

Unavoidable Cycles in Polynomial-Based Time-Invariant LDPC Convolutional Codes

Hua Zhou and Norbert Goertz

Institute of Telecommunications
Vienna University of Technology
Gusshausstrasse 25/389, 1040 Wien, Austria
Email: {Hua.Zhou, Norbert.Goertz}@nt.tuwien.ac.at

Abstract—Low-Density Parity-Check convolutional codes (LDPCccs) are very interesting for practical error-correction coding in wireless transmission as they have excellent performance and at the same time they allow for variable block size with low complexity encoding and decoding. As for all LDPC codes that are decoded by the sub-optimal (but highly efficient) Sum Product Algorithm, the cycles in the code graph are very important for the practical performance of the coding scheme.

Time-invariant LDPCccs can be defined by a polynomial syndrome former (transposed parity-check matrix in polynomial form), that can be derived from corresponding Quasi-Cyclic (QC) LDPC block codes. Given the polynomial syndrome former with certain structures, unavoidable cycles with lengths ranging from 6 to 12 will be shown to exist. We provide some rules for designing good codes with respect to the shortest cycle in the code-graph, the girth of the code, which is a crucial parameter for its decoding performance.

I. INTRODUCTION

LDPC convolutional codes (LDPCccs) were first proposed in [1]. Studies in [2] have shown that LDPC convolutional codes are suitable for practical implementation with continuous transmission as well as block transmission in frames of arbitrary size. It has been proved that, under pipeline decoding, LDPC convolutional codes have an error performance comparable to that of their block-code counterparts without an increase in computational complexity [3].

LDPC convolutional codes can be separated into two categories, time-invariant and time-variant LDPCccs, with respect to the structure of their syndrome formers. Time-invariant LDPCccs can be derived from Quasi-Cyclic (QC) LDPC block codes [4] and protographs [6], while time-variant ones normally are obtained by unwrapping the parity check matrices of LDPC block codes [1], [8].

In terms of the decoding performance of LDPC convolutional codes, free distance and cycle properties (girth and number of short cycles) are two major issues, which are related to the existence of an error floor and the convergence speed of the Sum Product Algorithm (SPA) [10]. In [5] and [9], cycle properties of time-invariant and time-variant LDPC convolutional codes derived from QC LDPC block codes have been analyzed, respectively. In this paper, we will investigate some unavoidable cycles in time-invariant LDPC convolutional codes. Given certain polynomial syndrome former structures $\mathbf{H}^T(D)$, some cycles can not be avoided,

no matter what specific monomials we pick when defining $\mathbf{H}^T(D)$. Consequently, these destructive structures have to be eliminated in the code design in order to achieve large girth.

The rest of the paper is organized as follows: In Section II, we review the definition and the syndrome former of an LDPC convolutional code. Section III reviews how to derive a time-invariant LDPC convolutional code from corresponding QC LDPC block code together with an example. In Section IV, we discuss the definition and conditions for cycles in a polynomial syndrome former of a time-invariant LDPC convolutional code. Finally, Section V presents some unavoidable cycles under different polynomial entry weights.

II. LDPC CONVOLUTIONAL CODES

A rate $R=b/c$ regular (m_s, J, K) LDPC convolutional code is the set of sequences \mathbf{v} satisfying the equation $\mathbf{v}\mathbf{H}^T=\mathbf{0}$, where

$$\mathbf{H}^T = \begin{bmatrix} \ddots & & \ddots & & & & \\ H_0^T(0) & \dots & H_{m_s}^T(m_s) & & & & \\ & \ddots & \vdots & \ddots & & & \\ & & H_0^T(t) & \dots & H_{m_s}^T(t+m_s) & & \\ & & \ddots & & \ddots & & \\ & & & & & \ddots & \end{bmatrix}. \quad (1)$$

This semi-infinite transposed parity check matrix \mathbf{H}^T , called the syndrome former, is made up of a set of binary matrices $H_i^T(t+i)$ of size $c \times (c-b)$. The syndrome former memory m_s together with the associated constraint length, defined as $v_s = (m_s + 1) \cdot c$, is proportional to the decoding complexity. However, to achieve capacity-approaching performance, a large value of m_s is required [3]. \mathbf{H}^T contains exactly J and K ones in each row and column, respectively. For time-invariant LDPC convolutional codes, the binary “sub-matrices” in \mathbf{H}^T are constant, i.e., $H_i^T(i) = H_i^T(i+t)$, $t = 1, 2, 3, \dots$, while for periodic time-variant LDPC convolutional codes with period T we have $H_i^T(i) = H_i^T(i+T)$.

