for a binary linear code under ML decoding, the minimum pseudoweight

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

is a very important characteristic for a binary linear code under MPI decoding, especially for large signal-to-noise ratios (SNRs).

As a consequence, the large SNR performance of MPI decoding can be worse than that predicted by the ML decoding union bound, which constitutes a major problem when trying to determine performance guarantees. Addressing this problem from the convolutional code perspective, i.e., studying the pseudocodeword problem described above for low-density parity-check convolutional codes (in the following more concisely called LDPC convolutional codes), constitutes the major topic of this paper.[6]

We investigate a class of time-invariant LDPC convolutional codes derived by "unwrapping" certain classes of quasi-cyclic (QC) LDPC block codes that are known to have good performance [12], [13]. Unwrapping a QC block code to obtain a time-invariant convolutional code represents a major link between QC block codes and convolutional codes. This link was first introduced in a paper by Tanner [14], where it was shown that the free distance of the unwrapped convolutional code, if nontrivial, cannot be smaller than the minimum distance of the underlying QC code. This idea was later extended in [15], [16]. More recently, a construction for LDPC convolutional codes based on QC-LDPC block codes was introduced by Tanner *et al.* [12], and a sliding-window MPI decoder was described. In that paper, it was noted that the (nontrivial) convolutional versions of these codes significantly outperformed their block code counterparts in the waterfall region of the bit-error rate (BER) curve, even though locally the graphical representations of the MPI decoders were essentially equivalent.

In the following sections, we will study the connections that exist between pseudocodewords in QC codes and pseudocodewords in the associated convolutional codes and show that this connection mimics the connection between the codewords in QC codes and their associated convolutional codes. One of our goals will be to formulate analytical results (or at least efficient procedures) that will allow us to bound the minimum pseudoweight of the pseudocodewords of the block and convolutional codes.

We also note that our results have a bearing on the characterization of the behavior of the so-called linear-programming (LP) decoder due to Feldman, Wainwright, and Karger [17], [18]. This connection [7], [19] is not surprising, given the central role of the fundamental polytope in the LP decoder formulation, where a certain linear cost function is minimized over the fundamental polytope.

A. Motivational Example

As a motivational example[7] we simulated a rate $R = 2/5$ $(3,5)$-regular LDPC convolutional code with syndrome former memory $m_{\mathrm{s}} = 21$, together with three wrapped block code versions: a $[155, 64]$ $(3,5)$-regular QC-LDPC block code, a $[200, 82]$ $(3,5)$-regular QC-LDPC block code, and a $[240, 98]$, $(3,5)$-regular QC-LDPC block code, with parity-check matrices of increasing circulant sizes $r = 31$, $r = 40$, and $r = 48$, respectively, while keeping the same structure within each $r \times r$ circulant [20]. (Note that increasing the circulant size of the QC code increases its complexity, i.e., its block length. Also note that each of the three block codes has rate slightly greater than $2/5$.) As in [12], a sum–product-algorithm-type sliding-window MPI decoder was used to decode the convolutional code. Conventional LDPC block code MPI decoders were employed to decode the QC-LDPC block codes. All decoders were allowed a maximum of 50 iterations. The resulting BER performance of these codes on a binary-input AWGNC is shown in Fig. 2.

We note that, particularly in the low-to-moderate SNR region, where the complete pseudoweight spectrum plays an important role, the unwrapped LDPC convolutional code performs between 0.5 and 1.0 dB better than the associated QC-LDPC block codes. Also, as the circulant size increases, the performance of the block codes approaches that of the convolutional code. These performance curves suggest that the pseudocodewords in the block code that result in decoding failures, when unwrapped, have larger pseudoweight in the convolutional code.

In order to underline the influence of pseudocodewords under MPI decoding, we consider the following experiment for a binary-input AWGNC. Let $\boldsymbol{\omega}$ be a *minimal pseudocodeword* for the above-mentioned $R = 2/5$ $(3,5)$-regular LDPC convolutional code. (In the simulations we took a minimal pseudocodeword that was found by using a heuristic algorithm that searches

[4]Recall that, when transmitting over a binary-input additive white Gaussian noise channel (AWGNC), the decision region of a codeword shares a facet with the decision region of the all-zero codeword if and only if that codeword is a minimal codeword.

[5]Pseudoweights are channel-dependent, e.g., the binary-input AWGNC pseudoweight for the pseudocodeword $\boldsymbol{\omega} \neq \boldsymbol{0}$ is given by [5]–[7], [11]

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

where $\|\cdot\|_1$ and $\|\cdot\|_2$ are, respectively, the 1-norm and 2-norm. Note that if $\boldsymbol{\omega}$ is a vector containing only zeros and ones, then $w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega}) = w_{\mathrm{H}}(\boldsymbol{\omega})$.

[6]Note that $w_{\mathrm{p}}^{\min}(\boldsymbol{H}) \leq w_{\mathrm{H}}^{\min}(\mathsf{C})$, and the larger the gap between $w_{\mathrm{p}}^{\min}(\boldsymbol{H})$ and $w_{\mathrm{H}}^{\min}(\mathsf{C})$, the greater role the minimum pseudoweight plays. In any case, the minimum Hamming weight is still important because it quantifies the impact of undetectable errors.

[7]The exact meaning of the technical terms in this paragraph will be discussed in Section II.

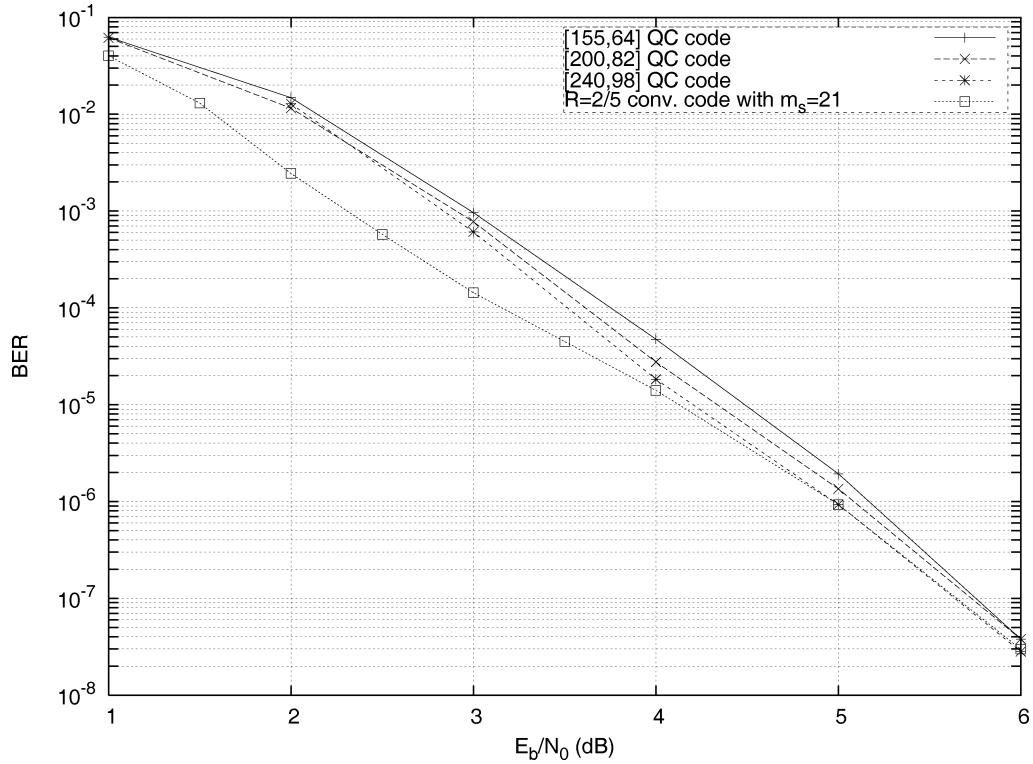

Fig. 2. The sum–product-algorithm-based MPI decoding performance of a rate $R = 2/5$ $(3, 5)$-regular LDPC convolutional code and three associated $(3, 5)$-regular QC-LDPC block codes. (Maximum of 50 iterations.)

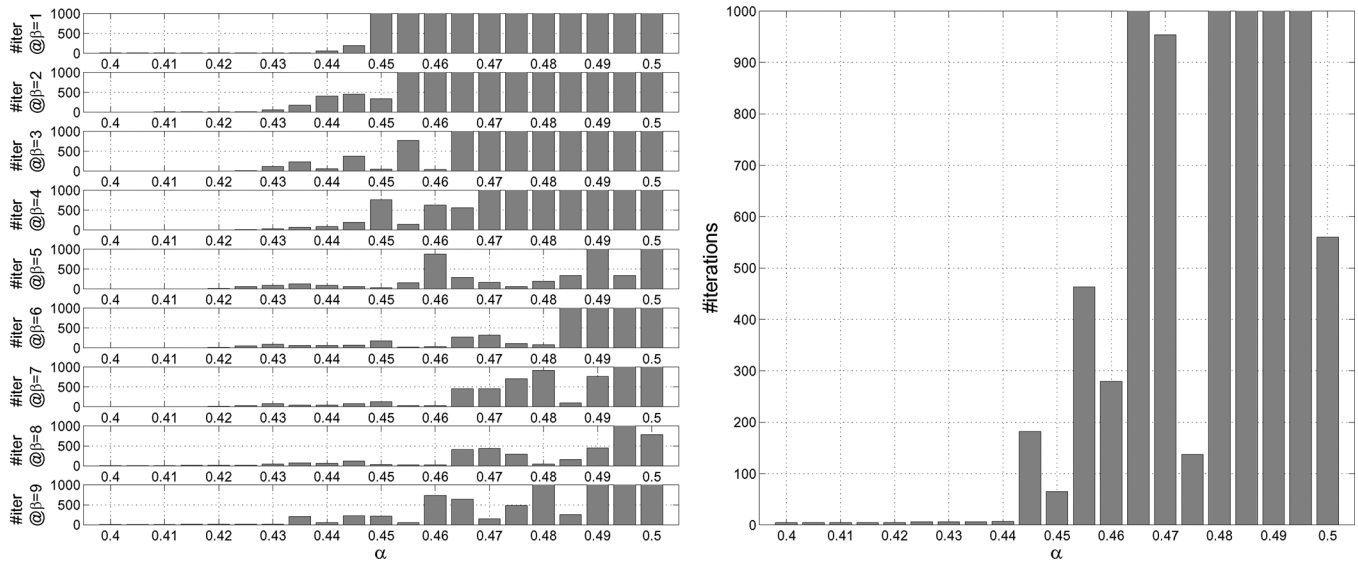

Fig. 3. Left: Number of iterations needed for a sum–product-algorithm-type MPI decoder to decide for the all-zero codeword. (1000 corresponds to no convergence.) Here, the β values $1.00, 2.00, 3.00, 4.00, 5.00, 6.00, 7.00, 8.00, 9.00$ correspond to SNRs E_{b}/N_0 of, respectively, -2.04 dB, , 0.96 dB, 2.73 dB, 3.97 dB, 4.94 dB, 5.74 dB, 6.41 dB, 6.99 dB, and 7.50 dB. Right: Number of iterations needed for a min-sum-algorithm-type MPI decoder to decide for the all-zero codeword. (1000 corresponds to no convergence.) For the simulated (α, β)-pairs, we did not observe convergence to the all-zero codeword for $\alpha > 1/2$, either for the sum–product-algorithm-type MPI decoder or for the min-sum-algorithm-type MPI decoder.

the fundamental polytope for minimal pseudocodewords with low pseudoweight.[8] For the given LDPC convolutional code and its parity-check matrix, we found a minimal pseudocodeword $\boldsymbol{\omega}$ that has AWGNC pseudoweight 18.1297 (which happens to be smaller than the free distance). Moreover, we define the log-likelihood ratio vector $\boldsymbol{\lambda}(\alpha, \beta)$ to be

$$\boldsymbol{\lambda}(\alpha,\beta) \triangleq \beta \cdot \left(\mathbf{1} - 2\alpha\sqrt{w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega})}\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|_2}\right).$$

[8]A published algorithm that performs a similar search is given in [21].

We then run the MPI decoder initialized with $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\alpha, \beta)$ and count how many iterations it takes until the decoder decides for the all-zero codeword as a function of α and β. The results are shown in Fig. 3.

The meaning of $\boldsymbol{\lambda}(\alpha, \beta)$ is the following. If $\alpha = 0$, then $\boldsymbol{\lambda}(\alpha, \beta)$ corresponds to the log-likelihood ratio vector that the receiver sees when the communication system operates at an SNR of $E_{\mathrm{b}}/N_0 = \beta/(4R)$ and when the noise vector that is added by the binary-input AWGNC happens to be the all-zero vector (see, e.g., the discussion in [7, Sec. 3]). For nonzero α,

the expression for $\boldsymbol{\lambda}(\alpha,\beta)$ has been defined such that the LP decoder has a decision boundary at $\alpha = 1/2$: for $\alpha < 1/2$ the all-zero codeword wins against the pseudocodeword $\boldsymbol{\omega}$ whereas for $\alpha > 1/2$ the all-zero codeword loses against the pseudocodeword $\boldsymbol{\omega}$ under LP decoding.[9]

We ran the MPI decoder for various choices of α and β: Fig. 3 (left) shows the number of iterations needed using a sum–product-algorithm-type MPI decoder whereas Fig. 3 (right) shows the number of iterations needed using a min-sum-algorithm-type MPI decoder. (Note that the decisions reached by the latter are independent of the choice of β, $\beta > 0$, the reason being that scaling the log-likelihood ratio vector by β simply scales all messages at all iterations by the same factor β.) In contrast to the min-sum-algorithm-type MPI decoder, the behavior of the sum–product-algorithm-type MPI decoder depends on the value of β, and so it is not surprising that we observe that the closeness of the decision boundary to $\alpha = 1/2$ is a function of β. In particular, note that for very small β's (i.e., very low SNRs), sum–product-algorithm-type MPI decoders have a weakness, since one can show (by looking at the message-update equations) that such decoders give back the correct codeword only if the hard-decision vector based on the received log-likelihood vector is the correct codeword. This results in a decision boundary that is further away from $\alpha = 1/2$ for very small β.

B. Paper Goals and Structure

In this paper, we provide a possible explanation for the performance differences between the different QC codes and the convolutional code observed in the motivational example above. Based on the results of [6], [7] that relate code performance to the existence of pseudocodewords, we examine the pseudocodeword weight spectra of QC-LDPC block codes and their associated convolutional codes, respectively. We will show that for a nontrivial LDPC convolutional code derived by unwrapping a nontrivial QC-LDPC block code,[10] the minimum pseudoweight of the convolutional code is at least as large as the minimum pseudoweight of the underlying QC code, i.e.,

$$w_{\mathrm{p}}^{\min}\left(\boldsymbol{H}_{\mathrm{QC}}^{(r)}\right) \leq w_{\mathrm{p}}^{\min}\left(\boldsymbol{H}_{\mathrm{conv}}\right).$$

This result, which parallels the well-known relationship between the free Hamming distance of nontrivial convolutional codes and the minimum Hamming distance of their nontrivial quasi-cyclic counterparts [14],[11] is based on the fact that every pseudocodeword in the convolutional code induces a pseudocodeword in the block code with pseudoweight no larger than that of the convolutional code's pseudocodeword. 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 Fig. 2.

[9]Note that the square root of the AWGNC pseudoweight is the (normalized) Euclidean distance to the LP decoding decision boundary in the log-likelihood ratio space.

[10]Nontrivial means here that the set of pseudocodewords contains nonzero pseudocodewords.

[11]Nontrivial means here that the set of codewords contains nonzero codewords.

The paper is structured as follows. In Section II, we develop the background necessary to describe the connection between pseudocodewords in unwrapped convolutional codes and those in the associated QC codes. In particular, after having reviewed the basics of QC codes in Section II-A and of convolutional codes in Section II-B, in Section II-C we briefly discuss the connection between convolutional codes and their associated QC codes, especially how codewords in the former can be used to construct codewords in the latter. Then, in Section II-D, we define the fundamental polytope/cone of a parity-check matrix and show how we can describe the fundamental cone of a polynomial parity-check matrix through polynomial inequalities. We end the section by showing how pseudocodewords in unwrapped convolutional codes yield pseudocodewords in the associated QC codes. In Section III, we compare the pseudoweights of unwrapped convolutional codes and their associated QC block codes. In particular, Section III-A introduces various channel pseudoweights and Section III-B presents the main result, namely, that the minimum AWGNC pseudoweight, the minimum binary-symmetric channel (BSC) pseudoweight, the minimum binary-erasure channel (BEC) pseudoweight, and the minimum max-fractional weight of a convolutional code are at least as large as the corresponding minimum pseudoweights of a wrapped QC block code. Section IV discusses a method to analyze problematic pseudocodewords, i.e., pseudocodewords with small pseudoweight. The method addresses the convolutional code case. It introduces two sequences of "truncated" pseudoweights and, respectively, "bounded pseudocodeword" pseudoweights, that play an important role in identifying the minimum pseudoweight of the convolutional code, similar to the role that column distances and row distances play in identifying the free distance. We end with some conclusions in Section V.

C. Notation

We will use the following notation. We let $\mathbb{F}_2$, $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_{++}$ be the Galois field of size 2, the field of real numbers, the set of nonnegative real numbers, and the set of positive real numbers, respectively.

We say that a polynomial $\omega_\ell(D) = \sum_i \omega_{\ell,i} D^i$ with real coefficients is nonnegative, and we write $\omega_\ell(D) \geq 0$, if all its coefficients $\omega_{\ell,i}$ satisfy $\omega_{\ell,i} \geq 0$. Similarly, a polynomial vector $\boldsymbol{\omega}(D) = \big(\omega_0(D), \omega_1(D), \ldots, \omega_{L-1}(D)\big)$ is nonnegative, and we write $\boldsymbol{\omega}(D) \geq \mathbf{0}$, if all its polynomial components $\omega_\ell(D)$ satisfy $\omega_\ell(D) \geq 0$ for all $\ell \in \{0, 1, \ldots, L-1\}$. Moreover, a polynomial matrix $\boldsymbol{A}(D)$ is nonnegative, and we write $\boldsymbol{A}(D) \geq \mathbf{0}$, if all its entries are nonnegative polynomials.

If $a(X) \in \mathbb{F}_2[X]$ is a polynomial and r is some positive integer, then $\big(a(X) \bmod_{\mathbb{F}_2} (X^r - 1)\big)$ denotes the polynomial $b(X) \in \mathbb{F}_2[X]$ of degree smaller than r such that $b(X) = a(X)$ in $\mathbb{F}_2[X]/\langle X^r - 1\rangle$.

Similarly, if $a(X) \in \mathbb{R}[X]$ is a polynomial and r is some positive integer, then $\big(a(X) \bmod_{\mathbb{R}} (X^r - 1)\big)$ denotes the polynomial $b(X) \in \mathbb{R}[X]$ of degree smaller than r such that $b(X) = a(X)$ in $\mathbb{R}[X]/\langle X^r - 1\rangle$. In this sense, $\big(a(X) \bmod (X^r - 1)\big) \geq 0$ means that $a(X)$ is such that the polynomial $b(X)$ of degree smaller than r with $b(X) = a(X)$ in $\mathbb{R}[X]/\langle X^r - 1\rangle$ has only

nonnegative coefficients. This notation is straightforwardly extended to polynomial vectors and matrices.

