

Girth Analysis of Polynomial-Based Time-Invariant LDPC Convolutional Codes

Hua Zhou and Norbert Goertz

Institute of Telecommunications
Vienna University of Technology
Gusshausstrasse 25/E389, 1040 Wien, Austria
Email: {Hua.Zhou, Norbert.Goertz}@nt.tuwien.ac.at

Abstract—Low-Density Parity-Check convolutional codes (LDPCccs) can be efficiently decoded by a pipelined sub-optimal Sum Product Algorithm. The latter may suffer, however, from convergence problems, due to cycles in the Tanner graph. To improve the decoding performance, we analyze the cycle properties, based on the connections between monomials in the polynomial syndrome former (transposed parity-check matrix in polynomial form) of time-invariant LDPCccs. Due to specific structures in the polynomial syndrome former, some cycles are *unavoidable* no matter what monomials are placed in the polynomial syndrome former. It is shown that large-weight entries in the polynomial syndrome former lead to small girth, while monomial or empty entries, which can break short unavoidable cycles, may result in large girth. A novel algorithm is proposed to generate “good” LDPCccs with respect to their cycle properties: destructive structures in the polynomial syndrome former leading to small girth are systematically avoided.

I. INTRODUCTION

Since LDPC convolutional codes (LDPCccs) were first proposed in [1], LDPCccs have been intensively investigated in recent years. Studies in [2] have shown that LDPC convolutional codes are suitable for practical implementation with continuous transmission as well as block transmission of 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], while time-variant ones are, e.g., obtained by unwrapping the parity check matrices of LDPC block codes [1], [7].

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) [9]. In [5] and [8], 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 explore fundamentally 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 briefly review the polynomial syndrome former of an LDPC convolutional code. Section III is devoted to the definition and conditions for cycles in a polynomial syndrome former of a time-invariant LDPC convolutional code. Section IV presents some unavoidable cycles under different polynomial entry weights and the existence of empty entries. Finally, in Section V we propose an algorithm to produce codes with “good” cycle properties.

II. LDPC CONVOLUTIONAL CODES

As described in [4], a particular category of time-invariant LDPC convolutional codes of rate $R = b/c$, derived from Quasi Cyclic (QC) LDPC block codes, is given by a polynomial syndrome former

$$\mathbf{H}^T(D) = \begin{bmatrix} D^{s_{11}} & D^{s_{12}} & \dots & D^{s_{1J}} \\ D^{s_{21}} & D^{s_{22}} & \dots & D^{s_{2J}} \\ \vdots & \vdots & \ddots & \vdots \\ D^{s_{K1}} & D^{s_{K2}} & \dots & D^{s_{KJ}} \end{bmatrix} \quad (1)$$

with common factors removed from each column of $\mathbf{H}^T(D)$. The syndrome former memory m_s is given by

$$m_s = \min\{s_{11}, \dots, s_{kj}, \dots, s_{KJ}\} \quad (2)$$

with $j \in \{2, \dots, J-1\}$ and $k \in \{2, \dots, K-1\}$. Together with the associated constraint length, defined as $v_s = (m_s + 1) \cdot c$, the syndrome former memory is proportional to the decoding complexity, and to achieve capacity-approaching performance, a large value of m_s is required [3]. The degrees/weights¹ of polynomial entries in (1) are all one. In literature, such a matrix with all degree-one polynomials as its elements is also called a “monomial matrix”. Time-invariant LDPCccs defined in (1) result in a regular (J, K) LDPC convolutional code with rate $R=b/c=(K-J)/K$: it has exactly J monomials in each row and K monomials in each column in the polynomial syndrome former. The notation “ D ” can be interpreted as a delay as common in convolutional codes. Note that it is not necessary for every polynomial entry to have a non-zero element. Empty

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.

entries in (1) may form an *irregular* LDPC convolutional code. In this paper, we will examine the girth properties of time-invariant LDPC convolutional codes with both *nonempty* and empty monomial entries in the polynomial syndrome former.

