Girth analysis and design of periodically time-varying SC-LDPC codes

Girth analysis and design of periodically time-varying SC-LDPC codes

Massimo Battaglioni, *Member, IEEE*, Franco Chiaraluce, *Senior Member, IEEE*, Marco Baldi, *Senior Member, IEEE* and Michael Lentmaier, *Senior Member, IEEE*

***Abstract*—Time-varying spatially coupled low-density parity-check (SC-LDPC) codes with very large period are characterized by significantly better error rate performance and girth properties than their time-invariant counterparts, but the number of parameters they require to be described is usually very large and unpractical. Time-invariant SC-LDPC codes, which can be seen as periodically time-varying codes with unitary period, are represented through a small number of parameters and designed exploiting few degrees of freedom, but their error rate performance and girth properties are sub-optimal. In this paper, we show that the limits of time-invariant SC-LDPC codes can be overcome by transforming them into time-varying SC-LDPC codes with very small period.**

In particular, we show that periodically time-varying SC-LDPC codes with small period may exhibit significantly better girth properties than the corresponding time-invariant codes by exploiting a larger number of degrees of freedom in the code design, which however scale at most linearly with the product of the code period and the size of the considered base matrix.

***Index Terms*—Convolutional codes, girth, LDPC codes, spatially coupled codes, time-invariant codes, time-varying codes.**

I. Introduction

Low-density parity-check (LDPC) codes are state-of-the-art error-correcting codes. It has been demonstrated that a subclass of LDPC codes, named spatially coupled low-density parity-check (SC-LDPC) codes, are capable of achieving the channel capacity for a large number of channels, thanks to the so-called threshold saturation phenomenon [2]. SC-LDPC codes are obtained by coupling together L disjoint LDPC code protographs; if $L \to \infty$, spatially coupled low-density parity-check convolutional codes (SC-LDPC-CCs) are obtained; otherwise, block codes are obtained for finite values of L.

SC-LDPC-CCs were first proposed in [3] under the name of low-density parity-check convolutional codes (LDPC-CCs), as the convolutional counterparts of low-density parity-check block codes (LDPC-BCs). As shown, for example, in [4, Theorem 1], many properties of an LDPC-CC, such as its free distance and girth, are at least as good as those of the underlying LDPC-BC. The decoding latency and complexity of these codes under pipeline or sliding window decoding [3], [5], [6] are proportional to their *constraint length* that, in turn, is proportional to the product between the code *block length* a, that is, the number of rows of the syndrome former matrices forming the parity-check matrix, and its *memory* m_s (see [7], [8], and the references therein). For this reason, either one of these parameters is usually kept small. In [3], time-varying SC-LDPC-CCs with excellent properties were designed by considering a small block length and letting the *period* T increase with the memory of the code. This approach was later generalized in [9], [10], where codes with the same constraint length as those in [3] were designed, but performance improvements were achieved. Unfortunately, for the same block length, the number of parameters required to describe a code increases with the period and makes the theoretical analysis and the code design quite problematic.

For this reason, time-invariant SC-LDPC-CCs, introduced in [11], have recently received a great deal of attention. In fact, they are time-varying SC-LDPC-CCs with unitary period and a small number of parameters is thus needed for their description, if the block length is kept small. For the binary erasure channel it was observed in [12] that time-invariant codes with small block length can achieve excellent error rate performance as the memory increases.

The error rate performance of an SC-LDPC code, as that of all LDPC codes, is adversely affected by the presence of cycles in their Tanner graph. Therefore, design approaches often aim at maximizing the minimum length of the cycles in the code Tanner graph, also known as *girth* of the graph. Actually, more subtle harmful objects exist that may cause the failure of iterative decoders: they are known as stopping sets [13], trapping sets [14], or absorbing sets [15], depending on the considered transmission channel and decoder. However, many harmful trapping sets originate from short cycles in the code Tanner graph (see [16], [17] and the references therein for an overview of the connection between cycles and trapping sets). A punctual removal of the cycles forming the most harmful objects is usually a sufficient condition for a significant improvement of the performance of LDPC codes, as shown in [18]–[20]. So, in order to have good performance in the high SNR region, a possible approach consists in removing all the cycles with a given short length [21]. Such an approach, which has been shown to achieve good results [22], [23], is also adopted in this work.

Many recent works have dealt with the girth properties of

The material in this paper has been presented in part at the 2019 IEEE International Symposium on Information Theory, Paris (France) [1].

M. Battaglioni, F. Chiaraluce and M. Baldi are with the Department of Information Engineering, Polytechnic University of Marche, Ancona, Italy (e-mail: {m.battaglioni,f.chiaraluce,m.baldi}@univpm.it).

Michael Lentmaier is with the Department of Electrical and Information Technology, Lund University, Lund, Sweden (e-mail: Michael.Lentmaier@eit.lth.se).

time-invariant SC-LDPC-CCs [23]–[30]. A time-invariant SC-LDPC-CC can be represented by a symbolic matrix $\mathbf{H}(D)$. Each entry of $\mathbf{H}(D)$ is a polynomial in the variable D with binary coefficients. According to the well-known isomorphism between $M \times M$ circulant matrices and polynomials modulo $x^M - 1$, quasi-cyclic low-density parity-check (QC-LDPC) codes also admit the same polynomial representation. Thus, the two worlds can be joined by assuming that an SC-LDPC-CC admits a parity-check matrix in quasi-cyclic form, considering $M \to \infty$ [11]. Furthermore, even if M is finite, the parity-check matrix of a QC-LDPC code has an equivalent representation as that of a time-invariant SC-LDPC-CC with tail-biting termination [31], which can be obtained by applying a proper row and column reordering. The above considerations permit us to take advantage of a number of results that were proven for QC-LDPC codes and apply them to the case of convolutional codes. In particular, we use and extend some results from [4], [32], [33], where QC-LDPC codes with very large girth are analyzed and designed.

We prove that, for a given block length, just increasing the value of the memory is not sufficient to improve the girth of time-invariant codes. However, we show that even allowing a period which is (slightly) larger than 1 permits to achieve larger girths than in the time-invariant case. This was also observed in [34], for a particular family of QC-SC-LDPC codes with period 2 and 3. QC-SC-LDPC codes are obtained by applying a further lifting procedure, using circulant matrices, on the parity-check matrix of an SC-LDPC code. Therefore, the initial SC-LDPC code, which can be obtained by lifting a protograph, acts as a protograph in its turn, thus yielding a code which is both quasi-cyclic (QC) and spatially coupled (SC). However, our analysis is not restricted to the QC case, so it is quite different from that in [34] and our results are more general than those in [34]. We derive a lower bound on the period required to overcome the upper bound on the girth of time-invariant fully-connected monomial codes, which is 12, and through numerical examples we show that our bound is tight, at least for codes that are $(3, a)$-regular. In the time-varying scenario, the degrees of freedom in the code design grow linearly with the period, with respect to the time-invariant setting. The price for this improvement is a larger number of parameters needed for the code representation but, if the period is kept small, the additional number of parameters is also small and the theoretical analysis is still feasible. In particular, we are able to provide explicit expressions to describe the relationship between the degrees of freedom in the code design and the period of the code. Moreover, we take advantage of a new and convenient representation of periodically time-varying SC-LDPC-CCs to design codes with the same block length as the time-invariant ones, but with larger girth (and memory). An intermediate step in this direction is made in [35], where the authors propose to design QC-LDPC codes based on pre-lifted protographs. Other works in which double or multi-step lifting procedures are considered are [22], [36]. We show here that the intermediate step of pre-lifting the initial protograph can be avoided, as long as "periodically" circulant matrices are used in a single comprehensive lifting procedure.

The paper is organized as follows. In Section II, we introduce the notation and describe the various types of LDPC codes considered in our study. In Section III, we find equivalent descriptions for time-invariant and periodically time-varying codes showing that both they allow a QC representation. In Section IV, the girth properties of SC-LDPC-CCs are discussed. In Section V, using the theoretical results derived in the previous sections, we demonstrate that suitably designed time-varying SC-LDPC-CCs may exhibit girth values significantly larger than those of the corresponding time-invariant codes. In Section VI, we provide methods for designing periodically time-varying codes with large girth. In Section VII we assess the error rate performance of the newly designed codes by considering some examples. Finally, Section VIII concludes the paper.

II. Background and Notation

In this section we provide a quick overview of the techniques commonly used to design LDPC codes and introduce the definitions and notation used throughout the paper.

A. LDPC Block Codes

An LDPC-BC with block length n and dimension k is defined as the null space of a sparse parity-check matrix $\mathbf{H}$ with size $r \times n$ and rank $n - k$, with $r \geq n - k$. The rate of such a code is $R = \frac{k}{n}$. The parity-check matrix of a binary LDPC-BC is the bi-adjacency matrix of the so-called Tanner graph [37]. A protograph is a small Tanner graph described by a $c \times a$ biadjacency matrix $\mathbf{B}$, known as a *base matrix*. Each entry $b_{i,j}$ indicates the number of edges between two nodes in the protograph. A protograph-based code can be obtained by "lifting" a protograph[1]. Such a code is described by an $Mc \times Ma$ parity-check matrix obtained by replacing each non-zero entry $b_{i,j}$ of the base matrix by a sum of $b_{i,j}$ non-overlapping permutation matrices of size M and each zero entry by an $M \times M$ zero matrix. An *ensemble* of codes is defined as the collection of all codes sharing the same base matrix and having the same block length.

The parity-check matrix $\mathbf{H}$ of QC-LDPC codes can be obtained from $\mathbf{B}$ by applying a circulant lifting, that is, replacing each non-zero entry of the base matrix by the sum of $b_{i,j}$ non-overlapping circulant matrices of size M. The rate of such a code is not smaller than $1 - \frac{c}{a}$. Its minimum distance is denoted as $d_{\min}$. It is well-known that, if we consider $M \to \infty$, we obtain time-invariant SC-LDPC-CCs [11]. Due to the highly redundant structure of $\mathbf{H}$, a common alternative representation which is often used in the literature to describe QC-LDPC codes exploits polynomials in $\mathbb{F}_2[x]/(x^M - 1)$, where $\mathbb{F}_2[x]/(x^M - 1)$ is the ring of polynomials with coefficients in the Galois field $\mathbb{F}_2$ modulo $x^M - 1$. In this case, the code is described by a $c \times a$ matrix with polynomial entries, that is

$$\mathbf{H}(x) \triangleq \begin{bmatrix} h_{0,0}(x) & \dots & h_{0,a-1}(x) \\ \vdots & \ddots & \vdots \\ h_{c-1,0}(x) & \dots & h_{c-1,a-1}(x) \end{bmatrix}, \tag{1}$$

[1] With a slight abuse of notation, throughout the paper we will also state that the base matrices corresponding to a given protograph can be lifted.

where each $h_{i,j}(x)$, $i=0,1,2,\ldots,c-1$, $j=0,1,2,\ldots,a-1$, is a polynomial $\in \mathbb{F}_2[x]/(x^M-1)$. According to the well-known isomorphism between $M \times M$ circulant matrices and polynomials modulo x^M-1, any polynomial $h_{i,j}(x)$ unambiguously describes an $M \times M$ circulant matrix. In particular, the exponents of $h_{i,j}(x)$ represent the positions of the non-zero elements of the first column (or row) of the corresponding circulant matrix.

B. SC-LDPC Convolutional Codes

Let us briefly recall the concept of time-invariant, periodically time-varying and time-varying codes. Time-varying SC-LDPC-CCs with asymptotic rate $R_\infty = \frac{a-c}{a}$ are characterized by a parity-check matrix

$$\mathbf{H}^{\mathrm{T}} = \begin{bmatrix} \ddots & & \ddots & & & \\ \mathbf{H}_0^{\mathrm{T}}(0) & \ldots & \mathbf{H}_{m_s}^{\mathrm{T}}(m_s) & \mathbf{0} & & \\ & \ddots & \vdots & \ddots & & \\ & \mathbf{0} & \mathbf{H}_0^{\mathrm{T}}(t) & \ldots & \mathbf{H}_{m_s}^{\mathrm{T}}(t+m_s) & \mathbf{0} \\ & & \ddots & & \ddots & \end{bmatrix}, \tag{2}$$

where $^{\mathrm{T}}$ denotes transposition and each block $\mathbf{H}_i(t)$, $i = 0,1,2,\ldots,m_s$, is a binary matrix with size $c \times a$, as in

$$\mathbf{H}_i(t) = \begin{bmatrix} h_i^{(0,0)}(t) & \ldots & h_i^{(0,a-1)}(t) \\ \vdots & \ddots & \vdots \\ h_i^{(c-1,0)}(t) & \ldots & h_i^{(c-1,a-1)}(t) \end{bmatrix}, \tag{3}$$

where a is defined as the *block length* of the code.

The parity-check matrix $\mathbf{H}$ is said to be (d_v, d_c)-regular if all its rows have Hamming weight d_c and all its columns have Hamming weight d_v. We also define

$$\mathbf{H_s}(t) \triangleq \left[\mathbf{H}_{m_s}^{\mathrm{T}}(t)|\mathbf{H}_{m_s-1}^{\mathrm{T}}(t)|\ldots|\mathbf{H}_0^{\mathrm{T}}(t)\right]$$

as the t-th syndrome former matrix. The variable m_s is the syndrome former memory order (sometimes simply addressed to as *memory*) of the code and $\nu_s = (m_s+1)a$ is its syndrome former constraint length. Since these codes are convolutional, $M \to \infty$. If $\mathbf{H}_i(t) = \mathbf{H}_i(t+T)$ for a finite value of T, the corresponding code is said to be *periodically time-varying* with *period* T. When $T=1$, we say that the code is *time-invariant*. Time-invariant codes are characterized by a unique $\mathbf{H_s}$.

The first approaches to the design of SC-LDPC-CCs showed that the parity-check matrix of these codes can be directly obtained from that of LDPC-BCs by applying suitable unwrapping techniques [3], [9], [11], without the need of introducing a protograph-based design. In particular, a cut-and-paste technique was proposed in [3], and later generalized in [9], [10], which permits to obtain time-varying SC-LDPC-CCs starting from LDPC-BCs. Another approach, proposed in [11], allows to construct time-invariant SC-LDPC-CCs by applying an $M-$lifting procedure, with $M \to \infty$, to the graph of QC-LDPC codes.

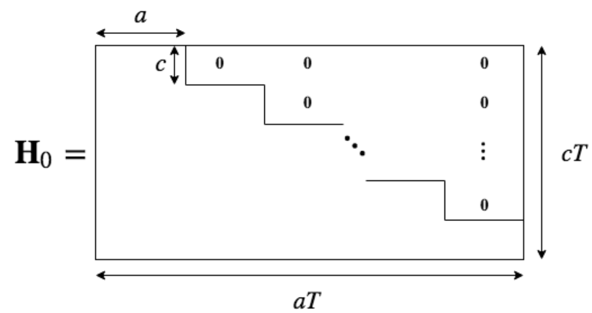

Fig. 1. Submatrix with a stair-like structure.