Finally, to a polynomial matrix $\boldsymbol{M}(D) \in \mathbb{F}_2[D]^{J\times L}$ we associate a weight matrix [13] in $\mathbb{Z}^{J\times L}$, where each entry denotes the number of nonzero coefficients in the corresponding entry of $\boldsymbol{M}(D)$. For example, $\boldsymbol{M}(D) = [1 + D + D^5, D]$ has weight matrix $(3,1)$. Then a polynomial matrix will be called *monomial* [12] if its weight matrix contains only ones, it will be called of *type I* [13] if its weight matrix contains only zeros and ones, and it will be called of *type II* [13] if its weight matrix contains only zeros, ones, and twos. (The weight matrix of $\boldsymbol{M}(X) \bmod_{\mathbb{F}_2} (X^r-1)$, where $\boldsymbol{M}(X)$ is a matrix in $\big(\mathbb{F}_2[X]/\langle X^r-1\rangle\big)^{J\times L}$, is defined analogously.)

II. Pseudocodewords for QC and Convolutional Codes

This section presents an important link between QC block codes and convolutional codes that was first introduced in a paper by Tanner [14] and later extended in [15], [16]. Similar to the connection between codewords in an unwrapped convolutional code and codewords in the underlying QC code, we will show that pseudocodewords in the convolutional code give pseudocodewords when projected onto an underlying QC code.

Note that all codes in this paper, also if not always explicitly stated, will be binary linear codes. Moreover, note that a parity-check matrix $\boldsymbol{H} \in \mathbb{F}_2^{m\times n}$ for a binary linear block code C of length n and dimension k must satisfy $\operatorname{rank}(\boldsymbol{H}) = n - k$, and usually the parity-check matrix $\boldsymbol{H}$ is chosen such that $m = n-k$. However, in the case of LDPC codes, it sometimes makes sense to use a parity-check matrix where m is strictly larger than $n - k$.

A. Quasi-Cyclic (QC) Block Codes

A binary linear block code whose length n factorizes as $n \triangleq r \cdot L$ for some integers $r \geq 2$ and $L \geq 1$ and whose set of codewords is invariant under cyclic shifts by L positions is called a quasi-cyclic (QC) code with period L. It follows that such a QC code $\overline{\mathsf{C}}_{\mathrm{QC}}^{(r)}$ can be described by a parity-check matrix of the form

$$\overline{\boldsymbol{H}}_{\mathrm{QC}}^{(r)} \triangleq \begin{bmatrix} \boldsymbol{H}_0 & \boldsymbol{H}_{r-1} & \cdots & \cdots & \boldsymbol{H}_2 & \boldsymbol{H}_1 \\ \boldsymbol{H}_1 & \boldsymbol{H}_0 & \cdots & \cdots & \boldsymbol{H}_3 & \boldsymbol{H}_2 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ \boldsymbol{H}_{r-2} & \boldsymbol{H}_{r-3} & \cdots & \cdots & \boldsymbol{H}_0 & \boldsymbol{H}_{r-1} \\ \boldsymbol{H}_{r-1} & \boldsymbol{H}_{r-2} & \cdots & \cdots & \boldsymbol{H}_1 & \boldsymbol{H}_0 \end{bmatrix} \tag{2}$$

with submatrices $\boldsymbol{H}_s \in \mathbb{F}_2^{J\times L}$, $s = 0, 1, \ldots, r-1$, where J is some positive integer. The generic variable name for a codeword in $\overline{\mathsf{C}}_{\mathrm{QC}}^{(r)}$ will be $\overline{\boldsymbol{c}}$.

Example 4: Consider the code $\overline{\mathsf{C}}_{\mathrm{QC}}^{(r)}$ described by the parity-check matrix

$$\overline{\boldsymbol{H}}_{\mathrm{QC}}^{(r)} = \left[\begin{array}{ccc|ccc|ccc} 1&1&0&0&0&0&0&0&0\\ 0&1&0&1&0&0&0&0&1\\ \hline 0&0&0&1&1&0&0&0&0\\ 0&0&1&0&1&0&1&0&0\\ \hline 0&0&0&0&0&0&1&1&0\\ 1&0&0&0&0&1&0&1&0 \end{array}\right].$$

Clearly, $\boldsymbol{H}$ has the form in (2) with $r = 3$, $J = 2$, $L = 3$, $n = r \cdot L = 9$, and

$$\boldsymbol{H}_0 \triangleq \begin{bmatrix} 1&1&0\\ 0&1&0 \end{bmatrix}, \boldsymbol{H}_1 \triangleq \begin{bmatrix} 0&0&0\\ 0&0&1 \end{bmatrix}, \boldsymbol{H}_2 \triangleq \begin{bmatrix} 0&0&0\\ 1&0&0 \end{bmatrix}$$

and therefore gives a QC code. A possible codeword $\overline{\boldsymbol{c}}$ in $\overline{\mathsf{C}}_{\mathrm{QC}}^{(r)}$ is, e.g.,

$$\overline{\boldsymbol{c}} = (1,1,1,\ 1,1,1,\ 0,0,0). \qquad \square$$

With a certain choice of row and column permutations of $\overline{\boldsymbol{H}}_{\mathrm{QC}}^{(r)}$ we can obtain a parity-check matrix $\boldsymbol{H}_{\mathrm{QC}}^{(r)}$ of a new code, henceforth called $\mathsf{C}_{\mathrm{QC}}^{(r)}$, that is of interest because of its particular description. Namely, let the parity-check matrix $\boldsymbol{H}_{\mathrm{QC}}^{(r)}$ be obtained from $\overline{\boldsymbol{H}}_{\mathrm{QC}}^{(r)}$ as follows: we start with $\overline{\boldsymbol{H}}_{\mathrm{QC}}^{(r)}$ and we take the first row in the first block of J rows, the first row in the second block of J rows, etc., then the second row in the first block, the second row in the second block, etc., and then we apply a similar procedure for the columns, i.e., we take the first column in the first block of L columns, the first column of the second block of L columns, etc., then the second column in the first block, the second column in the second block, etc. The generic variable name for a codeword in $\mathsf{C}_{\mathrm{QC}}^{(r)}$ will be $\boldsymbol{c}$. It is clear that by suitable permutation of a codeword $\overline{\boldsymbol{c}} \in \overline{\mathsf{C}}_{\mathrm{QC}}^{(r)}$ we obtain the corresponding codeword $\boldsymbol{c} \in \mathsf{C}_{\mathrm{QC}}^{(r)}$.

Example 5: Applying this procedure to the code $\overline{\mathsf{C}}_{\mathrm{QC}}^{(r)}$ in Example 4 we obtain the code $\mathsf{C}_{\mathrm{QC}}^{(r)}$ with parity-check matrix

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)} = \left[\begin{array}{ccc|ccc|ccc} 1&0&0&1&0&0&0&0&0\\ 0&1&0&0&1&0&0&0&0\\ 0&0&1&0&0&1&0&0&0\\ \hline 0&1&0&1&0&0&0&0&1\\ 0&0&1&0&1&0&1&0&0\\ 1&0&0&0&0&1&0&1&0 \end{array}\right] = \begin{bmatrix} \boldsymbol{I}_0 & \boldsymbol{I}_0 & \boldsymbol{0} \\ \boldsymbol{I}_2 & \boldsymbol{I}_0 & \boldsymbol{I}_1 \end{bmatrix}$$

where $\boldsymbol{I}_s$ is an $r \times r$ identity matrix whose columns have been left-cyclically shifted s times. After suitable permutation of the codeword $\overline{\boldsymbol{c}} \in \overline{\mathsf{C}}_{\mathrm{QC}}^{(r)}$ in Example 4 we obtain the codeword $\boldsymbol{c} = (1,1,0,\ 1,1,0,\ 1,1,0) \in \mathsf{C}_{\mathrm{QC}}^{(r)}$. $\square$

The Tanner graph of the code $\overline{\mathsf{C}}_{\mathrm{QC}}^{(r)}$ is shown in Fig. 1 (right) and is therefore a cubic cover of the Tanner graph of the code in Example 1. Note that some, but not all, finite covers of a code are QC codes.

This will actually be a recurring theme in this paper: graph covers are used in two different contexts. First, we will use graph covers to analyze MPI decoding algorithms (see, e.g., the introductory comments in Section I), and second, we will use graph covers to construct Tanner graphs that define codes that are of interest by themselves.

In general, it can easily be seen that the matrix $\boldsymbol{H}_{\mathrm{QC}}^{(r)}$ is composed of a $J \times L$ array of $r \times r$ circulant matrices over $\mathbb{F}_2$. This suggests a more compact description of $\boldsymbol{H}_{\mathrm{QC}}^{(r)}$. Namely,

replacing a circulant submatrix like $\boldsymbol{A} = \sum_{s=0}^{r-1} a_s \boldsymbol{I}_s$ by the polynomial $a(X) = \sum_{s=0}^{r-1} a_s X^s$, we obtain the matrix $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) \in (\mathbb{F}_2[X]/\langle X^r-1\rangle)^{J\times L}$, in the following called a polynomial parity-check matrix for $\mathsf{C}_{\mathrm{QC}}^{(r)}$. Similarly, codewords can be described by polynomial vectors. Namely, with the codeword $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_2^n$ in $\mathsf{C}_{\mathrm{QC}}^{(r)}$ we can associate the polynomial codeword

$$\boldsymbol{c}(X) = \big(c_0(X), c_1(X), \ldots, c_{L-1}(X)\big) \in (\mathbb{F}_2[X]/\langle X^r-1\rangle)^L$$

where

$$c_\ell(X) = \sum_{s=0}^{r-1} c_{\ell r+s} X^s$$

for $\ell = 0, 1, \ldots, L-1$. With these definitions, and the well-known equivalence of multiplying an $r \times r$ circulant matrix by a vector of length r on the one hand and the multiplication of two polynomials modulo X^r-1 on the other, it follows that

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)} \cdot \boldsymbol{c}^{\mathsf{T}} = \boldsymbol{0}^{\mathsf{T}} \pmod 2$$

if and only if

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) \cdot \boldsymbol{c}^{\mathsf{T}}(X) \bmod_{\mathbb{F}_2} (X^r-1) = \boldsymbol{0}^{\mathsf{T}}.$$

Example 6: For the code in Examples 4 and 5 we obtain the polynomial parity-check matrix

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) = \begin{bmatrix} X^0 & X^0 & 0 \\ X^2 & X^0 & X^1 \end{bmatrix}. \tag{3}$$

Moreover, with the codeword $\boldsymbol{c} = (1,1,0,\ 1,1,0,\ 1,1,0) \in \mathsf{C}_{\mathrm{QC}}^{(r)}$ in Example 5 we can associate the polynomial codeword $\boldsymbol{c}(X) = (1{+}X,\ 1{+}X,\ 1{+}X)$. It is easily verified that this polynomial vector indeed satisfies $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) \cdot \boldsymbol{c}^{\mathsf{T}}(X) \bmod_{\mathbb{F}_2} (X^r-1) = \boldsymbol{0}^{\mathsf{T}}$. □

An important equation relating the polynomial parity-check matrix $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)$ to the scalar matrices $\boldsymbol{H}_0, \boldsymbol{H}_1, \ldots, \boldsymbol{H}_{r-1}$ that appear in the description of $\overline{\boldsymbol{H}}_{\mathrm{QC}}^{(r)}$, cf. (2), is given by

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) = \boldsymbol{H}_0 + \boldsymbol{H}_1 X + \cdots + \boldsymbol{H}_{r-1} X^{r-1}.$$

B. Convolutional Codes

Let L be some positive integer and define the right shift by L positions of the semi-infinitely long vector $(\overline{c}_0, \overline{c}_1, \overline{c}_2, \ldots)$ to be the semi-infinitely long vector

$$(\underbrace{0, \ldots, 0}_{L \text{ zeros}}, \overline{c}_0, \overline{c}_1, \overline{c}_2, \ldots).$$

Then, a semi-infinitely long binary linear code that is invariant under shifts by L positions to the right is called a convolutional code with period L. It can easily be seen that such a convolutional code $\overline{\mathsf{C}}_{\mathrm{conv}}$ can be described by a semi-infinitely long parity-check matrix of the form

$$\overline{\boldsymbol{H}}_{\mathrm{conv}} \triangleq \begin{bmatrix} \boldsymbol{H}_0 & & & \\ \boldsymbol{H}_1 & \boldsymbol{H}_0 & & \\ \boldsymbol{H}_2 & \boldsymbol{H}_1 & \boldsymbol{H}_0 & \\ \vdots & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \ddots \\ \boldsymbol{H}_{r-1} & \boldsymbol{H}_{r-2} & \boldsymbol{H}_{r-3} & \ddots \\ & \boldsymbol{H}_{r-1} & \boldsymbol{H}_{r-2} & \ddots \\ & & \boldsymbol{H}_{r-1} & \ddots \\ & & & \ddots \end{bmatrix} \tag{4}$$

where r is some positive integer, $\boldsymbol{H}_s \in \mathbb{F}_2^{J\times L}$ $(s = 0, 1, \ldots, r-1)$ for some positive integer J $(J<L)$, and $R = (L-J)/L$ is the code rate. The generic variable name for a codeword in $\overline{\mathsf{C}}_{\mathrm{conv}}$ will be $\overline{\boldsymbol{c}}$. Moreover, let the syndrome former memory $m_s \le r-1$ of $\overline{\mathsf{C}}_{\mathrm{conv}}$ be the largest integer in $\{0, 1, \ldots, r-1\}$ such that $\boldsymbol{H}_{m_s} \neq \boldsymbol{0}$.

Example 7: Consider the convolutional code $\overline{\mathsf{C}}_{\mathrm{conv}}$ described by the parity-check matrix

$$\overline{\boldsymbol{H}}_{\mathrm{conv}} = \left[\begin{array}{ccc|ccc|ccc|ccc} 1&1&0 & & & & & & & & & \\ 0&1&0 & & & & & & & & & \\ \hline 0&0&0 & 1&1&0 & & & & & & \\ 0&0&1 & 0&1&0 & & & & & & \\ \hline 0&0&0 & 0&0&0 & 1&1&0 & & & \\ 1&0&0 & 0&0&1 & 0&1&0 & & & \\ \hline & & & 0&0&0 & 0&0&0 & \ddots&\ddots&\ddots \\ & & & 1&0&0 & 0&0&1 & \ddots&\ddots&\ddots \\ \hline & & & & & & 0&0&0 & \ddots&\ddots&\ddots \\ & & & & & & 1&0&0 & \ddots&\ddots&\ddots \\ \hline & & & & & & & & & \ddots&\ddots&\ddots \end{array}\right].$$

Clearly, $r \ge 3$, $J = 2$, $L = 3$, and

$$\boldsymbol{H}_0 \triangleq \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix},\ \boldsymbol{H}_1 \triangleq \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ \boldsymbol{H}_2 \triangleq \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

and $\boldsymbol{H}_s = \boldsymbol{0}$ for $3 \le s < r$. Because $\boldsymbol{H}_2 \neq \boldsymbol{0}$ and $\boldsymbol{H}_s = \boldsymbol{0}$ for $s \ge 3$, the syndrome former memory is $m_s = 2$. □

In the same way that a parity-check matrix $\boldsymbol{H}_{\mathrm{QC}}^{(r)}$ of a new code $\mathsf{C}_{\mathrm{QC}}^{(r)}$ was obtained by permuting the rows and the columns of the parity-check matrix $\overline{\boldsymbol{H}}_{\mathrm{QC}}^{(r)}$ of $\overline{\mathsf{C}}_{\mathrm{QC}}^{(r)}$, a parity-check matrix $\boldsymbol{H}_{\mathrm{conv}}$ of a new code $\mathsf{C}_{\mathrm{conv}}$ can be obtained by permuting the rows and the columns of the parity-check matrix $\overline{\boldsymbol{H}}_{\mathrm{conv}}$ of $\overline{\mathsf{C}}_{\mathrm{conv}}$. Of course, $\mathsf{C}_{\mathrm{conv}}$ and $\overline{\mathsf{C}}_{\mathrm{conv}}$ are equivalent codes. The generic variable name for a codeword in $\mathsf{C}_{\mathrm{conv}}$ will be $\boldsymbol{c}$. It is clear that, by suitable permutation of a codeword $\overline{\boldsymbol{c}} \in \overline{\mathsf{C}}_{\mathrm{conv}}$, we obtain the corresponding codeword $\boldsymbol{c} \in \mathsf{C}_{\mathrm{conv}}$.

Example 8: Applying this procedure to the code $\overline{\mathsf{C}}_{\mathrm{conv}}$ in Example 7 we obtain the code $\mathsf{C}_{\mathrm{conv}}$ with the parity-check matrix

shown at the bottom of the page, where $\boldsymbol{I}'_s$ is a semi-infinitely long quadratic matrix with ones in the sth subdiagonal and zeros everywhere else. □

In general, it can easily be seen that the matrix $\boldsymbol{H}_{\mathrm{conv}}$ is composed of a $J \times L$ array of semi-infinitely long quadratic Toeplitz matrices over $\mathbb{F}_2$. This suggests a more compact description of $\boldsymbol{H}_{\mathrm{conv}}$. Namely, replacing a Toeplitz submatrix like $\boldsymbol{A} = \sum_{s \geq 0} a_s \boldsymbol{I}'_s$ by the polynomial $a(D) = \sum_{s \geq 0} a_s D^s$, we obtain the matrix $\boldsymbol{H}_{\mathrm{conv}}(D) \in (\mathbb{F}_2[D])^{J \times L}$, in the following called a polynomial parity-check matrix for $\mathsf{C}_{\mathrm{conv}}$. Similarly, codewords can be described by polynomial vectors. Namely, with the codeword $\boldsymbol{c} \in (\mathbb{F}_2^{\infty})^L$ in $\mathsf{C}_{\mathrm{conv}}$ we can associate the polynomial codeword

$$\boldsymbol{c}(D) = \big(c_0(D), c_1(D), \ldots c_{L-1}(D)\big) \in (\mathbb{F}_2[D])^L.$$

With these definitions,[12] and the well-known equivalence of multiplying a semi-infinitely long Toeplitz matrix by a semi-infinitely long vector on the one hand and the multiplication of two polynomials on the other, it follows that

$$\boldsymbol{H}_{\mathrm{conv}} \cdot \boldsymbol{c}^{\mathsf{T}} = \boldsymbol{0}^{\mathsf{T}} \pmod 2$$

if and only if

$$\boldsymbol{H}_{\mathrm{conv}}(D) \cdot \boldsymbol{c}^{\mathsf{T}}(D) = \boldsymbol{0}^{\mathsf{T}} \pmod 2.$$

Example 9: For the code in Examples 7 and 8 we obtain the polynomial parity-check matrix

$$\boldsymbol{H}_{\mathrm{conv}}(D) = \begin{bmatrix} D^0 & D^0 & 0 \\ D^2 & D^0 & D^1 \end{bmatrix}. \tag{5}$$

□

Similar to the case of QC codes, an important equation relating the polynomial parity-check matrix $\boldsymbol{H}_{\mathrm{conv}}(D)$ to the scalar matrices $\boldsymbol{H}_0, \boldsymbol{H}_1, \ldots, \boldsymbol{H}_{r-1}$ that appear in the description of $\overline{\boldsymbol{H}}_{\mathrm{conv}}$, cf. (4), is given by

$$\boldsymbol{H}_{\mathrm{conv}}(D) = \boldsymbol{H}_0 + \boldsymbol{H}_1 D + \cdots + \boldsymbol{H}_{r-1} D^{r-1}.$$

[12]Strictly speaking, codewords are power series rather than polynomials. However, since the paper is primarily concerned with low-Hamming-weight codewords and low pseudoweight pseudocodewords, we use the terminology polynomial codewords and polynomial pseudocodewords.