For a periodic time-variant LDPC convolutional code with period T , given the time-domain syndrome former \mathbf{H}^T in (1) and syndrome former memory m_s , we define the corresponding polynomial-domain syndrome former (briefly called

polynomial syndrome former) as:

$$\mathbf{H}^T(D) = \begin{bmatrix} H_1^T(D) \\ H_2^T(D) \\ \vdots \\ H_t^T(D) \\ \vdots \\ H_T^T(D) \end{bmatrix}, \quad (2)$$

where each polynomial sub-matrix is

$$H_t^T(D) = \sum_{n=0}^{m_s} H_n^T(t+n) \cdot D^n \quad (3)$$

with $H_n^T(t+n)$ given in (1). Note that, in (2) the superscript T means transpose while the subscript T refers to the period. Finally, $\mathbf{H}^T(D)$ consists of T distinct polynomial sub-matrices, and each is of size $c \times (c-b)$. It is clear that, for time-invariant LDPC convolutional codes, i.e., $T=1$, there is only one polynomial sub-matrix in the polynomial syndrome former. In this paper, we will focus on time-invariant cases.

III. DERIVATION OF LDPC CONVOLUTIONAL CODES

A. Time-invariant LDPCccs

A particular category of time-invariant LDPC convolutional codes, described by polynomial matrices, is derived from Quasi Cyclic (QC) LDPC block codes. As described in [4], in the Galois Field $\text{GF}(m)$, with m prime, we assume that a and b are two nonzero elements with multiplicative orders $o(a) = k$ and $o(b) = j$, respectively. With the (s, t) th element $P_{s,t} = b^s a^t \pmod m$, $s=0,1,\dots,j-1$ and $t=0,1,\dots,k-1$, we form the $j \times k$ matrix \mathbf{P} of elements from $\text{GF}(m)$:

$$\mathbf{P} = \begin{bmatrix} 1 & a & a^2 & \dots & a^{k-1} \\ b & ab & a^2b & \dots & a^{k-1}b \\ \dots & \dots & \dots & \dots & \dots \\ b^{j-1} & ab^{j-1} & a^2b^{j-1} & \dots & a^{k-1}b^{j-1} \end{bmatrix} \quad (4)$$

(all products in the matrix are “modulo m ”).

The parity-check matrix \mathbf{H}_{QC} of a QC LDPC block code is generated from (4) as follows:

$$\mathbf{H}_{QC} = \begin{bmatrix} I_1 & I_a & I_{a^2} & \dots & I_{a^{k-1}} \\ I_b & I_{ab} & I_{a^2b} & \dots & I_{a^{k-1}b} \\ \dots & \dots & \dots & \dots & \dots \\ I_{b^{j-1}} & I_{ab^{j-1}} & I_{a^2b^{j-1}} & \dots & I_{a^{k-1}b^{j-1}} \end{bmatrix}. \quad (5)$$

The matrices $I_{P_{s,t}}$ are $m \times m$ identity matrices cyclically shifted to the left by $P_{s,t}-1$ positions. Note that, therefore, I_1 corresponds to an “unshifted” standard identity matrix.

As explained in [4], LDPC convolutional codes can be formed by a matrix that consists of polynomial entries, with each polynomial entry describing the first column of the corresponding shifted identity sub-matrix in \mathbf{H}_{QC} . For instance, if the first *column* of a *horizontally left-shifted* identity matrix is $[00010]^T$ (this means three *horizontal* shifts to the left), it is represented by the polynomial D^3 . Therefore, according to the

parity-check matrix in (5), the associated polynomial matrix for the LDPC convolutional codes is given by

$$\mathbf{H}(D) = \begin{bmatrix} D^0 & D^{a-1} & D^{a^2-1} & \dots & D^{a^{k-1}-1} \\ D^{b-1} & D^{ab-1} & D^{a^2b-1} & \dots & D^{a^{k-1}b-1} \\ \dots & \dots & \dots & \dots & \dots \\ D^{b^{j-1}-1} & D^{ab^{j-1}-1} & D^{a^2b^{j-1}-1} & \dots & D^{a^{k-1}b^{j-1}-1} \end{bmatrix}. \quad (6)$$

A polynomial matrix generated in this way will produce a time-invariant LDPC convolutional code, as the polynomial parity-check matrix is constant.