III. CYCLE PROPERTIES OF LDPCCCS

Short cycles in the factor graph of LDPC codes affect the independence of messages passed between check and variable nodes in sum product decoding. 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 Tanner graph to achieve good properties such as large minimum distance. Hence, we are interested to analyze cycle properties of LDPC codes. In this section, we will briefly review how cycles are formed in time-invariant LDPCccs with a polynomial syndrome former as in (1).

Monomials in the same row (column) in the polynomial syndrome former $\mathbf{H}^T(D)$ connect to each other without (with) delay: the “delay” incurred equals the difference of the “powers” of the two monomials that are connected. The formal definition of the delay between any two monomials $D^{s_{ij}}$ and $D^{s_{kl}}$ in the polynomial syndrome former (1) is given by

$$\Delta(s_{ij}, s_{kl}) \doteq \Delta(s_{ij}, s_{il}) + \Delta(s_{il}, s_{kl}), \quad (3)$$

where the delay has been decomposed into a horizontal move $s_{ij} \rightarrow s_{il}$ with zero delay, i.e.,

$$\Delta(s_{ij}, s_{il}) \doteq 0 \quad (4)$$

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

$$\Delta(s_{il}, s_{kl}) \doteq s_{kl} - s_{il}. \quad (5)$$

According to (4) and (5), a path \mathcal{P} – which is a sequence of pairs of monomials from the polynomial syndrome former – forms a cycle of length $2 \times L$, when

$$\sum_{\forall \{s, s'\} \in \mathcal{P}} \Delta(s, s') = 0 \quad (6)$$

with the path given by

$$\mathcal{P} = \left\{ \underbrace{\{s_{j_1 k_1}, s_{j_1 k_2}\}}_{\text{horizontal move}}, \underbrace{\{s_{j_1 k_2}, s_{j_2 k_2}\}}_{\text{vertical}}, \underbrace{\{s_{j_2 k_2}, s_{j_2 k_3}\}}_{\text{horizontal}}, \dots, \underbrace{\{s_{j_{L-1} k_{L-1}}, s_{j_{L-1} k_L}\}}_{\text{horizontal}}, \underbrace{\{s_{j_{L-1} k_L}, s_{j_L k_L}\}}_{\text{vertical}}, \underbrace{\{s_{j_L k_L}, s_{j_L k_1}\}}_{\text{horizontal}} \right\} \quad (7)$$

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 elements 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 (6) must be zero. This will be illustrated further in Section IV.

IV. UNAVOIDABLE CYCLES

According to the description in Section III, it seems that cycles can be eliminated by choosing a suitable power index of each monomial entry in $\mathbf{H}^T(D)$ that will not enable any path to satisfy the condition in (6). 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 such cycles, we investigate some unavoidable cycles with lengths ranging from 6 to 20 by characterizing them into three categories regarding the structure of polynomial syndrome former.

A. Unavoidable 12-cycles

In Fig. 1, we demonstrate by simple graphical means that *any* LDPC convolutional code that can be described by a time-invariant polynomial syndrome former has to have 12-cycles, if there is some 2×3 or 3×2 polynomial submatrix contained in (1) with nonempty monomial entries (which will be true for any useful code). It is important to note that this 2×3

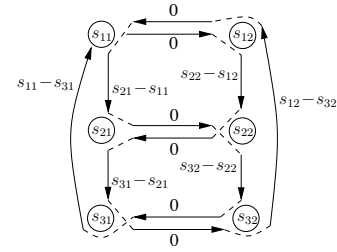


Fig. 1: Unavoidable 12-Cycles in a 3×2 polynomial submatrix or 3×2 polynomial submatrix does *not* need to be “compact”, i.e., there may be other rows and columns of (1) in between the rows and columns of the submatrix.