Remark 1 Periodically time-varying codes with period $T > 1$ can be seen as time-invariant codes, whose $\mathbf{H_s}$ is obtained as the concatenation of the first T syndrome former matrices of the periodically time-varying code. The block length of the time-invariant code is aT, and the number of parity symbols per time instant is cT. Notice that the converse, in general, is not necessarily true. Indeed, given $T > 1$, not all the time-invariant codes with block length aT and cT parity symbols per time instant can be seen as periodically time-varying codes with block length a and c parity symbols per time instant. More precisely, a time-invariant code with block length aT and cT parity symbols per time instant can be represented as a periodically time-varying code with period T, block length a and c parity symbols per time instant if and only if its $\mathbf{H}_0$ is as in Fig. 1.

The symbolic representation of the syndrome former matrix of time-invariant SC-LDPC-CCs exploits polynomials in $\mathbb{F}_2[D]$, where $\mathbb{F}_2[D]$ is the ring of polynomials with coefficients in the Galois field $\mathbb{F}_2$. As in the QC-LDPC case, the code is described by a $c \times a$ *symbolic matrix* having polynomial entries, that is

$$\mathbf{H}(D) \triangleq \begin{bmatrix} h_{0,0}(D) & \ldots & h_{0,a-1}(D) \\ \vdots & \ddots & \vdots \\ h_{c-1,0}(D) & \ldots & h_{c-1,a-1}(D) \end{bmatrix}, \tag{4}$$

where each $h_{i,j}(D)$, $i=0,1,2,\ldots,c-1$, $j=0,1,2,\ldots,a-1$, is a polynomial in $\mathbb{F}_2[D]$. If $\mathbf{H}(D)$ contains only monomial entries, the code is said a fully-connected monomial code. The code representation based on $\mathbf{H_s}$ can be converted into that based on $\mathbf{H}(D)$ as follows

$$h_{i,j}(D) = \sum_{m=0}^{m_s} h_m^{(i,j)} \cdot D^m. \tag{5}$$

The symbolic representation can also be used if the code is periodically time-varying with period T, by placing T suitably shifted symbolic matrices side-by-side, as in

$$\mathbf{H}(D) \triangleq \left[\; \mathbf{H}_0(D) \;\middle|\; D \cdot \mathbf{H}_1(D) \;\middle|\; \ldots \;\middle|\; D^{T-1} \cdot \mathbf{H}_{T-1}(D) \;\right] \tag{6}$$

where each $\mathbf{H}_i(D)$ is the symbolic matrix of $\mathbf{H_s}(i)$. Notice that the size of (6), which is a $c \times aT$ matrix, is reduced by a factor T with respect to that of the equivalent time-invariant code which, according to Remark 1 and (4), is a $cT \times aT$ matrix.

We define the *exponent matrix* $\mathbf{P}$ as the matrix containing the exponents of the entries of $\mathbf{H}(D)$, which has the following form

$$\mathbf{P} \triangleq \begin{bmatrix} \mathbf{p}_{0,0} & \cdots & \mathbf{p}_{0,a-1} \\ \vdots & \ddots & \vdots \\ \mathbf{p}_{c-1,0} & \cdots & \mathbf{p}_{c-1,a-1} \end{bmatrix}. \quad (7)$$

In particular, any $\mathbf{p}_{i,j}$ is a vector containing different entries, and its zth entry $p_{i,j,z}$ is obtained as the $\log_D$ of the zth monomial in $\mathbf{H}_i(D)$. We can assume without loss of generality that the entries of $\mathbf{p}_{i,j}$ are in ascending order, that is, $p_{i,j,z_0} < p_{i,j,z_1}$ when $z_0 < z_1$.

We define $\mathcal{W}(\cdot)$ as the Hamming weight function, extended to the case of inputs in the form of matrices or vectors with polynomial entries (in which case it returns a weight matrix or vector). The matrix $\mathcal{W}(\mathbf{H}(D))$ corresponds to the base matrix of the code; so, in the rest of the paper we denote $\mathcal{W}(\mathbf{H}(D))$ as $\mathbf{B}$. Similarly to block codes, we define an *ensemble of codes* $\mathcal{E}(\mathbf{B})$ as the collection of all codes characterized by the same $\mathbf{B}$. In the time-varying case, if $\mathcal{W}(\mathbf{H}_i(D)) = \mathcal{W}(\mathbf{H}_j(D))$, $\forall i \neq j$, we assume without loss of generality that $\mathbf{B} = \mathcal{W}(\mathbf{H}_0(D))$.

Example 1 Let us consider the following symbolic matrix, representing a time-varying SC-LDPC-CC with period $T = 2$, block length $a = 3$, $c = 2$ parity symbols per period, memory $m_s = 2$, constraint length $\nu_s = (m_s+1)a = 9$ and asymptotic rate $R_\infty = \frac{1}{3}$:

$$\mathbf{H}(D) = \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & D & D & D \\ 1 & D & D^2 & D & D^3 & D^2 \end{array}\right]. \quad (8)$$

According to (5), we have that

$$\mathbf{H}_0(0) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix},$$

$$\mathbf{H}_1(1) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

$$\mathbf{H}_2(2) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and $\mathbf{H}_0(1) = \mathbf{H}_0(0)$, $\mathbf{H}_1(2) = \mathbf{H}_2(2)$ and $\mathbf{H}_2(3) = \mathbf{H}_1(1)$. The resulting binary parity-check matrix is

$$\mathbf{H}^{\mathrm{T}} = \begin{bmatrix} \ddots & & & & & & & & \\ 1 & 1 & 0 & 0 & 0 & 0 & & & \\ 1 & 0 & 0 & 1 & 0 & 0 & & & \\ 1 & 0 & 0 & 0 & 0 & 1 & & & \\ & & 1 & 1 & 0 & 0 & 0 & 0 & \\ & & 1 & 0 & 0 & 0 & 0 & 1 & \\ & & 1 & 0 & 0 & 1 & 0 & 0 & \\ & & & & & & & & \ddots \end{bmatrix},$$

Notice that the considered code is fully-connected monomial, and can be compactly represented by the following exponent matrix

$$\mathbf{P} = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 1 & 3 & 2 \end{array}\right].$$

Finally, applying the Hamming weight function to (8) we notice that $\mathcal{W}(\mathbf{H}_0(D)) = \mathcal{W}(\mathbf{H}_1(D))$ and thus the base matrix can be written as

$$\mathbf{B} = \mathcal{W}(\mathbf{H}_0(D)) = \mathcal{W}(\mathbf{H}_1(D)) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Remark 2 Notice that, in principle, the passage from a periodically time-varying code with period T to a time-invariant code with larger block length may have significant drawbacks in terms of decoding latency and complexity. These two quantities, for belief propagation (BP)-based sliding window (SW) decoders [3], [5], [6] depend on the window size, which is usually chosen as an integer multiple of the syndrome former constraint length [7]. Indeed, on the one hand, the time-varying version of the code has syndrome former constraint length $\nu_s^{\mathrm{T.V.}} = (m_s+1)a$, which does not depend on the period. On the other hand, the time-invariant code has syndrome former constraint length $\nu_s^{\mathrm{T.I.}} = Ta(\lceil \frac{m_s}{T} \rceil + 1)$ which, differently from $\nu_s^{\mathrm{T.V.}}$, has a linear dependence on the period. This is a crucial difference with respect to previous works focusing on the design of QC-LDPC codes with large girth and symbolic matrix with size $Tc \times Ta$ [4], [22], [33], [35]. Indeed, even though the $c \times Ta$ symbolic matrix of a time-varying code can also be written as a $Tc \times Ta$ symbolic matrix, the latter guarantees that a stair-like structure as in Fig. 1 is maintained in $\mathbf{H}_0$. In other words, the $Tc \times Ta$ symbolic matrix of the time-invariant code corresponding to a time-varying code with period T has to satisfy certain constraints.

Clearly, the design of a time-invariant code described by an "unconstrained" $Tc \times Ta$ symbolic matrix is more general than that just described in Remark 2, which has to ensure that a stair-like structure in $\mathbf{H}_0$ is maintained. However, for the reasons explained above, it would results in a less compact description and in a larger syndrome former constraint length. The use of pre-lifted protographs [22], [35], [36] helps in this direction, as it reduces the number of possible $Tc \times Ta$ symbolic matrices. Still, the number of symbolic matrices obtainable using pre-lifting increases more than linearly with the block length. Moreover, in the general case, the block length increases linearly with the pre-lifting size. In our case, exploiting time-varying codes, the block length is fixed and the number of parameters required for description grows at most linearly with the period T.

C. Tail Biting SC-LDPC Convolutional Codes

For practical reasons, SC-LDPC-CCs need to be terminated at some point. Let us consider the following section of the semi-infinite parity-check matrix (2)

$$\mathbf{H}_{[0,L]}^{\mathrm{T}} = \begin{bmatrix} \mathbf{H}_0^{\mathrm{T}}(0) & \ldots & \mathbf{H}_{m_s}^{\mathrm{T}}(m_s) & & \\ & \ddots & \vdots & \ddots & \\ & & \mathbf{H}_0^{\mathrm{T}}(L) & \ldots & \mathbf{H}_{m_s}^{\mathrm{T}}(L+m_s) \end{bmatrix}. \quad (9)$$

An SC-LDPC-CC terminated in tail-biting fashion, or briefly a tail-biting SC-LDPC-CC, with coupling length $L >$

$$
\tilde{\mathbf{H}}^{\mathrm{T}}_{[0,L]} = \begin{bmatrix}
\mathbf{H}^{\mathrm{T}}_0(0) & \mathbf{H}^{\mathrm{T}}_1(1) & \cdots & \mathbf{H}^{\mathrm{T}}_{m_s}(m_s) & \mathbf{0} & \\
\mathbf{0} & \mathbf{H}^{\mathrm{T}}_0(1) & \cdots & \cdots & \mathbf{H}^{\mathrm{T}}_{m_s}(m_s+1) & \mathbf{0} \\
 & & \ddots & \vdots & \ddots & \\
\mathbf{H}^{\mathrm{T}}_{m_s}(L+1) & \mathbf{0} & & \mathbf{H}^{\mathrm{T}}_0(L-m_s+1) & \cdots & \mathbf{H}^{\mathrm{T}}_{m_s-1}(L) \\
\mathbf{H}^{\mathrm{T}}_{m_s-1}(L+1) & \mathbf{H}^{\mathrm{T}}_{m_s}(L+2) & \mathbf{0} & & \ddots & \vdots \\
\vdots & & & & \mathbf{H}^{\mathrm{T}}_0(L-1) & \mathbf{H}^{\mathrm{T}}_1(L) \\
\mathbf{H}^{\mathrm{T}}_1(L+1) & \cdots & \mathbf{H}^{\mathrm{T}}_{m_s}(L+m_s) & \mathbf{0} & & \mathbf{H}^{\mathrm{T}}_0(L)
\end{bmatrix} \tag{10}
$$

m_s, is obtained by wrapping back the last $m_s c$ columns of (9) after L times instants. The corresponding matrix is as in (10).

Note that the rate of the corresponding code is the same as the asymptotic rate of the initial time-varying SC-LDPC-CC.

D. Cycles

Each symbol 1 in the parity-check matrix corresponds to a segment in the code Tanner graph. Thus, we define a *walk* in $\mathbf{H}$ as a sequence of alternating horizontal and vertical segments between symbols 1 in the same row and columns, respectively. A *path* in $\mathbf{H}$ is a walk in which all the symbols 1 are distinct. A *cycle* of length λ in $\mathbf{H}$ is defined by

- a path of length $\lambda-1$, in which the starting and the ending symbols 1 are in the same row (or in the same column) of $\mathbf{H}$
- an additional horizontal (or vertical) segment connecting the starting and the ending symbols 1.

A *simple cycle* in $\mathbf{H}$ is a cycle which does not contain any other cycle. Simple cycles are formed by a single vertical segment and a single horizontal segment in each column and row of $\mathbf{H}$ they involve. The *girth* g of a code is defined as the length of its shortest cycle. In general, cycles can be grouped in two categories: the avoidable cycles and the unavoidable cycles. The avoidable cycles can be indeed avoided with a proper choice of the entries of $\mathbf{P}$, whereas unavoidable cycles only depend on $\mathbf{B}$.

It is shown in [38], and later generalized in [39] for codes characterized by $\mathbf{H}(D)$ including generic polynomial entries, that a necessary and sufficient condition for the existence of a cycle with length $2k$ in the Tanner graph of a QC-LDPC block code is

$$
\sum_{i=0}^{k-1} \left(p_{m_i,n_i,l_i} - p_{m_i,n_{i+1},l'_i} \right) = 0 \mod M, \tag{11}
$$

where $n_k = n_0$, $l_i \neq l'_i$ if $n_i = n_{i+1}$ and $l'_i \neq l_{i+1}$ if $m_i = m_{i+1}$.

III. A Bridge between Block and Convolutional LDPC Codes

As mentioned in the previous sections, both QC-LDPC codes and time-invariant SC-LDPC-CCs admit the same symbolic representation. In this section we show that the same connection exists between periodically time-varying SC-LDPC-CCs and a class of block codes based on *periodically circulant matrices*. We begin our analysis recalling the two well-known alternative representations of the parity-check matrix of a QC-LDPC block code, and establishing some relationships between these codes and SC-LDPC-CCs.[2]

A. Time-Invariant Codes

Remark 3 The parity-check matrix of a QC-LDPC code with rate $R = \frac{a-c}{a}$, expressed as a circulants block matrix, has an equivalent form, characterizing a tail-biting time-invariant SC-LDPC convolutional code.

The two representations, are:

$$
\mathbf{H}_{\mathrm{QC}} = \begin{bmatrix}
\mathbf{Q}^{(0,0)} & \cdots & \mathbf{Q}^{(0,a-1)} \\
\vdots & \ddots & \vdots \\
\mathbf{Q}^{(c-1,0)} & \cdots & \mathbf{Q}^{(c-1,a-1)}
\end{bmatrix}, \tag{12}
$$

$$
\mathbf{H}^{\mathrm{tb}} = \begin{bmatrix}
\mathbf{H}_0 & \mathbf{H}_{M-1} & \cdots & \mathbf{H}_1 \\
\mathbf{H}_1 & \mathbf{H}_0 & \ddots & \mathbf{H}_2 \\
\vdots & \ddots & \ddots & \vdots \\
\mathbf{H}_{M-1} & \mathbf{H}_{M-2} & \cdots & \mathbf{H}_0
\end{bmatrix}, \tag{13}
$$

respectively. We must note that each $\mathbf{Q}^{(i,j)}$ is a square circulant matrix with size M, whereas each $\mathbf{H}_i$ is a $c \times a$ matrix; so, $\mathbf{H}_{\mathrm{QC}}$ and $\mathbf{H}^{\mathrm{tb}}$ have the same size.