$\mathbf{H}(D)$ is the parity-check matrix of a regular (j, k) LDPC convolutional code in polynomial form; it has exactly j monomials in each column and k monomials in each row. The notation “ D ” can be interpreted as a delay as common in convolutional codes. The degrees/weights¹ of the polynomial entries in (6) are all one. In literature, such a matrix with all degree-one polynomials as its elements is also called a “monomial matrix”.

B. Code example of a time-invariant LDPCccs

We use the method summarized above to generate a LDPC convolutional code that will be used to illustrate cycle formation; the code was also used as an example in [4], [7]. We pick the field $\text{GF}(7)$ and the elements $a=2$ and $b=6$. As $k=3$ is the smallest positive integer $k > 0$ for which $a^k \pmod 7=1$, the order of a is $k \doteq o(a)=3$; similarly, the order of b is $j \doteq o(b)=2$. From this we obtain by (5)

$$\mathbf{H}_{QC} = \begin{bmatrix} I_1 & I_2 & I_4 \\ I_6 & I_5 & I_3 \end{bmatrix}_{2.7 \times 3.7} \quad (7)$$

for the parity-check matrix of the QC LDPC block code. By (6) we obtain the parity-check matrix

$$\mathbf{H}(D) = \begin{bmatrix} D^0 & D^1 & D^3 \\ D^5 & D^4 & D^2 \end{bmatrix}_{2 \times 3} \equiv \begin{bmatrix} 1 & D & D^3 \\ D^3 & D^2 & 1 \end{bmatrix}_{2 \times 3} \quad (8)$$

in polynomial form for the LDPC convolutional code. The code has a code rate of $R=1-j/k=1-2/3=1/3$. We have removed the common factor D^2 from the lower row of $\mathbf{H}(D)$. Removing such common factor will lead to smaller syndrome former memory² $m_s = 3$ (compared to $m_s = 5$ in the original form of $\mathbf{H}(D)$) and is, therefore, favorable for a practical implementation [7].

IV. CYCLE PROPERTIES OF LDPCCCS

It is well known that “short” cycles in the factor graph of LDPC block codes affect the decoding performance. The larger the girth – the shortest cycle in a factor graph – the better the decoding performance will be. A “good” code, however, will have to have cycles in the code graph to achieve good properties such as large minimum distance. Hence, we are

¹The degree or weight of a polynomial is defined as the number of additive terms with different powers of D that are involved. For instance, the degree of $1+D+D^2$ is three, while the degree of $D+D^3$ is two. When the degree of a polynomial is one it is also called a monomial.

²the syndrome former memory m_s is equal to the largest power among all the polynomial entries D^l in $\mathbf{H}(D)$.

interested to analyze cycle properties of LDPC codes. In this section, we will briefly review how cycles are formed in time-invariant LDPC codes regarding the example in (8).

Without loss of generality, we will below consider the transpose of the $\mathbf{H}(D)$ matrix, which is called polynomial syndrome former. A cycle of length 12 is shown in the polynomial syndrome former $\mathbf{H}^T(D)$ in Fig. 1.

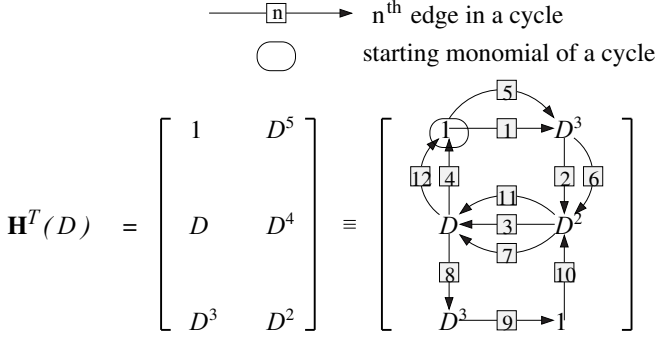


Fig. 1: Polynomial domain transposed parity-check matrix $\mathbf{H}^T(D)$

Monomials from the same row or column in the polynomial syndrome former $\mathbf{H}^T(D)$ connect to each other with or without delay: the “delay” incurred equals the difference of the “powers” of the two monomials that are connected. This is illustrated for our code example in Fig. 2: the elements s_{ij} are the powers of D from the syndrome former matrix at the right-hand side of Fig. 1. The formal definition of the delay