We note that, instead of using semi-infinite parity-check matrices to describe convolutional codes, we could also have used infinite matrices, where we impose the condition that we only consider codewords that have zero entries for negative time indices.

C. A Link Between QC Block Codes and Convolutional Codes

Given the close resemblance of the contents of Section II-A and Section II-B, it is not surprising that there is a natural connection between QC block codes and convolutional codes (see also [12], [14]–[16], [22]). More precisely, with any QC block code $\mathsf{C}_{\mathrm{QC}}^{(r)}$ of length $r \cdot L$, given by a $J \times L$ polynomial matrix parity-check matrix

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) = \boldsymbol{H}_0 + \boldsymbol{H}_1 X + \cdots + \boldsymbol{H}_{r-1} X^{r-1}$$

with polynomial operations performed modulo $X^r - 1$, we can associate a rate $(L-J)/L$ convolutional code $\mathsf{C}_{\mathrm{conv}}$ given by the $J \times L$ polynomial parity-check matrix $\boldsymbol{H}_{\mathrm{conv}}(D) \in (\mathbb{F}_2[D])^{J \times L}$ with

$$\boldsymbol{H}_{\mathrm{conv}}(D) = \boldsymbol{H}_{\mathrm{QC}}^{(r)}(D) = \boldsymbol{H}_0 + \boldsymbol{H}_1 D + \cdots + \boldsymbol{H}_{r-1} D^{r-1}$$

where the change of indeterminates indicates the lack of modulo $D^r - 1$ operations. On the other hand, in the same way as we obtained $\mathsf{C}_{\mathrm{conv}}$ from $\mathsf{C}_{\mathrm{QC}}^{(r)}$, we can, upon choosing a positive integer r, obtain a QC code $\mathsf{C}_{\mathrm{QC}}^{(r)}(X)$ from a convolutional code $\mathsf{C}_{\mathrm{conv}}(D)$.[13]

In the following, we will say that the parity-check matrix $\overline{\boldsymbol{H}}_{\mathrm{conv}}$ in (4) is obtained by "unwrapping" the submatrices of the parity-check matrix $\overline{\boldsymbol{H}}_{\mathrm{QC}}^{(r)}$ in (2), and that the parity-check matrix $\overline{\boldsymbol{H}}_{\mathrm{QC}}^{(r)}$ is obtained by "wrapping" the submatrices of the parity-check matrix $\overline{\boldsymbol{H}}_{\mathrm{conv}}$.

Example 10: Considering again the QC code in Example 6 and the convolutional code in Example 9, we see that their polynomial parity-check matrices are connected by $\boldsymbol{H}_{\mathrm{conv}}(D) = \boldsymbol{H}_{\mathrm{QC}}^{(r)}(D)$. □

Example 11: Consider the QC code $\mathsf{C}_{\mathrm{QC}}^{(r)}$ defined by the polynomial parity-check matrix

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) = \begin{pmatrix} X^1 & X^2 & X^4 \\ X^6 & X^5 & X^3 \end{pmatrix}$$

[13]We will say more about the choice of r in Assumption 23.

$$\boldsymbol{H}_{\mathrm{conv}} = \left[\begin{array}{cccc|cccc|cccc} 1 & 0 & 0 & & 1 & 0 & 0 & & 0 & 0 & 0 & \\ 0 & 1 & 0 & \ddots & 0 & 1 & 0 & \ddots & 0 & 0 & 0 & \ddots \\ 0 & 0 & 1 & \ddots & 0 & 0 & 1 & \ddots & 0 & 0 & 0 & \ddots \\ & \ddots & \ddots & \ddots & & \ddots & \ddots & \ddots & & \ddots & \ddots & \ddots \\ \hline 0 & 0 & 0 & & 1 & 0 & 0 & & 0 & 0 & 0 & \\ 0 & 0 & 0 & \ddots & 0 & 1 & 0 & \ddots & 1 & 0 & 0 & \ddots \\ 1 & 0 & 0 & \ddots & 0 & 0 & 1 & \ddots & 0 & 1 & 0 & \ddots \\ & \ddots & \ddots & \ddots & & \ddots & \ddots & \ddots & & \ddots & \ddots & \ddots \end{array}\right] = \begin{bmatrix} \boldsymbol{I}'_0 & \boldsymbol{I}'_0 & \boldsymbol{0} \\ \boldsymbol{I}'_2 & \boldsymbol{I}'_0 & \boldsymbol{I}'_1 \end{bmatrix}$$

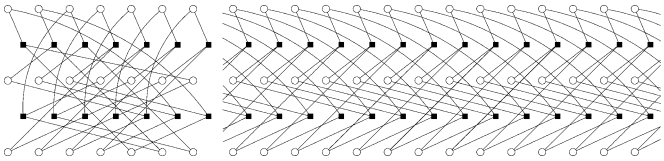

Fig. 4. Left: The finite Tanner graph of the QC code in Example 11 defined by $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)$. (Here $r = 7$, and so, because $J = 2$ and $L = 3$, there are $Jr = 14$ check nodes and $Lr = 21$ variable nodes.) Right: Part of the semi-infinite Tanner graph of the convolutional code in Example 11 defined by $\boldsymbol{H}_{\mathrm{conv}}(D)$.

and the convolutional code $\mathsf{C}_{\mathrm{conv}}$ defined by the polynomial parity-check matrix

$$\boldsymbol{H}_{\mathrm{conv}}(D) = \begin{pmatrix} D^1 & D^2 & D^4 \\ D^6 & D^5 & D^3 \end{pmatrix}.$$

Clearly, $\boldsymbol{H}_{\mathrm{conv}}(D)$ and $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)$ are connected by $\boldsymbol{H}_{\mathrm{conv}}(D) = \boldsymbol{H}_{\mathrm{QC}}^{(r)}(D)$ for $r \geq 7$. The effect that the "wrapping"/"unwrapping" process has on the parity-check matrices $\overline{\boldsymbol{H}}_{\mathrm{QC}}^{(r)}$ and $\overline{\boldsymbol{H}}_{\mathrm{conv}}$ can also be observed in the Tanner graph representation of the two parity-check matrices in Fig. 4. □

We turn now our attention to codewords in these codes. For any nonzero codeword $\boldsymbol{c}(D)$ with finite support in the convolutional code, its r *wrap-around*, defined by the vector

$$\boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X^r - 1) \in \left(\mathbb{F}_2[X]/\langle X^r - 1\rangle\right)^L,$$

is a codeword in the associated QC-code, since

$$\begin{aligned} \boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) \cdot \left(\boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X^r - 1)\right)^{\mathsf{T}} \bmod_{\mathbb{F}_2} (X^r - 1) \\ = \left(\boldsymbol{H}_{\mathrm{conv}}(X) \bmod_{\mathbb{F}_2} (X^r - 1)\right) \\ \cdot \left(\boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X^r - 1)\right)^{\mathsf{T}} \bmod_{\mathbb{F}_2} (X^r - 1) \\ = \boldsymbol{H}_{\mathrm{conv}}(X) \cdot \boldsymbol{c}^{\mathsf{T}}(X) \bmod_{\mathbb{F}_2} (X^r - 1) \\ = \boldsymbol{0}^{\mathsf{T}}. \end{aligned}$$

Example 12: Consider the polynomial vector

$$\boldsymbol{c}(D) \triangleq \begin{pmatrix} D + D^3 \\ D + D^3 \\ 1 + D^4 \end{pmatrix}^{\mathsf{T}} \in \mathrm{Nullsp}\left(\boldsymbol{H}_{\mathrm{conv}}(D)\right)$$

which is a codeword in the convolutional code $\mathsf{C}_{\mathrm{conv}}$ described in Example 9.

- For $r = 3$, the wrap-around

  $$\boldsymbol{c}^{(3)}(X) \triangleq \boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X^r - 1) \in \left(\mathbb{F}_2[X]/\langle X^r - 1\rangle\right)^3$$

  gives the polynomial codeword

  $$\boldsymbol{c}^{(3)}(X) \triangleq (1 + X, 1 + X, 1 + X) \in \mathrm{Nullsp}\left(\boldsymbol{H}_{\mathrm{QC}}^{(3)}(X)\right)$$

  and the equivalent codeword

  $$\boldsymbol{c}^{(3)} \triangleq (1,1,0,\ 1,1,0,\ 1,1,0) \in \mathrm{Nullsp}\left(\boldsymbol{H}_{\mathrm{QC}}^{(3)}\right).$$

  Here, the polynomial parity-check matrix $\boldsymbol{H}_{\mathrm{QC}}^{(3)}(X)$ and the binary parity-check matrix $\boldsymbol{H}_{\mathrm{QC}}^{(3)}$ were discussed in Examples 6 and 5, respectively.
- For $r = 1$, the wraparound

  $$\boldsymbol{c}^{(1)}(X) \triangleq \boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X - 1) \in \left(\mathbb{F}_2[X]/\langle X^r - 1\rangle\right)^3$$

  gives the polynomial codeword

  $$\boldsymbol{c}^{(1)}(X) \triangleq (0,0,0) \in \mathrm{Nullsp}\left(\boldsymbol{H}_{\mathrm{QC}}^{(1)}(X)\right)$$

  which is equivalent to the codeword

  $$\boldsymbol{c}^{(1)}(X) \triangleq (0,0,0) \in \mathrm{Nullsp}\left(\boldsymbol{H}_{\mathrm{QC}}^{(1)}\right).$$

  Here, the parity-check matrix $\boldsymbol{H}_{\mathrm{QC}}^{(1)}$ equals the parity-check matrix $\boldsymbol{H}$ that was discussed in Example 1. □

One can show that the Hamming weights of the codeword $\boldsymbol{c}(D)$ and its wrapped-around version $\boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X^r - 1)$ are linked by the inequality

$$w_{\mathrm{H}}\left(\boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X^r - 1)\right) \leq w_{\mathrm{H}}(\boldsymbol{c}(D))$$

which gives the inequality [14], [15]

$$d_{\min}\left(\mathsf{C}_{\mathrm{QC}}^{(r)}\right) \leq d_{\mathrm{free}}\left(\mathsf{C}_{\mathrm{conv}}\right), \quad \text{for all } r \geq m_s + 1$$

where we have assumed that all codes are nontrivial. (As we will also see in the proof of Theorem 13, when establishing this inequality one needs to take special care of codewords that map to the all-zero codeword.)

We will also make use of the following inequalities.

Theorem 13: Let r be a positive integer, let $\mathsf{C}_{\mathrm{conv}}$ be a convolutional code defined by $\boldsymbol{H}_{\mathrm{conv}}(D)$, and for any nonnegative integer k let $\mathsf{C}_{\mathrm{QC}}^{(2^k r)}$ be a QC code defined by

$$\boldsymbol{H}_{\mathrm{QC}}^{(2^k r)}(X) \triangleq \boldsymbol{H}_{\mathrm{conv}}(X) \bmod_{\mathbb{F}_2} (X^{2^k r} - 1).$$

Then

$$d_{\min}\left(\mathsf{C}_{\mathrm{QC}}^{(r)}\right) \leq d_{\min}\left(\mathsf{C}_{\mathrm{QC}}^{(2r)}\right) \leq d_{\min}\left(\mathsf{C}_{\mathrm{QC}}^{(4r)}\right) \leq \cdots$$

and

$$\lim_{k \to \infty} d_{\min}\left(\mathsf{C}_{\mathrm{QC}}^{(2^k r)}\right) = d_{\mathrm{free}}\left(\mathsf{C}_{\mathrm{conv}}\right).$$

Proof: See Appendix A. □

We conclude this section by mentioning that there are also other ways to unwrap a QC code in order to obtain a convolutional code; see, for example, the unwrapping method that was discussed in [23] based on ideas from [24].

D. The Fundamental Polytope and Cone of the Parity-Check Matrices of QC and Convolutional Codes

In this subsection, we introduce our main objects of study, the fundamental polytope and cone of a parity-check matrix [6], [7], [17], [18].

Definition 14 (see [6], [7], [17], [18]): Let $\boldsymbol{H}$ be a binary matrix of size $m \times n$, let $\mathcal{I} \triangleq \mathcal{I}(\boldsymbol{H}) \triangleq \{0, \ldots, n-1\}$ be the set of column indices of $\boldsymbol{H}$, and let $\mathcal{J} \triangleq \mathcal{J}(\boldsymbol{H}) \triangleq \{0, \ldots, m-1\}$

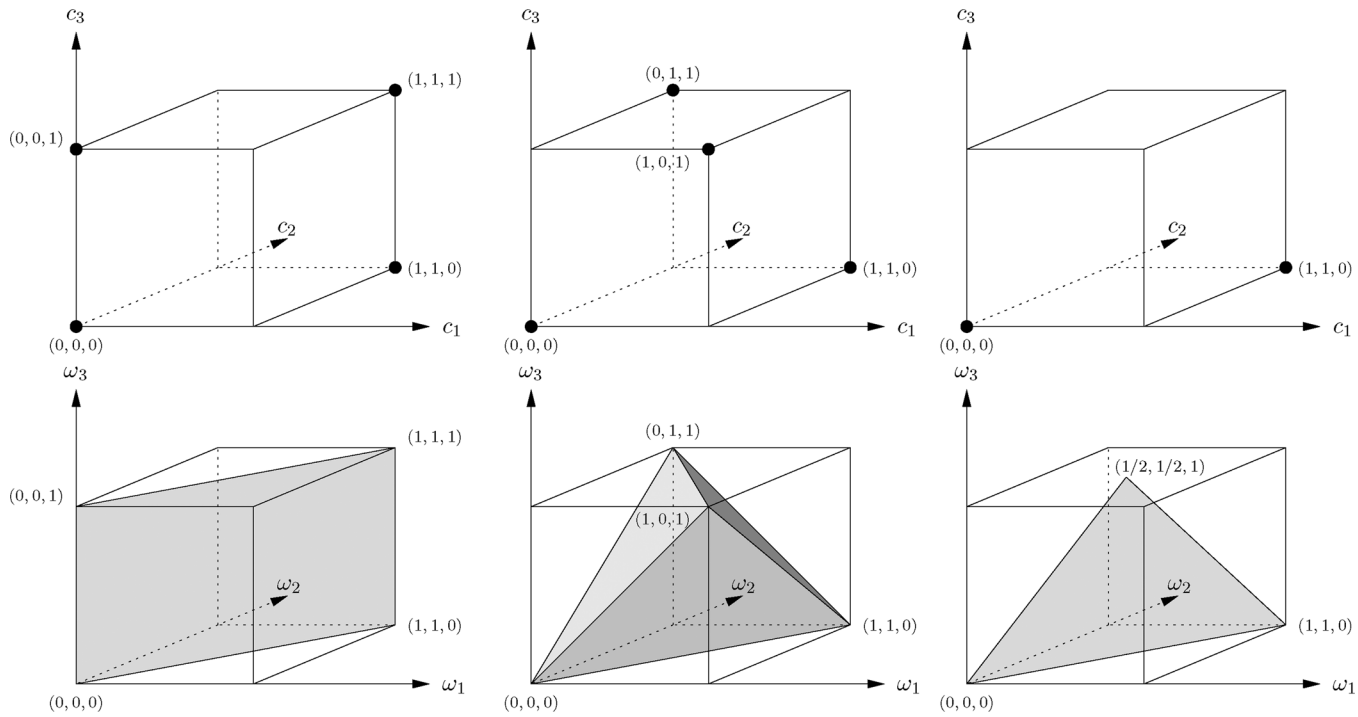

Fig. 5. Visualization of the codes and polytopes that appear in Example 15. Top row, left to right: code C_1, code C_2, and code $\mathsf{C} = \mathsf{C}_1 \cap \mathsf{C}_2$. Bottom row, left to right: convex hull $\mathrm{ConvHull}(\mathsf{C}_1)$ of code C_1, convex hull $\mathrm{ConvHull}(\mathsf{C}_2)$ of code C_2, and fundamental polytope $\mathcal{P}(\boldsymbol{H}) = \mathrm{ConvHull}(\mathsf{C}_1) \cap \mathrm{ConvHull}(\mathsf{C}_2)$.

be the set of row indices of $\boldsymbol{H}$, respectively. For each $j \in \mathcal{J}$, let $\mathcal{I}_j \triangleq \mathcal{I}_j(\boldsymbol{H}) \triangleq \{i \in \mathcal{I} \mid h_{ji} = 1\}$ be the support of the jth row of $\boldsymbol{H}$. The fundamental polytope $\mathcal{P}(\boldsymbol{H})$ of $\boldsymbol{H}$ is then defined as

$$\mathcal{P}\boldsymbol{H} \triangleq \bigcap_{j=1}^{m} \mathrm{ConvHull}(\mathsf{C}_j)$$

with

$$\mathsf{C}_j \triangleq \left\{\boldsymbol{c} \in \{0,1\}^n \;\middle|\; \boldsymbol{r}_j \boldsymbol{c}^{\mathsf{T}} = 0 \pmod 2\right\}$$

where $\boldsymbol{r}_j$ is the jth row of $\boldsymbol{H}$, and $\mathrm{ConvHull}(\mathsf{C}_j)$ is the convex hull of C_j, defined as the set of convex combinations of codewords in C_j when seen as points in $\mathbb{R}^n$. □

Expressed in terms of inequalities, the fundamental polytope can be described as follows. A vector $\boldsymbol{\omega} = (\omega_0, \ldots, \omega_{n-1}) \in \mathbb{R}^n$ is in the fundamental polytope $\mathcal{P}(\boldsymbol{H})$ if and only if

- for all $i \in \mathcal{I}(\boldsymbol{H})$
$$0 \le \omega_i \le 1; \tag{6}$$
- for all $j \in \mathcal{J}(\boldsymbol{H})$ and for all odd-sized sets $\mathcal{S} \subseteq \mathcal{I}_j(\boldsymbol{H})$
$$\sum_{i \in \mathcal{S}} \omega_i - \sum_{i \in (\mathcal{I}_j \setminus \mathcal{S})} \omega_i \le |\mathcal{S}| - 1. \tag{7}$$

Example 15: Consider the code C from Example 1 that is described by the parity-check matrix $\boldsymbol{H}$ in (1). The corresponding codes C_1 and C_2 are, respectively

$$\mathsf{C}_1 = \{(0,0,0),\ (1,1,0),\ (0,0,1),\ (1,1,1)\}$$
$$\mathsf{C}_2 = \{(0,0,0),\ (1,1,0),\ (0,1,1),\ (1,0,1)\}.$$

They are depicted as $\mathbb{R}^3$ point sets in the top row of Fig. 5, along with the code C, where

$$\mathsf{C} = \mathsf{C}_1 \cap \mathsf{C}_2 = \{(0,0,0),\ (1,1,0)\}.$$

On the other hand, the bottom row of Fig. 5 shows the convex hull $\mathrm{ConvHull}(\mathsf{C}_1)$ of C_1, the convex hull $\mathrm{ConvHull}(\mathsf{C}_2)$ of C_2, and the fundamental polytope $\mathcal{P}(\boldsymbol{H})$, where

$$\begin{aligned}\mathcal{P}(\boldsymbol{H}) &= \mathrm{ConvHull}(\mathsf{C}_1) \cap \mathrm{ConvHull}(\mathsf{C}_2)\\ &= \left\{ \boldsymbol{\omega} \in \mathbb{R}^3 \;\middle|\; \begin{array}{rl} 0 \le \omega_1 \le 1 & +\omega_1 - \omega_2 \le 0 \\ 0 \le \omega_2 \le 1 & -\omega_1 + \omega_2 \le 0 \\ 0 \le \omega_3 \le 1 & +\omega_1 - \omega_2 - \omega_3 \le 0 \\ & -\omega_1 + \omega_2 - \omega_3 \le 0 \\ & -\omega_1 - \omega_2 + \omega_3 \le 0 \\ & +\omega_1 + \omega_2 + \omega_3 \le 2 \end{array} \right\}\\ &= \mathrm{ConvHull}\left(\left\{(0,0,0),\ (1,1,0),\ \left(\frac{1}{2},\frac{1}{2},1\right)\right\}\right).\end{aligned}$$

We clearly see that the fundamental polytope $\mathcal{P}(\boldsymbol{H})$ is strictly larger than the convex hull $\mathrm{ConvHull}(\mathsf{C})$ of the code C. □

Note that the fundamental polytope is a function of a parity-check matrix. So different parity-check matrices for the same code can yield different fundamental polytopes. Note also that the fundamental polytope is usually an n-dimensional object. However, when the parity-check matrix contains rows of weight one or two, as happens in the above example, then there is a loss in dimensionality.