The parity-check matrix as in (13) was first described in [40], which is the first work dealing with QC-LDPC codes. The equivalent circulants block form has been introduced, instead, in [41]. In order to switch from one form to the other a reordering of the rows and the columns of the parity-check matrix is required. The procedure to pass from the blocks circulant form to the circulants block form, addressed to as **Procedure 1** in the rest of the paper, is as follows

1) Apply a permutation $\pi_1 : \{0, 1, \ldots, aM-1\} \rightarrow \{0, 1, \ldots, aM-1\}$ to the columns of $\mathbf{H}^{\mathrm{tb}}$ such that, for the j-th column, $\pi_1(j) = \lfloor \frac{j}{M} \rfloor + (j \mod M)a$, $j = 0, 1, \ldots, aM-1$
2) Apply a permutation $\pi_2 : \{0, 1, \ldots, cM-1\} \rightarrow \{0, 1, \ldots, cM-1\}$ to the rows of the parity-check matrix obtained in step 1) such that, for the i-th row, $\pi_2(i) = \lfloor \frac{i}{M} \rfloor + (i \mod M)c$, $i = 0, 1, \ldots, cM-1$.

The following lemma also holds.

[2]For the sake of brevity, the most straightforward proofs are omitted in this section.

Lemma 1 Given a QC-LDPC code described by (12), the elements of any circulant matrix $\mathbf{Q}^{(i',j')}$, $i' = 0,\dots,c-1$, $j' = 0,\dots,a-1$, must satisfy

$$q_{i,0}^{(i',j')} = 0 \quad \forall i > m_s,$$

where m_s is the syndrome former memory order of the underlying tail-biting SC-LDPC-CC.

In conclusion, by applying a circular lifting to a given base matrix, we can either obtain a QC-LDPC code or a tail-biting time-invariant SC-LDPC-CC. Let us now generalize this representation to periodically time-varying SC-LDPC-CCs with period T.

B. Periodically Time-Varying Codes

In order to generalize the above considerations to periodically time-varying codes with period T, we need to introduce the concept of *periodically circulant matrix* with period T.

An $M \times M$ periodically circulant permutation matrix $\mathbf{P}$ with period T is a permutation matrix such that $p_{[(i+T) \mod M,(j+T) \mod M]} = p_{i,j}$, for all $0 \le i,j \le M-1$. In order to describe unambiguously a periodically circulant permutation matrix, at most T integers are needed, corresponding to the positions of the T symbols 1 in the first T columns. If a matrix can be obtained as a sum of periodically circulant permutation matrices with non-overlapping support, we generically define it as a *periodically circulant matrix.*

Example 2 The following periodically circulant permutation matrix of size 6 and period 2

$$\left[\begin{array}{cc:cc:cc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right]$$

is unambiguously described by its first two columns. The other columns can be obtained by progressively shifting the first two columns by $T = 2$ positions downward. Since this matrix is also a permutation matrix, two integers (the supports of the first and of the second column, i.e., 0 and 3 in this example, respectively) are enough to represent it.

Let us consider a tail-biting periodically time-varying SC-LDPC convolutional code with period T and block length $n = aM$, where M is an integer multiple of T. The following lemmas hold.

Lemma 2 Let us consider a parity-check matrix having the following form

$$\mathbf{H}^{\mathrm{tb}} = \begin{bmatrix} \mathbf{H}_0(0) & \mathbf{H}_{M-1}(0) & \dots & \mathbf{H}_1(0) \\ \mathbf{H}_1(1) & \mathbf{H}_0(1) & \ddots & \mathbf{H}_2(1) \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{H}_{M-1}(T-1) & \mathbf{H}_{M-2}(T-1) & \dots & \mathbf{H}_0(T-1) \end{bmatrix} \tag{14}$$

describing a tail-biting periodically time-varying SC-LDPC convolutional code with period T and block length $n = aM$, such that $M = kT$, $k > 1$. By properly reordering its rows and columns with Procedure 1, the following parity-check matrix can be obtained

$$\mathbf{H}_{\mathrm{PM}} = \begin{bmatrix} \mathbf{P}^{(0,0)} & \dots & \mathbf{P}^{(0,a-1)} \\ \vdots & \ddots & \vdots \\ \mathbf{P}^{(c-1,0)} & \dots & \mathbf{P}^{(c-1,a-1)} \end{bmatrix}, \tag{15}$$

where each $\mathbf{P}^{(i',j')}$ is an $M \times M$ periodically circulant matrix with period T. Furthermore, the following condition holds for the elements of each of the ac periodically circulant matrices with period T:

$$p_{i,j}^{(i',j')} = 0, \forall i < j, i > j + m_s \quad \text{with} \quad j < T.$$

Lemma 3 Given a parity-check matrix as in (14), with block length $n = aM$, being $M = kT$, $k > 1$, by properly reordering its rows and columns, a parity-check matrix in form

$$\mathbf{H}_{\mathrm{QC}} = \begin{bmatrix} \mathbf{Q}^{(0,0)} & \dots & \mathbf{Q}^{(0,T(a-1))} \\ \vdots & \ddots & \vdots \\ \mathbf{Q}^{(T(c-1),0)} & \dots & \mathbf{Q}^{(T(c-1),T(a-1))} \end{bmatrix} \tag{16}$$

can always be obtained, where each $\mathbf{Q}^{(i',j')}$ is an $\frac{M}{T} \times \frac{M}{T}$ circulant matrix.

Proof: As discussed in Remark 1, any periodically time-varying code with period T and asymptotic rate $R = \frac{a-c}{a}$ can be seen as a time-invariant code with block length Ta and Tc parity symbols per time instant. Given this, the above lemma easily follows from Remark 3. ∎

It follows from Lemmas 2 and 3 that a tail-biting time-varying SC-LDPC-CC with period T and $R_\infty = \frac{a-c}{a}$ can be obtained by either lifting a smaller base matrix with size $c \times a$ with relatively large periodically circulant matrices, or lifting a larger base matrix with size $Tc \times Ta$ with relatively small circulant matrices.

Example 3 Let us consider the parity-check matrix of a periodically time-varying code with $T = 2$, $a = 3$, $c = 2$, $M = 6$ in tail-biting form, as in (17).

According to Remark 1, (17) is also the parity-check matrix of a time-invariant code with $a = 6$, $c = 4$ and $M = 3$. Applying the reordering procedure to these two different versions, we obtain (18) and (19), respectively.

Summarizing, (17) can be obtained by either lifting

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

with periodic permutation matrices of size 6, obtaining (18), or by lifting

$$\mathbf{B} = \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right] \tag{20}$$

$$\mathbf{H}^{\mathrm{tb}} = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc}
0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1\\ \hline
0&1&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0\\ \hline
0&0&0&1&0&0&0&0&1&0&0&0&0&0&0&0&1&0\\
0&1&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\ \hline
1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1\\
0&0&0&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0\\ \hline
0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&1&1&0&0&0&0&0\\ \hline
0&0&0&0&0&1&1&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0&0&0
\end{array}\right] \tag{17}$$

$$\mathbf{H}_{\mathrm{PM}} = \left[\begin{array}{cccccc|cccccc|cccccc}
0&0&0&0&0&1&0&0&0&1&0&0&1&0&0&0&0&0\\
0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&1&0&0\\
0&1&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0\\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1\\
0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0\\
0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0\\ \hline
1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1\\
0&0&0&0&0&1&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0\\
0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1&0
\end{array}\right] \tag{18}$$

$$\mathbf{H}_{\mathrm{QC}} = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc}
0&0&0&0&0&0&1&0&0&0&0&1&0&1&0&0&0&0\\
0&0&0&0&0&0&0&1&0&1&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&1&0&0&0&0&0\\ \hline
1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1\\
0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0\\ \hline
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\
1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0\\ \hline
0&0&0&0&0&0&1&0&0&0&0&1&0&0&1&0&0&0\\
0&0&0&0&0&0&0&1&0&1&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&1&0&0&0&0
\end{array}\right] \tag{19}$$

with circulant permutation matrices of size 3, obtaining (19). Finally, we remark that (18) can be represented with 10 integers (in general, at most 12 integers are required to represent a periodically time-varying code in tail-biting form with $a = 3$, $c = 2$ and $T = 2$, obtained by lifting (20)). Instead, a time-invariant code in tail-biting form with blocklength $a' = Ta = 6$, same code rate (that, is, $c' = Tc = 4$) would in general require twice as many integers for representation, i.e., 24 integers, as indicated by the number of circulant matrices in (19).

So, we have shown that both time-invariant and time-varying codes terminated in tail-biting fashion, for a finite value of M, or unterminated, for $M \to \infty$, allow a QC representation. This permits us to derive theoretical results taking advantage of many previous results that were proven for QC-LDPC codes and that, according to the above discussion, can be easily extended to our scenario. Furthermore, as we will show in Section VI, the analysis in this section permits us to extend the design methods we also provide to the block codes scenario.

IV. Girth Properties of SC-LDPC-CCs

In this section we first show how the girth properties of QC-LDPC codes and SC-LDPC-CCs are related. We derive sufficient conditions to guarantee that the girth of well-designed SC-LDPC-CCs is maintained in the underlying block code. Then, we discuss the limits of time-invariant codes.

A. *Girth and free girth*

The following result was given in [4, Theorem 1].

Lemma 4 The girth of a time-invariant SC-LDPC-CC, denoted as g_{free}, is lower bounded by the girth of the underlying QC-LDPC code, denoted as g_{QC}, that is,

$$g_{\text{QC}} \le g_{\text{free}}.$$

In the following theorem we provide and prove a sufficient condition on the block length of any QC-LDPC code such that its girth is the same as that of the overlying SC-LDPC-CC.

Theorem 1 Given a QC-LDPC code, whose $\mathbf{H}$ is a $c \times a$ array of $M \times M$ circulant matrices, with girth g_{QC} and its convolutional counterpart with girth g_{free}, if

$$M > \frac{g_{\text{free}} - 2}{2} m_s, \tag{21}$$

then $g_{\text{QC}} = g_{\text{free}}$.

Proof: See Appendix A. ■

Equation (21) only represents a sufficient condition and therefore it does not always provide the smallest value of M such that $g_{\text{QC}} = g_{\text{free}}$. However, assuming we have designed a time-invariant code with girth g_{free}, satisfying (21) allows to obtain a QC-LDPC code with the same girth. This permits us to design block and convolutional codes with the same girth simultaneously. In order to evaluate the tightness of (21) to the actual minimum values of M, denoted as $M_{\min}$, which are required to achieve $g_{\text{QC}} = g_{\text{free}}$, $\forall M > M_{\min} - 1$, we have considered some codes from [42] and obtained the results shown in Table I.

TABLE I
MINIMUM VALUES OF THE CIRCULANT SIZE M SUCH THAT $g_{\text{QC}} = g_{\text{free}}$, COMPARED TO THE SUFFICIENT VALUE GIVEN BY (21).

R	g_{free}	$\frac{g_{\text{free}}-2}{2} m_s + 1$	$M_{\min}$
$\frac{1}{4}$	8	10	9
$\frac{2}{5}$	8	19	17
$\frac{3}{6}$	8	22	18
$\frac{1}{4}$	10	45	45
$\frac{2}{5}$	10	77	75
$\frac{3}{6}$	10	125	123
$\frac{1}{4}$	12	101	97
$\frac{2}{5}$	12	211	195
$\frac{3}{6}$	12	541	466

We remark that (21) is independent of the code rate and the content of the parity-check matrix. So, the above result can be extended to codes obtained by lifting a base matrix not only with circulant matrices, but also with periodically circulant matrices having period T.

Corollary 1 Given a tail-biting time-varying SC-LDPC-CC with finite period T, reordered with Procedure 1, ($c \times a$ array of $M' \times M'$ periodically circulant matrices with period T), having girth g, and its convolutional version with girth g_{free}, if

$$M' > \frac{g_{\text{free}} - 2}{2} m_s T, \tag{22}$$

then $g = g_{\text{free}}$.

Proof: Equation (22) is simply obtained by substituting M and m_s in (21) with the corresponding parameters of the equivalent time-invariant code, according to Remark 1. ■

Theorem 1 and Corollary 1 allow to extend our analysis and design methods, focused on SC-LDPC-CCs, also to block codes: if a time-invariant (time-varying, respectively) SC-LDPC-CC with girth g^* has been designed, then a block code with girth g^* can be obtained by lifting the same base matrix with the same circulant (periodically circulant, respectively) matrices, but assuming $M = \frac{g^*-2}{2} m_s T + 1$ instead of $M \to \infty$.

B. Limits of Time-Invariant SC-LDPC-CCs

In this section we first derive some results on the cycle properties of base matrices, which are needed for the proof of the main statement, that is, Theorem 2.

Lemma 5 Given a $c \times a$ base matrix $\mathbf{B}$, if $\mathbf{B}$ does not contain any cycle (i.e., it is *acyclic*), then it has at least one row or one column with unitary weight.

Proof: The matrix $\mathbf{B}$ has finite size, so there exists at least one path in $\mathbf{B}$ which has the largest possible length. If multiple paths with this length exist, let us consider any of them without loss of generality. We denote the first and last symbols 1 of this path as s_0 and s_{end}, respectively. Let us suppose that $\mathbf{B}$ does not contain any row and any column with unitary weight. Then, s_0 (and also s_{end}) can be connected to another symbol 1 which does not belong do the longest path. Thus, the longest path can be lengthened, which is a contradiction. It follows that $\mathbf{B}$ must have at least one row or one column with unitary weight. ■

Lemma 6 Given a $c \times a$ (d_v, d_c)-regular base matrix $\mathbf{B}$, such that $d_c > 1$ and $d_v > 1$, $\mathbf{B}$ contains at least one cycle.

Proof: Follows from Lemma 5. In fact, if d_c and d_v are larger than 1, the bipartite graph corresponding to the given base matrix has no vertices with unitary degree; so, it cannot be acyclic. ■

Lemma 7 A $c \times a$ (d_v, d_c)-regular base matrix $\mathbf{B}$, such that $d_c > d_v > 1$, contains at least two cycles.