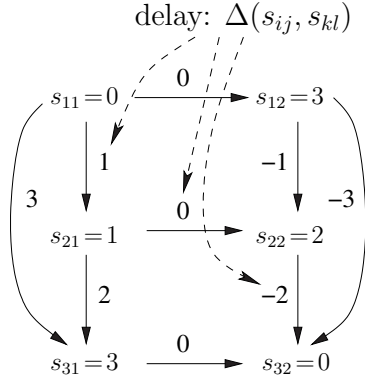


Fig. 2: Delays for code example derived from polynomial syndrome former matrix

is given by

$$\Delta(s_{ij}, s_{kl}) \doteq \Delta(s_{ij}, s_{il}) + \Delta(s_{il}, s_{kl}) \quad (9)$$

where it has decomposed a possible move from one element s_{ij} in the “matrix” in Fig. 2 to any other element s_{kl} into a horizontal move $s_{ij} \rightarrow s_{il}$ with zero delay, i.e.,

$$\Delta(s_{ij}, s_{il}) \doteq 0 \quad (10)$$

and a vertical move $s_{il} \rightarrow s_{kl}$ with the delay

$$\Delta(s_{il}, s_{kl}) \doteq s_{kl} - s_{il}. \quad (11)$$

Note that the delay $\Delta(s_{il}, s_{kl})$ for the vertical move in Fig. 2 has a sign, which identifies whether we move “forward” or “backward” in time when progressing along a vertical edge.

A path \mathcal{P} – which is a sequence of pairs of elements from the polynomial syndrome former – forms a cycle of length $2 \times L$, when

$$\sum_{\forall \{s, s'\} \in \mathcal{P}} \Delta(s, s') = 0 \quad (12)$$

with the path given by

$$\mathcal{P} = \left\{ \underbrace{\{s_{j_1 k_1}, s_{j_1 k_2}\}}_{\text{horizontal move}}, \underbrace{\{s_{j_1 k_2}, s_{j_2 k_2}\}}_{\text{vertical}}, \underbrace{\{s_{j_2 k_2}, s_{j_2 k_3}\}}_{\text{horizontal}}, \dots, \underbrace{\{s_{j_{L-1} k_{L-1}}, s_{j_{L-1} k_L}\}}_{\text{horizontal}}, \underbrace{\{s_{j_{L-1} k_L}, s_{j_1 k_1}\}}_{\text{vertical}} \right\} \quad (13)$$

with $j_x \in \{1, \dots, j\}$ and $k_y \in \{1, \dots, k\}$ and $x \in \{1, \dots, L\}$ and $y \in \{1, \dots, L\}$. It is important to note that the first and the last element in the path have to be the same in order for \mathcal{P} to form a cycle. This condition is, however, only necessary but not sufficient, as in addition the sum in (12) must be zero!

We will illustrate the concept by an example: the 12-cycle is described by a “closed path” in the polynomial syndrome former as depicted in Fig. 1. We apply the path from Fig. 1 to Fig. 2 and calculate the sum of the delays which are accumulated along the path:

$$\sum_{\forall \{s, s'\} \in \mathcal{P}} \Delta(s, s') = \begin{cases} \text{horizontal moves} & \text{vertical moves} \\ \Delta(s_{11}, s_{12}) & + \Delta(s_{12}, s_{22}) & + \\ \Delta(s_{22}, s_{21}) & + \Delta(s_{21}, s_{11}) & + \\ \Delta(s_{11}, s_{12}) & + \Delta(s_{12}, s_{22}) & + \\ \Delta(s_{22}, s_{21}) & + \Delta(s_{21}, s_{31}) & + \\ \Delta(s_{31}, s_{32}) & + \Delta(s_{32}, s_{22}) & + \\ \Delta(s_{22}, s_{21}) & + \Delta(s_{21}, s_{11}) & \end{cases} \quad (14)$$

With (10) we have “zero” delay for all horizontal moves and we use (11) for the vertical moves. We obtain

$$\sum_{\forall \{s, s'\} \in \mathcal{P}} \Delta(s, s') = \begin{cases} 0 & + & (2-3) & + \\ 0 & + & (0-1) & + \\ 0 & + & (2-3) & + \\ 0 & + & (3-1) & + \\ 0 & + & (2-0) & + \\ 0 & + & (0-1) & \end{cases} = 0, \quad (15)$$

which confirms the condition in (12): this 12-cycle does indeed exist.

