A Protograph-Based Design of Quasi-Cyclic Spatially Coupled LDPC Codes

Li Chen †, Shiyuan Mo †, Daniel J. Costello, Jr. ‡, David G. M. Mitchell §, Roxana Smarandache ‡
† School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China
‡ Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA
§ Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM, USA
Emails: chenli55@mail.sysu.edu.cn, moshiy@mail2.sysu.edu.cn, {costello.2, rsmarand}@nd.edu, dgmm@nmsu.edu

***Abstract*—Spatially coupled (SC) low-density parity-check (LDPC) codes can achieve capacity approaching performance with low message recovery latency when using sliding window (SW) decoding. An SC-LDPC code constructed from a protograph can be generated by first *coupling* a chain of block protographs and then *lifting* the coupled protograph using permutation matrices. This paper introduces a systematic design of SC-LDPC codes to eliminate 4-cycles in the coupled protograph. Using a quasi-cyclic (QC) lifting, we obtain QC-SC-LDPC codes of girth at least eight. Coupling a chain of block protographs implies *spreading* edges from one protograph to the others. Our protograph-based design can be viewed as guiding the edge spreading and also the graph-lifting process. Simulation results show the design leads to improved decoding performance, particularly in the error floor, compared to random designs.**

***Index Terms*—Cycles, LDPC codes, protographs, spatially coupled codes, sliding window decoding.**

I. Introduction

Since the original work of Thorpe [1], it has been recognized that protographs provide an efficient method of constructing low-density parity-check (LDPC) codes. Analyzing the iterative decoding thresholds and minimum distance properties of small protographs sheds light on constructing code ensembles with good asymptotic properties by applying a *graph-lifting* procedure [2]. If the permutation matrices used in the lifting procedure are circulants (shifted identity matrices), a quasi-cyclic (QC) ensemble results, a desirable property for practical implementation. Another important aspect of code design is to maximize the girth of the Tanner graph. For protograph-based constructions of QC-LDPC codes, this can be accomplished by applying the Fossorier condition [3] to the graph-lifting. The protograph-based method has also been used to construct good spatially coupled LDPC (SC-LDPC) codes [4]. An *edge-spreading* procedure is first applied to a chain of block protographs in order to introduce memory. This results in a two-step code design procedure, first the edge spreading and then the graph-lifting, to achieve good asymptotic properties and a large girth, respectively.

Several constructions for QC-SC-LDPC codes have been proposed in [5]–[8]. In this paper, we take a new protograph-based systematic design approach to insure large girth for QC-SC-LDPC codes. The idea depends on the fact that the girth of a lifted graph is lower bounded by the girth of its base graph. Hence, starting from a block protograph with good asymptotic properties, we design the edge spreading in two stages to maximize the girth and minimize the number of short cycles in the SC protograph. Then, in the lifting phase, we use circulants to further improve the girth by applying the Fossorier condition. Besides improving our ability to find protographs with large girth and a small number of short cycles, the two-stage approach makes it easier to apply the Fossorier condition, since the SC protograph has already been designed to achieve these objectives.

The edge-spreading procedure can be interpreted as decomposing a base matrix $\mathbf{B}$ (corresponding to a block protograph) into a number of submatrices, which are used to form an SC base matrix $\mathbf{B}_{\text{SC}}$. In our approach, we identify several component blocks of $\mathbf{B}_{\text{SC}}$ that guide the design of the submatrices, leading to an SC protograph with good girth properties. By further performing a graph-lifting of $\mathbf{B}_{\text{SC}}$ using the Fossorier condition to generate an SC parity-check matrix $\mathbf{H}_{\text{SC}}$, we show that it is possible to achieve a girth of at least eight. Simulation results show that substantial performance gains, particularly in the error floor, are achieved using the two-stage design approach compared to random designs.

II. SC-LDPC Codes

The construction of a protograph-based SC-LDPC code can be described as a two-step procedure – first protograph coupling and then lifting [4]. A protograph [1] is a small bipartite graph with n_c check nodes and n_v variable nodes, where $n_c < n_v$. It can be represented by a base matrix

$$\mathbf{B} = [\mathbf{B}(r,s)]_{n_c \times n_v}, \tag{1}$$

where $\mathbf{B}(r,s)$ is the row-r column-s entry, $1 \le r \le n_c$ and $1 \le s \le n_v$. The entries represent the number of edges that connect check node r to variable node s in the protograph. For example, Fig. 1(a) shows a protograph defined by $\mathbf{B} = [3\ 3]$. We first replicate the protograph as an infinite chain as shown in Fig. 1(b), then spread edges from the variable nodes of the protograph at time instant t by connecting them to check nodes at time instants $t+1$ to $t+\omega$. Replicating this spreading over all protographs yields an SC protograph with *coupling width* ω, as shown in Fig. 1(c). This edge spreading can be interpreted as decomposing $\mathbf{B}$ into $\omega + 1$ submatrices of the same size, i.e., $\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_\omega$, such that