For binary-input output-symmetric channels and binary linear codes, we can assume for analysis purposes that the all-zero codeword was sent, and so it is sufficient to study the fundamental polytope around the origin, which essentially amounts to studying the conic hull of the fundamental polytope.

Definition 16 (see [6], [7], [17], [18]): The fundamental cone $\mathcal{K}(\boldsymbol{H})$ of $\boldsymbol{H}$ is defined to be

$$\mathcal{K}(\boldsymbol{H}) = \mathrm{ConicHull}(\mathcal{P}(\boldsymbol{H}))$$

i.e., the conic hull of the fundamental polytope $\mathcal{P}(\boldsymbol{H})$, which is the object that is obtained by stretching the fundamental polytope $\mathcal{P}(\boldsymbol{H})$ to infinity, with the stretching center at the origin. Note that if $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$, then also $\alpha \cdot \boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$ for any real $\alpha > 0$. Moreover, for any $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H})$, there exists an $\alpha > 0$ (in fact, a whole interval of α's) such that $\alpha \cdot \boldsymbol{\omega} \in \mathcal{P}(\boldsymbol{H})$.

Expressed in terms of inequalities, the fundamental cone can be described as follows. A vector $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n) \in \mathbb{R}^n$ is in the fundamental polytope $\mathcal{K}(\boldsymbol{H})$ if and only if

- for all $i \in \mathcal{I}(\boldsymbol{H})$
$$0 \leq \omega_i; \tag{8}$$
- for all $j \in \mathcal{J}(\boldsymbol{H})$ and for all $i' \in \mathcal{I}_j(\boldsymbol{H})$
$$\omega_{i'} \leq \sum_{i \in (\mathcal{I}_j \setminus \{i'\})} \omega_i. \tag{9}$$

Example 17: Consider again the code C from Example 1 that is described by the parity-check matrix $\boldsymbol{H}$ in (1). The inequalities describing the fundamental cone $\mathcal{K}(\boldsymbol{H})$ are obtained by taking the homogeneous inequalities from the fundamental polytope description in Example 15, i.e.,

$$\begin{aligned}\mathcal{K}(\boldsymbol{H}) &= \mathrm{ConicHull}(\mathsf{C}_1) \cap \mathrm{ConicHull}(\mathsf{C}_2)\\ &= \left\{ \boldsymbol{\omega} \in \mathbb{R}^3 \left| \begin{array}{rl} & +\omega_1 - \omega_2 \leq 0 \\ 0 \leq \omega_1 & -\omega_1 + \omega_2 \leq 0 \\ 0 \leq \omega_2 & +\omega_1 - \omega_2 - \omega_3 \leq 0 \\ 0 \leq \omega_3 & -\omega_1 + \omega_2 - \omega_3 \leq 0 \\ & -\omega_1 - \omega_2 + \omega_3 \leq 0 \end{array} \right. \right\}\\ &= \mathrm{ConicHull}\left(\left\{ (1,1,0),\ \left(\frac{1}{2}, \frac{1}{2}, 1\right) \right\} \right). \end{aligned}$$ □

We have already mentioned in Section I that the fundamental polytope is the central object for LP decoding. We have also mentioned that the fundamental polytope says a lot about the codewords that live in finite covers of a Tanner graph. Let us now be more precise about this latter statement.

Definition 18: Let $\mathcal{G}$ be the Tanner graph of a length-n code C defined by the parity-check matrix $\boldsymbol{H}$. For some positive integer M, consider an M-cover $\tilde{\mathcal{G}}$ of $\mathcal{G}$. Let

$$\tilde{\boldsymbol{c}} = \big(\tilde{c}_{0,0}, \ldots, \tilde{c}_{0,M-1},\ \ldots\ \tilde{c}_{n-1,0}, \ldots, \tilde{c}_{n-1,M-1}\big)$$

be a codeword in $\tilde{\mathcal{G}}$, where $\tilde{c}_{i,m}$ corresponds to the mth copy of the ith variable node in $\mathcal{G}$. The (scaled) pseudocodeword associated with $\tilde{\boldsymbol{c}}$ is the rational vector

$$\boldsymbol{\omega}(\tilde{\boldsymbol{c}}) = \big(\omega_0(\tilde{\boldsymbol{c}}),\ \ldots\ , \omega_{n-1}(\tilde{\boldsymbol{c}})\big)$$

with entries

$$\omega_i(\tilde{\boldsymbol{c}}) = \frac{1}{M} \sum_{m=0}^{M-1} \tilde{c}_{i,m}$$

where the sum is taken in $\mathbb{R}$ (not in $\mathbb{F}_2$). The vector $M \cdot \boldsymbol{\omega}(\tilde{\boldsymbol{c}})$ (with purely integer entries) will be called the unscaled pseudocodeword associated with $\tilde{\boldsymbol{c}}$.

Remark 19: The main statement in [6], [7] is the following. Let $\mathcal{G}$ be the Tanner graph of a length-n code C that is defined by the parity-check matrix $\boldsymbol{H}$. Then the set of all (scaled) pseudocodewords that are associated with all possible codewords in all possible finite covers of $\mathcal{G}$ equals $\mathcal{P}(\boldsymbol{H}) \cap \mathbb{Q}^n$. Because $\mathcal{P}(\boldsymbol{H}) \cap \mathbb{Q}^n$ is dense in $\mathcal{P}(\boldsymbol{H})$ and because all vertices of $\mathcal{P}(\boldsymbol{H})$ are in $\mathbb{Q}^n$, it is sufficient to consider $\mathcal{P}(\boldsymbol{H})$ instead of the more complicated $\mathcal{P}(\boldsymbol{H}) \cap \mathbb{Q}^n$.

In the following, we will call any vector in the fundamental cone a pseudocodeword, and two pseudocodewords that are equal up to a positive scaling constant will be considered to be equivalent. These conventions are motivated by the fact that for any $\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H}) \setminus \mathcal{P}(\boldsymbol{H})$ we can always find a scaling factor $\alpha > 0$ such that $\alpha \cdot \boldsymbol{\omega} \in \mathcal{P}(\boldsymbol{H})$. (In any case, in the following it will be clear from the context if a pseudocodeword is meant to be a point in the fundamental polytope $\mathcal{P}(\boldsymbol{H})$ or merely a point in the fundamental cone $\mathcal{K}(\boldsymbol{H})$.)

Example 20: Consider again the code C from Example 1 that is described by the parity-check matrix $\boldsymbol{H}$ in (1) and whose Tanner graph $\mathcal{G}$ was shown in Fig. 1 (left). As discussed in Example 2 , $\tilde{\boldsymbol{c}} = (1,1,0,\ 1,1,0,\ 1,1,0)$ is a codeword in the triple cover of $\mathcal{G}$ shown in Fig. 1 (right). Its associated scaled pseudocodeword is $\boldsymbol{\omega}(\tilde{\boldsymbol{c}}) = (2/3, 2/3, 2/3)$ and its associated unscaled pseudocodeword is $3 \cdot \boldsymbol{\omega}(\tilde{\boldsymbol{c}}) = (2,2,2)$. It is an easy matter to verify that $\boldsymbol{\omega}(\tilde{\boldsymbol{c}})$ is indeed in the fundamental polytope $\mathcal{P}(\boldsymbol{H})$, whose defining inequalities were displayed in Example 15. Moreover, it is equally easy to verify that both $\boldsymbol{\omega}(\tilde{\boldsymbol{c}})$ and $3 \cdot \boldsymbol{\omega}(\tilde{\boldsymbol{c}})$ are in the fundamental cone $\mathcal{K}(\boldsymbol{H})$, whose defining inequalities were displayed in Example 17. □

From (8) and (9) it follows that for any parity-check matrix $\boldsymbol{H}$ there is a matrix $\boldsymbol{K}$ such that

$$\boldsymbol{\omega} \in \mathcal{K}(\boldsymbol{H}) \iff \boldsymbol{K} \cdot \boldsymbol{\omega}^{\mathsf{T}} \geq \boldsymbol{0}^{\mathsf{T}}.$$

(Clearly, the rows in $\boldsymbol{K}$ corresponding to (8) form an $n \times n$ identity matrix and the rows in $\boldsymbol{K}$ corresponding to (9) form a submatrix containing zeros, ones, and minus ones.) In the case of QC codes, pseudocodewords can be written as polynomial vectors $\boldsymbol{\omega}(X) \in (\mathbb{R}[X]/\langle X^r - 1\rangle)^L$ and the fundamental cone can be described with the help of polynomial matrices. Namely, for any $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)$, there is a polynomial matrix $\boldsymbol{K}_{\mathrm{QC}}^{(r)}(X)$ over $\mathbb{R}[X]/\langle X^r - 1\rangle$ such that

$$\big(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r - 1)\big) \in \mathcal{K}\big(\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)\big)$$

if and only if

$$\Big(\boldsymbol{K}_{\mathrm{QC}}^{(r)}(X) \cdot \boldsymbol{\omega}^{\mathsf{T}}(X) \bmod_{\mathbb{R}} (X^r - 1)\Big) \geq \boldsymbol{0}^{\mathsf{T}}. \tag{10}$$

Similarly, in the case of convolutional codes, pseudocodewords can be written as polynomial vectors $\boldsymbol{\omega}(D) \in (\mathbb{R}[D])^L$ and the fundamental cone can be described with the help of polynomial matrices. Namely, for any $\boldsymbol{H}_{\mathrm{conv}}(D)$ there is a polynomial matrix $\boldsymbol{K}_{\mathrm{conv}}(D)$ over $\mathbb{R}[D]$ such that

$$\boldsymbol{\omega}(D) \in \mathcal{K}\big(\boldsymbol{H}_{\mathrm{conv}}(D)\big)$$

if and only if

$$\boldsymbol{K}_{\mathrm{conv}}(D) \cdot \boldsymbol{\omega}^{\mathsf{T}}(D) \geq \boldsymbol{0}^{\mathsf{T}}. \tag{11}$$

For the correct interpretation of expressions like (10) and (11), we remind the reader of the notational conventions that were introduced in Section I-C.

We now briefly focus on the case of **monomial** parity-check matrices, since the above fundamental-cone description is

$$\boldsymbol{\Omega}(D) \triangleq \begin{bmatrix} -3D^2 - D^3 & +4D + D^2 & +3 + D + 4D^2 + D^3 & +3 + 4D + D^2 \\ +3D^2 + D^3 & -4D - D^2 & +3 + D + 4D^2 + D^3 & +3 + 4D + D^2 \\ +3D^2 + D^3 & +4D + D^2 & -3 - D - 4D^2 - D^3 & +3 + 4D + D^2 \\ +3D^2 + D^3 & +4D + D^2 & +3 + D + 4D^2 + D^3 & -3 - 4D - D^2 \end{bmatrix}$$

$$\boldsymbol{H}_{\mathrm{conv}}(D) \cdot \boldsymbol{\Omega}^{\mathsf{T}}(D) = \begin{bmatrix} 6 + 9D + 3D^2 & 6 + D + D^2 + 2D^3 & 7D + D^2 & D + 7D^2 + 2D^3 \\ 4D^2 + 4D^3 + 8D^4 + 2D^5 & 2D^2 + 4D^3 + 8D^4 + 2D^5 & 4D^2 + 4D^3 & 10D^2 \\ 6D^3 + 2D^4 + 8D^5 + 2D^6 & 6D^2 + 8D^3 + 2D^4 + 2D^6 & 6D^2 + 2D^3 & 8D^5 + 2D^6 \end{bmatrix}$$

$$\begin{aligned} &\boldsymbol{H}^{(5)}_{\mathrm{QC}}(X)\boldsymbol{\Omega}^{\mathsf{T}}(X) \bmod_{\mathbb{F}_2} (X^5 - 1) \\ &\quad = \begin{bmatrix} 6 + 9X + 3X^2 & 6 + X + X^2 + 2X^3 & 7X + X^2 & X + 7X^2 + 2X^3 \\ 2 + 4X^2 + 4X^3 + 8X^4 & 2 + 2X^2 + 4X^3 + 8X^4 & 4X^2 + 4X^3 & 10X^2 \\ 8 + 2X + 6X^3 + 2X^4 & 2X + 6X^2 + 8X^3 + 2X^4 & 6X^2 + 2X^3 & 8 + 2X \end{bmatrix}. \end{aligned}$$

particularly simple and useful in this case.[14] A vector $\boldsymbol{\omega}$ is a pseudocodeword in the fundamental cone of a monomial matrix $\boldsymbol{H}_{\mathrm{conv}}(D)$ if and only if the associated polynomial vector $\boldsymbol{\omega}(D) = \big(\omega_0(D), \ldots, \omega_{L-1}(D)\big)$ satisfies the following inequalities.

- For all $\ell \in \{0, \ldots, L-1\}$
$$\omega_\ell(D) \geq 0.$$
- For all $j \in \{0, \ldots, J-1\}$, for all $\ell' \in \{0, \ldots, L-1\}$,
$$h_{j\ell'}(D)\omega_{\ell'}(D) \leq \sum_{\ell \in \{0,\ldots,L-1\} \setminus \{\ell'\}} h_{j\ell}(D)\omega_\ell(D).$$
(Here, $h_{j\ell}(D)$ is the entry in row j and column ℓ of $\boldsymbol{H}_{\mathrm{conv}}(D)$, which, by assumption, is a monomial.)

Equivalently, if $\boldsymbol{\Omega}(D)$ is an $L \times L$ matrix with ℓth row vector

$$\big(\omega_0(D), \ldots, \omega_{\ell-1}(D), -\omega_\ell(D), \omega_{\ell+1}(D), \ldots, \omega_{L-1}(D)\big)$$

for $\ell \in \{0, \ldots, L-1\}$, then $\boldsymbol{\omega}(D)$ is a pseudocodeword if and only if

$$\boldsymbol{\omega}(D) \geq \mathbf{0} \quad \text{and} \quad \boldsymbol{H}_{\mathrm{conv}}(D) \cdot \boldsymbol{\Omega}^{\mathsf{T}}(D) \geq \mathbf{0}.$$

Example 21: Let

$$\boldsymbol{H}_{\mathrm{conv}}(D) \triangleq \begin{bmatrix} D^0 & D^0 & D^0 & D^0 \\ D^0 & D^1 & D^2 & D^3 \\ D^0 & D^4 & D^3 & D^2 \end{bmatrix}$$

be a polynomial parity-check matrix of a rate-$1/4$ convolutional code. The following vector:

$$\begin{aligned} \boldsymbol{\omega}(D) &\triangleq \begin{pmatrix} 3D^2 + D^3 \\ 4D + D^2 \\ 3 + D + 4D^2 + D^3 \\ 3 + 4D + D^2 \end{pmatrix}^{\mathsf{T}} \\ &= \begin{pmatrix} 0 \\ 0 \\ 3 \\ 3 \end{pmatrix}^{\mathsf{T}} + \begin{pmatrix} 0 \\ 4 \\ 1 \\ 4 \end{pmatrix}^{\mathsf{T}} D + \begin{pmatrix} 3 \\ 1 \\ 4 \\ 1 \end{pmatrix}^{\mathsf{T}} D^2 + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}^{\mathsf{T}} D^3 \end{aligned}$$

corresponding to the scalar vector

$$\overline{\boldsymbol{\omega}} = (0, 0, 3, 3,\ 0, 4, 1, 4,\ 3, 1, 4, 1,\ 1, 0, 1, 0,\ \ldots),$$

is a pseudocodeword for the convolutional code, since $\boldsymbol{\omega}(D) \geq \mathbf{0}$ and the matrix $\boldsymbol{\Omega}(D)$ (shown in the first equation at the top of this page) gives $\boldsymbol{H}_{\mathrm{conv}}(D) \cdot \boldsymbol{\Omega}^{\mathsf{T}}(D)$ (shown in the second equation at the top of the page), which satisfies

$$\boldsymbol{H}_{\mathrm{conv}}(D) \cdot \boldsymbol{\Omega}^{\mathsf{T}}(D) \geq \mathbf{0}. \qquad \square$$

Similarly, for a QC code that is described by a **monomial** $J \times L$ parity-check matrix $\boldsymbol{H}^{(r)}_{\mathrm{QC}}(X)$, a vector $\boldsymbol{\omega}(X)$ is a pseudocodeword in the corresponding fundamental cone if and only if the associated polynomial vector $\boldsymbol{\omega}(X) = \big(\omega_0(X), \ldots, \omega_{L-1}(X)\big)$ satisfies

$$\big(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r - 1)\big) \geq \mathbf{0}$$

and

$$\Big(\boldsymbol{H}^{(r)}_{\mathrm{QC}}(X) \cdot \boldsymbol{\Omega}^{\mathsf{T}}(X) \bmod_{\mathbb{R}} (X^r - 1)\Big) \geq \mathbf{0}.$$

Example 22: Let $\boldsymbol{H}^{(5)}_{\mathrm{QC}}(X)$ be the $r = 5$ parity-check matrix obtained from $\boldsymbol{H}_{\mathrm{conv}}(D)$ in Example 21 for a QC block code of length $n = 20$. Then, for the polynomial vector

$$\boldsymbol{\omega}^{(5)}(X) \triangleq \begin{pmatrix} 3X^2 + X^3 \\ 4X + X^2 \\ 3 + X + 4X^2 + X^3 \\ 3 + 4X + X^2 \end{pmatrix}^{\mathsf{T}}$$

we obtain $\boldsymbol{H}^{(5)}_{\mathrm{QC}}(X)\boldsymbol{\Omega}^{\mathsf{T}}(X) \bmod_{\mathbb{R}} (X^5 - 1)$ (shown in the third equation at the top of this page). Since $\boldsymbol{\omega}^{(5)}(X) \geq \mathbf{0}$ and $(\boldsymbol{H}^{(5)}_{\mathrm{QC}}(X)\boldsymbol{\Omega}^{\mathsf{T}}(X) \bmod_{\mathbb{R}} (X^5 - 1)) \geq \mathbf{0}$, $\boldsymbol{\omega}^{(5)}(X)$ is a pseudocodeword. Finally, we note that Example 25 will discuss more cases where $\boldsymbol{\omega}(D)$ is wrapped into a codeword $\boldsymbol{\omega}^{(r)}(X)$. $\square$

For the rest of the paper we will assume that the following conditions hold.