V. UNAVOIDABLE CYCLES

According to the description in Section IV, it seems that cycles can be eliminated by choosing suitable power index of each monomial entry in $\mathbf{H}^T(D)$ that will not enable any path to satisfy the condition in (12). However, we find that there are some unavoidable cycles, no matter what the “powers” of the

monomials in the polynomial syndrome former are. To better understand the existence of these cycles, we will, as a novelty, illustrate the unavoidable cycles with lengths ranging from 6 to 12 from a geometric perspective.

A. Unavoidable 12-cycles

Similar to the 12-cycle path in Fig. 1, in Figs. 3 and 4 we demonstrate by simple graphical means that *any* LDPC convolutional code that can be described by a time-invariant monomial syndrome former matrix has to have 12-cycles, if the underlying QC code has $j \geq 2$ and $k \geq 3$ (which will be true for any useful code). We use 3×2 (in Fig. 3) and

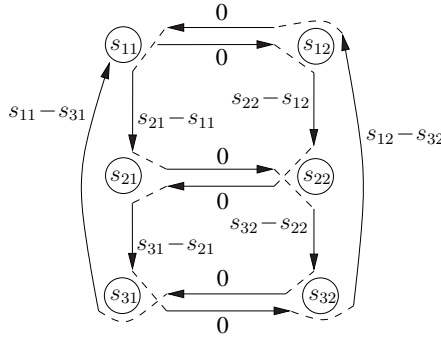


Fig. 3: Unavoidable 12-Cycles in a Monomial Syndrome Former

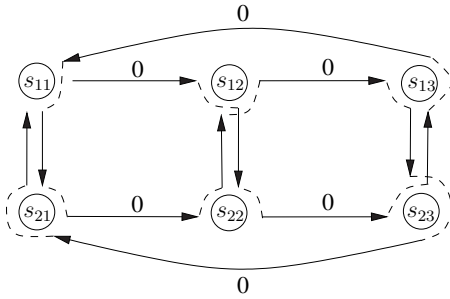


Fig. 4: Unavoidable 12-Cycles in a Monomial Syndrome Former

2×3 (in Fig. 4) submatrices of a possible syndrome former to demonstrate how 12-cycles are formed. Those cycles appear, regardless of the “powers” of the monomials in the matrix: it is just a structural property, common to all time-invariant LDPC convolutional codes based on a QC block code construction by circulant matrices.

In Fig. 3 we only need to form the sum of the delays along the path starting at element s_{11} : no matter which values the

elements s_{ij} have, the path delay sum is always zero, i.e.,

$$\sum_{\forall \{s, s'\} \in \mathcal{P}'} \Delta(s, s') = \left\{ \begin{array}{l} 0 + (s_{22} - s_{12}) + \\ 0 + (s_{31} - s_{21}) + \\ 0 + (s_{12} - s_{32}) + \\ 0 + (s_{21} - s_{11}) + \\ 0 + (s_{32} - s_{22}) + \\ 0 + (s_{11} - s_{31}) \end{array} \right\} = 0, \quad (16)$$

therefore, we have a cycle. The same applies to the second case depicted in Fig. 4. The situation is even simpler than in the first case, as all vertical branches (which incur a delay) are traversed twice and in opposite directions: therefore, the sum of the delays is again zero.

In Fig. 5 we show a full list of all such submatrices for a $(3, 5)$ LDPC convolutional code. From Fig. 5 we conclude

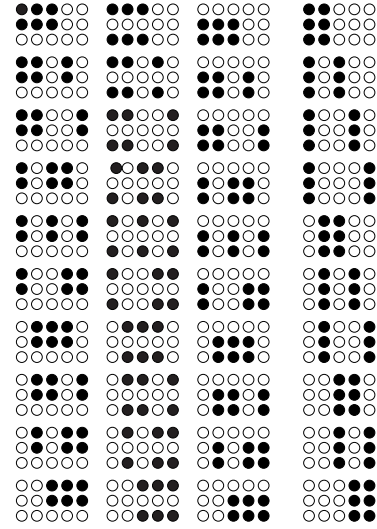


Fig. 5: 12-Cycle-Submatrices in the syndrome former of a $(3, 5)$ LDPC convolutional code

that a lower bound for the number of structurally different non-repetitive 12-cycles of an (j, k) LDPC convolutional code with $j \geq 2$ and $k \geq 3$ can be computed analytically according to

$$N_{12\text{-Cycles}} \geq \binom{j}{2} \cdot \binom{k}{3} + \binom{j}{3} \cdot \binom{k}{2}. \quad (17)$$