$$\mathbf{B}(r,s) = \sum_{i=0}^{\omega} \mathbf{B}_i(r,s), \tag{2}$$

978-1-5090-4096-4/17/$31.00 ©2017 IEEE

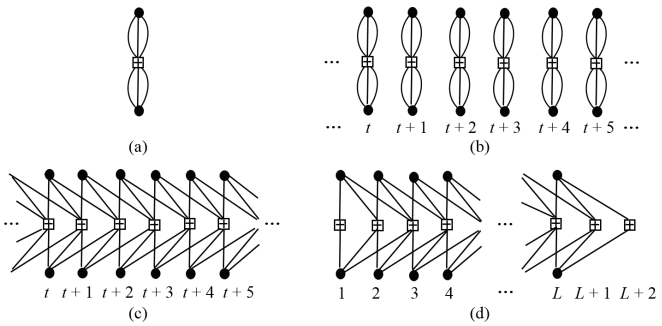

Fig. 1. (a) Block protograph for $\mathbf{B} = [3, 3]$, (b) an infinite chain of block protographs, (c) coupling the protographs with $\omega = 2$, and (d) a finite chain of L coupled protographs.

so the coupled protograph maintains the check node and variable node degrees of the original protograph. If the original protograph has a regular structure that exhibits uniform check node and variable node degrees, as in Fig. 1(a), the constructed SC protograph will also be regular. In practice, an SC-LDPC code has finite length. It can be obtained from a finite number L of coupled block protographs, where L is called the *coupling length.* The coupled protograph contains Ln_v variable nodes and $(L+\omega)n_c$ check nodes, as shown in Fig. 1(d), and the corresponding SC base matrix is

$$\mathbf{B}_{\text{SC}}^{(L)} = \begin{bmatrix} \mathbf{B}_0 & & & & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & & & & \\ \vdots & \vdots & \ddots & & & & \\ \mathbf{B}_\omega & \mathbf{B}_{\omega-1} & \cdots & \mathbf{B}_0 & & & \\ & \ddots & & & \ddots & & \\ & & \mathbf{B}_\omega & \mathbf{B}_{\omega-1} & \cdots & \mathbf{B}_0 \\ & & & \ddots & & \vdots \\ & & & & \mathbf{B}_\omega & \mathbf{B}_{\omega-1} \\ & & & & & \mathbf{B}_\omega \end{bmatrix}. \quad (3)$$

Note that the first and last ωn_c check nodes have reduced degrees, which is an important feature to realize the excellent thresholds of SC-LDPC codes [4], [10].

Let $\mathbf{P}_a$ denote an $a \times a$ binary permutation matrix and $\mathbf{I}_a$ denote the $a \times a$ identity matrix. Furthermore, let $\mathbf{I}_a^{(\theta)}$ denote the shifted identity matrix with each row of $\mathbf{I}_a$ cyclically shifted to the left by θ positions. Finally, let $\mathbf{\Xi}_{a\times b}$ denote an $a \times b$ binary matrix with a minimum row weight of one and with as small a maximum column weight as possible.

The parity-check matrix $\mathbf{H}_{\text{SC}}^{(L)}$ of an SC-LDPC code can be obtained by an M-fold matrix expansion from $\mathbf{B}_{\text{SC}}^{(L)}$ that corresponds to an M-fold graph-lifting of the coupled protograph [9]. In the lifted graph, each check node and variable node is replaced by M copies of the original node and each edge is replaced by M edges connecting M pairs of check and variable nodes. For $\mathbf{B}_{\text{SC}}^{(L)} = [\mathbf{B}_{\text{SC}}^{(L)}(r,s)]_{(L+\omega)n_c \times Ln_v}$, $\mathbf{H}_{\text{SC}}^{(L)}$ is generated by replacing each nonzero entry in $\mathbf{B}_{\text{SC}}^{(L)}$ by a sum of $\mathbf{B}_{\text{SC}}^{(L)}(r,s)$ permutation matrices $\mathbf{P}_M$ and replacing each zero entry by the $M \times M$ all zero matrix. The constraint length and design rate of the code are $Mn_v(\omega+1)$ and $R_{\text{SC}}^{(L)} = 1 - \frac{(L+\omega)n_c}{Ln_v}$, respectively, where $\lim_{L\to\infty} R_{\text{SC}}^{(L)} = 1 - \frac{n_c}{n_v}$. $\mathbf{H}_{\text{SC}}^{(L)}$ defines the Tanner graph of a particular SC-LDPC code. This lifting approach leads to the following characterization for the girth (denoted g) of the Tanner graph.