We use a 3×2 (see Fig. 1) submatrix of a possible syndrome former as an example to reveal how 12-cycles are formed. If we start at any monomial in the submatrix and follow the path indicated by the arrows we get back to the same monomial after 12 steps. We now apply (4) – (7) to Fig. 1: we only need to sum up all the delays (which have a sign!) 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.,

$$\sum_{\forall \{s, s'\} \in \mathcal{P}'} \Delta(s, s') = \left\{ \begin{array}{l} 0 + (s_{22} - s_{12}) + \\ 0 + (s_{31} - s_{21}) + \\ 0 + (s_{12} - s_{32}) + \\ 0 + (s_{21} - s_{11}) + \\ 0 + (s_{32} - s_{22}) + \\ 0 + (s_{11} - s_{31}) \end{array} \right\} = 0, \quad (8)$$

therefore, we have a cycle. Those cycles appear, regardless of the “powers” of the monomials in the polynomial syndrome former: they are a structural property, common to all time-invariant LDPC convolutional codes. Hence, we conclude that a lower bound for the number of structurally different non-repetitive 12-cycles of a ($J \geq 2, K \geq 3$) time-invariant LDPC convolutional code with nonempty entries in (1) can be evaluated analytically according to

$$N_{12\text{-Cycles}} \geq \binom{J}{2} \cdot \binom{K}{3} + \binom{J}{3} \cdot \binom{K}{2}. \quad (9)$$

This is a consequence of counting, how many different 2×3 and 3×2 submatrices can be built from a $K \times J$ syndrome former. For a $(3,5)$ LDPC convolutional code we obtain $N_{12\text{-Cycles}} \geq \binom{3}{2} \cdot \binom{5}{3} + \binom{3}{3} \cdot \binom{5}{2} = 40$. It should be noted, however, that depending on the choice of the polynomials in the syndrome former, the number of 12-cycles may be much larger than the lower bound. In Table I, we show the cycle calculations for a couple of code examples. The $(3,5)$ -code with $m/m_s = 2311/2005$ is a particularly interesting: this code fulfills the lower bound (9) 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 does, however, not make any 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 LDPCccs with entry weight *larger* than one will be considered. In [6] and [11], two girth theorems have been graphically and arithmetically proved for the case that the weight of any polynomial entry is larger than one, respectively.

Property 1: If the weight of any polynomial entry in a polynomial syndrome former is larger than two then the girth of corresponding time-invariant LDPC convolutional codes is upper bounded by six.

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 girth of corresponding time-invariant LDPC convolutional codes is upper bounded by eight.

Property 3: If the weight of any polynomial entry in a polynomial syndrome former is larger than one, then the girth of corresponding time-invariant LDPC convolutional code is upper bounded by ten.

C. Unavoidable cycles with length larger than 12

Above we have discussed the unavoidable cycles with length up to 12 for time-invariant LDPC convolutional codes with nonempty entries in the polynomial syndrome former. However, to achieve a girth larger than 12, empty entries are required to break shorter cycles in the polynomial syndrome former. Similar to the unavoidable 12-cycles in Fig. 1, we illustrate by Figs. 2 – 7 some structures that lead to unavoidable cycles with lengths 14 to 20. A circle in these figures refers to a nonempty monomial entry. We take Fig. 2 as an example: it is a submatrix of size 3×3 with two empty entries in positions S_{12} and S_{33} . We sum up all the delays along the path in each figure: it satisfies the condition in (6) regardless of the values of the monomials. In terms of all these unavoidable cycles and the structure of the submatrices of the polynomial syndrome former, we conclude the following property for time-invariant LDPCccs with empty entries in $\mathbf{H}^T(D)$:

Property 4: For time-invariant LDPC convolutional codes with polynomial syndrome former,

- (i) if there is an $m \times n$ submatrix with $|m-n| = 1$ containing two rows or two columns with weight \geq three;
- (ii) if there is an $m \times n$ submatrix with $|m-n| = 1$ containing one row or one column with weight \geq four;

(iii) if there is an $m \times n$ submatrix with $m = n$ containing one row and one column with weight \geq three,

and all the other rows and columns have weight \geq two, then the girth of this code is upper bounded by $(m+n+1) \times 2$.