For a $(3, 5)$ LDPC convolutional code we obtain $N_{12\text{-Cycles}} \geq \binom{3}{2} \cdot \binom{5}{3} + \binom{3}{3} \cdot \binom{5}{2} = 40$ which is confirmed by the full list in Fig. 5. It should be noted, however, that depending on the choice of the polynomials in the syndrome former matrix the number of 12-cycles may be much larger than the lower bound. In Table I in [5], it shows the cycle calculations of a couple of code examples. The $(3,5)$ -code with $m/m_s = 2311/2005$ is a particularly interesting example: this code fulfills the lower bound (17) and it does not have any 4,6,8,10-cycles. Therefore this time-invariant LDPC convolutional code is “perfect” with respect to its cycle properties and cannot be improved. This, however, does not make a statement about the free distance or weight distribution of this code.

B. Unavoidable 6- 8- and 10-cycles

The time-invariant LDPC convolutional codes derived from QC LDPC block codes in this paper only consist of monomial entries (i.e., their weight is one) in the polynomial syndrome former $\mathbf{H}^T(D)$. In this subsection, cycle properties of LDPC-codes with entry weight larger than one will be considered. In [7], two girth theorems have been arithmetically proved for the case that the weight of any polynomial entry is larger than one. Similar to the unavoidable 12-cycles, we will focus on the structure of the polynomial syndrome former to investigate short unavoidable cycles.

Property 1: If the weight of any polynomial entry in a polynomial syndrome former is larger than two then the corresponding LDPC convolutional codes will have girth less than or equal to six.

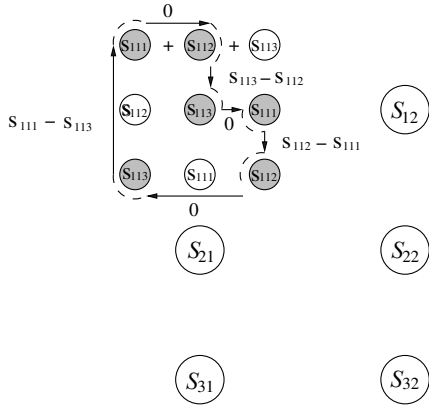


Fig. 6: Unavoidable 6-cycles in a polynomial matrix with an entry weight of three

An unavoidable 6-cycle is shown in Fig. 6: the small circles arranged in the 3×3 block represent the polynomial entry $D^{s_{11}}$ with weight 3, i.e., $D^{s_{11}} = D^{s_{111}} + D^{s_{112}} + D^{s_{113}}$. In other words, this polynomial entry consists of three monomials. Due to the property that monomials in the same row or column connect to each other with delays (10) or (11), this applies to the weight-three polynomial entry as well. For convenience in the graphical representation of the connections, $D^{s_{11}}$ is arranged in a 3×3 matrix form. Filled circles can be viewed as vertices in a cycle. After summing up all the delays along the path, powers involved in the path are cancelled by “themselves”, and, according to (12), we hence have a cycle. Fig. 6, therefore, shows that 6-cycles always exist in a polynomial syndrome former with *any* polynomial entry with weight larger than two.

Property 2: If the weight of any two polynomial entries in the same column or row in a polynomial syndrome former is larger than one, then the corresponding LDPC convolutional codes have girth less than or equal to eight.

Similar as above, we show in Fig. 7 and Fig. 8 that, if we accumulate the delays along the path, the result is zero and we now obtain unavoidable 8-cycles. Note that this property is only valid when these two polynomial entries are placed

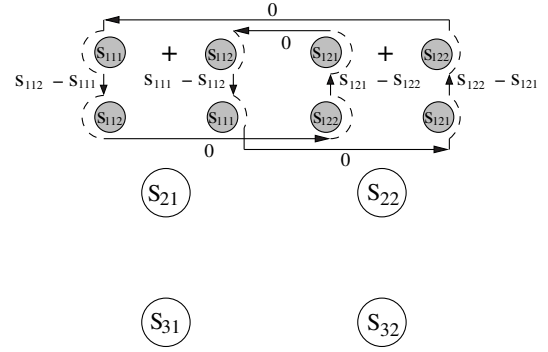


Fig. 7: Unavoidable 8-cycles in a polynomial matrix with two polynomial entries with weight two in the same row