Lemma 1. The girth of the Tanner graph of $\mathbf{H}_{\text{SC}}^{(L)}$ is lower bounded by girth of the protograph of $\mathbf{B}_{\text{SC}}^{(L)}$.

This lemma motivates the design in Section III.

III. Design of QC-SC-LDPC Codes

Based on Lemma 1, the proposed approach aims to first eliminate (or reduce the number of) 4-cycles in $\mathbf{B}_{\text{SC}}^{(L)}$. Then, using a systematic lifting, we try to construct $\mathbf{H}_{\text{SC}}^{(L)}$ with $g \geq 8$. Due to the diagonal nature of $\mathbf{B}_{\text{SC}}^{(L)}$ (see (3)), a careful examination of its structure is needed in the design.

A. Preliminaries

We consider the common case when the base matrix is all one, i.e., $\mathbf{B} = \mathbf{1}_{n_c \times n_v}$, e.g., $\mathbf{B} = \mathbf{1}_{3\times 6}$. A 4-cycle in a coupled protograph or Tanner graph corresponds to four nonzero entries that form a rectangular array in $\mathbf{B}_{\text{SC}}^{(L)}$ or $\mathbf{H}_{\text{SC}}^{(L)}$, respectively. This leads to the following lemma.

Lemma 2. In $\mathbf{B}_{\text{SC}}^{(L)}$, 4-cycles may exist in the following patterns: 1) one of its submatrices; 2) two submatrices that appear in the same row or the same column of $\mathbf{B}_{\text{SC}}^{(L)}$; 3) four submatrices that appear in a rectangular array of $\mathbf{B}_{\text{SC}}^{(L)}$.

Lemma 2 defines the various patterns in which 4-cycles can appear in $\mathbf{B}_{\text{SC}}^{(L)}$. We now decompose $\mathbf{B}_{\text{SC}}^{(L)}$ as follows:

- The *representative block* $\mathbf{B}_{\text{R}}$ is defined as

$$\mathbf{B}_{\text{R}} \triangleq \begin{bmatrix} \mathbf{B}_\omega & \mathbf{B}_{\omega-1} & \cdots & \mathbf{B}_0 \\ & \mathbf{B}_\omega & \cdots & \mathbf{B}_1 \\ & & \ddots & \vdots \\ & & & \mathbf{B}_\omega \end{bmatrix}, \quad (4)$$

with size $(\omega+1)n_c \times (\omega+1)n_v$. Any combination of one, two, or four submatrices in $\mathbf{B}_{\text{SC}}^{(L)}$ described by conditions 1), 2), or 3) in Lemma 2, respectively, will also appear in $\mathbf{B}_{\text{R}}$. Therefore, if $\mathbf{B}_{\text{R}}$ does not contain 4-cycles, neither will $\mathbf{B}_{\text{SC}}^{(L)}$. The following two definitions are based on $\mathbf{B}_{\text{R}}$.

- A *constituent block* $\mathbf{B}_{\text{C}}$ is defined as

$$\mathbf{B}_{\text{C}} \triangleq \begin{bmatrix} \mathbf{B}_{\beta-1} & \cdots & \mathbf{B}_1 & \mathbf{B}_0 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{B}_\omega & \cdots & \mathbf{B}_\alpha & \mathbf{B}_{\alpha-1} \end{bmatrix}, \quad (5)$$

where $\omega = \alpha + \beta - 2$ and $\alpha, \beta > 1$, with size $\alpha n_c \times \beta n_v$. $\mathbf{B}_{\text{C}}$ is obtained by forming a rectangular matrix from $\mathbf{B}_{\text{R}}$ that contains $\mathbf{B}_0$ in the upper right corner and one of the $\mathbf{B}_\omega$ submatrices along the diagonal (excepting the upper left and lower right corners) in the lower left corner. Hence, there are $\omega - 1$ choices for $\mathbf{B}_{\text{C}}$, and when $\omega = 2$, $\mathbf{B}_{\text{C}}$ is unique. Note that each possible constituent block includes every submatrix. We define the weight $wt(\mathbf{B}_i)$ of a submatrix $\mathbf{B}_i$ as the number of times $\mathbf{B}_i$ is included in a particular $\mathbf{B}_{\text{C}}$.