This property offers an overview of girth bounds for time-invariant LDPC convolutional codes. For a polynomial syndrome former with low entry density we expect the code, at least to some extent, to have large girth. However, low entry density in $\mathbf{H}^T(D)$ comes with low column and row weights in the time-domain syndrome former, which reduces the free distance of convolutional codes. Therefore, there is a tradeoff between girth and free distance.

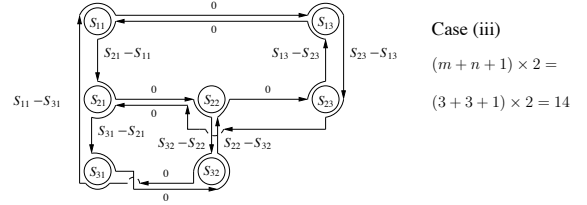


Fig. 2: Unavoidable 14-cycles in a 3×3 polynomial submatrix

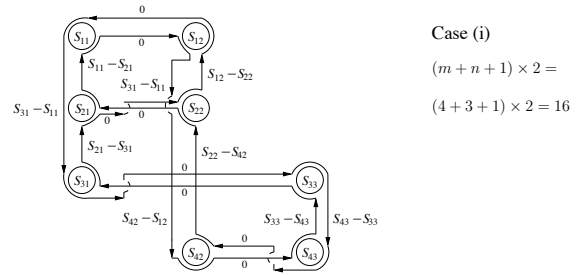


Fig. 3: Unavoidable 16-cycles in a 4×3 polynomial submatrix

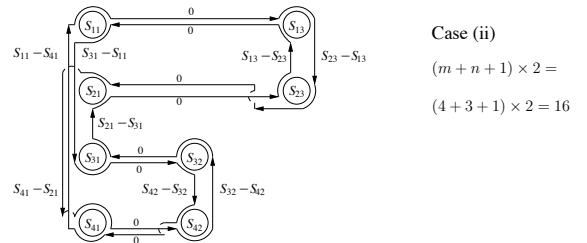


Fig. 4: Unavoidable 16-cycles in a 4×3 polynomial submatrix

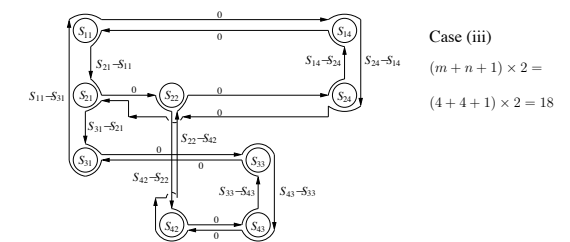


Fig. 5: Unavoidable 18-cycles in a 4×4 polynomial submatrix

V. GOOD LDPCCCS

A. The Design Algorithm

Based on the structural properties of unavoidable cycles (see Section IV), we propose an algorithm to generate polynomial syndrome formers of time-invariant LDPC convolutional codes

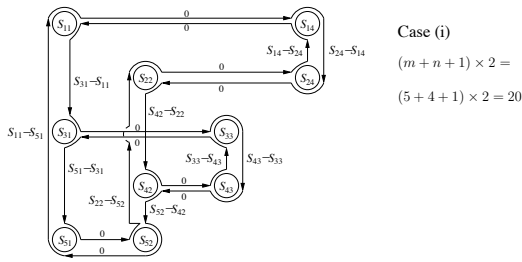


Fig. 6: Unavoidable 20-cycles in a 5×4 polynomial submatrix

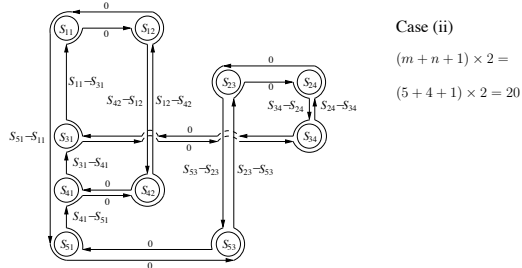


Fig. 7: Unavoidable 20-cycles in a 5×4 polynomial submatrix with a given (desired) girth. A flow chart of this algorithm is depicted in Fig. 8. It contains two steps:

- sequentially test each monomial entry in the polynomial syndrome former by means of the cycle counter algorithm in [8] until the girth of this code equals the given girth.
- try to reduce the value of syndrome former m_s as well as the number of smallest cycles, while keep the girth constant.

The main principle of the algorithm is based on the idea that if each monomial in the polynomial syndrome former is not involved in a cycle of length l , then the associated code will be free of l -cycles. Given the desired girth and considering *Properties 1 – 4*, we generate a polynomial syndrome former structure that will not contain any unavoidable cycles smaller than the selected girth. Afterwards, we apply the above two steps to place suitable entries in the polynomial syndrome former. In the first step, each monomial is sequentially tested by assigning a power index smaller than m_s to it, until no cycles smaller than the girth passing through this monomial exist. If no power indices smaller than the syndrome former memory m_s are available, m_s is increased until a suitable power index can be found. In the second step, given the achieved polynomial syndrome former from the first step, we sequentially test each monomial to check whether there is a monomial power index (smaller than the current m_s) that will result in a reduction of the number (indicated by Ng in Fig. 8) of cycles with length equal to the girth. This process is repeated until the maximum number of iterations is reached. An iteration is defined as the procedure of testing all the monomials in $\mathbf{H}^T(D)$. After each iteration, the syndrome former memory m_s is updated according to (2) and the number of cycles with length equal to the girth is calculated. In each step, an efficient cycle counter algorithm is used, the details of which can be found in [8].

B. Simulation Results

In Section IV we have shown that the maximum possible girth of a $(3, 5)$ time-invariant LDPC convolutional code with-

out empty entries in $\mathbf{H}^T(D)$ is twelve and that the minimum amount of 12-cycles is $\binom{3}{2} \cdot \binom{5}{3} + \binom{3}{3} \cdot \binom{5}{2} = 40$. In Table I, the $(3, 5)$ -code with $m/m_s = 2311/2005$ confirms these properties. However, a large syndrome former memory m_s causes high decoding complexity. To save on that, we apply the algorithm in Fig. 8 to produce a 5×3 polynomial syndrome former of a $(3, 5)$ LDPC convolutional code with no empty entries. Since there are submatrices of size 3×2 and 2×3 , the maximum achievable girth of this codes is twelve. Rather than $m_s = 2005$, we obtain by applying our algorithm the polynomial syndrome former

$$\mathbf{H}^T(D) = \begin{bmatrix} D^{166} & D^{12} & D^{27} \\ D^{181} & D^{95} & 1 \\ D^{19} & 1 & D^{185} \\ 1 & D^{154} & D^{117} \\ D^{58} & D^{138} & D^{170} \end{bmatrix} \quad (10)$$

of a $(3, 5)$ LDPC convolutional code with much smaller syndrome former memory $m_s = 185$. Yet, the code maintains good cycle properties, i.e., it has girth 12 and 40 (smallest possible number) 12-cycles as shown in Table I in the bottom row.