Proof: It follows from Lemma 6 that $\mathbf{B}$ contains at least one cycle. Let us suppose that $\mathbf{B}$ contains a single cycle with length λ, and assume that this cycle involves columns with index in $\{c_0, \ldots, c_{\frac{\lambda}{2}-1}\}$. Moreover, let us suppose that we remove any of these columns from $\mathbf{B}$, obtaining the matrix $\mathbf{B}^*$; the corresponding bipartite graph is acyclic, according to the initial assumption. However, $\mathbf{B}^*$ has $a-1$ columns with weight $d_v > 1$, λ rows with weight $d_c - 1 > 1$ and $c - \lambda$ rows with weight $d_c > 1$. This contradicts Lemma 5, which demands at least a row or a column with unitary weight in acyclic matrices. Hence, $\mathbf{B}$ must contain at least two cycles. ■

The following result, proven in [32] for QC-LDPC block codes, can also be applied to SC-LDPC-CCs.

Lemma 8 If two cycles of length λ_1 and λ_2 in a base matrix $\mathbf{B}$ have ρ symbols 1 in common, then there is an unavoidable cycle of length $2(\lambda_1 + \lambda_2 - \rho)$ in the parity-check matrix of the codes in $\mathcal{E}(\mathbf{B})$.

Proof: The unavoidable cycles of this type have been shown in [32] to satisfy (11) independently of M. Hence, they also exist in the convolutional case. ■

In [25], [36], some results on unavoidable cycles yielded by base matrices that do not contain zero entries are presented. This is not sufficient to cover our general case, and we also need to include base matrices containing zero entries in our analysis. This way, our analysis covers all possible base matrices of regular codes. This is done in the following lemmas.

Lemma 9 Given a $c \times a$ (d_v, d_c)-regular base matrix $\mathbf{B}$, such that $d_c > d_v > 1$, if it contains:

1) a 3, or an entry larger than 3, then the girth of any code in $\mathcal{E}(\mathbf{B})$ is bounded above by 6;
2) one pair of symbols 2, then the girth of any code in $\mathcal{E}(\mathbf{B})$ is bounded above by $4(\mathcal{L}+2)$, where $\mathcal{L}$ is the number of symbols in the shortest path between the two symbols 2, starting and ending with a horizontal segment[3];
3) a 2, then the girth of any code in $\mathcal{E}(\mathbf{B})$ is bounded above by:
   a) $2\lambda + 2$ if this symbol 2 is involved in at least one cycle, where λ is the length of the shortest cycle in which the symbol 2 is involved[4];
   b) $2(\lambda + 2\mathcal{L})$ if this symbol is not involved in any cycle, where $\mathcal{L}$ is the length of a path connecting the symbol 2 to a cycle with length λ.

Proof: See Appendix B. ■

The above lemma covers all possible cases in which the matrix contains at least one entry larger than 1. If the entries of $\mathbf{B}$ are restricted to be in $\{0, 1\}$, then we can take advantage of the following result, given in [26].

Lemma 10 If the base matrix $\mathbf{B}$ includes:

- an $m \times n$ submatrix, with $m = n + 1$, containing two rows or two columns with weight greater than 2, or
- an $m \times n$ submatrix, with $m = n + 1$, containing one row or one column with weight greater than 3, or
- an $m \times n$ submatrix, with $m = n$, containing one row and one column with weight greater than or equal to 3,

and all the other rows and columns have weight greater than or equal to 2, then the girth of any time-invariant code in $\mathcal{E}(\mathbf{B})$ is bounded above by $2(m + n + 1)$.

Proof: The proof is the same as in [26, Property 4]. ■

[3]Notice that if the two symbols are in the same row or column, then $\mathcal{L} = 0$ and the girth is bounded above by 8, as stated in [25].

[4]We remark that, in [25], zero entries are not allowed; so, $\lambda = 4$ and the girth is bounded above by 10.

Summarizing, we have shown in Lemmas 8, 9 and 10 that, under certain conditions holding for a given base matrix $\mathbf{B}$, the parity-check matrix of codes in $\mathcal{E}(\mathbf{B})$ contains unavoidable cycles. Still, this does not demonstrate that at least one of such conditions holds for any arbitrarily chosen base matrix with finite size and regular row/column weight, for any (finite or infinite) value of the code memory. We prove this fact in the following theorem.

Theorem 2 Given a $c \times a$ (d_v, d_c)-regular base matrix $\mathbf{B}$, such that $d_c > d_v > 1$, where d_c and d_v are finite, the parity-check matrix of any time-invariant code in $\mathcal{E}(\mathbf{B})$ contains at least an unavoidable cycle. Therefore, the girth of any time-invariant code in $\mathcal{E}(\mathbf{B})$ is bounded above by a finite value, say g^*, corresponding to the length of this unavoidable cycle, for any value of m_s.

Proof: See Appendix C. ■

In conclusion, we have shown that if the size of the base matrix is finite, even if $m_s \to \infty$, unavoidable cycles of length g^* always exist in any regular time-invariant code in $\mathcal{E}(\mathbf{B})$, upper bounding the value of its girth. In other words, just increasing the memory of a regular time-invariant code, and keeping a fixed block length, is not sufficient to achieve unbounded girth.

Notice that, according to the proof in Appendix C, given two cycles with length λ_1 and λ_2, the exact value of g^* depends on how they are connected. Summarizing, and assuming that $\mathbf{B}$ does not contain entries larger than 1, we have

- $g^* = 2(\lambda_1 + \lambda_2 - \rho)$ if the two cycles share at least ρ symbols 1
- $g^* = 2(\lambda_1 + \lambda_2)$ if the two cycles do not share symbols 1 but have a segment on the same row (or column);
- $g^* = 2(\lambda_1 + \lambda_2 + 1)$ if the two cycles are connected by a path of length 2;
- $g^* = 2(\lambda_1 + \lambda_2 + \mathcal{L} - 1)$ if the two cycles are connected by a path of length $\mathcal{L}$ starting with a horizontal (or vertical) segment and ending with a horizontal (or vertical) segment;
- $g^* = 2(\lambda_1 + \lambda_2 + \mathcal{L})$ if the two cycles are connected by a path with length $\mathcal{L}$ starting with a horizontal (or vertical) segment and ending with a vertical (or horizontal) segment.

The tightest, i.e., smallest, value of g^* depends on the single code cycle properties and on the considered cycles.

The following corollary is a straightforward consequence of Remark 1 and Theorem 2.

Corollary 2 Given a $c \times Ta$ (d_v, d_c)-regular base matrix, $d_c > d_v > 1$, where d_c, d_v and T are finite, and symbolic matrix as in (6), such that $\mathcal{W}(\mathbf{H}_i(D)) = \mathcal{W}(\mathbf{H}_j(D))$, $\forall i \neq j$, then the parity-check matrix of any time-invariant code in $\mathcal{E}(\mathbf{B})$ contains at least one unavoidable cycle. Therefore, the girth of any periodically time-varying code in $\mathcal{E}(\mathbf{B} = \mathcal{W}(\mathbf{H}_0(D)))$ is bounded above by a finite value, say g^*, corresponding to the length of this unavoidable cycle, for any value of m_s.

Proof: The statement can be proved by following the proof of Theorem 2, but considering a $Tc \times Ta$ matrix, according to Remark 1. ■

Thus, also regular periodically time-varying codes with finite period T have limited girth, that cannot be improved by just increasing their memory. In sight of this, asymptotic analysis techniques [43], [44], which rely on the assumption of infinite girth, may not be the most suited tool to predict the performance of the codes we design. Still, as we show next, introducing a small periodicity in the code can yield an improvement of the girth properties. Therefore, prediction of the performance through asymptotic analysis tools would be more appropriate on periodically time-varying codes than on time-invariant codes, as the former are closer to codes with infinite girth than the latter.

We remark that, as shown in [45], block codes can achieve infinite girth for infinite block length. However, by unwrapping a block code, we can obtain a time-varying code with a girth that is lower bounded by that of the block code. In this construction, if the length of the block code increases, both memory and period of the time-varying code increase [46]. Therefore, also a time-varying code can achieve infinite girth if both the period and the memory tend to infinity. We have proved in Theorem 2 and Corollary 4 that assuming a finite value of the period is instead a sufficient condition to have finite girth.

However, we will verify in the next section that, for a given base matrix, the length of the unavoidable cycles affecting time-invariant codes lower bounds that of the unavoidable cycles affecting periodically time-varying codes. This is a straightforward consequence of Lemma 2: time-invariant codes and periodically time-varying codes can be obtained by lifting the same base matrix. In the former case, circulant matrices have to be used, whereas periodically circulant matrices are allowed in the second case. Clearly, periodically circulant matrices grant a larger number of degrees of freedom in the design, as they are characterized by the shifting of the first T initial columns, whereas circulant matrices are obtained by shiftings of their first column only. Exploiting periodicity, we can break short unavoidable cycles caused by the structure of the base matrix. The avoidable cycles are removed with a proper choice of the exponent matrix $\mathbf{P}$. In the next section we consider a specific family of codes and derive theoretical and numerical results which validate the above statements.

V. Codes with Girth larger than 12

In this section we prove that periodically time-varying codes with $T > 1$ may have better girth than the time-invariant ones with the same block length. We indeed consider symbolic matrices with the same size $c \times a$, keeping in mind that periodically time-varying codes with $T > 1$ contain T symbolic matrices, leading to more degrees of freedom in the code design. In particular, we show how unavoidable cycles with length 12 can be avoided through a proper management of the period T. We have chosen this particular length because unavoidable cycles with length 12 are the shortest ones that can occur in the parity-check matrix of codes without parallel edges. So, as a benchmark, we consider fully-connected monomial codes. The reason is threefold: firstly, it is shown in Lemma 9 that entries of $\mathbf{B}$ which are larger than 1 have a negative impact on the girth[5]; secondly, full connection guarantees that the code block length is relatively small and that $\mathbf{H}(D)$ can be represented with the minimum possible number of parameters; thirdly, unavoidable cycles with length 12 derive from

$$\mathbf{B}_{12} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{23}$$

and full-connection puts us in the worst-case scenario, as the base matrix of fully-connected monomial codes is filled with submatrices of the type (23). For all these reasons, if we consider a $c \times a$ base matrix containing some zero entries, the analysis of unavoidable cycles with length 12 is less complex than that of a $c \times a$ base matrix containing only 1's. In particular, the number of submatrices of the type (23) decreases as the number of zero entries in the base matrix increases. In a (c, a)-regular fully-connected monomial code, using an exhaustive approach,

$$\binom{a}{3}\binom{c}{2} + \binom{a}{2}\binom{c}{3}$$

submatrices of the type (23) have to be tested. For example, a single void entry reduces this quantity to

$$\binom{a-1}{2}(c-1) + (a-1)\binom{c-1}{2};$$

so, the search space becomes smaller. The effect of more void entries in the base matrix, which further reduce the search space, depends on their position within the base matrix and it is not analyzed here.

A. Impact of the Period T on the Girth

Let us consider a time-invariant code in $\mathcal{E}(\mathbf{B})$, where $\mathbf{B}$ contains (23). The parity-check matrix of this code exhibits unavoidable cycles of length 12, i.e., its girth is bounded above by 12.

From now on, we consider periodically time-varying codes which are described by the same $\mathbf{B}$ as that of time-invariant codes. In other words, we assume that $\mathcal{W}(\mathbf{H}_i(D)) = \mathcal{W}(\mathbf{H}_j(D))$, $\forall i \neq j$. This assumption is pessimistic, as we are not exploiting all degrees of freedom in the choice of $\mathbf{B}$. Increasing the girth by allowing different base matrices would indeed be an easier problem.

Let us now consider $\mathcal{E}(\mathbf{B}_{12})$. We demonstrate that time-varying codes randomly picked from $\mathcal{E}(\mathbf{B}_{12})$ can achieve girth larger than 12, even for small values of T. This confirms the potential of time-varying SC-LDPC-CCs over the time-invariant ones.

Lemma 11 A time-varying code with period $T = 2$ in $\mathcal{E}(\mathbf{B}_{12})$ has $g \leq 20$, and a solution holding with the equality sign always exists for a proper choice of the parameters.

[5]Entries of the base matrix larger than 1 affect positively the code free distance, as shown in [36]. However, free distance is not the focus of this work.

Proof: According to the Remark 1, a time-varying code with period $T = 2$ can be studied by means of a $2c \times 2a$ symbolic matrix. Referring to (23), we have a 4×6 $(2,3)$-regular base matrix. Any matrix with 4 rows can contain a cycle with at most length 8.

According to the results in [33], if the girth of a base matrix $\mathbf{B}$ is g_b, any time-invariant code in $\mathcal{E}(\mathbf{B})$ can have girth larger than or equal to $3g_b$. Let us show first that the 4×6 $(2,3)$-regular base matrix can have, at most, girth $g_b = 6$. In order to achieve $g_b > 4$, all the columns of the base matrix must be different. In a 4×6 $(2,3)$-regular base matrix there are exactly $\binom{4}{2} = 6$ different columns that can be used; this means that by using them all, all the cycles of length 4 can be avoided. On the other hand, using all the different columns, cycles of length 6 arise. Hence, the girth of any code in $\mathcal{E}(\mathbf{B}_{12})$ can be larger than or equal to 18. It is shown in [33] that the only submatrices leading to unavoidable cycles of length 18 are

$$\mathbf{B}_{18}^1 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{B}_{18}^2 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \tag{24}$$

which contain at least one column (and one row) with weight larger than 2; thus, the 4×6 $(2,3)$-regular matrix cannot contain them. It follows that $g \geq 20$ can be achieved.

In order to prove that $g \leq 20$, let us consider the submatrices that lead to unavoidable cycles of length 20, which are

$$\mathbf{B}_{20}^1 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad \mathbf{B}_{20}^2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{25}$$

Both matrices in (25) contain a cycle of length 8. It follows that in order to have $g > 20$ the 4×6 $(2,3)$-regular matrix should not contain cycles of length 8. On the other hand, $\mathbf{B}$ cannot contain cycles of length larger than 8, as it only has 4 rows; so, if $\mathbf{B}$ does not contain cycles of length 8, it only contains cycles of length 4 and 6. Then, if $\mathbf{B}$ contains only cycles of length 4 and 6, they combine yielding cycles of length smaller than or equal to 18, according to Lemma 8. Thus, we can conclude that $g \leq 20$, which holds with the equality sign if and only if $\mathbf{B}$ contains at least one cycle of length 8. ■

Generalizing (23), we have the following $2 \times a$ base matrix

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \end{bmatrix}. \tag{26}$$

Corollary 3 Any time-varying code with period $T = 2$ in the ensemble described by (26) can achieve

$$g \leq \begin{cases} 14 & \text{if} \quad a = 4, \\ 12 & \text{if} \quad a \geq 4. \end{cases} \tag{27}$$

and a solution holding with the equality sign always exists for a proper choice of the parameters.