- *Excluded patterns* $\mathbf{B}_{\text{E}}^{(j)}$ are defined as

$$\mathbf{B}_{\mathrm{E}}^{(1)} = \begin{bmatrix}\mathbf{B}_\omega & \mathbf{B}_0\end{bmatrix},\ \mathbf{B}_{\mathrm{E}}^{(2)} = \begin{bmatrix}\mathbf{B}_0 \\ \mathbf{B}_\omega\end{bmatrix},$$

$$\mathbf{B}_{\mathrm{E}}^{(j)} = \begin{bmatrix}\mathbf{B}_{a_j} & \mathbf{B}_{b_j} \\ \mathbf{B}_{c_j} & \mathbf{B}_{d_j}\end{bmatrix},\ j = 3,4,\ldots,n_{\mathrm{E}}, \tag{6}$$

where $a_j, b_j, c_j, d_j \in \{0,1,\ldots,\omega\}$. Block $\mathbf{B}_{\mathrm{E}}^{(1)}$ (resp. $\mathbf{B}_{\mathrm{E}}^{(2)}$) is the $n_c \times 2n_v$ (resp. $2n_c \times n_v$) single row (resp. column) "pattern" (submatrix) that appears in $\mathbf{B}_{\mathrm{R}}$ but not $\mathbf{B}_{\mathrm{C}}$. Similarly, $\mathbf{B}_{\mathrm{E}}^{(j)}$, $j = 3,4,\ldots,n_{\mathrm{E}}$, are the $2n_c \times 2n_v$ rectangular patterns appearing in $\mathbf{B}_{\mathrm{R}}$ but not $\mathbf{B}_{\mathrm{C}}$. The number of excluded patterns n_{E} depends on ω and the chosen $\mathbf{B}_{\mathrm{C}}$, while the particular set of excluded patterns depends on the chosen $\mathbf{B}_{\mathrm{C}}$. Note that when $\omega = 2$, there are only two excluded patterns $\mathbf{B}_{\mathrm{E}}^{(1)}$ and $\mathbf{B}_{\mathrm{E}}^{(2)}$, since all $2n_c \times 2n_v$ rectangular patterns appear in $\mathbf{B}_{\mathrm{C}}$. The following example illustrates the above definitions.

Example 1. When $\omega = 3$, we have

$$\mathbf{B}_{\mathrm{R}} = \begin{bmatrix}\mathbf{B}_3 & \mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 \\ & \mathbf{B}_3 & \mathbf{B}_2 & \mathbf{B}_1 \\ & & \mathbf{B}_3 & \mathbf{B}_2 \\ & & & \mathbf{B}_3\end{bmatrix}.$$

$\mathbf{B}_{\mathrm{C}}$ can be defined as

$$\mathbf{B}_{\mathrm{C}} = \begin{bmatrix}\mathbf{B}_2 & \mathbf{B}_1 & \mathbf{B}_0 \\ \mathbf{B}_3 & \mathbf{B}_2 & \mathbf{B}_1\end{bmatrix},$$

where $wt(\mathbf{B}_0) = wt(\mathbf{B}_3) = 1$ and $wt(\mathbf{B}_1) = wt(\mathbf{B}_2) = 2$. The excluded patterns are

$$\mathbf{B}_{\mathrm{E}}^{(1)} = \begin{bmatrix}\mathbf{B}_3 & \mathbf{B}_0\end{bmatrix},\ \mathbf{B}_{\mathrm{E}}^{(2)} = \begin{bmatrix}\mathbf{B}_0 \\ \mathbf{B}_3\end{bmatrix},\ \mathbf{B}_{\mathrm{E}}^{(3)} = \begin{bmatrix}\mathbf{B}_1 & \mathbf{B}_0 \\ \mathbf{B}_3 & \mathbf{B}_2\end{bmatrix}.$$

Note that $\mathbf{B}_{\mathrm{C}}$ can also be of size $3n_c \times 2n_v$, which also results in three excluded patterns.

The above definitions lead to the following theorem.

Theorem 3. The coupled protograph of $\mathbf{B}_{\mathrm{SC}}^{(L)}$ does not have any 4-cycles if the chosen $\mathbf{B}_{\mathrm{C}}$ and associated $\mathbf{B}_{\mathrm{E}}^{(j)}, j = 1,2,\ldots,n_{\mathrm{E}}$, do not contain any 4-cycles.

Proof: The result follows directly from Lemma 2. For condition 1), $\mathbf{B}_{\mathrm{C}}$ includes all possible submatrices. For conditions 2) and 3), $\mathbf{B}_{\mathrm{C}}$ and $\mathbf{B}_{\mathrm{E}}^{(j)}$ contain all possible patterns of submatrices that can result in 4-cycles in $\mathbf{B}_{\mathrm{SC}}^{(L)}$. ■

B. Design of $\mathbf{B}_{SC}^{(L)}$ *- Stage 1*

Design Rule 1 Initialize the Submatrices (Stage 1)

1.1: Let $\mathbf{B}_0 = [\mathbf{I}_{n_c} \ \ \Xi_{n_c \times (n_v - n_c)}]$.