Fig. 9 presents the decoding performance of a set of $(3, 5)$ time-invariant LDPCccs: the dashed curves correspond to those LDPCccs derived from corresponding QC-LDPC block codes [4] and the solid curve describes the performance of the proposed code in (11) that was found with the optimization algorithm. All codes have a polynomial syndrome former of size 5×3 with no empty entries and each entry being of weight one. The polynomial syndrome former matrices of these $(3, 5)$ codes with syndrome former memories 21, 57, 126, 134 and 204 are defined by $\mathbf{H}_1^T(D)$, $\mathbf{H}_2^T(D)$, $\mathbf{H}_3^T(D)$, $\mathbf{H}_4^T(D)$ and $\mathbf{H}_5^T(D)$, respectively. They are given as

$$\mathbf{H}_1^T(D) = \begin{bmatrix} 1 & 1 & D^{18} \\ D^1 & D^5 & D^{12} \\ D^3 & D^{15} & 1 \\ D^7 & D^4 & D^7 \\ D^{15} & D^{13} & D^{21} \end{bmatrix}, \mathbf{H}_2^T(D) = \begin{bmatrix} 1 & 1 & D^{35} \\ D^8 & D^{43} & D^{45} \\ D^{19} & D^3 & D^{13} \\ D^{57} & D^9 & D^{30} \\ D^{33} & D^2 & 1 \end{bmatrix},$$

$$\mathbf{H}_3^T(D) = \begin{bmatrix} 1 & D^{28} & D^{116} \\ D^7 & D^{101} & D^{36} \\ D^{63} & D^{81} & 1 \\ D^{58} & D^{72} & D^{14} \\ D^{18} & 1 & D^{126} \end{bmatrix}, \mathbf{H}_4^T(D) = \begin{bmatrix} 1 & D^{23} & D^{127} \\ D^{41} & 1 & D^{109} \\ D^{134} & D^{120} & D^{77} \\ D^{58} & D^{92} & 1 \\ D^{124} & D^2 & D^{24} \end{bmatrix}$$

and

$$\mathbf{H}_5^T(D) = \begin{bmatrix} 1 & 1 & D^{171} \\ D^{86} & D^{85} & 1 \\ D^{97} & D^9 & D^{65} \\ D^{90} & D^{145} & D^{177} \\ D^{204} & D^{168} & D^{40} \end{bmatrix}.$$

In Table I, m indicates LDPCccs that were derived from QC LDPC block codes and m_s refers to the syndrome former memories of the LDPCccs with common factors removed from $\mathbf{H}^T(D)$. In terms of cycle properties, the proposed code (gray-shaded in the last row of Table I) is superior to those derived from QC-LDPC block codes. This is also verified by the improvement in the waterfall region as shown in the bit-error rate comparison in Fig. 9. Because of poor free distance, all codes suffer from error floors around 10^{-5} . The simulation was carried out over a binary phase-shift keying (BPSK) modulation AWGN channel with a maximum of 50 iterations