Proof: Carried out with similar arguments as those in the proof of Lemma 11. ■

We notice from (27) that, as expected, increasing a, for fixed values of the period $T = 2$, yields a drop of the upper bound on the girth which can be reached with a suitable choice of the parameters. We thus need to increase T, if we aim at breaking the unavoidable cycles of length 12 affecting time-invariant codes in the ensemble described by (26). The following result holds.

Lemma 12 A necessary and sufficient condition to obtain a $(2,a)$-regular time-varying code having period T with girth $g > 12$ from the ensemble described by (26) is $T \geq \lceil \frac{a}{2} \rceil$.

Proof: According to Remark 1, we can represent any $(2,a)$-regular time-varying code with a $2T \times aT$ base matrix $\mathbf{B}$. In order to avoid the unavoidable cycles of length 12, entailed by (23), we need to ensure that $\mathbf{B}$ does not contain the same column more than twice. The number of possible different columns is T^2, which can be repeated up to two times, yielding $2T^2$ possibilities. If the number of possibilities is smaller than the number of columns of the base matrix, the latter cannot be constructed. Thus, it must be $2T^2 \geq aT$, that is $T \geq \frac{a}{2}$. Choosing $T = \frac{a}{2}$ when a is even and $T = \lceil \frac{a}{2} \rceil$ when a is odd ensures that the above condition is verified. ■

The above lemma can be generalized to any value of $c > 2$ as follows.

Corollary 4 A necessary condition to obtain a (c,a)-regular time-varying code with period T, with $c > 2$ and girth $g > 12$ from $\mathcal{E}(\mathbf{B})$, where $\mathbf{B}$ is an all-ones $c \times a$ matrix, is $T \geq \lceil \frac{a}{2} \rceil$.

Proof: Carried out with similar arguments as those in the proof of Lemma 12. ■

B. Degrees of Freedom in the Code Design

Once the value of T which guarantees that girth larger than 12 can be obtained for a proper choice of $\mathbf{P}$ has been found, we need to determine the number of parameters N_p required to actually design a periodically time-varying code free of unavoidable cycles with length up to 12. In other words, we need to determine which entries of the base matrix are to be lifted with circulant permutation matrices (with $M \to \infty$) and which are to be lifted with periodically circulant permutation matrices with period T (and $M \to \infty$).

In general, a periodically time-varying fully-connected monomial code with period T needs at most Tac entries to be defined, since its symbolic matrix (6) contains T sub-matrices with size $c \times a$. This assumes that all the entries of the base matrix are lifted with periodic permutation matrices. Nevertheless, as we show in the following, some entries of the base matrix are lifted with circulant permutation matrices, whereas others are lifted with periodic permutation matrices; so, the cost in terms of representation might actually be smaller than Tac.

For the sake of convenience, we introduce a $c \times a$ matrix $\mathbf{L}$ with binary entries. If $l_{i,j}$ is 0, then $b_{i,j}$ is lifted with a circulant permutation matrix, which can be represented by a single integer (the position of the only 1 in the first column). If, instead, $l_{i,j}$ is 1, then $b_{i,j}$ is lifted with a periodically circulant permutation matrix, which can be represented by T integers. In other words, the matrix $\mathbf{L}$ has the same size as $\mathbf{B}$ and catches which entries are lifted with periodic permutation matrices, requiring T integers each for representation, and which ones are lifted with circulant matrices, requiring 1 integer each for representation. The knowledge of $\mathbf{L}$, or better, of its Hamming weight, and T permits to calculate the number of integers required to design and describe a periodically time-varying SC-LDPC-CC with period T and girth larger than 12. The following considerations hold.

Lemma 13 Periodically time-varying fully-connected monomial SC-LDPC-CCs with period T are characterized by $N_p = \mathcal{W}(\mathbf{L})T + (ac - \mathcal{W}(\mathbf{L}))$, where $\mathcal{W}(\mathbf{L})$ is the Hamming weight of $\mathbf{L}$.

Proof: Follows from the fact that the symbols 1 in $\mathbf{L}$ represent periodically circulant permutation matrices with period T, described by T integers, and the 0's in $\mathbf{L}$ represent circulant permutation matrices, described by a single integer. ∎

Corollary 5 Time-invariant fully-connected monomial SC-LDPC-CCs have $N_p = ac$.

Proof: Follows from the fact that $\mathbf{L}$ contains only zeros in the time-invariant case. ∎

Our goal is obviously that of reducing N_p as much as possible, still keeping the conditions which guarantee a certain girth fulfilled. In other words, we need to minimize $\mathcal{W}(\mathbf{L})$ in order to achieve a certain girth using the smallest possible number of degrees of freedom.

Example 4 Let us consider (23) and any $\mathbf{L}$ such that $\mathcal{W}(\mathbf{L}) = 1$. Then, a periodically time-varying code with period $T = 2$, $N_p = 7$ and $g = 20$ in $\mathcal{E}(\mathbf{B}_{12})$ can be obtained for a proper choice of the entries of $\mathbf{P}$. This means that exploiting at least 7 degrees of freedom instead of the 6 of the time-invariant case, the girth can be increased from 12 to 20. This is a straightforward consequence of Lemma 11.

The following lemmas hold.

Lemma 14 In order to prevent the occurrence of unavoidable cycles with length 12 in codes picked from the ensemble described by (26), it has to be $\mathcal{W}(\mathbf{L}) \geq a - 2$, and at least one matrix $\mathbf{L}$ such that the above inequality is satisfied with the equality sign always exists.

Proof: Clearly, $\mathbf{L}$ can have at most two all-zeros columns, or unavoidable cycles due to (23) would arise. This proves $\mathcal{W}(\mathbf{L}) \geq a - 2$. The proof of the second part of the theorem is conducted by evidence: let us consider

$$\mathbf{L} = \left[\begin{array}{cc|ccccc} 0 & 0 & 1 & 0 & 1 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & 1 & \dots \end{array}\right];$$

then, this matrix excludes the presence of cycles with length 12, for a proper choice of $\mathbf{P}$. ∎

Lemma 15 Let us consider a time-varying code with period T picked from the ensemble described by (26). In order to achieve $g > 12$, it has to be

$$N_p \geq (a-2)\left\lceil \frac{a}{2} \right\rceil + a + 2, \tag{28}$$

and at least one matrix $\mathbf{L}$ such that (28) holds with the equality sign always exists.

Proof: Equation (28) is obtained considering that, according to Lemma 12, $T = \lceil \frac{a}{2} \rceil$ is required, at least, to have $g > 12$. The rest follows from Lemmas 13 and 14. ∎

Let us now consider the following $3 \times a$ base matrix

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \end{bmatrix}. \tag{29}$$

The following lemmas hold.

Lemma 16 In order to prevent the occurrence of unavoidable cycles with length 12 in periodically time-varying codes with period T, randomly picked from the ensemble described by (29), it has to be

$$\mathcal{W}(\mathbf{L}) \geq \begin{cases} 3 & \text{if} \quad a = 4, \\ W_{\mathbf{L},3,a} & \text{otherwise}, \end{cases}$$

where $W_{\mathbf{L},3,a} \triangleq \min\{a, 6\} + 2\max\{0, a-6\}$. Furthermore, a solution with the equality sign can always be found for a proper choice of $\mathbf{L}$ (the proper choice is discussed in the proof).

Proof: Let us consider the possible columns of $\mathbf{L}$. Keeping in mind that we aim at minimizing its weight, we have two possible cases:

1) if the all-zeros column appears in $\mathbf{L}$, then each one of the other three columns with weight 1 can appear at most once, and each other column with weight 2 can appear as many times we need. The resulting minimum weight of $\mathbf{L}$ is therefore $W_1 \triangleq 3 + 2\max\{0, a-4\}$
2) if the all-zeros column does not appear in $\mathbf{L}$, then each one of the other three columns with weight 1 can appear at most six times (twice each), and each other column with weight 2 can appear as many times we need. The resulting minimum weight of $\mathbf{L}$ is therefore $W_2 \triangleq \min\{a, 6\} + 2\max\{0, a-6\}$.

Comparing W_1 and W_2, it is easy to observe that W_1 is smaller than W_2 only if $a = 4$. ∎

Lemma 17 Let us consider a time-varying code with period T, randomly picked from the ensemble described by (29). In order to achieve girth larger than 12, it has to be

$$N_p \geq \begin{cases} 12 + 3(T-1) & \text{if} \quad a = 4, \\ TW_{\mathbf{L},3,a} + 3a - W_{\mathbf{L},3,a} & \text{otherwise}, \end{cases} \tag{30}$$

and it is possible to find at least one matrix $\mathbf{L}$ such that a solution with the equality sign exists.

Proof: Easily follows from Lemmas 13 and 16. ∎

Let us now consider the following $4 \times a$ base matrix

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \end{bmatrix}. \tag{31}$$

The following lemmas hold.

Lemma 18 In order to prevent the occurrence of unavoidable cycles with length 12 in codes picked from the ensemble described by (31), a necessary condition that has to be fulfilled is

$$\mathcal{W}(\mathbf{L}) \geq \begin{cases} W_{\mathbf{L},4,a,0} & \text{if} \quad 5 \leq a \leq 7, \\ W_{\mathbf{L},4,a,1} & \text{if} \quad 8 \leq a \leq 9, \\ W_{\mathbf{L},4,a,2} & \text{if} \quad a = 10, \\ W_{\mathbf{L},4,a,3} & \text{if} \quad a \geq 11. \end{cases}$$

where

$$W_{\mathbf{L},4,a,0} \triangleq 3 + 2\min\{a-3, 3\} + 3\max\{0, a-6\},$$

$$W_{\mathbf{L},4,a,1} \triangleq 14 + 3\max\{0, a-8\},$$

$$W_{\mathbf{L},4,a,2} \triangleq 19$$

$$W_{\mathbf{L},4,a,3} \triangleq 2\min\{a, 12\} + 3(a-12)$$

Furthermore, it is possible to find at least one matrix $\mathbf{L}$ such that the above inequality is satisfied with the equality sign.

Proof: See Appendix D. ∎

Lemma 19 Let us consider a time-varying code with period T and girth larger than 12 randomly picked from the ensemble described by (31). It is possible to find a matrix $\mathbf{L}$ such that this code can be represented with

$$N_p = \begin{cases} TW_{\mathbf{L},4,a,0} + 4a - W_{\mathbf{L},4,a,0} & \text{if} \quad 5 \leq a \leq 7, \\ TW_{\mathbf{L},4,a,1} + 4a - W_{\mathbf{L},4,a,1} & \text{if} \quad 8 \leq a \leq 9, \\ TW_{\mathbf{L},4,a,2} + 4a - W_{\mathbf{L},4,a,2} & \text{if} \quad a = 10, \\ TW_{\mathbf{L},4,a,3} + 4a - W_{\mathbf{L},4,a,3} & \text{if} \quad a \geq 11 \end{cases} \tag{32}$$

integers and solutions with smaller values of N_p do not exist.

Proof: Easily follows from Lemmas 13 and 18. ∎

For the general case of $c > 4$, we derive some *sufficient* conditions on N_p.

Lemma 20 Let us consider an all-ones $c \times a$ matrix, $c < a$. If the following condition is satisfied

$$\mathcal{W}(\mathbf{L}) \geq W_{\mathbf{L},c,a}$$

where

$$W_{\mathbf{L},c,a} \triangleq (c-2)\min\{a, c(c-1)\} + (c-1)\max\{0, a-c(c-1)\},$$

then it is possible to find at least one $\mathbf{L}$ such that the occurrence of unavoidable cycles with length 12 is prevented.

Proof: Let us divide $\mathbf{L}$ in two parts as follows

$$\mathbf{L} = \left[\mathbf{L}_1 | \mathbf{L}_2\right].$$

There are $\binom{c}{2}$ vectors with length c and weight $c-2$ that can be chosen as different columns of $\mathbf{L}$. Each of these columns can be chosen twice, without causing the occurrence of unavoidable cycles with length 12. We fill $\mathbf{L}_1$ with these columns, which are, at most, $c(c-1)$. Each of them obviously contributes with $c-2$ symbols 1 to the total weight of $\mathbf{L}$. $\mathbf{L}_2$ is filled with columns chosen as random vectors with length c and weight $c-1$, which cannot cause unavoidable cycles with length 12. Combining these considerations the lemma is eventually proved. ∎

Theorem 3 A periodically time-varying code with period T and girth larger than 12 can be designed using

$$N_p = TW_{\mathbf{L},c,a} + ac - W_{\mathbf{L},c,a}$$

degrees of freedom, where $W_{\mathbf{L},c,a}$ has been defined in Lemma 20, for a proper choice of $\mathbf{L}$.

Proof: Easily follows from Lemmas 13 and 20. ∎

Keeping this in mind, in the following section we investigate the design of SC-LDPC-CCs.

VI. Design Methods

In this section we first show how to design periodically time-varying codes with small period T and same block length as the time-invariant ones, but larger girth. We also show that, in order to achieve much larger girths, a larger memory is required.

As claimed in Lemma 11, codes with girth $g = 20$ in $\mathcal{E}(\mathbf{B}_{12})$ can be obtained considering a period $T = 2$. The following symbolic matrix, describing a time-varying SC-LDPC-CC with period $T = 2$ and girth $g = 20$, has been found by means of a random search

$$\mathbf{H}(D) = \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & D & D & D \\ 1 & D & D^6 & D & D^2 & D^{17} \end{array}\right]. \tag{33}$$

Notice that, according to Lemma 15, at least 7 integers are needed to design a code with this girth. In our case, we have used exactly 7 integers: 3 integers are sufficient to describe the entries at position $(0, j_0)$, $j_0 \in \{0, 1, \dots, 5\}$, 2 integers are sufficient for the entries at position $(1, j_1)$, $j_1 \in \{0, 1, 3, 4\}$ and 2 integers are needed for the entries at positions $(1, 2)$ and $(1, 5)$. The corresponding $\mathbf{L}$ is indeed

$$\mathbf{L} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In other words, the girth can be improved from 12 (in the time-invariant case) to 20 (in the time-varying case) by considering 7 integers instead of 6 to design the symbolic matrix, when $\mathcal{E}(\mathbf{B}_{12})$ is considered.

Notice that solutions with smaller memory than (33), which is $m_s = 16$, can be found. The design procedure we use works on a per-column basis and is summarized in Algorithm 1. In words, $\mathbf{P}$ is initialized as an empty matrix; then, as a test, a column is appended to $\mathbf{P}$ and the girth of the corresponding code is evaluated. If the girth is not smaller than the target girth, then the new column is fixed and the procedure is repeated until $\mathbf{P}$ has aT columns, or all the possible columns have been tested without success. In the latter case, m_s is increased by one and the procedure is restarted.