1.2: Initialize $\mathbf{B}_\omega$ such that there is no 4-cycle in $\mathbf{B}_{\mathrm{E}}^{(2)}$, the minimum row weight of $\mathbf{B}_\omega$ is two, and the maximum column weight of $\mathbf{B}_\omega$ is one.

1.3: Initialize $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_{\omega-1}$ such that

1) $\mathbf{B}(r,s) = \sum_{i=0}^{\omega} \mathbf{B}_i(r,s)$, i.e., (2) is satisfied;

2) There is no 4-cycle in any of these submatrices or in the excluded patterns $\mathbf{B}_{\mathrm{E}}^{(j)}$ $(j = 3,4,\ldots,n_{\mathrm{E}})$.

Based on Theorem 3, we aim to design the submatrices such that neither the chosen $\mathbf{B}_{\mathrm{C}}$ nor its associated $\mathbf{B}_{\mathrm{E}}^{(j)}$ contain any 4-cycles. The proposed design includes two stages: Stage 1 initializes the submatrices based on $\mathbf{B}_{\mathrm{E}}^{(j)}$; Stage 2 modifies the submatrices based on $\mathbf{B}_{\mathrm{C}}$.

Given a base matrix $\mathbf{B} = \mathbf{1}_{n_c \times n_v}$ and coupling width ω, the Stage 1 design is given in Design Rule 1. It insures the submatrices and the excluded patterns do not contain any 4-cycles. Step **1.1** insures the minimum row weight of $\mathbf{B}_0$ is two and its maximum column weight is one if $n_v - n_c \geq n_c$. If $n_v - n_c < n_c$, some columns of $\mathbf{B}_0$ will have weight greater than one. The design of $\mathbf{B}_\omega$ in Step **1.2** also insures that $\mathbf{B}_{\mathrm{E}}^{(1)}$ does not contain any 4-cycles. Requiring $\mathbf{B}_0$ and $\mathbf{B}_\omega$ to have a minimum row weight of two is due to the fact that they are the only submatrices in the top and bottom rows of $\mathbf{B}_{\mathrm{SC}}^{(L)}$, respectively, and a row weight of at least two is needed to assist the startup and termination of sliding window (SW) decoding [10], [11]. Also, if possible, restricting the maximum column weight of $\mathbf{B}_\omega$ to one simplifies the design of Step **1.3** for the cases where $\mathbf{B}_\omega$ is included in $\mathbf{B}_{\mathrm{E}}^{(j)}(j = 3,4,\ldots,n_{\mathrm{E}})$. The remaining submatrices $\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_{\omega-1}$ are then initialized based on $\mathbf{B}_{\mathrm{E}}^{(j)}(j = 3,4,\ldots,n_{\mathrm{E}})$, $\mathbf{B}_0$, and $\mathbf{B}_\omega$. The following example illustrates the procedure.

Example 2. Given $\mathbf{B} = \mathbf{1}_{3\times 6}$ and $\omega = 3$, $\mathbf{B}_{\mathrm{R}}$, $\mathbf{B}_{\mathrm{C}}$, $\mathbf{B}_{\mathrm{E}}^{(1)}$, $\mathbf{B}_{\mathrm{E}}^{(2)}$, and $\mathbf{B}_{\mathrm{E}}^{(3)}$ are given in *Example 1.* We can employ Design Rule 1 to initialize the submatrices. First, we can let

$$\mathbf{B}_0 = [\mathbf{I}_3 \ \ \Xi_{3\times 3}] = \begin{bmatrix}1&0&0&0&1&0\\0&1&0&1&0&0\\0&0&1&0&0&1\end{bmatrix}.$$

Placing $\mathbf{B}_0$ into $\mathbf{B}_{\mathrm{E}}^{(2)}$, we can initialize

$$\mathbf{B}_3 = \begin{bmatrix}0&1&1&0&0&0\\1&0&0&0&0&1\\0&0&0&1&1&0\end{bmatrix}$$

so that neither $\mathbf{B}_{\mathrm{E}}^{(1)}$ nor $\mathbf{B}_{\mathrm{E}}^{(2)}$ contain any 4-cycles. In order to initialize $\mathbf{B}_1$ and $\mathbf{B}_2$, we place both $\mathbf{B}_0$ and $\mathbf{B}_3$ into $\mathbf{B}_{\mathrm{E}}^{(3)}$. To satisfy (2), we must place zeros into certain positions in $\mathbf{B}_1$ and $\mathbf{B}_2$, leaving 12 unspecified positions in $\mathbf{B}_{\mathrm{E}}^{(3)}$. We can then initialize $\mathbf{B}_1$ and $\mathbf{B}_2$ as

$$\mathbf{B}_1 = \begin{bmatrix}0&0&0&1&0&1\\0&0&1&0&1&0\\1&1&0&0&0&0\end{bmatrix},\ \mathbf{B}_2 = \begin{bmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{bmatrix},$$

where this choice insures $\mathbf{B}_{\mathrm{E}}^{(3)}$ does not contain any 4-cycles.