Assumption 23: Let $\mathsf{C}_{\mathrm{conv}}(D)$ be a convolutional code defined by the polynomial parity-check matrix $\boldsymbol{H}_{\mathrm{conv}}(D)$.

[14] Monomial matrices were defined in Section I-C.

When wrapping this code to obtain a QC code $\mathsf{C}_{\mathrm{QC}}^{(r)}(X)$ defined by the polynomial parity-check matrix $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) \triangleq \boldsymbol{H}_{\mathrm{conv}}(X) \bmod_{\mathbb{F}_2} (X^r-1)$, we allow only positive integers r such that the weight matrix associated with $\boldsymbol{H}_{\mathrm{conv}}(D)$ equals the weight matrix associated with $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)$. A similar condition is imposed when wrapping the polynomial parity-check matrix $\boldsymbol{H}_{\mathrm{QC}}^{(2r)}(X)$ of a code $\mathsf{C}_{\mathrm{QC}}^{(2r)}(X)$ to obtain the polynomial parity-check $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)$ of the wrapped code $\mathsf{C}_{\mathrm{QC}}^{(r)}(X)$.

These conditions essentially guarantee that in the wrapping process only exponents are changed, i.e., no nonzero coefficients are added. For example, the above condition is satisfied for $\boldsymbol{H}_{\mathrm{conv}}(D) = (1+D^2+D^4, D)$ and for $r = 3$ because $\boldsymbol{H}_{\mathrm{conv}}(X) \bmod_{\mathbb{F}_2} (X^3 - 1) = (1 + X^2 + X, X) = (1 + X + X^2, X)$, and so the weight matrix of $\boldsymbol{H}_{\mathrm{QC}}^{(3)}(X)$ equals the weight matrix of $\boldsymbol{H}_{\mathrm{conv}}(D)$. However, for $r = 2$, the above condition is not satisfied because $\boldsymbol{H}_{\mathrm{conv}}(X) \bmod_{\mathbb{F}_2} (X^2-1) = (1+1+1, X) = (1, X)$, and so the weight matrix of $\boldsymbol{H}_{\mathrm{QC}}^{(2)}(X)$ does not equal the weight matrix of $\boldsymbol{H}_{\mathrm{conv}}(D)$.

As a consequence, these conditions guarantee that the Tanner graphs associated with $\mathsf{C}_{\mathrm{conv}}(D)$ and $\mathsf{C}_{\mathrm{QC}}^{(r)}(X)$ look locally the same, i.e., the variable node degrees of corresponding variable nodes are the same and the check node degrees of corresponding check nodes are the same. Moreover, simple cycles in the Tanner graph of $\mathsf{C}_{\mathrm{conv}}(D)$ map to (possibly non-simple) cycles in the Tanner graph of $\mathsf{C}_{\mathrm{QC}}^{(r)}(X)$. (Because of the transient behavior of the Tanner graph of the convolutional code around the zero time index, the statement in this paragraph holds only for convolutional code time indices that are large enough.)

In general, the conditions in Assumption 23 are satisfied for any $r \geq m_s + 1$, where m_s is the syndrome former memory of $\boldsymbol{H}_{\mathrm{conv}}(D)$. Moreover, there are a variety of classes of polynomial parity-check matrices $\boldsymbol{H}_{\mathrm{conv}}(D)$ such that the above conditions are satisfied for any positive r. One such class is the class of monomial matrices, and another such class is the class of type-I polynomial matrices (see Section I-C).

Note that in Theorem 13 we did not need Assumption 23. So that theorem holds under more general conditions. However, if we want the Tanner graphs of the wrapped codes to have the same variable and check node degrees as the unwrapped codes, the conditions in Assumption 23 must be imposed.

In the following, we would like to better understand how pseudocodewords in $\mathcal{K}(\boldsymbol{H}_{\mathrm{conv}}(X))$ are connected to pseudocodewords in $\mathcal{K}(\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X))$; in particular, we would like to see if we can make statements similar to the statements in Theorem 13 and in the paragraphs preceding it. We will see that such statements can indeed be made. In fact, many of the upcoming observations can be seen as special cases of the following observations.

- Let $\mathcal{G}$ be some graph, let $\tilde{\mathcal{G}}$ be some M-cover of $\mathcal{G}$; and let $\tilde{\mathcal{G}}'$ be some M'-cover of $\tilde{\mathcal{G}}$. Then $\tilde{\mathcal{G}}'$ is an $(M' \cdot M)$-cover of $\mathcal{G}$.
- If $\mathcal{G}^{(Mr)}$ and $\mathcal{G}^{(r)}$ are the Tanner graphs of $\boldsymbol{H}_{\mathrm{QC}}^{(Mr)}(X)$ and $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)$, respectively, then $\mathcal{G}^{(Mr)}$ is an M-cover of $\mathcal{G}^{(r)}$.

Therefore, the fundamental cone of $\boldsymbol{H}_{\mathrm{QC}}^{(Mr)}(X)$, which by Remark 19 is characterized by the finite covers of $\mathcal{G}^{(Mr)}$, is tightly connected to the fundamental cone of $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)$, which again by Remark 19 is characterized by the finite covers of $\mathcal{G}^{(r)}$.

Theorem 24: Assume that the conditions in Assumption 23 hold. Let $\boldsymbol{\omega}(D)$ be a pseudocodeword in the convolutional code defined by $\boldsymbol{H}_{\mathrm{conv}}(D)$, i.e., $\boldsymbol{\omega}(D) \in \mathcal{K}(\boldsymbol{H}_{\mathrm{conv}}(D))$. Then its r wraparound polynomial vector is a pseudocodeword in the associated QC code defined by $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)$, i.e.,

$$\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1) \in \mathcal{K}(\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X))$$

where

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) \triangleq \boldsymbol{H}_{\mathrm{conv}}(X) \bmod_{\mathbb{F}_2} (X^r-1).$$

Proof: See Appendix B. □

Note that for a nonzero polynomial pseudocodeword $\boldsymbol{\omega}(D) \in \mathcal{K}(\boldsymbol{H}_{\mathrm{conv}}(D))$, the wrapped polynomial vector $\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)$ is never the all-zero vector. This is in contrast to some nonzero polynomial codewords $\boldsymbol{c}(D) \in \mathsf{C}_{\mathrm{conv}}(D)$ where the wrapped polynomial vector $\boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X^r-1)$ equals the all-zero vector. The different behavior comes from the fact that in the first case we are operating in a polynomial ring over $\mathbb{R}$, whereas in the second case we are operating in a polynomial ring over $\mathbb{F}_2$.

Example 25: The polynomial vector

$$\boldsymbol{\omega}(D) \triangleq \begin{pmatrix} 3D^2 + D^3 \\ 4D + D^2 \\ 3 + D + 4D^2 + D^3 \\ 3 + 4D + D^2 \end{pmatrix}^{\mathsf{T}}$$

is in the fundamental cone of the polynomial parity-check matrix $\boldsymbol{H}_{\mathrm{conv}}(D)$ from Example 21. Because $\boldsymbol{H}_{\mathrm{conv}}(D)$ is monomial, Assumption 23 holds for any $r \geq 1$, and so, by applying Theorem 24 we see that

$$\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1) \in \mathcal{K}(\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X))$$

for all any $r \geq 1$. The corresponding wrapped pseudocodewords are as follows.

For all $r \geq 4$

$$\boldsymbol{\omega}^{(r)}(X) \triangleq \boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1) = \begin{pmatrix} 3X^2 + X^3 \\ 4X + X^2 \\ 3 + X + 4X^2 + X^3 \\ 3 + 4X + X^2 \end{pmatrix}^{\mathsf{T}}.$$

For $r = 3$

$$\boldsymbol{\omega}^{(3)}(X) \triangleq \boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^3-1) = \begin{pmatrix} 1 + 3X^2 \\ 4X + X^2 \\ 4 + X + 4X^2 \\ 3 + 4X + X^2 \end{pmatrix}^{\mathsf{T}}.$$

For $r = 2$

$$\boldsymbol{\omega}^{(2)}(X) \triangleq \boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^2-1) = \begin{pmatrix} 3 + X \\ 1 + 4X \\ 7 + 2X \\ 4 + 4X \end{pmatrix}^{\mathsf{T}}.$$

For $r = 1$

$$\boldsymbol{\omega}^{(1)}(X) \triangleq \boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X-1) = \begin{pmatrix} 4 \\ 5 \\ 9 \\ 8 \end{pmatrix}^{\mathsf{T}}. \qquad \square$$

III. Pseudoweight Comparison Between QC Codes and Convolutional Codes

This section begins with an introductory subsection in which various channel pseudoweights are defined and continues with the main result that the minimum AWGNC, BSC, BEC, and max-fractional pseudoweights of a convolutional code are at least as large as the corresponding pseudoweights of a wrapped QC block code.

A. Definition of Pseudoweights

Definition 26 (see [5]–[7], [11], [17], [18]): Let $\boldsymbol{\omega} = (\omega_0, \ldots, \omega_{n-1})$ be a nonzero vector in $\mathbb{R}_+^n$. The AWGNC pseudoweight of the vector $\boldsymbol{\omega}$ is defined to be

$$w_{\mathrm{p}}^{\mathrm{AWGNC}}(\boldsymbol{\omega}) \triangleq \frac{\|\boldsymbol{\omega}\|_1^2}{\|\boldsymbol{\omega}\|_2^2}$$

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

$$f(\xi) \triangleq \omega_i' \quad (i < \xi \leq i+1,\ 0 < \xi \leq n-1)$$
$$F(\xi) \triangleq \int_0^{\xi} f(\xi')\, \mathrm{d}\xi'$$

and

$$e \triangleq F^{-1}\left(\frac{F(n-1)}{2}\right) = F^{-1}\left(\frac{\|\boldsymbol{\omega}\|_1}{2}\right).$$

Then the BSC pseudoweight $w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega})$ of the vector $\boldsymbol{\omega}$ is defined to be $w_{\mathrm{p}}^{\mathrm{BSC}}(\boldsymbol{\omega}) \triangleq 2e$. The BEC pseudoweight of the vector $\boldsymbol{\omega}$ is defined to be

$$w_{\mathrm{p}}^{\mathrm{BEC}}(\boldsymbol{\omega}) = \left|\mathrm{supp}(\boldsymbol{\omega})\right|$$

where $\mathrm{supp}(\boldsymbol{\omega})$ is the set of all indices i corresponding to nonzero components ω_i of $\boldsymbol{\omega}$. The fractional weight of a vector $\boldsymbol{\omega} \in [0,1]^n$ is defined to be

$$w_{\mathrm{frac}}(\boldsymbol{\omega}) = \|\boldsymbol{\omega}\|_1.$$

Finally, the max-fractional weight of a vector $\boldsymbol{\omega} \in \mathbb{R}_+^n$ is defined to be

$$w_{\text{max-frac}}(\boldsymbol{\omega}) \triangleq \frac{\|\boldsymbol{\omega}\|_1}{\|\boldsymbol{\omega}\|_\infty}$$

where $\|\boldsymbol{\omega}\|_\infty$ is the infinity or maximum norm. For $\boldsymbol{\omega} = \mathbf{0}$ we define all of the above pseudoweights, fractional weights, and max-fractional weights to be zero. $\square$

A detailed discussion of the motivation and significance of these definitions can be found in [7]. Note that, whereas the fractional weight has an operational meaning only for vertices of the fundamental polytope, the other measures have an operational meaning for any vector in the fundamental polytope or cone. Note also that in this paper the quantities w_{frac} and $w_{\text{max-frac}}$ are defined for any vector in $\mathbb{R}_+^n$, whereas [17] defined w_{frac} and $w_{\text{max-frac}}$ to be the quantities that we will call $w_{\mathrm{frac}}^{\min}$ and $w_{\text{max-frac}}^{\min}$.

Example 27: Let

$$\boldsymbol{\omega}(D) = \begin{pmatrix} 3D^2 + D^3 \\ 4D + D^2 \\ 3 + D + 4D^2 + D^3 \\ 3 + 4D + D^2 \end{pmatrix}^{\mathsf{T}}$$

be the pseudocodeword in Example 21 and let

$$\overline{\boldsymbol{\omega}} = (0,0,3,3,\ 0,4,1,4,\ 3,1,4,1,\ 1,0,1,0,\ 0,0,0,0,\ \ldots)$$

be its scalar vector description. Then

$$w_{\mathrm{p}}^{\mathrm{AWGNC}}(\overline{\boldsymbol{\omega}}) = \frac{\|\overline{\boldsymbol{\omega}}\|_1^2}{\|\overline{\boldsymbol{\omega}}\|_2^2} = \frac{(3\cdot 4 + 3\cdot 3 + 5\cdot 1)^2}{3\cdot 16 + 3\cdot 9 + 5\cdot 1} = 8.45.$$

In order to compute $w_{\mathrm{p}}^{\mathrm{BSC}}(\overline{\boldsymbol{\omega}})$, we let

$$\overline{\boldsymbol{\omega}}' = (4,4,4,3,3,3,1,1,1,1,1,0,0,0,0,0,\ldots)$$

be the vector that lists the components of $\overline{\boldsymbol{\omega}}$ in non-increasing order. We obtain $w_{\mathrm{p}}^{\mathrm{BSC}}(\overline{\boldsymbol{\omega}}) = 2e = \frac{20}{3} = 6.67$, since we need to add up $e = \frac{10}{3}$ ordered components of $\overline{\boldsymbol{\omega}}'$ to obtain $\frac{\|\overline{\boldsymbol{\omega}}\|_1}{2} = 13$. Moreover

$$w_{\mathrm{p}}^{\mathrm{BEC}}(\overline{\boldsymbol{\omega}}) = |\mathrm{supp}(\overline{\boldsymbol{\omega}})| = 11$$

and, finally, $\|\overline{\boldsymbol{\omega}}\|_\infty = \max_i \overline{\omega}_i$, from which it follows that

$$w_{\text{max-frac}}(\overline{\boldsymbol{\omega}}) = \frac{\|\overline{\boldsymbol{\omega}}\|_1}{\|\overline{\boldsymbol{\omega}}\|_\infty} = \frac{26}{4} = 6.5. \qquad \square$$

One measure of the effect that pseudocodewords have on the performance of a code is given by the minimum *pseudoweight* [6], [7], [17], [18]

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

where $\mathcal{V}(\mathcal{P}(\boldsymbol{H})) \setminus \{\mathbf{0}\}$ is the set of all nonzero vertices in the fundamental polytope $\mathcal{P}(\boldsymbol{H})$ and the pseudoweights are the appropriate ones for each channel (AWGNC, BSC, and BEC pseudoweights) or the fractional and max-fractional weights.

Computing these values can be quite challenging, since the task of finding the set of vertices of $\mathcal{P}(\boldsymbol{H})$ is in general very complex. However, in the case of four of the above pseudoweights (the minimum AWGNC, BSC, and BEC pseudoweights and the minimum max-fractional weight) there is a computationally simpler description, given by

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

for the appropriate pseudoweight. (Note that there is no such statement for the minimum fractional weight; see, e.g., [7], [18]).

Example 28: Let $\mathsf{C}_{\mathrm{QC}}^{(5)}$ be the length-20 QC code defined by the polynomial parity-check matrix $\boldsymbol{H}_{\mathrm{QC}}^{(5)}(X)$ of Example 22. The minimum AWGNC pseudoweight is $w_{\mathrm{p}}^{\mathrm{AWGNC,\,min}}(\boldsymbol{H}_{\mathrm{QC}}^{(5)}) = 6$ and equals the minimum Hamming weight. (The minimum AWGNC pseudoweight was obtained using a vertex enumeration program for polytopes that lists all the minimal pseudocodewords of a code [25].) □

Remark 29: In [7], it was shown that for a code C defined by a parity-check matrix $\boldsymbol{H}$, the following inequalities hold:

$$\begin{aligned} w_{\mathrm{frac}}^{\min}(\boldsymbol{H}) &\le w_{\text{max-frac}}^{\min}(\boldsymbol{H}) \le w_{\mathrm{p}}^{\mathrm{AWGNC,\,min}}(\boldsymbol{H}) \\ &\le w_{\mathrm{p}}^{\mathrm{BEC,\,min}}(\boldsymbol{H}) \le w_{\mathrm{H}}^{\min}(\mathsf{C}) \\ w_{\mathrm{frac}}^{\min}(\boldsymbol{H}) &\le w_{\text{max-frac}}^{\min}(\boldsymbol{H}) \le w_{\mathrm{p}}^{\mathrm{BSC,\,min}}(\boldsymbol{H}) \\ &\le w_{\mathrm{p}}^{\mathrm{BEC,\,min}}(\boldsymbol{H}) \le w_{\mathrm{H}}^{\min}(\mathsf{C}). \end{aligned}$$

Therefore, $w_{\mathrm{frac}}^{\min}(\boldsymbol{H})$ and $w_{\text{max-frac}}^{\min}(\boldsymbol{H})$ can serve as lower bounds for $w_{\mathrm{p}}^{\mathrm{AWGNC,\,min}}(\boldsymbol{H})$, $w_{\mathrm{p}}^{\mathrm{BSC,\,min}}(\boldsymbol{H})$, and $w_{\mathrm{p}}^{\mathrm{BEC,\,min}}(\boldsymbol{H})$.

B. Minimum Pseudoweights

In what follows, we compare the minimum pseudoweights and the minimum max-fractional weight of a QC block code to the same quantities for its corresponding convolutional code, which we assume to have a fundamental cone containing nonzero vectors.[15] In order to analyze the minimum pseudoweights and the minimum max-fractional weight, it is sufficient to analyze the weights of the nonzero vectors in the fundamental cone. Throughout this subsection, without loss of generality, all pseudocodewords $\boldsymbol{\omega}(D)$ are assumed to have finite support.[16]

Theorem 30: We assume that the conditions in Assumption 23 hold. For the AWGNC, BSC, and BEC pseudoweights, if $\boldsymbol{\omega}(D) \in \mathcal{K}(\boldsymbol{H}_{\mathrm{conv}}(D))$, then

$$w_{\mathrm{p}}\big(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r - 1)\big) \le w_{\mathrm{p}}\big(\boldsymbol{\omega}(D)\big).$$

Therefore, if the fundamental cone of the convolutional code is not trivial (i.e., it contains nonzero vectors), we obtain

$$w_{\mathrm{p}}^{\min}\left(\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)\right) \le w_{\mathrm{p}}^{\min}\big(\boldsymbol{H}_{\mathrm{conv}}(D)\big).$$

Proof: See Appendix C. □

Theorem 30 implies that low-pseudoweight vectors in the block code may correspond to higher pseudoweight vectors in the convolutional code, but the opposite is not possible. This suggests that the pseudocodewords in the block code that result in decoding failures may not cause such failures in the convolutional code.