Algorithm 1

Input block length a, number of parity symbols per period c, memory order m_s, period T, girth g

procedure EXPONENT_MATRIX_SEARCH(a, c, m_s, T, g)
  $\mathbf{P} \leftarrow [\ \ ]$
  while $\mathbf{P}$ has less than $aT+1$ columns **do**
    for $i \leftarrow 0$ **to** T **do**
      $\mathcal{S}^{(i)} \leftarrow$ All possible vectors with length c and entries in $[i, i+1, \ldots, i+m_s]$ containing the value i at least once.
      $x \leftarrow 1$
      while $x \leq aT$ **do**
        $i \leftarrow \lfloor \frac{x}{a} \rfloor$
        for $j_i \leftarrow 0$ **to** $(m_s+1)^c - m_s^c$ **do**
          $\mathbf{P}_{\text{test}} \leftarrow [\mathbf{P} \quad \mathcal{S}^{(i)}_{j_i}]$
          $g^* =$Girth_Evaluation($\mathbf{P}$)
          if $g^* \geq g$ **then**
            $\mathbf{P} \leftarrow [\mathbf{P} \quad \mathcal{S}^{(i)}_{j_i}]$
            if $x = aT$ **then**
              return $\mathbf{P}$
            else
              $x \leftarrow x+1$
        $x \leftarrow x-1$
    $m_s \leftarrow m_s+1$

We remark that $\mathbf{H}_i(D)$, $0 \leq i \leq T-1$, in general, contains entries with exponents in $[i, m_s+i]$ but, as shown in [47] for QC-LDPC codes and later generalized in [23] for SC-LDPC-CCs, it is always possible to obtain $\mathbf{H}_i(D)$ such that all its columns contain at least once x^i. This considerably reduces the search space. The algorithm we use in order to determine the girth of a given code is inspired by the cycle-counting algorithm proposed in [48, Algorithm 1]. Nevertheless, assessing the *multiplicity* of cycles goes beyond the scope of this work; so, we just focus on the *existence* of cycles with a given length.

By applying Algorithm 1, which is exhaustive on a reduced search space, we have found the exponent matrix with the smallest possible m_s, which is

$$\mathbf{P} = \left[\begin{array}{ccc|ccc} 0 & 0 & 4 & 1 & 1 & 7 \\ 0 & 3 & 0 & 6 & 7 & 1 \end{array}\right], \tag{34}$$

corresponding to a periodically time-varying code with period 2 and $m_s = 6$. Nevertheless, being

$$\mathbf{L} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix},$$

we have used 9 integers for the code design, which is larger than the lower bound given by (28).

We generalize the above arguments also considering $d_v = c = 3$. We have designed a $(3,4)$-regular time-varying code with period $T=2$ and girth $g=14$ by means of a guess-and-test algorithm, inspired by that proposed in [26]. Its exponent matrix is as follows

$$\mathbf{P} = \left[\begin{array}{cccc|cccc} 16 & 112 & 79 & 0 & 262 & 69 & 213 & 1 \\ 142 & 25 & 0 & 0 & 133 & 26 & 1 & 183 \\ 0 & 92 & 160 & 133 & 1 & 1 & 58 & 225 \end{array}\right]. \tag{35}$$

The corresponding $\mathbf{L}$ is as follows

$$\mathbf{L} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

If we increase a, obtaining a $(3,5)$-regular code, we need $T \geq 3$ to ensure that the potential girth is larger than 12. We have found a solution with the equality sign, that is, such that $T=3$: the exponent matrix of a periodically time-varying code with $g=14$, $a=5$, $c=T=3$ is (36).

There also exists a solution for $(3,6)$-regular time-varying codes with period $T=3$ to achieve $g>12$. The exponent matrix, found by means of the same guess-and-test algorithm as above, defining a time-varying code with $T=3$ and $g=14$ is (37). Notice that, referring to all the above examples, the bounds given in Lemma 11 and Corollary 4 are satisfied with the equality sign, showing that they are tight, at least for small values of c.

VII. Error rate performance

In this section we assess the performance of the newly designed codes described in Section VI. For this purpose, we consider some code examples and estimate their performance in terms of bit error rate (BER) under iterative decoding through Monte Carlo simulations of binary phase shift keying (BPSK) modulated transmission over the additive white Gaussian noise (AWGN) channel. We have made the parity-check matrices of the codes we consider publicly available in [49]. We consider a BP-based decoder performing 150 iterations. For practical reasons, the codes are terminated after 24 000 time instants. Owing to termination and to the possible presence of some linearly dependent rows in the parity-check matrix of the considered codes, the asymptotic code rate for SC-LDPC-CCs and the design rate for QC-LDPC block codes might not coincide with the actual code rate. However, the effect of such a deviation from the ideal rate on the fairness of our comparisons is negligible.

Let us consider a code according to (35), i.e., a $(3,4)$-regular time-varying code with period $T=2$, $a=4$, $c=3$, $m_s = 261$, $R_\infty = \frac{1}{4}$ and girth $g=14$, and compare its performance to those of codes with the same parameters but smaller girths, i.e., $g=6$ and $g=12$. These codes are obtained by feeding Algorithm 1 with the parameters listed above. Their exponent matrices are

$$\mathbf{P}_6 = \left[\begin{array}{cccc|cccc} 261 & 96 & 50 & 226 & 92 & 183 & 119 & 73 \\ 210 & 247 & 182 & 102 & 157 & 66 & 89 & 219 \\ 182 & 164 & 139 & 194 & 13 & 197 & 182 & 237 \end{array}\right]$$

$$\mathbf{P} = \left[\begin{array}{ccccc|ccccc|ccccc} 2032 & 108 & 1575 & 375 & 0 & 1 & 1283 & 1 & 1 & 2272 & 708 & 1163 & 2 & 771 & 648 \\ 0 & 2180 & 0 & 0 & 226 & 1859 & 546 & 1768 & 1171 & 1179 & 2 & 2 & 1890 & 1594 & 2 \\ 1737 & 0 & 1358 & 801 & 187 & 794 & 1 & 1032 & 1556 & 1 & 1380 & 2300 & 2457 & 2 & 454 \end{array}\right] \quad (36)$$

$$\mathbf{P} = \left[\begin{array}{cccccc|cccccc|cccccc} 0 & 300 & 81 & 36 & 0 & 195 & 1 & 157 & 302 & 2 & 1 & 305 & 2 & 4 & 1447 & 2029 & 2 & 2 \\ 0 & 120 & 49 & 64 & 209 & 5 & 3 & 150 & 105 & 300 & 230 & 304 & 1109 & 1204 & 2032 & 2 & 2923 & 2921 \\ 51 & 1 & 0 & 2 & 103 & 2 & 282 & 2 & 1 & 77 & 3 & 3 & 534 & 1450 & 2 & 2923 & 1609 & 3485 \end{array}\right] \quad (37)$$

and

$$\mathbf{P}_{12} = \left[\begin{array}{cccc|cccc} 71 & 148 & 26 & 244 & 260 & 249 & 9 & 17 \\ 32 & 251 & 42 & 194 & 209 & 36 & 93 & 239 \\ 224 & 110 & 151 & 226 & 49 & 81 & 124 & 263 \end{array}\right],$$

respectively.

Simulation results are shown in Fig. 2. We notice that, as expected, the performance improves as the girth increases. In addition, we consider two time-invariant codes with $T = 1$, $a = 4$, $c = 3$, $m_s = 261$, $R_\infty = \frac{1}{4}$ and $g = 12$. In order to distinguish between them, they are addressed to as Code 1 and Code 2. Their exponent matrices are

$$\mathbf{P}_{C1} = \begin{bmatrix} 139 & 0 & 122 & 263 \\ 0 & 245 & 0 & 65 \\ 119 & 211 & 31 & 0 \end{bmatrix}$$

and

$$\mathbf{P}_{C2} = \begin{bmatrix} 266 & 68 & 0 & 186 \\ 217 & 174 & 0 & 163 \\ 0 & 0 & 91 & 0 \end{bmatrix},$$

respectively. Finally, we consider the QC-LDPC block code designed in [50] with girth $g = 14$ and design rate $R = \frac{1}{4}$, based on the Steiner Triple System. Notice that the blocklength of this code, which is 34 260, is much larger than the constraint length of the time-varying SC-LDPC code with the same girth, which is 1 052. Despite this, the latter code maintains a significant gain over the QC-LDPC block code.

Time-varying codes also show a significant improvement over the time-invariant ones with the same parameters. A justification of such a large gap is provided by the analysis of the error patterns yielding decoding failures for these codes. In particular, many decoding failures of the time-invariant codes with girth 12 are caused by codewords with weight 24; this was also observed in [27]. It is shown in [36] that the free distance of $(3, 4)$-regular time-invariant fully-connected monomial SC-LDPC codes is indeed upper bounded by 24. Instead, time-varying codes are characterized by larger symbolic matrices, yielding larger upper bounds on the code free distance. For example, the error patterns of (35) do not contain any codeword with weight smaller than 106. So, even though we are aware that a careful analysis of the dominant trapping sets is required, it can be immediately noticed that a very large difference between the structure and the free distance of the codes has a major impact also on the code performance. A thorough analysis of the free distance of periodically time-varying codes with $T > 1$ is out of the scope of this paper, and is left for future works.

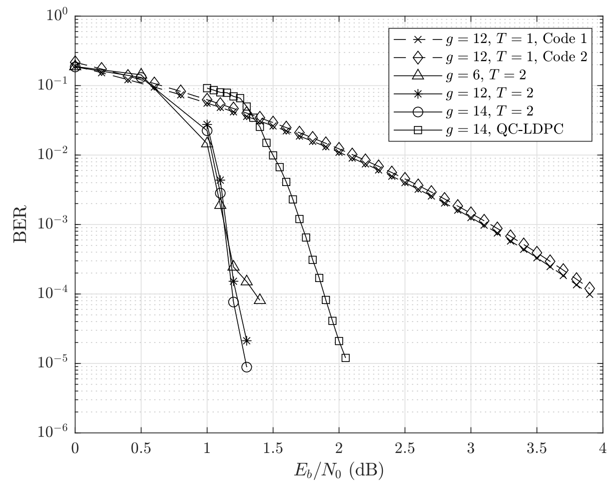

Fig. 2. Comparison of the BER resulting from Monte Carlo simulations of QC-LDPC codes, time-invariant codes and time-varying codes with $T = 2$ for different girth values. In all cases $a = 4$, $c = 3$, $m_s = 261$ and $R = \frac{1}{4}$.

Aiming at evaluating the trade-off between the description complexity (in terms of integers required to represent the code, i.e., N_p) and the error rate performance, in Table II we show the value of $\frac{E_b}{N_0}$ at which $\mathrm{BER} = 10^{-4}$, for all the considered codes, and we denote it as $\left(\frac{E_b}{N_0}\right)^*$. Notice that the periodically time-varying codes with $T = 2$ and block length $a = 4$ are represented by 24 integers instead of the 12 integers required for the time-invariant codes with block length $a = 4$; however, we remark that a time-invariant code with block length $Ta = 8$ and the same code rate would require 48 integers for representation.

TABLE II
$\frac{E_b}{N_0}$ AT WHICH $\mathrm{BER} = 10^{-4}$, FOR CODES WITH DIFFERENT PERIOD AND GIRTH.

T	g	$\left(\frac{E_b}{N_0}\right)^*$ (dB)
0	14	1.84
1	12	3.9 - Code 1
1	12	3.91 - Code 2
2	6	1.37
2	12	1.22
2	14	1.19

Table II allows us to remark that there is a significant gap, of

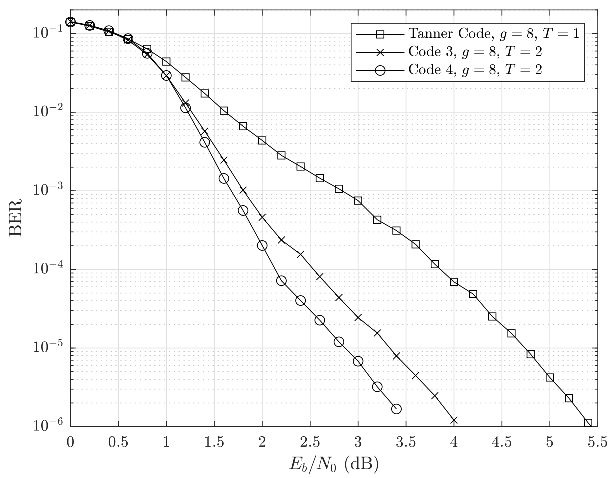

Fig. 3. Comparison of the BER resulting from Monte Carlo simulations of time-invariant codes and time-varying codes with $T = 2$. In all cases $a = 5$, $c = 3$, $g = 8$, $m_s = 28$ and $R = \frac{2}{5}$.

about 2.7 dBs when $\mathrm{BER} = 10^{-4}$ between the time invariant codes, which can be represented with 12 integers, and the periodically time-varying ones, which can be represented with 24 integers.

In order to provide a further example, let us consider a time-invariant code with girth $g = 8$, described by the following exponent matrix from [11]

$$\mathbf{P} = \begin{bmatrix} 1 & 2 & 4 & 8 & 16 \\ 5 & 10 & 20 & 9 & 18 \\ 25 & 19 & 7 & 14 & 28 \end{bmatrix}, \tag{38}$$

which corresponds to an SC-LDPC code with $R_\infty = 2/5$ and girth 8. Its simulated performance is reported in Fig. 3. For carrying out a fair comparison, we have designed two time-varying codes with $T = 2$, the same memory and girth, but a smaller multiplicity of cycles with length 8. We have named these codes Code 3 and Code 4, which are described by the following exponent matrices

$$\mathbf{P}_{C3} = \left[\begin{array}{ccccc|ccccc} 0 & 0 & 1 & 20 & 0 & 1 & 1 & 2 & 21 & 1 \\ 25 & 13 & 0 & 1 & 11 & 26 & 14 & 1 & 2 & 12 \\ 21 & 14 & 28 & 0 & 3 & 22 & 15 & 19 & 1 & 4 \end{array}\right], \tag{39}$$

$$\mathbf{P}_{C4} = \left[\begin{array}{ccccc|ccccc} 20 & 0 & 27 & 0 & 13 & 1 & 28 & 19 & 12 & 19 \\ 21 & 20 & 17 & 15 & 28 & 21 & 18 & 1 & 24 & 3 \\ 0 & 28 & 0 & 16 & 0 & 29 & 1 & 26 & 1 & 1 \end{array}\right]. \tag{40}$$

The Tanner time-invariant SC-LDPC code has 3 cycles of length 8 per node, whereas Code 3 and Code 4 have 1.2 and 0.8 cycles of length 8 per node, respectively. This, along with the potentially larger minimum distance of time-varying codes, justifies the better performance observed for Code 4 with respect to the other codes.