Finally, we place the initialized submatrices into $\mathbf{B}_{\mathrm{C}}$ to check if it contains any 4-cycles. If not, the coupled protograph of $\mathbf{B}_{\mathrm{SC}}^{(L)}$ has $g \geq 6$, and the design is complete. Otherwise, we proceed to Stage 2 to eliminate (or reduce the number of) 4-cycles that remain in $\mathbf{B}_{\mathrm{C}}$.

C. Design of $\mathbf{B}_{SC}^{(L)}$ *- Stage 2*

This stage modifies the initialized submatrices to remove the remaining 4-cycles in $\mathbf{B}_{\mathrm{C}}$. In order to distinguish between an entry in $\mathbf{B}_{\mathrm{C}}$ and one in $\mathbf{B}_i$, we use $\mathbf{B}_{\mathrm{C}}(x,y)$ to denote its row-x column-y entry, for $1 \leq x \leq \alpha n_c$ and $1 \leq y \leq \beta n_v$. The Stage 2 design is given in Design Rule 2.

Design Rule 2 Modify the submatrices (Stage 2)

2.1: Identify a 4-cycle in $\mathbf{B}_\mathrm{C}$ with entries

$$\mathbf{B}_\mathrm{C}(x_1,y_1)=1,\ \mathbf{B}_\mathrm{C}(x_1,y_2)=1,$$
$$\mathbf{B}_\mathrm{C}(x_2,y_1)=1,\ \mathbf{B}_\mathrm{C}(x_2,y_2)=1.$$

2.2: Say the four entries belong to submatrices $\mathbf{B}_{i_1}$, $\mathbf{B}_{i_2}$, $\mathbf{B}_{i_3}$, and $\mathbf{B}_{i_4}$, where $(i_1,i_2,i_3,i_4)\in\{0,1,\ldots,\omega\}$. Denote these entries as

$$\mathbf{B}_{i_1}(r_1,s_1)=1,\ \mathbf{B}_{i_2}(r_1,s_2)=1,$$
$$\mathbf{B}_{i_3}(r_2,s_1)=1,\ \mathbf{B}_{i_4}(r_2,s_2)=1.$$

2.3: Among these four entries, identify those that have not previously been *flipped*. Pick one that belongs to a submatrix of highest weight and denote it $\mathbf{B}_{i'}(r',s')$, where $i'\in\{i_1,i_2,i_3,i_4\}$ and $(r',s')\in\{(r_1,s_1),(r_1,s_2),(r_2,s_1),(r_2,s_2)\}$. Further, denote the appearance of $\mathbf{B}_{i'}(r',s')$ in $\mathbf{B}_\mathrm{C}$ as $\mathbf{B}_\mathrm{C}(x',y')$, where $(x',y')\in\{(x_1,y_1),(x_1,y_2),(x_2,y_1),(x_2,y_2)\}$. *Flip down* both $\mathbf{B}_{i'}(r',s')$ and $\mathbf{B}_\mathrm{C}(x',y')$ such that

$$\mathbf{B}_{i'}(r',s'):\ 1\ \rightarrow\ 0, \tag{7}$$
$$\mathbf{B}_\mathrm{C}(x',y'):\ 1\ \rightarrow\ 0. \tag{8}$$

Also flip down all other entries in $\mathbf{B}_\mathrm{E}^{(j)}(j=1,2,\ldots,n_\mathrm{E})$ and $\mathbf{B}_\mathrm{C}$ that correspond to entry $\mathbf{B}_{i'}(r',s')$.

2.4: *Flip up* $\mathbf{B}_i(r',s')$ in one of the other submatrices

$$\mathbf{B}_i(r',s'):\ 0\ \rightarrow\ 1, \tag{9}$$

where $i=0,1,\ldots,\omega$ and $i\neq i'$, conditioned on

1) The entry has not been previously flipped;
2) The flipping does not create new 4-cycles in $\mathbf{B}_i$ or in any $\mathbf{B}_\mathrm{E}^{(j)}$ that includes $\mathbf{B}_i$;
3) The number of 4-cycles contained in $\mathbf{B}_\mathrm{C}$ does not increase after a complete flipping (down and up) process.

Also flip up all other entries in $\mathbf{B}_\mathrm{E}^{(j)}(j=1,2,\ldots,n_\mathrm{E})$ and $\mathbf{B}_\mathrm{C}$ that correspond to entry $\mathbf{B}_i(r',s')$.