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

Theorem 31: We assume that the conditions in Assumption 23 hold. If $\boldsymbol{\omega}(D) \in \mathcal{K}(\boldsymbol{H}_{\mathrm{conv}}(D))$, then

$$w_{\text{max-frac}}\big(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r - 1)\big) \le w_{\text{max-frac}}(\boldsymbol{\omega}(D)).$$

Therefore

$$w_{\text{max-frac}}^{\min}\left(\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)\right) \le w_{\text{max-frac}}^{\min}\big(\boldsymbol{H}_{\mathrm{conv}}(D)\big).$$

Proof: See Appendix D. □

In the case of the fractional weight, it is easy to see that for any $\boldsymbol{\omega}(D) \in \mathcal{V}(\mathcal{P}(\boldsymbol{H}_{\mathrm{conv}}(D))) \setminus \{\mathbf{0}\}$, we have

$$\|\boldsymbol{\omega}(D)\|_1 = \|\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r - 1)\|_1$$

and hence

$$w_{\mathrm{frac}}(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r - 1)) = w_{\mathrm{frac}}(\boldsymbol{\omega}(D)).$$

When comparing the minimum fractional weight of the convolutional and QC codes, we encounter a computationally harder case, since these values must be computed over the set of nonzero pseudocodewords that are vertices of the fundamental polytope. This is not an easy task, because a vertex pseudocodeword in the convolutional code might not map into a vertex pseudocodeword in the QC code.

The following theorem, however, can be established. To help clarify the idea, we recall that $w_{\mathrm{frac}}^{\min}(\boldsymbol{H}_{\mathrm{QC}}^{(r)})$ has the following operational meaning [17], [18]. Let $\mathcal{E}_{\mathrm{QC}}^{(r)} \subseteq \mathcal{I}(\boldsymbol{H}_{\mathrm{QC}}^{(r)})$ be the set of positions where bit flips occurred when using the code $\mathsf{C}_{\mathrm{QC}}^{(r)}$ for data transmission over a BSC with crossover probability p, $0 \le p < 1/2$. If $|\mathcal{E}_{\mathrm{QC}}^{(r)}| < \frac{1}{2} w_{\mathrm{frac}}^{\min}(\boldsymbol{H}_{\mathrm{QC}}^{(r)})$, then LP decoding succeeds. Similarly, $w_{\mathrm{frac}}^{\min}(\boldsymbol{H}_{\mathrm{conv}})$ implies the following. If $\mathcal{E}_{\mathrm{conv}} \subseteq \mathcal{I}(\boldsymbol{H}_{\mathrm{conv}})$ is the set of positions where bit flips occurred when using the code $\mathsf{C}_{\mathrm{conv}}$ for data transmission over a BSC, then $|\mathcal{E}_{\mathrm{conv}}| < \frac{1}{2} w_{\mathrm{frac}}^{\min}(\boldsymbol{H}_{\mathrm{conv}})$ guarantees that LP decoding is correct.

Theorem 32: Assume that the conditions in Assumption 23 hold, that $\mathsf{C}_{\mathrm{conv}}$ is used for data transmission over a BSC with crossover probability p, where $0 \le p < 1/2$, and that bit flips occur at positions $\mathcal{E}_{\mathrm{conv}} \subseteq \mathcal{I}(\boldsymbol{H}_{\mathrm{conv}})$. If

$$|\mathcal{E}_{\mathrm{conv}}| < \frac{1}{2} w_{\mathrm{frac}}^{\min}(\boldsymbol{H}_{\mathrm{QC}}^{(r)}) \tag{12}$$

then LP decoding succeeds. (Note that, on the right-hand side of inequality (12), we have $w_{\mathrm{frac}}^{\min}(\boldsymbol{H}_{\mathrm{QC}}^{(r)})$ and not $w_{\mathrm{frac}}^{\min}(\boldsymbol{H}_{\mathrm{conv}})$.)

Proof: See Appendix E. □

As discussed at the end of [7, Sec. 6], $w_{\mathrm{frac}}^{\min}(\boldsymbol{H})$ and $w_{\text{max-frac}}^{\min}(\boldsymbol{H})$ can give, especially for long codes, quite conservative lower bounds on $w_{\mathrm{p}}^{\mathrm{BSC,\,min}}(\boldsymbol{H})$. (For example, the guarantees on the error correction capabilities of the LP decoder implied by $w_{\mathrm{frac}}^{\min}(\boldsymbol{H})$ and $w_{\text{max-frac}}^{\min}(\boldsymbol{H})$ are not good enough to prove the results in [26].) However, there are polynomial-time algorithms that compute $w_{\mathrm{frac}}^{\min}(\boldsymbol{H})$ and $w_{\text{max-frac}}^{\min}(\boldsymbol{H})$ [17], [18].

Remark 33: It is not difficult to adapt Theorems 30 and 31 such that similar conclusions can be drawn with respect to a QC block code with the same structure but a larger circulant size that is a multiple of r. Using similar arguments to the ones in

[15] Obviously, this condition is a very weak technical requirement.

[16] With suitable modifications, this can easily be generalized to $\boldsymbol{\omega}(D)$ with $\|\boldsymbol{\omega}(D)\|_1 < \infty$. Note that such polynomial vectors also fulfill $\|\boldsymbol{\omega}(D)\|_2 < \infty$.

TABLE I
THE PSEUDOWEIGHTS OF THE PSEUDOCODEWORDS IN EXAMPLE 25

	$w_{\mathrm{p}}^{\mathrm{AWGNC}}$	$w_{\mathrm{p}}^{\mathrm{BSC}}$	$w_{\mathrm{p}}^{\mathrm{BEC}}$	$w_{\mathrm{max-frac}}$
$\boldsymbol{\omega}(D)$	8.45	6.67	11	6.5
$\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)$ for all $r \geq 4$	8.45	6.67	11	6.5
$\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^3-1)$	7.86	6.5	10	6.5
$\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^2-1)$	6.09	5	8	3.71
$\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X-1)$	3.63	3	4	2.89

the proofs of these theorems and Assumption 23, we obtain the following more general inequalities that hold for the AWGNC, BSC, BEC, max-fractional, and fractional pseudoweights. If $\boldsymbol{\omega}(D) \in \mathcal{K}(\boldsymbol{H}_{\mathrm{conv}}(D))$, then

$$\begin{aligned} w_{\mathrm{p}}\big(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\big) &\leq w_{\mathrm{p}}\big(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^{2r}-1)\big) \\ &\leq w_{\mathrm{p}}\big(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^{4r}-1)\big) \\ &\leq \cdots \\ &\leq w_{\mathrm{p}}\big(\boldsymbol{\omega}(D)\big), \quad \text{for all } r \geq 1. \end{aligned}$$

In addition, for the AWGNC, BSC, BEC, and max-fractional minimum pseudoweights, the following holds for any $m \geq 1$:

$$w_{\mathrm{p}}^{\min}\big(\boldsymbol{H}_{\mathrm{QC}}^{(r)}\big) \leq w_{\mathrm{p}}^{\min}\big(\boldsymbol{H}_{\mathrm{QC}}^{(mr)}\big).$$

(Empirically, for $r < r'$, r' not a multiple of r, it very often holds that $w_{\mathrm{p}}^{\min}\big(\boldsymbol{H}_{\mathrm{QC}}^{(r)}\big) \leq w_{\mathrm{p}}^{\min}\big(\boldsymbol{H}_{\mathrm{QC}}^{(r')}\big)$; however, this is not always the case. A similar statement can be made about the relationship between the minimum Hamming distances of the corresponding codes.)

Example 34: To illustrate how the pseudoweights of the pseudocodewords in the convolutional code and their projections onto the QC codes satisfy the pseudoweight inequalities in Remark 33 for all the defined pseudoweights $w_{\mathrm{p}}^{\mathrm{AWGNC}}$, $w_{\mathrm{p}}^{\mathrm{BSC}}$, $w_{\mathrm{p}}^{\mathrm{BEC}}$, and $w_{\text{max-frac}}$, we computed the pseudoweights of the pseudocodewords in Example 25. Table I contains these results. □

Next, we illustrate some of the bounds on the minimum pseudoweight of codes derived in this section. We take a tower of three QC codes together with their convolutional version and compute their minimum pseudoweights which, according to the bounds derived above, form an increasing sequence, upper-bounded by the minimum pseudoweight of the convolutional version. However, due to the large code parameters, we were only able to compute the minimum pseudoweight of the code of length 20. For the other QC codes, we used the methods of [6], [7] to give lower and upper bounds.

Example 35: Consider the $(3, 4)$-regular QC-LDPC code of length $4r$ given by the scalar parity-check matrix

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)} = \begin{bmatrix} \boldsymbol{I}_0 & \boldsymbol{I}_0 & \boldsymbol{I}_0 & \boldsymbol{I}_0 \\ \boldsymbol{I}_0 & \boldsymbol{I}_1 & \boldsymbol{I}_2 & \boldsymbol{I}_3 \\ \boldsymbol{I}_0 & \boldsymbol{I}_4 & \boldsymbol{I}_3 & \boldsymbol{I}_2 \end{bmatrix}$$

or, equivalently, the polynomial-parity check matrix

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & X & X^2 & X^3 \\ 1 & X^4 & X^3 & X^2 \end{bmatrix}.$$

For $r = 5$, we obtain a $[20, 7, 6]$ code with rate $R = 0.35$. By increasing r we obtain other QC codes. By taking r to be $5 \cdot 2^m$, $m = 1, 2, \ldots$, we obtain a tower of QC codes whose graphs form a sequence of covers of the Tanner graph of the $[20, 7, 6]$ code. For $r \geq 9$, all the codes have minimum distance 10, and hence the free distance of the associated rate $R = 1/4$ convolutional code is $d_{\mathrm{free}} = 10$, strictly larger than the minimum distance 6 of the $[20, 7, 6]$ code.

For the $[20, 7, 6]$ code we ran a vertex enumeration program [25] for polytopes that lists all the minimal pseudocodewords and found that the minimum pseudoweight of the $[20, 7, 6]$ code is $w_{\mathrm{p}}^{\mathrm{AWGNC,\,min}} = 6.00$. The larger parameters of the other three codes allowed us to only lower- and upper-bound their minimum pseudoweights.[17] For the [40,12,10] QC code, we obtained $6.05 \leq w_{\mathrm{p}}^{\mathrm{AWGNC,\,min}} \leq 9.09$ and for the $[80, 22, 10]$ and the $[160, 42, 10]$ codes, we obtained $6.20 \leq w_{\mathrm{p}}^{\mathrm{AWGNC,\,min}} \leq 9.09$.[18] The increase in the lower bound from 6, which is the minimum weight of the $[20, 7, 6]$ code, to 6.05 and 6.20 shows that the minimum AWGNC pseudoweight of the $[20, 7, 6]$ code is less than that of the $[40, 12, 10]$, $[80, 22, 10]$, and $[160, 42, 10]$ codes. This increase in the lower bounds from 6.05 to 6.20 suggests, but does not prove, the existence of an increasing sequence of minimum AWGNC pseudoweights for these codes, according to the results of this subsection. For comparison, we simulated the four QC codes together with the associated convolutional code. The results for an AWGNC are given in Fig. 6, and we note that they are consistent with the suggested increasing sequence of minimum pseudoweights.

For completeness, we mention that the techniques in [17] and [18] allow us to efficiently compute the minimum max-fractional weight for the above-mentioned codes: we obtain $w_{\text{max-frac}}^{\min} = 4.67$ for the length-20 code, $w_{\text{max-frac}}^{\min} = 5.31$ for the length-40 code, $w_{\text{max-frac}}^{\min} = 5.33$ for the length-80 code, and $w_{\text{max-frac}}^{\min} = 5.33$ for the length-160 code. Applying the results that were mentioned in Remark 29, we see that these values yield weaker lower bounds on $w_{\mathrm{p}}^{\mathrm{AWGNC,\,min}}$ than the ones given in the previous paragraph.

We conclude this example by looking at $\boldsymbol{H}_{\mathrm{QC}}^{(r)}$ for $r < 5$. Using the vertex enumeration program in [25], we were able to compute the minimum pseudoweights of the QC codes of length $4r$ given by the parity-check matrices $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)$ in Example 35 for $r = 4$ down to $r = 1$. The corresponding results are presented in Table II. □

IV. ANALYSIS OF PROBLEMATIC PSEUDOCODEWORDS IN CONVOLUTIONAL CODES

Studying pseudocodewords of small pseudoweight, and, in particular (since the minimum pseudoweight is upper-bounded by the minimum Hamming weight), studying pseudocodewords

[17]The lower bounds are obtained by applying the techniques that were presented in [27].

[18]Using some more sophisticated lower bounds from [27], one can actually show that $7.19 \leq w_{\mathrm{p}}^{\mathrm{AWGNC,\,min}}$ for the length-40, the length-80, and the length-160 code.

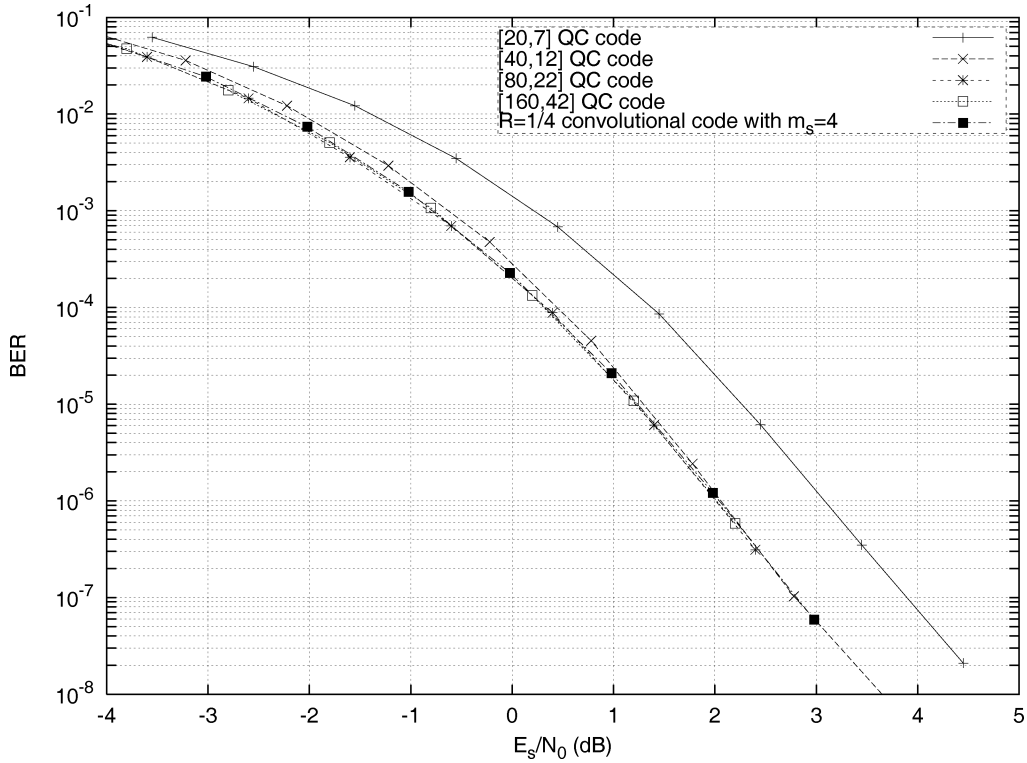

Fig. 6. The performance of a rate $R = 1/4$ $(3,4)$-regular LDPC convolutional code and four associated $(3,4)$-regular QC-LDPC block codes. (Note that the horizontal axis is E_s/N_0 and not the more common $E_b/N_0 = (1/R) \cdot E_s/N_0$; this former choice yields an easier comparison between the different curves.)

TABLE II
THE MINIMUM PSEUDOWEIGHTS OF THE CODES $\mathsf{C}_{\mathrm{QC}}^{(r)}$ GIVEN BY THE PARITY-CHECK MATRICES $\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X)$ IN EXAMPLE 35, FOR $r = 4$ DOWN TO $r = 1$

	$w_{\mathrm{p}}^{\mathrm{AWGNC,min}}$	$w_{\mathrm{p}}^{\mathrm{BSC,min}}$	$w_{\mathrm{p}}^{\mathrm{BEC,min}}$	$w_{\mathrm{max-frac}}^{\mathrm{min}}$
$r=4$	4	4	4	4
$r=3$	4	4	4	3
$r=2$	2	2	2	2
$r=1$	2	2	2	2

of pseudoweight smaller than the minimum Hamming weight, represents an important problem in the performance analysis of LDPC codes because it allows us to identify potential failures in MPI decoding.

Upper and lower bounds on the minimum pseudoweight of a convolutional code can be obtained by exploiting the "sliding" structure of its semi-infinite parity-check matrix $\boldsymbol{H}_{\mathrm{conv}}$ and some of its submatrices, which allows relatively easy computations by taking advantage of the increased sparseness compared to the corresponding parity-check matrix of an underlying QC code. On the one hand, this technique allows us to find certain low-weight pseudocodewords, and on the other hand, it illustrates the advantage of using a convolutional code structure over a block code structure in pseudocodeword analysis. In addition, similar to the expected increase in minimum distance when going from a QC code to its unwrapped convolutional version, we also expect an increase in the minimum pseudoweight of the convolutional code, leading to better performance compared to the original QC code. Our theoretical results and experimental observations point strongly in this direction. We now briefly explain our technique.

Similar to associating with a convolutional code [28] an increasing sequence of *column distances* $(d_\ell^{\mathrm{c}})_{\ell \geq 0}$ and a decreasing sequence of *row distances* $(d_\ell^{\mathrm{r}})_{\ell \geq 0}$, having the property that

$$d_0^{\mathrm{c}} \leq d_1^{\mathrm{c}} \leq \cdots \leq d_{\mathrm{free}} \leq \cdots \leq d_1^{r} \leq d_0^{r},$$

we define two sequences of pseudoweights that prove helpful in identifying the overall minimum pseudoweight.