VIII. Conclusion and future work

We have shown that periodically time-varying SC-LDPC codes with small period allow achieving significant improvements with respect to the corresponding time-invariant counterparts in terms of girth, at the expense of an increase in the number of degrees of freedom that is at most linear with the product of the code period and the size of the considered base matrix. Actually, for practical parameter choices, this upper bound is shown to be loose, thus yielding limited design complexity. In fact, we have shown that just increasing the memory of time-invariant codes is not sufficient to improve their girth beyond a fixed upper threshold. In principle, this is only possible by increasing both the period and the memory. For this reason, we have focused our attention on time-varying SC-LDPC-CCs, for which we have proposed a new and efficient representation that allows an easy design of codes with the same block length as the time-invariant ones, but with larger girth. Clearly, in order to achieve a larger girth, a larger memory is required as well.

As a challenge for future works, we believe that it is worth investigating the effect of irregularity on our analysis. On the one hand, from a theoretical standpoint, we believe that statements such as Theorem 2 and Corollary 3 also hold if the code is irregular, as long as its parity-check matrix does not contain either rows or columns with unitary weight. On the other hand, from a numerical standpoint, using masking techniques [51], [52] could lead to irregular codes with girths even larger than those achieved by introducing a small periodicity in the code design, at the cost of a more complex theoretical analysis and representation.

We believe that further insights may be provided by an investigation of the free distance of periodically time-varying SC-LDPC codes with $T > 1$, which is potentially larger than that of the time-invariant SC-LDPC codes characterized by the same code rate and values of a, c, m_s.

Appendix A
Proof of Theorem 1

We take advantage of the equivalence between the circulants block form and the blocks circulant form of the parity-check matrix of QC-LDPC codes. A parity-check matrix as in (13) is considered.

We consider cycles in the parity-check matrix as defined in Section II-D. We need to distinguish between two types of cycles:

1) those without any symbol 1 in the tail[6] of (13) (named type-1 cycles in the following),
2) those with at least one symbol 1 in the tail of (13) (named type-2 cycles in the following).

A cycle, with length λ, of type-1 can span at most

$$N_p^{(1)} = \left\lfloor \frac{\lambda}{4} \right\rfloor m_s + 1 \tag{41}$$

periods. This can be shown by considering that a type-1 cycle with length λ has $\frac{\lambda}{2}$ horizontal segments and $\frac{\lambda}{2}$ vertical

[6]The tail of the parity-check matrix of a tail-biting code is the portion above its main block diagonal.

segments and any horizontal segment can span at most m_s+1 periods. The cycle spanning the largest number of periods has horizontal segments spanning m_s+1 periods each. Starting from any of the leftmost symbols 1, we notice that, in this scenario, $\frac{\lambda}{2}$ segments are required to reach any of the rightmost symbols 1, and as many are needed to go back to the untouched leftmost symbol 1. Thus,

$$N_p^{(1)} = \left\lfloor \frac{\lambda/2}{2} \right\rfloor (m_s+1) - \left(\left\lfloor \frac{\lambda/2}{2} \right\rfloor - 1 \right) = \left\lfloor \frac{\lambda}{4} \right\rfloor m_s + 1.$$

Type-2 cycles, in their turn, can be separated into

- cycles with an even number of symbols 1 in the tail of (13),
- cycles with an odd number of symbols 1 in the tail of (13).

Any type-2 cycle with an even number of symbols 1 in the tail of the parity-check matrix can be mapped into a type-1 cycle, and fulfills (41). On the other hand, any type-2 cycle with an odd number of symbols 1 in the tail of the parity-check matrix can span at most

$$N_p^{(2)} = \frac{\lambda-2}{2} m_s + 1 \tag{42}$$

periods. In order to prove (42) we firstly need to consider that, for any finite M, horizontal segments of type-2 cycles can span up to M periods in $\mathbf{H}$. This eventually means that, given M, the shortest cycle spanning all the available M periods has one single symbol 1 in the tail of the matrix. Given this, any type-2 cycle with length λ has one horizontal segment and one vertical segment meeting in the tail, and $\frac{\lambda-2}{2}$ segments in the diagonal band of $\mathbf{H}$. It easily follows that

$$N_p^{(2)} = \frac{\lambda-2}{2}(m_s+1) - \frac{\lambda-2}{2} + 1 = \frac{\lambda-2}{2} m_s + 1.$$

Comparing $N_p^{(1)}$ with $N_p^{(2)}$, we notice that

$$\left\lfloor \frac{\lambda}{4} \right\rfloor m_s + 1 \leq \frac{\lambda}{4} m_s + 1 \leq \frac{\lambda-2}{2} m_s + 1$$

for $\lambda \geq 4$, which is always true by definition. We can conclude that $N_p^{(2)}$ defines the largest number of periods that a cycle with length λ can span in the parity-check matrix of a QC-LDPC code.

Notice that $g_{\text{QC}} = g_{\text{free}}$ if and only if no cycle with length $\lambda < g_{\text{free}}$ exists in the parity-check matrix of the QC-LDPC code. So, if a cycle of type-1 does not exist in the $\mathbf{H}$ of the convolutional code, it cannot exist in the $\mathbf{H}$ of the QC-LDPC code. On the other hand, if we choose M in such a way that the longest cycle with length $\lambda = g_{\text{free}} - 2$ is too short to have at least one symbol 1 in the tail of $\mathbf{H}$, the free girth is guaranteed. Thus, it must be

$$M > \frac{g_{\text{free}} - 2 - 2}{2} m_s + 1 + m_s - 1 = \frac{g_{\text{free}} - 2}{2} m_s.$$

Appendix B

Appendix B - Proof of Lemma 9

We consider a generic code in the ensemble $\mathcal{E}(\mathbf{B})$ for each of the above cases.

1) If $\mathbf{B}$ contains a 3, then $\mathbf{H}(D)$ contains the polynomial $h_{i,j}(D) = D^{k_1} + D^{k_2} + D^{k_3}$, for some i and j, which, according to (11), yields an unavoidable cycle with length 6, due to the following equation
$$(k_1 - k_2) + (k_2 - k_3) + (k_3 - k_1) = 0.$$
2) Let us consider the case in which a row of $\mathbf{B}$ contains one pair of symbols 2. The same proof also holds if such pair exists in the same column. The symbolic matrix contains $h_{i,j_1}(D) = D^{k_1} + D^{k_2}$ and $h_{i,j_2}(D) = D^{m_1} + D^{m_2}$, for some i, j_1, j_2. Then, the following equation
$$(k_1 - k_2) + (m_1 - m_2) + (k_2 - k_1) + (m_2 - m_1) = 0 \tag{43}$$
leads to an unavoidable cycle with length 8. If the two symbols 2 are neither in the same row nor in the same column, then there must exist a path which connects them, which starts and ends with a horizontal segment, as the matrix has finite size and $d_c > 2$. Then, the equation describing an unavoidable cycle contains 8 terms due to the symbols 2, as in (43), and $4\mathcal{L}$ additional terms due to the $\mathcal{L}$ symbols connecting them.
3) The symbol 2 either is involved in at least one cycle or belongs to a path connected to a cycle, since the matrix has finite size. In the first case, we have the following configuration in $\mathbf{B}$ (after a reordering of the row and column indices, if needed)
$$\begin{bmatrix} 2 & \cdots \\ 1 & \ddots \\ \vdots & \end{bmatrix},$$
which corresponds to
$$\begin{bmatrix} D^{k_1} + D^{k_2} & \cdots \\ D^{m_1} & \ddots \\ \vdots & \end{bmatrix}$$
in $\mathbf{H}(D)$. This leads to an unavoidable cycle described by the equation
$$(k_1 - k_2) + (k_2 - m_1) + (m_1 - k_1) + \ldots = 0,$$
where the omitted portion of the equation contains the terms relative to the remaining $\lambda - 2$ symbols in the shortest cycle in which the symbol 2 is involved, covered in both possible ways, so that they cancel each other. There are thus $6 + 2(\lambda - 2) = 2\lambda + 2$ terms in the equation.
If, on the other hand, the symbol 2 does not belong to any cycle, then there exists a path with length $\mathcal{L}$ starting from it, which connects the symbol 2 to a cycle, say with length λ. As $d_c > 2$, we can consider the path to begin with a horizontal segment. If the path ends with a horizontal segment, then the equation of type (11)

leading to an unavoidable cycle contains 4 terms due the symbol 2, 2λ terms due to the cycle and $4(\mathcal{L}-1)$ terms due to the path. So, the unavoidable cycle has length $2(\lambda+2\mathcal{L})$. Similarly, if the path ends with a vertical segment, the equation of type (11) yielding an unavoidable cycle contains 4 terms due to the symbol 2, $4(\mathcal{L}-2)$ terms due to the first $\mathcal{L}-2$ symbols in the path, and $2(\lambda+2)$ terms due to the cycle and the last symbol in the path. So, also in this case, the unavoidable cycle has length $2(\lambda+2\mathcal{L})$.

Appendix C
Proof of Theorem 2

According to Lemma 9, if the base matrix contains entries larger than 1, then the girth of any code in $\mathcal{E}(\mathbf{B})$ is bounded above by $g^* \in \{6,8,10,12\}$, independently of the value of m_s, and the theorem is proved. If the entries of the $\mathbf{B}$ are binary, according to Lemma 7, $\mathbf{B}$ contains at least two cycles, say with length λ_1 and λ_2. There are two possibilities:

- The two cycles share at least ρ symbols 1, with $\rho \geq 1$.
- The two cycles do not share any symbol 1.

In case the cycles share symbols 1, according to Lemma 8, the girth of the codes in $\mathcal{E}(\mathbf{B})$ is bounded above by $g^* = 2(\lambda_1+\lambda_2-\rho)$, independently of the value of m_s, and the theorem is proved. In case the cycles do not share symbols 1, next we analyze all the possible cases for positioning of the two cycles, which do not have any symbol 1 in common and do not form any other cycle.

1) The two cycles have a horizontal (or vertical) segment in the same row (or column).
Then, considering the $(\frac{\lambda_1}{2}+\frac{\lambda_2}{2}-1)\times(\frac{\lambda_1}{2}+\frac{\lambda_2}{2})$ (or vice versa) submatrix which contains the two cycles, we can easily notice that it has one row (or column) with weight 4, and no rows/columns with weight 1. So, according to Lemma 10, the girth of any code in $\mathcal{E}(\mathbf{B})$ is bounded above by $g^* = 2(\lambda_1+\lambda_2)$, regardless of m_s.
2) The two cycles are connected by a path with length 2, formed by a horizontal segment and a vertical segment. Then, considering the $(\frac{\lambda_1}{2}+\frac{\lambda_2}{2})\times(\frac{\lambda_1}{2}+\frac{\lambda_2}{2})$ submatrix which contains the two cycles and the remaining symbol 1 in the path, we can easily notice that it has one row and one column with weight 3, and no rows/columns with weight 1. So, according to Lemma 10, the girth of any code in $\mathcal{E}(\mathbf{B})$ is bounded above by $g^* = 2(\lambda_1+\lambda_2+1)$, for any m_s.
3) The two cycles are connected by a path with length $\mathcal{L}$ starting with a horizontal (or vertical) segment and ending with a horizontal (or vertical) segment. Clearly, $\mathcal{L}$ is odd and the path involves $\mathcal{L}+1$ symbols: each of the two cycles contain one of these symbols and the remaining $\mathcal{L}-1$ symbols in the path, instead, do not belong to the two considered cycles. Then, considering the $(\frac{\lambda_1}{2}+\frac{\lambda_2}{2}+\frac{\mathcal{L}-1}{2})\times(\frac{\lambda_1}{2}+\frac{\lambda_2}{2}+\frac{\mathcal{L}-3}{2})$ (or vice versa) submatrix which contains the two cycles and the path, we can easily notice that it has two rows (two columns) with weight 3, and no rows/columns with weight 1. So, according to Lemma 10, the girth of any code in $\mathcal{E}(\mathbf{B})$ is bounded above by $g^* = 2(\lambda_1+\lambda_2+\mathcal{L}-1)$, for any m_s.
4) The two cycles are connected by a path with length $\mathcal{L}$ starting with a horizontal (or vertical) segment and ending with a vertical (or horizontal) segment. Clearly, $\mathcal{L}$ is even and the path involves $\mathcal{L}+1$ symbols in total: as above, two of them belong to the considered cycles and $\mathcal{L}-1$ of them do not. Then, considering the $(\frac{\lambda_1}{2}+\frac{\lambda_2}{2}+\frac{\mathcal{L}-1}{2})\times(\frac{\lambda_1}{2}+\frac{\lambda_2}{2}+\frac{\mathcal{L}-1}{2})$ submatrix which contains the two cycles and the path, we notice that it has one row and one column with weight 3, and no rows/columns with weight 1. So, according to Lemma 10, the girth of any code in $\mathcal{E}(\mathbf{B})$ is bounded above by $g^* = 2(\lambda_1+\lambda_2+\mathcal{L})$, for any value of m_s.

It can be easily shown that these are the only possibilities for the position of the two cycles, without creating other cycles.

Appendix D
Proof of Lemma 18

Let us consider all the different columns that $\mathbf{L}$ can contain. The following considerations hold:

1) If the all-zeros column appears in $\mathbf{L}$, it is possible to use at most 6 columns with weight 2 and as many columns with weight 3 we wish, in order to design $\mathbf{L}$. The minimum weight of the resulting $\mathbf{L}$ is $W_3 \triangleq 2\min\{6, a-1\}+3\max\{0, a-7\}$.
2) If the all-zeros column does not appear in $\mathbf{L}$ and all the four columns with weight 1 appear in $\mathbf{L}$, we can fill the remaining part of $\mathbf{L}$ with as many columns with weight 3 as necessary. The minimum weight of the resulting $\mathbf{L}$ is $W_4 \triangleq 4+3(a-4)$.
3) If the all-zeros column is not used and 3 columns with weight 1 are used, we can fill the remaining part of $\mathbf{L}$ with 3 columns with weight 2 and as many columns with weight 3 as necessary. The minimum weight of the resulting $\mathbf{L}$ is $W_5 \triangleq 3+2\min\{a-3, 3\}+3\max\{0, a-6\}$.
4) If the all-zeros column is not used and 2 columns with weight 1 are used, we can fill the remaining part of $\mathbf{L}$ with 6 columns with weight 2 and as many columns with weight 3 as necessary. The minimum weight of the resulting $\mathbf{L}$ is $W_6 \triangleq 2+2\min\{a-2, 6\}+3\max\{0, a-8\}$.
5) If the all-zeros column is not used and 1 column with weight 1 is used, we can fill the remaining part of $\mathbf{L}$ with 9 columns with weight 2 and as many columns with weight 3 as necessary. The minimum weight of the resulting $\mathbf{L}$ is $W_7 \triangleq 1+2\min\{a-1, 9\}+3\max\{0, a-10\}$.
6) If the all-zeros column and the columns with weight 1 are not used, we can fill the remaining part of $\mathbf{L}$ with 12 columns with weight 2 and as many columns with weight 3 as necessary. The minimum weight of the resulting $\mathbf{L}$ is $W_8 \triangleq 2\min\{a, 12\}+3\max\{0, a-12\}$.