2.5: If the flipping of Step **2.4** succeeds, go to Step **2.6**; else, reflip $\mathbf{B}_\mathrm{C}(x',y')$ and $\mathbf{B}_{i'}(r',s')$ to their original values, i.e.,

$$\mathbf{B}_\mathrm{C}(x',y'):\ 0\ \rightarrow\ 1, \tag{10}$$
$$\mathbf{B}_{i'}(r',s'):\ 0\ \rightarrow\ 1. \tag{11}$$

Also reflip all other entries in $\mathbf{B}_\mathrm{E}^{(j)}(j=1,2,\ldots,n_\mathrm{E})$ and $\mathbf{B}_\mathrm{C}$ that correspond to entry $\mathbf{B}_{i'}(r',s')$, and go to Step **2.3**.

2.6: Repeat Steps **2.1** to **2.4** until all 4-cycles are removed or there are no more eligible bits to flip.

In Step **2.2**, i_1, i_2, i_3 and i_4 do not need to be distinct. In Step **2.3**, we prioritize the "flipping down" of a nonzero entry of an identified submatrix that has the highest weight in $\mathbf{B}_\mathrm{C}$. In doing so, we remove the most nonzero entries in $\mathbf{B}_\mathrm{C}$, so that more 4-cycles are likely to be removed. However, it is possible that none of the four entries allows a complete flipping, in which case the remaining 4-cycle is labelled *dormant*. It will be targeted again if some other complete flipping occurs and this *dormant* 4-cycle still exists. But for small coupling widths ω, it may not be possible to eliminate all 4-cycles in $\mathbf{B}_\mathrm{C}$. Intuitively, the above design can be seen as spreading the $n_c n_v$ ones in $\mathbf{B}$ over the $\omega+1$ submatrices in such a way that 4-cycles are eliminated. With a larger ω, it is easier to design a $\mathbf{B}_\mathrm{R}$ that does not contain any 4-cycles. For a particular set of initial submatrices $\mathbf{B}_0,\mathbf{B}_1,\ldots,\mathbf{B}_\omega$ designed in Stage 1, if the Stage 2 design does not eliminate 4-cycles in $\mathbf{B}_\mathrm{C}$, or if it results in a $\mathbf{B}_0$ and/or $\mathbf{B}_\omega$ with minimum row weight less than two, the design can be repeated with a different set of initial submatrices. Although for $\omega>2$, there are multiple choices for $\mathbf{B}_\mathrm{C}$, each of which is associated with a different set of excluded patterns, our experience has shown that the choice of $\mathbf{B}_\mathrm{C}$ does not affect whether or not 4-cycles can be eliminated, but a different set of initial submatrices may help.

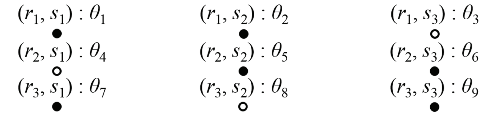

Fig. 2. A 3×3 grid of nonzero entries corresponding to a 6-cycle in $\mathbf{B}_\mathrm{SC}^{(L)}$, where $r_1<r_2<r_3$ and $s_1<s_2<s_3$.

D. QC Lifting Based on $\mathbf{B}_{SC}^{(L)}$

Given a designed $\mathbf{B}_\mathrm{SC}^{(L)}$, we can employ a systematic lifting using circulants in an attempt to further increase the girth. In this paper, we pay particular attention to the removal of all 6-cycles (and any remaining 4-cycles) so that the resulting $\mathbf{H}_\mathrm{SC}^{(L)}$ is QC and has $g\geq 8$. Note that any 6-cycle can be represented by a 3×3 grid of nonzero entries in $\mathbf{B}_\mathrm{SC}^{(L)}$, as shown in Fig. 2. To remove a $2k$-cycle, $k=2,3,\ldots$, circulants can be chosen according to the Fossorier condition (Theorem 2.1 of [3]). For example, if the six nonzero entries that constitute a 6-cycle in Fig. 2 (indicated by the solid circles) are lifted with different circulants $\mathbf{I}_M^{(\theta)}$, where the shifting factors satisfy

$$(\theta_1-\theta_7)\neq(\theta_2-\theta_5)+(\theta_6-\theta_9)\mod M, \tag{12}$$

then there are no 6-cycles in the lifted subgraph associated with this grid. In this case, we say that the cycle in the protograph is "removed". There are another five possible configurations of 6-cycles, and similar constraints on their shifting factors can be employed to remove the cycles.