We recall that an encoder polynomial generator matrix $\boldsymbol{G}_{\mathrm{conv}}(D)$ of a rate $R = K/L$ convolutional code with encoder memory m_{e} has associated with it a semi-infinite sliding generator matrix $\overline{\boldsymbol{G}}_{\mathrm{conv}}$. Let $\overline{\boldsymbol{G}}_{\mathrm{conv}}^{(j,i)}$ denote the $jK \times iL$ submatrix of $\overline{\boldsymbol{G}}_{\mathrm{conv}}$ with rows indexed by the first j block rows of $\overline{\boldsymbol{G}}_{\mathrm{conv}}$ and columns indexed by the first i block columns of $\overline{\boldsymbol{G}}_{\mathrm{conv}}$, and let

$$d_{j,i} = \min \left\{ w_{\mathrm{H}} \left(\overline{\boldsymbol{u}} \cdot \overline{\boldsymbol{G}}_{\mathrm{conv}}^{(j,i)} \right) \Big| \begin{matrix} \overline{\boldsymbol{u}} = (\overline{\boldsymbol{u}}_0, \ldots, \overline{\boldsymbol{u}}_{j-1}) \in \mathbb{F}_2^{jK} \\ \overline{\boldsymbol{u}}_0 \neq 0 \end{matrix} \right\}.$$

Then, the sequence of matrices

$$\overline{\boldsymbol{G}}_{\mathrm{conv}}^{(1,1)}, \overline{\boldsymbol{G}}_{\mathrm{conv}}^{(2,2)}, \ldots, \overline{\boldsymbol{G}}_{\mathrm{conv}}^{(\ell,\ell)}, \ldots,$$

gives us an increasing sequence of so-called column distances $(d_\ell^{\mathrm{c}})_{\ell \geq 0}$, $d_\ell^{\mathrm{c}} \triangleq d_{\ell,\ell}$, and the sequence of matrices

$$\overline{\boldsymbol{G}}_{\mathrm{conv}}^{(1,m_{\mathrm{e}}+1)}, \overline{\boldsymbol{G}}_{\mathrm{conv}}^{(2,m_{\mathrm{e}}+2)}, \ldots, \overline{\boldsymbol{G}}_{\mathrm{conv}}^{(\ell,m_{\mathrm{e}}+\ell)}, \ldots,$$

gives us a decreasing sequence of so-called row distances $(d_\ell^{\mathrm{r}})_{\ell \geq 0}$, $d_\ell^{\mathrm{r}} \triangleq d_{\ell,\ell+m_{\mathrm{e}}}$.

The column distance d_ℓ^c is a "truncated" distance, i.e., it measures the minimum of the Hamming weights of the vectors of length ℓL that constitute the first ℓL components of some codeword with a nonzero initial block component. The row distance d_ℓ^r is a "bounded codeword" distance, i.e., it measures the minimum of the Hamming weights of the codewords with nonzero initial block component and support contained in the first $(m_\mathrm{e}+\ell)L$ positions. The column distances and row distances represent valuable lower and upper bounds, respectively, on the free distance that become increasingly tight with increasing ℓ, and, in the limit, become equal to the free distance. If similar sequences could be defined for pseudoweights, they would prove helpful in identifying the overall minimum pseudoweight.

With this in mind, we define corresponding sequences of "truncated" pseudoweights and "bounded pseudocodeword" pseudoweights.

Let $\boldsymbol{H}_\mathrm{conv}(D)$ be a polynomial parity-check matrix for a convolutional code C_conv with syndrome former memory m_s, and let $\overline{\boldsymbol{H}}_\mathrm{conv}$ be its semi-infinite sliding parity-check matrix. Similar to the above notation, let $\overline{\boldsymbol{H}}_\mathrm{conv}^{(j,i)}$ be the $jJ \times iL$ submatrix of $\overline{\boldsymbol{H}}_\mathrm{conv}$ formed by the first j block rows of $\overline{\boldsymbol{H}}_\mathrm{conv}$ and the first i block columns of $\overline{\boldsymbol{H}}_\mathrm{conv}$. We will consider two sequences of such submatrices, namely

$$\overline{\boldsymbol{H}}_\mathrm{conv}^{(1,1)},\ \overline{\boldsymbol{H}}_\mathrm{conv}^{(2,2)},\ \ldots,\overline{\boldsymbol{H}}_\mathrm{conv}^{(\ell,\ell)},\ \ldots$$

and

$$\overline{\boldsymbol{H}}_\mathrm{conv}^{(m_s+1,1)},\ \overline{\boldsymbol{H}}_\mathrm{conv}^{(m_s+2,2)},\ \ldots,\overline{\boldsymbol{H}}_\mathrm{conv}^{(m_s+\ell,\ell)},\ \ldots.$$

In the second sequence, the first matrix that has a nonzero null space with certainty is $\overline{\boldsymbol{H}}_\mathrm{conv}^{(m_s+m_\mathrm{e}+1,m_\mathrm{e}+1)}$, since there is a nonzero polynomial codeword of degree m_e (associated with a scalar codeword of length $(m_\mathrm{e}+1)L$). Since

$$\overline{\boldsymbol{H}}_\mathrm{conv}^{(m_s+m_\mathrm{e}+1+\ell,m_\mathrm{e}+1+\ell)} \cdot \left(\overline{\boldsymbol{G}}_\mathrm{conv}^{(1+\ell,m_\mathrm{e}+1+\ell)}\right)^\mathsf{T} = \mathbf{0}$$

these matrices act like parity-check matrices in computing the row distances, with $\overline{\boldsymbol{H}}_\mathrm{conv}^{(m_s+m_\mathrm{e}+1+\ell,m_\mathrm{e}+1+\ell)}$ giving the ℓth row distance, for all $\ell \geq 0$. There might be nonzero nulls spaces earlier in the sequence, so we denote by $\overline{\boldsymbol{H}}_\mathrm{conv}^{(m_s+\mu,\mu)}$ the first matrix with a nonzero null space. Similarly, $\overline{\boldsymbol{H}}_\mathrm{conv}^{(\ell,\ell)}$, $\ell \geq 0$, will act like parity-check matrices in computing the ℓth column distances. So by computing the null spaces of these parity-check matrices we get upper and lower bounds on d_free for the convolutional code that are similar to the column and row distances defined from the generator matrix.

We remark also that if we wrap the convolutional code $\mathrm{mod}_{\mathbb{F}_2}(X^r-1)$, with $r \geq m_s+\mu$, then the matrix $\overline{\boldsymbol{H}}_\mathrm{conv}^{(m_s+\mu,\mu)}$ is a submatrix of the parity-check matrix $\overline{\boldsymbol{H}}_\mathrm{QC}^{(r)}$ of the QC code that remains unchanged after the wrapping. Hence, a codeword of minimum weight for the matrix $\overline{\boldsymbol{H}}_\mathrm{conv}^{(m_s+\mu,\mu)}$ will, if extended by zeros, be a codeword in the QC code. If this codeword has weight equal to the minimum distance of the QC code, then the free distance of the convolutional code is equal to the weight of this codeword. The minimum distance of the QC code could be smaller, however, and in this case the free distance will be upper-bounded by the weight of this codeword and lower-bounded by the minimum distance of the QC code.

In what follows, we will mimic the theory of row distances and column distances of a convolutional code to bound the minimum pseudoweight of the convolutional code. In the case of *row* distances/weights, the theory carries over from the Hamming distance case to the pseudoweight case and we obtain the following upper bounds on the minimum pseudoweight of the convolutional code:

$$\begin{aligned} w_\mathrm{p}^\mathrm{min}\left(\overline{\boldsymbol{H}}_\mathrm{conv}\right) &\leq \ldots \\ &\leq w_\mathrm{p}^\mathrm{min}\left(\overline{\boldsymbol{H}}_\mathrm{conv}^{(m_s+2,2)}\right) \\ &\leq w_\mathrm{p}^\mathrm{min}\left(\overline{\boldsymbol{H}}_\mathrm{conv}^{(m_s+1,1)}\right). \end{aligned}$$

So, by computing vectors in the fundamental cones of these parity-check matrices, we obtain upper bounds on $w_\mathrm{p}^\mathrm{min}(\overline{\boldsymbol{H}}_\mathrm{conv})$ that are similar to the row Hamming distances defined from the generator matrix.

In the case of *column* distances/weights, the theory does in general *not* carry over from the Hamming distance case to the pseudoweight case. Only for BEC pseudoweights do we obtain the following lower bounds on the minimum pseudoweight of the convolutional code:

$$\begin{aligned} w_\mathrm{p}^{\mathrm{BEC,\,min*}}\left(\overline{\boldsymbol{H}}_\mathrm{conv}^{(1,1)}\right) &\leq w_\mathrm{p}^{\mathrm{BEC,\,min*}}\left(\overline{\boldsymbol{H}}_\mathrm{conv}^{(2,2)}\right) \\ &\leq \ldots \\ &\leq w_\mathrm{p}^{\mathrm{BEC,\,min*}}\left(\overline{\boldsymbol{H}}_\mathrm{conv}\right) \end{aligned}$$

where $w_\mathrm{p}^{\mathrm{BEC,\,min*}}(\,\cdot\,)$ denotes the minimum BEC pseudoweight of all pseudocodewords in the corresponding fundamental cone that have at least one nonzero component in the first block.

Example 36: The pseudocodeword in Example 21 was obtained by attempting to compute small degree nonzero vectors in the fundamental cone of $\mathcal{K}(\overline{\boldsymbol{H}}_\mathrm{conv})$ using the above technique. The first nonzero "row pseudoweight" is 4, and the vector

$$\begin{pmatrix} 3D^2+D^3 \\ 4D+D^2 \\ 3+D+4D^2+D^3 \\ 3+4D+D^2 \end{pmatrix}^\mathsf{T}$$

is in the fundamental cone of $\overline{\boldsymbol{H}}_\mathrm{conv}^{(8,4)}$. Its AWGNC pseudoweight is 8.45, which is an upper bound on the minimum pseudoweight of the convolutional code. The free distance of this code is 10. The reduced pseudocodeword

$$\begin{pmatrix} 3X^2+X^3 \\ 4X+X^2 \\ 3+X+4X^2+X^3 \\ 3+4X+X^2 \end{pmatrix}^\mathsf{T}$$

modulo $\mathrm{mod}_\mathbb{R}(X^r-1)$, for $r = 5, 10, 20, 40$ has the same AWGNC pseudoweight 8.45, larger than the minimum distance of the $[20,7,6]$ code, which makes this pseudocodeword irrelevant,[19] but smaller than the minimum distances of the $[40,12,10]$, $[80,22,10]$, and $[160,42,10]$ codes. The upper

[19]Irrelevant in the sense that it does not improve upon the upper bound on the AWGNC pseudoweight that is implied by the minimum Hamming weight.

bound $w_{\mathrm{p}}^{\mathrm{AWGNC,\,min}} \leq 9.09$ in Example 35 therefore becomes $w_{\mathrm{p}}^{\mathrm{AWGNC,\,min}} \leq 8.45$ based on this pseudocodeword. □

This computational method has been applied successfully to larger codes as well. An example is the rate $R = 2/5$ LDPC convolutional code with syndrome former memory $m_{\mathrm{s}} = 21$ that was simulated in Fig. 2. The code was constructed by unwrapping a $[155, 64]$ $(3, 5)$-regular QC-LDPC block code with minimum Hamming distance 20. The convolutional code has free distance 24,[20] which already suggests a possible performance improvement compared to the QC code. Following the approach described above, we constructed a class of pseudocodewords among which the minimum AWGNC pseudoweight was 17.85. Thus, this class of pseudocodewords contains vectors of weight less than the free distance, which makes them relevant to the performance analysis of iterative decoding. Consequently, an upper bound on the minimum AWGNC pseudoweight of the convolutional code is 17.85, and, from the way we constructed this class of pseudocodewords, we believe it is a very tight bound. Projecting this pseudocodeword onto the QC codes obtained by wrapping the convolutional code gives upper bounds on the minimum AWGNC pseudoweight of these codes as well (in some cases tighter than the ones obtained using the methods of [6], [7]). The upper bound in [6], [7] for the minimum AWGNC pseudoweight of the $[155, 64]$ code is 16.4. These upper bounds, together with the simulation results in Fig. 2, suggest that the minimum AWGNC pseudoweight of the convolutional code is strictly greater than the minimum AWGNC pseudoweight of the $[155, 64]$ QC code. An evaluation of the exact values of the minimum AWGNC pseudoweight in these cases does not seem possible, however, due to the large complexity of such a task. Also note that if an upper bound on the minimum AWGNC pseudoweight of the convolutional code smaller than 16.4 could be found, it would decrease the upper bound on the minimum AWGNC pseudoweight of the $[155, 64]$ $(3, 5)$-regular QC-LDPC block code as well.

V. Conclusion

For an LDPC convolutional code derived by unwrapping a QC-LDPC block code, we have shown that the free pseudoweight of the convolutional code is at least as large as the minimum pseudoweight of the underlying QC code. This result suggests that the pseudoweight 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 Figs. 2 and 6. In order to analyze problematic pseudocodewords, i.e., pseudocodewords with small pseudoweight, we also presented a method of analysis that introduces sequences of "truncated" pseudoweights and "bounded pseudocodeword" pseudoweights which lower- and upper-bound the minimum pseudoweight of the convolutional code. This is similar to the method of using column and row distances to bound the free distance of a convolutional code from below and above.

[20]The free distance of this convolutional code was obtained by Bocharova *et al.* at the Department of Information Technology, Lund University, Lund, Sweden, using a program called BEAST (see [29]).

Appendix A
Proof of Theorem 13

Let $\boldsymbol{H}_{\mathrm{QC}}^{(2r)}(X)$ be a polynomial parity-check matrix of $\mathsf{C}_{\mathrm{QC}}^{(2r)}$ and let

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) \triangleq \boldsymbol{H}_{\mathrm{QC}}^{(2r)}(X) \bmod_{\mathbb{F}_2} (X^r - 1)$$

be the corresponding parity-check matrix of $\mathsf{C}_{\mathrm{QC}}^{(r)}$. Let $\boldsymbol{c}(X)$ be a nonzero codeword in $\mathsf{C}_{\mathrm{QC}}^{(2r)}$ of weight equal to the minimum distance $d_{\min}\big(\mathsf{C}_{\mathrm{QC}}^{(2r)}\big)$. (Without loss of generality, we can assume that the degree of $\boldsymbol{c}(X)$ is smaller than $2r$.) We have

$$\boldsymbol{H}_{\mathrm{QC}}^{(2r)}(X) \cdot \boldsymbol{c}^{\mathsf{T}}(X) \bmod_{\mathbb{F}_2} (X^{2r} - 1) = \boldsymbol{0}^{\mathsf{T}}$$

and since $(X^r - 1) \,|\, (X^{2r} - 1)$, it follows that

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) \cdot \boldsymbol{c}^{\mathsf{T}}(X) \bmod_{\mathbb{F}_2} (X^r - 1) = \boldsymbol{0}^{\mathsf{T}}.$$

If $\boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X^r - 1) \neq \boldsymbol{0}$, we obtain

$$\begin{aligned} d_{\min}\big(\mathsf{C}_{\mathrm{QC}}^{(r)}\big) &\leq w_{\mathrm{H}}\big(\boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X^r - 1)\big) \\ &\leq w_{\mathrm{H}}\big(\boldsymbol{c}(X)\big) = d_{\min}\big(\mathsf{C}_{\mathrm{QC}}^{(2r)}\big). \end{aligned}$$

If $\boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X^r - 1) = \boldsymbol{0}$, we can write $\boldsymbol{c}(X) = (X^r - 1)\boldsymbol{c}_1(X)$, where $\boldsymbol{c}_1(X) \bmod_{\mathbb{F}_2} (X^r - 1) \neq \boldsymbol{0}$ (otherwise, $\boldsymbol{c}(X) \bmod_{\mathbb{F}_2} (X^{2r} - 1) = \boldsymbol{0}$). (Without loss of generality, we can assume that the degree of $\boldsymbol{c}_1(X)$ is smaller than r.) Hence

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) \cdot (X^r - 1) \cdot \boldsymbol{c}_1^{\mathsf{T}}(X) \bmod_{\mathbb{F}_2} (X^{2r} - 1) = \boldsymbol{0}^{\mathsf{T}}$$

or equivalently

$$\boldsymbol{H}_{\mathrm{QC}}^{(r)}(X) \cdot \boldsymbol{c}_1^{\mathsf{T}}(X) \bmod_{\mathbb{F}_2} (X^r - 1) = \boldsymbol{0}^{\mathsf{T}}$$

which implies that $\boldsymbol{c}_1(X)$ is a nonzero codeword in $\mathsf{C}_{\mathrm{QC}}^{(r)}$. Therefore

$$\begin{aligned} d_{\min}\left(\mathsf{C}_{\mathrm{QC}}^{(2r)}\right) &= w_{\mathrm{H}}\big(\boldsymbol{c}(X)\big) = w_{\mathrm{H}}\big((X^r - 1)\boldsymbol{c}_1(X)\big) \\ &\geq 2 \cdot w_{\mathrm{H}}\big(\boldsymbol{c}_1(X)\big) > w_{\mathrm{H}}\big(\boldsymbol{c}_1(X)\big) \\ &\geq d_{\min}\left(\mathsf{C}_{\mathrm{QC}}^{(r)}\right). \end{aligned}$$

Mimicking this argument, we also obtain $d_{\mathrm{free}}\left(\mathsf{C}_{\mathrm{conv}}\right) \geq d_{\min}\left(\mathsf{C}_{\mathrm{QC}}^{(r)}\right)$, and hence the desired inequalities

$$\begin{aligned} d_{\min}\left(\mathsf{C}_{\mathrm{QC}}^{(r)}\right) &\leq d_{\min}\big(\mathsf{C}_{\mathrm{QC}}^{(2r)}\big) \leq d_{\min}\big(\mathsf{C}_{\mathrm{QC}}^{(4r)}\big) \\ &\leq \ldots \leq d_{\mathrm{free}}\left(\mathsf{C}_{\mathrm{conv}}\right). \end{aligned}$$

From the way we construct the semi-infinite sliding matrix of the convolutional code by unwrapping the scalar parity-check matrices of the QC block codes, we can see that there exists a QC code of circulant size r large enough so that its minimum distance is equal to the free distance of the convolutional code. This assures the limit equality in the theorem statement.

We conclude the proof with the remark that part of the derivation depends on the fact that the characteristic of $\mathbb{F}_2$ is 2, and so $(X^r-1)^2 = (X^{2r}-1)$ (in $\mathbb{F}_2[X]$).

Appendix B
Proof of Theorem 24

We have seen that for any $\boldsymbol{H}_{\rm conv}(D)$ describing a convolutional code there is a matrix $\boldsymbol{K}_{\rm conv}(D)$ such that

$$\boldsymbol{\omega}(D) \in \mathcal{K}\big(\boldsymbol{H}_{\rm conv}(D)\big)$$

if and only if

$$\boldsymbol{K}_{\rm conv}(D) \cdot \boldsymbol{\omega}^{\mathsf{T}}(D) \geq \mathbf{0}^{\mathsf{T}}.$$

By reducing $\boldsymbol{K}_{\rm conv}(D)$ modulo D^r-1, we obtain a matrix

$$\boldsymbol{K}_{\rm QC}^{(r)}(X) \triangleq \boldsymbol{K}_{\rm conv}(X) \bmod_{\mathbb{R}} (X^r-1)$$

with the property that a polynomial vector $\boldsymbol{\omega}'(X)$ satisfies

$$\boldsymbol{\omega}'(X) \in \mathcal{K}\left(\boldsymbol{H}_{\rm QC}^{(r)}(X)\right)$$

if and only if

$$\boldsymbol{K}_{\rm QC}^{(r)}(X) \cdot \boldsymbol{\omega}'^{\mathsf{T}}(X) \bmod_{\mathbb{R}} (X^r-1) \geq \mathbf{0}^{\mathsf{T}}.$$

Reducing $\boldsymbol{K}_{\rm conv}(D) \cdot \boldsymbol{\omega}^{\mathsf{T}}(D) \geq \mathbf{0}^{\mathsf{T}}$ modulo D^r-1, we obtain

$$\boldsymbol{K}_{\rm QC}^{(r)}(X) \cdot \big(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\big)^{\mathsf{T}} \bmod_{\mathbb{R}} (X^r-1) \geq \mathbf{0}^{\mathsf{T}}$$

which proves the claim.