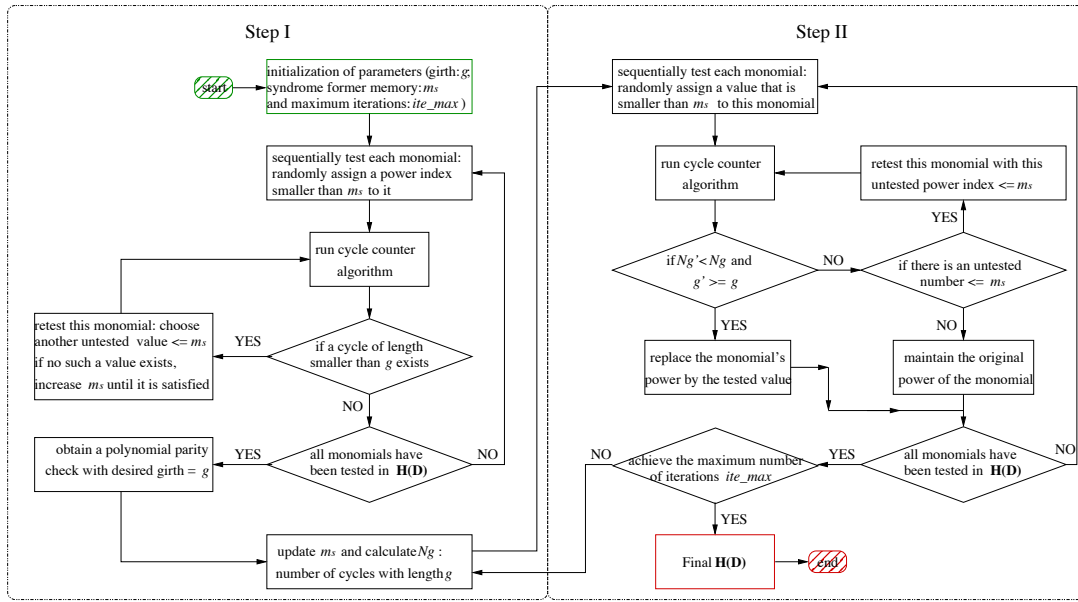


Fig. 8: Algorithm for code design

TABLE I: Cycle properties of (3,5) time-invariant LDPCccs

m/m_s	girth	8-cycles	10-cycles	12-cycles
31/21	8	11	62	351
61/57	10	0	21	148
151/126	10	0	3	55
181/134	12	0	0	67
241/204	12	0	0	52
2311/2005	12	0	0	40
-/185	12	0	0	40

Note: common factors have been removed from $\mathbf{H}^T(D)$.

of the on-demand variable node activation [10] sum product pipeline [11] decoding algorithm for LDPC convolutional codes.

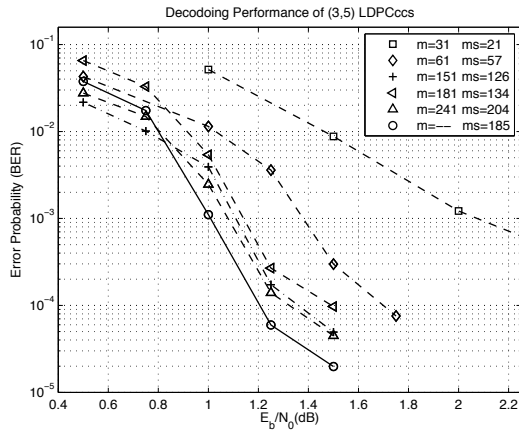


Fig. 9: Decoding performance of (3,5) LDPCccs

VI. CONCLUSION

In this paper, the cycle formation in polynomial syndrome formers of time-invariant LDPC convolutional codes has been discussed. Moreover, we investigated unavoidable cycles with

lengths ranging from 6 to 20 that are caused by polynomial structures in the syndrome former, 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 syndrome former $\mathbf{H}^T(D)$, as this will help to break small cycles and, hence, allows for larger girth. Furthermore, in order to get an LDPC convolutional code with girth larger than 12, some empty entries in polynomial syndrome former are required.

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, Sept. 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, Sept. 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, Dec. 2004.
- [5] Hua Zhou, Norbert Goertz, "Cycle analysis of time-invariant LDPC convolutional codes," *17th International Conference on Telecommunications*, Doha, Qatar, Apr., 2010.
- [6] Hua Zhou, Norbert Goertz, "Unavoidable Cycles in Polynomial-Based Time-Invariant LDPC Convolutional Codes," *European Wireless*, Vienna, Austria, Apr., 2011.
- [7] 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, Jun., 2007.
- [8] Hua Zhou, Norbert Goertz, "Cycle analysis of time-variant LDPC convolutional codes," *6th International Symposium on Turbo Codes & Iterative Information Processing*, Brest, France, Sept., 2010.
- [9] S. Lin and D. J. Costello, "Error Control Coding," Pearson Prentice Hall, 2004
- [10] Ali Emre Pusane, Michael Lentmaier, Kamil Sh. Zigangirov, and Daniel J. Costello Jr., "Reduced complexity decoding strategies for LDPC convolutional codes," *ISIT 2004*, Chicago, USA, Jun., 2004.
- [11] A. Sridharan, "Design and analysis of LDPC convolutional codes," Phd Dissertation, University of Notre Dame, Indiana, USA, Feb. 2005.