Then, we have compared W_3 to W_8 for increasing $a > 4$, choosing the function (or one of the functions) providing the smallest $\mathcal{W}(\mathbf{L})$.

REFERENCES

[1] M. Battaglioni, M. Baldi, F. Chiaraluce, and M. Lentmaier, “Girth properties of time-varying SC-LDPC convolutional codes,” in *Proc. IEEE International Symposium on Information Theory (ISIT 2019)*, Paris, France, Jul. 2019, pp. 2599–2603.

[2] K. Kudekar, T. Richardson, and R. L. Urbanke, “Spatially coupled ensembles universally achieve capacity under belief propagation,” *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.

[3] 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.

[4] I. E. Bocharova, F. Hug, R. Johannesson, B. D. Kudryashov, and R. V. Satyukov, “Searching for voltage graph-based LDPC tailbiting codes with large girth,” *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2265–2279, Apr. 2012.

[5] M. Lentmaier, M. M. Prenda, and G. Fettweis, “Efficient message-passing scheduling for terminated LDPC convolutional codes,” in *Proc. IEEE International Symposium on Information Theory (ISIT 2011)*, St. Petersburg, Russia, Aug. 2011, pp. 1826–1830.

[6] A. R. Iyengar, M. Papaleo, P. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, “Windowed decoding of protograph-based LDPC convolutional codes over erasure channels,” *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2303–2320, Apr. 2012.

[7] D. J. Costello, L. Dolecek, T. E. Fuja, J. Kliewer, D. G. M. Mitchell, and R. Smarandache, “Spatially coupled sparse codes on graphs: theory and practice,” *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 168–176, Jul. 2014.

[8] M. Battaglioni, A. Tasdighi, M. Baldi, M. H. Tadayon, and F. Chiaraluce, “Compact QC-LDPC block and SC-LDPC convolutional codes for low-latency communications,” in *Proc. 2018 IEEE 29th International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC)*, Bologna, Italy, Sep. 2018.

[9] A. E. Pusane, R. Smarandache, P. O. Vontobel, and D. J. Costello, “Deriving good LDPC convolutional codes from LDPC block codes,” *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 835–857, Feb. 2011.

[10] K. Liu, M. El-Khamy, and J. Lee, “Finite-length algebraic spatially-coupled quasi-cyclic LDPC codes,” *IEEE J. Select. Areas Commun.*, vol. 34, no. 2, pp. 329–344, Feb. 2016.

[11] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, “LDPC block and convolutional codes based on circulant matrices,” *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 2966–2984, Dec. 2004.

[12] D. Achlioptas, H. Hassani, W. Liu, and R. Urbanke, “Time-invariant LDPC convolutional codes,” in *Proc. IEEE International Symposium on Information Theory (ISIT 2017)*, Aachen, Germany, Jun. 2017, pp. 366–370.

[13] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, “Finite-length analysis of low-density parity-check codes on the binary erasure channel,” *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.

[14] T. Richardson, “Error floors of LDPC codes,” in *Proc. 41st Annual Allerton Conf.*, Monticello, IL, Oct. 2003, pp. 1426–1435.

[15] L. Dolecek, Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, “Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes,” *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 181–201, Jan. 2010.

[16] Y. Hashemi and A. Banihashemi, “On characterization of elementary trapping sets of variable-regular LDPC codes,” *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5188–5203, Sep. 2014.

[17] ——, “New characterization and efficient exhaustive search algorithm for leafless elementary trapping sets of variable-regular LDPC codes,” *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6713–6736, Dec. 2016.

[18] X. Tao, Y. Li, Y. Liu, and Z. Hu, “On the construction of LDPC codes free of small trapping sets by controlling cycles,” *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 9–12, Jan. 2018.

[19] M. Sadeghi and F. Amirzade, “Edge-coloring technique to analyze elementary trapping sets of spatially-coupled LDPC convolutional codes,” *IEEE Commun. Lett.*, Early Access, 2019.

[20] S. Naseri and A. H. Banihashemi, “Construction of girth-8 QC-LDPC codes free of small trapping sets,” *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1904–1908, Nov 2019.

[21] Y. Wang, J. S. Yedidia, and S. C. Draper, “Construction of high-girth QC-LDPC codes,” in *Proc. 5th Int. Symp. on Turbo Codes and Related Topics*, Lausanne, Switzerland, Oct. 2008, pp. 180–185.

[22] Y. Wang, S. C. Draper, and J. S. Yedidia, “Hierarchical and high-girth QC LDPC codes,” *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4553–4583, Jul. 2013.

[23] M. Battaglioni, A. Tasdighi, G. Cancellieri, F. Chiaraluce, and M. Baldi, “Design and analysis of time-invariant SC-LDPC convolutional codes with small constraint length,” *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 918–931, Mar. 2018.

[24] H. Zhou and N. Goertz, “Cycle analysis of time-invariant LDPC convolutional codes,” in *Proc. 2010 17th International Conference on Telecommunications (ICT 2010)*, Doha, Qatar, Apr. 2010, pp. 23–28.

[25] ——, “Unavoidable cycles in polynomial-based time-invariant LDPC convolutional codes,” in *Proc. 11th European Wireless Conference on Sustainable Wireless Technologies*, Vienna, Austria, Apr. 2011, pp. 39–44.

[26] ——, “Girth analysis of polynomial-based time-invariant LDPC convolutional codes,” in *Proc. 2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Vienna, Austria, Apr. 2012, pp. 104–108.

[27] M. Battaglioni, M. Baldi, and G. Cancellieri, “Connections between low-weight codewords and cycles in spatially coupled LDPC convolutional codes,” *IEEE Trans. Commun.*, vol. 66, no. 8, pp. 3268–3280, Aug. 2018.

[28] M. H. Tadayon, A. Tasdighi, M. Battaglioni, M. Baldi, and F. Chiaraluce, “Efficient search of compact QC-LDPC and SC-LDPC convolutional codes with large girth,” *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1156–1159, Jun. 2018.

[29] M. Sadeghi, “Optimal search for girth-8 quasi cyclic and spatially coupled multiple-edge LDPC codes,” *IEEE Commun. Lett.*, vol. 23, no. 9, pp. 1466–1469, Sep. 2019.

[30] S. Naseri and A. H. Banihashemi, “Spatially coupled LDPC codes with small constraint length and low error floor,” *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 254–258, Feb. 2019.

[31] M. B. Tavares, K. S. Zigangirov, and G. P. Fettweis, “Tail-biting LDPC convolutional codes,” in *Proc. IEEE International Symposium on Information Theory (ISIT 2007)*, Nice, France, Jun. 2007, pp. 2341–2345.

[32] S. Myung, K. Yang, and J. Kim, “Quasi-cyclic LDPC codes for fast encoding,” *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2894–2901, Aug. 2005.

[33] S. Kim, J.-S. No, H. Chung, and D.-J. Shin, “Quasi-cyclic low-density parity-check codes with girth larger than 12,” *IEEE Trans. Inf. Theory*, vol. 53, no. 8, pp. 2885–2891, Aug. 2007.

[34] S. J. Chandrasetty, V. A. Johnson and G. Lechner, “Memory-efficient quasi-cyclic spatially coupled low-density parity-check and repeat-accumulate codes,” *IET Commun.*, vol. 8, no. 17, pp. 3179–3188, Nov. 2014.

[35] D. G. M. Mitchell, R. Smarandache, and D. J. Costello, “Quasi-cyclic LDPC codes based on pre-lifted protographs,” *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5856–5874, Oct. 2014.

[36] R. Smarandache and P. O. Vontobel, “Quasi-cyclic LDPC codes: Influence of proto- and Tanner-graph structure on minimum Hamming distance upper bounds,” *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 585–607, Feb. 2012.

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

[38] M. P. C. Fossorier, “Quasi-cyclic low-density parity-check codes from circulant permutation matrices,” *IEEE Trans. Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[39] M. Sadeghi and F. Amirzade, “Analytical lower bound on the lifting degree of multiple-edge QC-LDPC codes with girth 6,” *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1528–1531, Aug. 2018.

[40] R. Townsend and E. J. Weldon, “Self-orthogonal quasi-cyclic codes,” *IEEE Trans. Inf. Theory*, vol. IT-13, no. 2, pp. 183–195, Apr. 1967.

[41] M. Karlin, “New binary coding results by circulants,” *IEEE Trans. Inf. Theory*, vol. IT-15, no. 1, pp. 81–92, Jan. 1969.

[42] M. Battaglioni, A. Tasdighi, G. Cancellieri, F. Chiaraluce, and M. Baldi. (2017, Jul.) Design and analysis of time-invariant SC-LDPC convolutional codes with small constraint length: exponent matrices. [Online]. Available: https://doi.org/10.6084/m9.figshare.5166373

[43] T. Richardson and R. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[44] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, “Iterative decoding threshold analysis for LDPC convolutional codes,” *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

[45] R. G. Gallager, “Low-density parity-check codes,” *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[46] M. Lentmaier, D. V. Truhachev, and K. S. Zigangirov, “To the theory of low-density convolutional codes. II,” *Problems of Information Transmission*, vol. 32, pp. 288–306, 2001.

[47] A. Tasdighi, A. H. Banihashemi, and M.-R. Sadeghi, "Efficient search of girth-optimal QC-LDPC codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 4, pp. 1552–1564, Apr. 2016.

[48] H. Zhou and N. Goertz, "Cycle analysis of time-variant LDPC convolutional codes," in *Proc. 6th Int. Symp. Turbo Codes Iterative Information Processing*, Paris, France, Sep. 2010, pp. 48–52.

[49] M. Battaglioni, F. Chiaraluce, M. Baldi, and M. Lentmaier. (2021) Dataset related to "Girth analysis and design of periodically time-varying SC-LDPC codes". [Online]. Available: http://doi.org/10.5281/zenodo.4467759

[50] M. Esmaeili and M. Gholami, "Structured quasi-cyclic LDPC codes with girth 18 and column-weight $J \geq 3$," *AEU - International Journal of Electronics and Communications*, vol. 64, no. 3, pp. 202–217, 2010.

[51] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, Inc., 2004.

[52] J. Xu, L. Chen, I. Djurdjevic, S. Lin, and K. Abdel-Ghaffar, "Construction of regular and irregular LDPC codes: Geometry decomposition and masking," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 121–134, Jan 2007.

Massimo Battaglioni (Member, IEEE) was born in Macerata, Italy, in 1991. He received a Laurea degree in Electronic Engineering in 2013, a Laurea Magistrale degree in Electronic Engineering (summa cum laude) in 2015 and a PhD in Information Engineering in 2019, by Marche Polytechnic University. Since 2019 he is a postdoctoral researcher at the Department of Information Engineering of Marche Polytechnic University. In 2017 he has been a visiting student at the Electrical and Information Technology Department, LTH, Lund University, Sweden. In 2018 he has been a visiting student at the Klipsch School of Electrical and Computer Engineering, NMSU, Las Cruces, NM, USA and at the School of Electrical and Electronic Engineering, University College Dublin, Ireland. His research activity is focused on coding techniques for communications reliability and cryptography, with particular attention to block and convolutional LDPC codes for symmetric and asymmetric channels and their application to cryptography. He currently serves as an Editor for IEEE Communications Letters.

Franco Chiaraluce (Senior Member, IEEE) was born in Ancona, Italy, in 1960. He received the Laurea degree (summa cum laude) in electronic engineering from the University of Ancona in 1985. Since 1987, he has been with the Department of Electronics and Automatics, University of Ancona. He is currently a Full Professor in Telecommunications at the Marche Polytechnic University, Ancona, Italy. He has coauthored more than 300 scientific papers and 3 books, and he holds 3 patents. On his research topics, he collaborates with national and international companies. His main research interests involve various aspects of communication systems theory and design, with a special emphasis on error correcting codes, cryptography, and physical layer security. He is also a member of the IEICE.

Marco Baldi (Senior Member, IEEE) received the Laurea degree (Hons.) in electronics engineering and the Ph.D. degree in electronics, computer, and telecommunications engineering from the Università Politecnica delle Marche, Ancona, Italy. Since 2019, he has been an Associate Professor with the Università Politecnica delle Marche, Department of Information Engineering, where he also coordinates the local node of the CINI Cybersecurity National Laboratory and takes part in the Research and Service Center for Privacy and Cybersecurity (CRiSPY). He has coauthored over 150 scientific articles and one book, and holds four patents. His research is focused on coding and cryptography for information security and reliability. He is a member of AEIT, CINI, CNIT, IEEE Communications Society, IEEE Information Theory Society and an IEEE senior member. He serves as a Senior Associate Editor for IEEE Communications Letters and an Associate Editor for the EURASIP Journal on Wireless Communications and Networking and the MDPI Information journal.

Michael Lentmaier (Senior Member, IEEE) is an Associate Professor at the Department of Electrical and Information Technology at Lund University, which he joined in January 2013. His research interests include design and analysis of coding systems, graph based iterative algorithms and Bayesian methods applied to decoding, detection and estimation in communication systems. He received the Dipl.-Ing. degree in electrical engineering from University of Ulm, Germany in 1998, and the Ph.D. degree in telecommunication theory from Lund University, Sweden in 2003. He then worked as a Post-Doctoral Research Associate at University of Notre Dame, Indiana and at University of Ulm. From 2005 to 2007 he was with the Institute of Communications and Navigation of the German Aerospace Center (DLR) in Oberpfaffenhofen, where he worked on signal processing techniques in satellite navigation receivers. From 2008 to 2012 he was a senior researcher and lecturer at the Vodafone Chair Mobile Communications Systems at TU Dresden, where he was heading the Algorithms and Coding research group. He is a senior member of the IEEE and served as an editor for IEEE Communications Letters (2010-2013), IEEE Transactions on Communications (2014-2017), and IEEE Transactions on Information Theory (2017-2020). He was awarded the Communications Society & Information Theory Society Joint Paper Award (2012) for his paper "Iterative Decoding Threshold Analysis for LDPC Convolutional Codes".