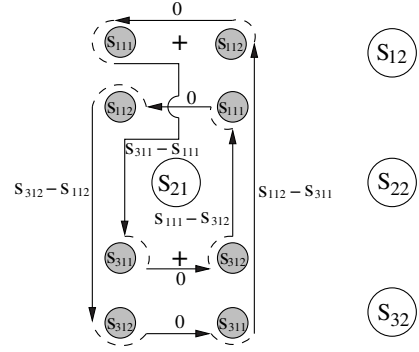


Fig. 8: Unavoidable 8-cycles in a polynomial matrix with two polynomial entries with weight two in the same column

in the same row or column. A unique weight-two polynomial entry in $\mathbf{H}^T(D)$ will cause unavoidable 10-cycles as shown in Fig. 9; the formal statement is as follows:

Property 3: If the weight of any polynomial entry in a polynomial syndrome former is larger than one, then the corresponding LDPC convolutional code has girth less than or equal to ten.

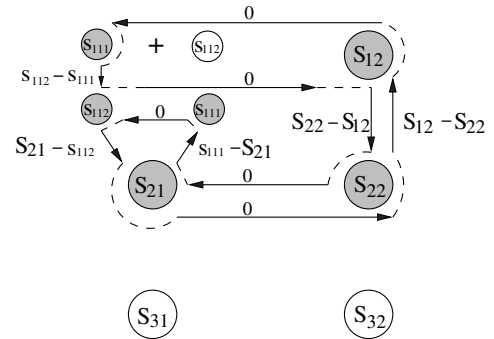


Fig. 9: Unavoidable 10-cycles in a polynomial matrix with a weight of two of any polynomial entry

VI. CONCLUSION

In this paper, the cycle formation in polynomial-domain syndrome formers of time-invariant LDPC convolutional codes, derived from corresponding QC LDPC block codes, has been

discussed. Moreover, we investigated some unavoidable cycles with lengths ranging from 6 to 12 that are caused by certain polynomial structures, no matter what the specific choice of the matrix coefficients is. Based on the analysis we conjecture that, to obtain a "good" LDPC convolutional code with respect to its (large) girth, rather than polynomial entries with large weight, monomials are preferred in the polynomial syndrome former $\mathbf{H}^T(D)$ since this will avoid small cycles and, hence, allows for larger girth. Furthermore, in order to get an LDPC convolutional code with girth larger than 12, the polynomial syndrome former should not contain any submatrices of size 2×3 or 3×2 as in Fig. 5.

REFERENCES

- [1] A. J. Felström and K. Sh. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrices," *IEEE Trans. Information Theory*, VOL. 45, NO. 6, September 1999.
- [2] S. Bates, D. G. Elliot, and R. Swamy, "Termination sequence generation circuits for low-density parity-check convolutional codes," in *IEEE Trans. Circuits and Systems*, VOL. 53, NO. 9, September 2006.
- [3] D. J. Costello, Jr., A. E. Pusane, S. Bates, and K. Sh. Zigangirov, "A comparison between LDPC block and convolutional codes," in *Proc. Information Theory and Applications Workshop*, San Diego, USA, 2006.
- [4] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello Jr. "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans. Information Theory*, VOL. 50, NO. 12, December 2004.
- [5] Hua Zhou, Norbert Goertz, "Cycle analysis of time-invariant LDPC convolutional codes," *17th International Conference on Telecommunications*, Doha, Qatar, April, 2010.
- [6] D. G. M. Mitchell, A. E. Pusane, K. Sh. Zigangirov, and D. J. Costello, "Asymptotically good LDPC convolutional codes based on protographs," *IEEE Intern. Symp. on Inform. Theory*, Toronto, Canada, July 6, 2008.
- [7] A. Sridharan, "Design and analysis of LDPC convolutional codes," Phd Dissertation, University of Notre Dame, Indiana, USA, Feb 2005.
- [8] A. E. Pusane, R. Smarandache, P. O. Vontobel, D.J. Costello "On deriving good LDPC convolutional codes from QC LDPC block codes," *IEEE Intern. Symp. on Inform. Theory*, Nice, France, June, 2007.
- [9] Hua Zhou, Norbert Goertz, "Cycle analysis of time-variant LDPC convolutional codes," *6th International Symposium on Turbo Codes & Iterative Information Processing*, Brest, France, September, 2010.
- [10] S. Lin and D. J. Costello, "Error Control Coding," Pearson Prentice Hall, 2004