Appendix C
Proof of Theorem 30

In the following, we analyze separately the AWGNC, BSC, and BEC pseudoweights of $\boldsymbol{\omega}(D)$ and of its r wraparound $\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)$. Let $\boldsymbol{\omega}(D) = \big(\omega_0(D), \ldots, \omega_{L-1}(D)\big)$ be a pseudocodeword. By assumption, $\boldsymbol{\omega}(D)$ has finite support, i.e., there exists an integer t such that the maximal degree of any $\omega_\ell(D)$, $\ell \in \{0, \ldots, L-1\}$, is smaller than t. We will first show that

$$\|\boldsymbol{\omega}(D)\|_1 = \|\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\|_1$$

which we then use several times in the proof. Let

$$\begin{aligned}
\omega_\ell(D) &\triangleq \sum_{i=0}^{t-1} \omega_{\ell,i} D^i \\
&= \sum_{i=0}^{r-1} \omega_{\ell,i} D^i + D^r \sum_{i=0}^{r-1} \omega_{\ell,i+r} D^i + D^{2r} \sum_{i=0}^{r-1} \omega_{\ell,i+2r} D^i \\
&\quad + \cdots + D^{\lfloor (t-1)/r \rfloor r} \sum_{i=0}^{r-1} \omega_{\ell,i+\lfloor (t-1)/r \rfloor r} D^i \\
&= \sum_{i'=0}^{\lfloor (t-1)/r \rfloor} D^{i'r} \sum_{i=0}^{r-1} \omega_{\ell,i+i'r} D^i \\
&= \sum_{i=0}^{r-1} \sum_{i'=0}^{\lfloor (t-1)/r \rfloor} D^{i'r} \omega_{\ell,i+i'r} D^i \,.
\end{aligned}$$

Then

$$\omega_\ell(X) \bmod_{\mathbb{R}} (X^r-1) = \sum_{i=0}^{r-1} \left(\sum_{i'=0}^{\lfloor (t-1)/r \rfloor} \omega_{\ell,i+i'r} \right) X^i$$

from which we obtain the norm equality

$$\begin{aligned}
\|\boldsymbol{\omega}(D)\|_1 &= \|\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\|_1 \\
&= \sum_{\ell=0}^{L-1} \sum_{i=0}^{r-1} \sum_{i'=0}^{\lfloor (t-1)/r \rfloor} \omega_{\ell,i+i'r} \,.
\end{aligned}$$

A. Binary-Input Additive White Gaussian Noise Channel (AWGNC)

Since

$$\|\boldsymbol{\omega}(D)\|_1 = \|\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\|_1$$

and

$$\begin{aligned}
\|\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\|_2^2 &= \sum_{\ell=0}^{L-1} \sum_{i=0}^{r-1} \left(\sum_{i'=0}^{\lfloor (t-1)/r \rfloor} \omega_{\ell,i+i'r} \right)^2 \\
&\geq \sum_{\ell=0}^{L-1} \sum_{i=0}^{r-1} \sum_{i'=0}^{\lfloor (t-1)/r \rfloor} \omega_{\ell,i+i'r}^2 \\
&= \|\boldsymbol{\omega}(D)\|_2^2
\end{aligned}$$

we obtain

$$w_{\rm p}^{\rm AWGNC}(\boldsymbol{\omega}(D)) \geq w_{\rm p}^{\rm AWGNC}\big(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\big).$$

B. Binary-Symmetric Channel (BSC)

In order to compare the BSC pseudoweight of the two vectors, we first need to arrange the components in decreasing order. Let

$$M_0 \geq M_1 \geq \cdots \geq M_{tL-1}$$

and

$$m_0 \geq m_1 \geq \cdots \geq m_{rL-1}$$

be lists of all the potentially nonzero coefficients of all the components of $\boldsymbol{\omega}(D)$ and $\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)$, respectively, in nonincreasing order. (In order to simplify the exposition in the following, we assume, without loss of generality, that t is such that $t \geq r$.) Since

$$\|\boldsymbol{\omega}(D)\|_1 = \|\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\|_1$$

we obtain that

$$\frac{\|\boldsymbol{\omega}(D)\|_1}{2} = \frac{\|\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\|_1}{2} \triangleq M$$

which gives $\sum_{i=0}^{tL-1} M_i = \sum_{i=0}^{rL-1} m_i = 2M$. Hence, the two sequences of nonnegative integers form two partitions, λ and μ, respectively, of $2M$. In order to prove the theorem statement for the BSC pseudoweight, it is therefore enough to show that $\sum_{i=0}^{i'-1} M_i \leq \sum_{i=0}^{i'-1} m_i$ for all $i' = 1, 2, \ldots, rL$, i.e., that μ majorizes λ [30].

We show first that $m_0 \geq M_0$. Suppose the contrary, i.e., $m_0 < M_0$. Since $m_i \leq m_0$ for all $i \in \{0, \ldots, rL-1\}$, we

obtain that $m_i < M_0$ for all $i \in \{0,\ldots,rL-1\}$. But m_i, $i \in \{0,\ldots,rL-1\}$, was obtained by adding over $\mathbb{R}_+$ a certain subset of the set $\{M_{i'} \mid i' \in \{0,\ldots,tL-1\}\}$. So there should be at least one $m_{i''}$ that has M_0 in its composition, and hence $m_{i''} \geq M_0$. This is a contradiction, from which we obtain $m_0 \geq M_0$.

We finish the proof by induction. Namely, we want to show that from $\sum_{i=0}^{i'-1} M_i \leq \sum_{i=0}^{i'-1} m_i$ for some $i' \in \{1,\ldots,rL-1\}$, it follows that $\sum_{i=0}^{i'} M_i \leq \sum_{i=0}^{i'} m_i$. If $M_{i'} \leq m_{i'}$, then this induction step clearly holds. So, assume that $M_{i'} > m_{i'}$. Since

$$m_{rL-1} \leq \cdots \leq m_{i'} < M_{i'} \leq M_{i'-1} \leq \cdots \leq M_0$$

we can deduce that $m_{i'}$, and in fact all m_i with $i' \leq i \leq rL-1$, cannot contain any M_i with $0 \leq i \leq i'$ in their composition. Hence, all possible M_i, $0 \leq i \leq i'$, have occurred in the composition of m_i for $0 \leq i \leq i'-1$, which gives $\sum_{i=0}^{i'} m_i \geq \sum_{i=0}^{i'-1} m_i \geq \sum_{i=0}^{i'} M_i$. This proves that μ majorizes λ and we obtain

$$w_{\mathrm{p}}^{\mathrm{BSC}}\big(\boldsymbol{\omega}(D)\big) \geq w_{\mathrm{p}}^{\mathrm{BSC}}\big(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\big).$$

C. Binary Erasure Channel (BEC)

Since the components of the vector $\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)$ are obtained by adding in $\mathbb{R}$ certain nonnegative components of $\boldsymbol{\omega}(D)$, it follows that

$$\left|\mathrm{supp}(\boldsymbol{\omega}(D))\right| \geq \left|\mathrm{supp}\left(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\right)\right|$$

and we obtain $w_{\mathrm{p}}^{\mathrm{BEC}}\big(\boldsymbol{\omega}(D)\big) \geq w_{\mathrm{p}}^{\mathrm{BEC}}\big(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\big)$.

Appendix D
Proof of Theorem 31

We have

$$\|\boldsymbol{\omega}(D)\|_1 = \|\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\|_1$$

and

$$\begin{aligned}\|\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)\|_\infty &= \max_{\ell=0}^{L-1} \max_{i=0}^{r-1} \sum_{i'=0}^{\lfloor (t-1)/r \rfloor} \omega_{\ell,i+i'r} \\ &\geq \max_{\ell=0}^{L-1} \max_{i=0}^{r-1} \max_{i'=0}^{\lfloor (t-1)/r \rfloor} \omega_{\ell,i+i'r} \\ &= \|\boldsymbol{\omega}(D)\|_\infty\end{aligned}$$

which leads to

$$w_{\text{max-frac}}(\boldsymbol{\omega}(X) \bmod_{\mathbb{R}} (X^r-1)) \leq w_{\text{max-frac}}(\boldsymbol{\omega}(D)).$$

It now follows that

$$w_{\text{max-frac}}^{\min}(\boldsymbol{H}_{\mathrm{QC}}^{(\mathrm{r})}) \leq w_{\text{max-frac}}^{\min}(\boldsymbol{H}_{\mathrm{conv}}).$$

Appendix E
Proof of Theorem 32

We know that

$$\begin{aligned} w_{\mathrm{frac}}^{\min}\left(\boldsymbol{H}_{\mathrm{QC}}^{(r)}\right) &\overset{(*)}{\leq} w_{\text{max-frac}}^{\min}\left(\boldsymbol{H}_{\mathrm{QC}}^{(r)}\right) \\ &\overset{(**)}{\leq} w_{\text{max-frac}}^{\min}\left(\boldsymbol{H}_{\mathrm{conv}}\right) \end{aligned} \tag{13}$$

where step $(*)$ follows from [17], [18] (see also [7]) and step $(**)$ follows from Theorem 31.

Similar to the comments before Theorem 32, remember that $w_{\text{max-frac}}^{\min}(\boldsymbol{H}_{\mathrm{QC}}^{(r)})$ has the following meaning [17], [18]. Let $\mathcal{E}_{\mathrm{conv}} \subseteq \mathcal{I}(\boldsymbol{H}_{\mathrm{conv}})$ be the set of positions where the bit flips occurred when using $\mathsf{C}_{\mathrm{conv}}$ for data transmission over a BSC. If $|\mathcal{E}_{\mathrm{conv}}| < \frac{1}{2} w_{\text{max-frac}}^{\min}(\boldsymbol{H}_{\mathrm{conv}})$, then LP decoding succeeds.

Now, because the theorem statement assumes that $|\mathcal{E}_{\mathrm{conv}}| < \frac{1}{2} w_{\mathrm{frac}}^{\min}(\boldsymbol{H}_{\mathrm{QC}}^{(r)})$, using (13) we have $|\mathcal{E}_{\mathrm{conv}}| < \frac{1}{2} w_{\text{max-frac}}^{\min}(\boldsymbol{H}_{\mathrm{conv}})$ and so, according to the meaning of $w_{\text{max-frac}}^{\min}(\boldsymbol{H}_{\mathrm{conv}})$, LP decoding succeeds.

Acknowledgment

The authors gratefully acknowledge the constructive comments made by the reviewers.

References

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[3] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[4] R. M. Tanner, "A recursive approach to low-complexity codes," *IEEE Trans. Inf. Theory*, vol. IT–27, no. 5, pp. 533–547, Sep. 1981.

[5] N. Wiberg, "Codes and Decoding on General Graphs," Ph.D. dissertation, Linköping Univ., Linköping, Sweden, 1996.

[6] R. Koetter and P. O. Vontobel, "Graph covers and iterative decoding of finite-length codes," in *Proc. 3rd Int. Symp. on Turbo Codes and Related Topics*, Brest, France, Sep. 2003, pp. 75–82.

[7] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," *IEEE Trans. Inf. Theory*, 2007, accepted for publication.

[8] W. S. Massey, *Algebraic Topology: An Introduction*, reprint of the 1967 ed. New York: Springer-Verlag, 1977, vol. 56, Graduate Texts in Mathematics.

[9] H. M. Stark and A. A. Terras, "Zeta functions of finite graphs and coverings," *Adv. Math.*, vol. 121, no. 1, pp. 124–165, 1996.

[10] R. Smarandache and P. O. Vontobel, "Pseudo-codeword analysis of Tanner graphs from projective and Euclidean planes," *IEEE Trans. Inform. Theory*, vol. 53, no. 7, pp. 2376–2393, Jul. 2007.

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

[12] 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. Inf. Theory*, vol. 50, no. 12, pp. 2966–2984, Dec. 2004.

[13] R. Smarandache and P. O. Vontobel, "On regular quasi-cyclic LDPC codes from binomials," in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, Jun./Jul. 2004, p. 274.

[14] R. M. Tanner, "Convolutional Codes From Quasi-Cyclic Codes: A Link Between the Theories of Block and Convolutional Codes," UC Santa Cruz, Santa Cruz, CA, 1987, Tech Rep. UCSC-CRL-87-21.

[15] 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)*. Providence, RI: Amer. Math. Soc., 1993, vol. 14, DIMACS Ser. Discrete Math. Theoret. Comput. Sci, pp. 189–198.
[16] M. Esmaeili, T. A. Gulliver, N. P. Secord, and S. A. Mahmoud, "A link between quasi-cyclic codes and convolutional codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 1, pp. 431–435, Jan. 1998.
[17] J. Feldman, "Decoding Error-Correcting Codes via Linear Programming," Ph.D. dissertation, MIT, Cambridge, MA, 2003.
[18] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 954–972, Mar. 2005.
[19] P. O. Vontobel and R. Koetter, "On the relationship between linear programming decoding and min-sum algorithm decoding," in *Proc. Int. Symp. Information Theory and its Applications (ISITA)*, Parma, Italy, Oct. 2004, pp. 991–996.
[20] R. Smarandache, A. E. Pusane, P. O. Vontobel, and D. J. Costello, Jr., "Pseudo-codewords in LDPC convolutional codes," in *Proc. IEEE Int. Symp. Information Theory*, Seattle, WA, Jul. 2006, pp. 1364–1368.
[21] M. Chertkov and M. G. Stepanov, "An efficient pseudocodeword search algorithm for linear programming decoding of LDPC codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1514–1520, Apr. 2008.
[22] A. Sridharan and D. J. Costello, Jr., "A new construction for low density parity check convolutional codes," in *Proc. IEEE Information Theory Workshop*, Bangalore, India, Oct. 2002, p. 212.
[23] A. E. Pusane, R. Smarandache, P. O. Vontobel, and D. J. Costello, Jr., "On deriving good LDPC convolutional codes from QC LDPC block codes," in *Proc. IEEE Int. Symp. Information Theory*, Nice, France, Jun. 2007, pp. 1221–1225.
[24] A. Jiménez-Felström and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.
[25] D. Avis, "LRS: A revised implementation of the reverse search vertex enumeration algorithm," in *Polytopes—Combinatorics and Computation*, G. Kalai and G. M. Ziegler, Eds. Basel, Switzerland: Birkhäuser-Verlag, 2000, pp. 177–198.
[26] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright, "LP decoding corrects a constant fraction of errors," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 82–89, Jan. 2007.
[27] P. O. Vontobel and R. Koetter, "Lower bounds on the minimum pseudo-weight of linear codes," in *Proc. IEEE Int. Symp. Information Theory*, Chicago, IL, Jun./Jul. 2004, p. 70.
[28] S. Lin and D. J. Costello, Jr., *Error Control Coding*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2004.
[29] I. E. Bocharova, M. Handlery, R. Johannesson, and B. D. Kudryashov, "A BEAST for prowling in trees," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1295–1302, Jun. 2004.
[30] A. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications*. San Diego, CA: Academic, 1979.

Roxana Smarandache (S'96–A'01–M'04) has completed her undergraduate studies in mathematics at the University of Bucharest, Romania, in 1996, with a B.S. thesis on number theory. From 1996 to 2001, she pursued the Ph.D. degree in mathematics at the University of Notre Dame, Notre Dame, IN, which she completed in July 2001. Her thesis was in coding theory, with the subject of algebraic convolutional codes.

She is an Associate Professor in the Department of Mathematics and Statistics at San Diego State University, San Diego, CA. During the academic year 1999–2000, she was for six months a Visiting Scholar at EPFL, Lausanne, Switzerland, in the Department of Communication Systems. During the academic year 2005–2006, she was on a leave at the University of Notre Dame, holding a Visiting Assistant Professor position in the Department of Mathematics. During 2008–2009, she was on sabbatical leave during which she visited the Mathematics Department at the University of Zurich, Switzerland, and the Mathematics and Electrical Engineering Departments at the University of Notre Dame. Her research topics are mainly related to coding theory. Her recent interests include low-density parity-check codes, iterative and linear programming decoding, and convolutional codes.

Ali E. Pusane (S'99–M'08) received the B.Sc. and M.Sc. degrees in electronics and communications engineering from Istanbul Technical University, Istanbul, Turkey, in 1999, and 2002, respectively. He received the M.Sc. degree in electrical engineering in 2004, another M.Sc. degree in applied mathematics in 2006, and the Ph.D. degree in electrical engineering in 2008, all from the University of Notre Dame, Notre Dame, IN.

His research interests include error control coding, coded modulation, and information theory.

Pascal O. Vontobel (S'96–M'97) received the Diploma degree in electrical engineering in 1997, the Post-Diploma degree in information techniques in 2002, and the Ph.D. degree in electrical engineering in 2003, all from ETH Zurich, Zurich, Switzerland.

From 1997 to 2002, he was a Research and Teaching Assistant at the Signal and Information Processing Laboratory at ETH Zurich. After being a Postdoctoral Research Associate at the University of Illinois at Urbana-Champaign, at the University of Wisconsin-Madison (Visiting Assistant Professor), and at the Massachusetts Institute of Technology, he joined the Information Theory Research Group at Hewlett-Packard Laboratories in Palo Alto, CA, in the summer of 2006 as a Research Scientist. His research interests lie in information theory, communications, and signal processing.

Dr. Vontobel was awarded the ETH medal for his Ph.D. dissertation.

Daniel J. Costello, Jr. (S'62–M'69–SM'78–F'86–LF'08) received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1969.

He joined the Illinois Institute of Technology as an Assistant Professor in 1969. In 1985, he became Professor at the University of Notre Dame and later served as Department Chair. In 2000, he was named Bettex Professor of Electrical Engineering at Notre Dame. His research interests are in error control coding and coded modulation. He has more than 300 technical publications and has coauthored a textbook entitled *Error Control Coding: Fundamentals and Applications*, the 2nd edition of which was published in 2004.

Dr. Costello has been a member of the IEEE Information Theory Society Board of Governors since 1983, and in 1986 he served as President. He also served as Associate Editor for two IEEE TRANSACTIONS—COMMUNICATIONS and INFORMATION THEORY—and as Co-Chair of the ISIT's in Kobe, Japan (1988), Ulm, Germany (1997), and Chicago, IL (2004). In 1999, he received the Humboldt Research Prize from Germany, and in 2000 he was selected by the IEEE Information Theory Society as a recipient of a Third-Millennium Medal. He was corecipient of the 2009 IEEE Donald G. Fink Prize Paper Award, which recognizes an outstanding survey, review, or tutorial paper in any IEEE publication issued during the previous calendar year.