To proceed, we first identify all the nonzero entries that result in cycles we wish to remove. (The remaining nonzero entries can be lifted by randomly generated circulants.) Then, the identified cycles are removed sequentially by selecting circulants according to the the Fossorier condition. However, since nonzero entries in the protograph often participate in multiple cycles, care is needed to make sure that cycle removal does not create cycles elsewhere in the graph. In our design, the entries and shifting factors are recorded after removing a cycle. When the next cycle is targeted, we first check to see if any of its nonzero entries have been previously lifted. If so, they are not changed, and the shifting factors of the other nonzero entries are chosen such that the Fossorier condition is satisfied (if possible). For sufficiently large M, our study has found that there is enough freedom to choose the shifting factors to construct QC-SC-LDPC codes with $g\geq 8$.

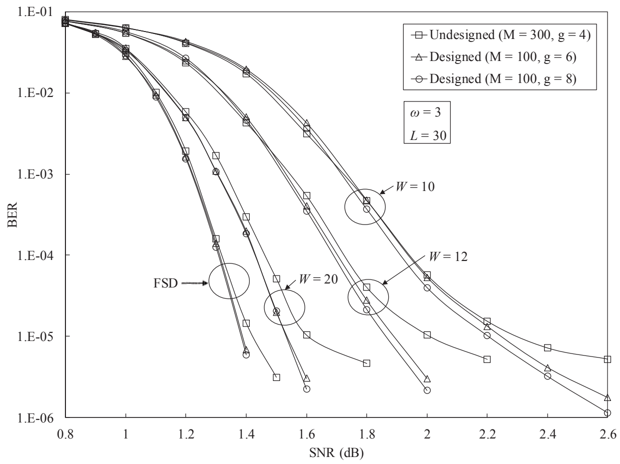

Fig. 3. Performance of SC-LDPC codes with $(n_c, n_v) = (3, 6)$.

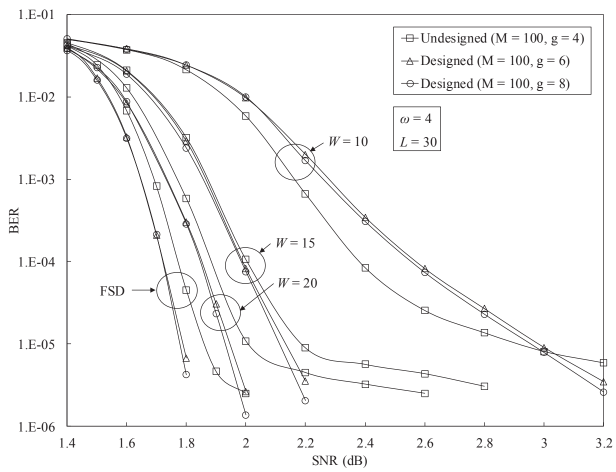

Fig. 4. Performance of SC-LDPC codes with $(n_c, n_v) = (3, 8)$.

IV. Simulation Results

We compare the performance of our designed SC-LDPC codes and undesigned (randomly chosen) codes. The designed codes have girth six and eight, obtained by a random lifting and a QC lifting, respectively, so the designed codes with $g = 8$ are QC-SC-LDPC codes. Simulations were performed over the AWGN channel using BPSK modulation. SW decoding [10], [11] with different window sizes W was used, the maximum iteration number per window position was 100, and the soft bit error rate (BER) stopping rule threshold was 1×10^{-6}. When $W = L$, the decoding window covers the entire Tanner graph of the code and SW decoding is equivalent to standard flooding schedule decoding (FSD). The FSD was simulated with a maximum iteration number of 1000.

Fig. 3 shows the performance of designed and undesigned SC-LDPC codes with $(n_c, n_v) = (3, 6)$ and $\omega = 3$. The undesigned code was constructed from the base matrix $\mathbf{B} = [3\ 3]$ with $\mathbf{B}_0 = [1\ 1]$, $\mathbf{B}_1 = [1\ 0]$, $\mathbf{B}_2 = [0\ 1]$, and $\mathbf{B}_3 = [1\ 1]$, and $M = 300$ and 100 for the undesigned and designed codes, respectively, so they have the same overall frame length. Base matrix and associated submatrices of the designed codes are given in *Example 2*. $L = 30$ and $R_{\text{SC}}^{(L)} = 0.45$ in all cases. The results show that the designed codes substantially outperform the undesigned code, particularly in the error floor, and that performing a QC lifting further improves the performance. Fig. 4 shows the performance of designed and undesigned SC-LDPC codes with $(n_c, n_v) = (3, 8)$ and $\omega = 4$. For the undesigned code, the 3×8 binary submatrices were chosen such that $\mathbf{B}_0$ and $\mathbf{B}_4$ have a minimum row weight of two, while $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{B}_3$ were chosen randomly to satisfy (2), and in this case $R_{\text{SC}}^{(L)} = 0.575$. Again, the results show the designed codes outperform the undesigned code. When $W = 10$, the designed codes outperform only in the error floor. This is because the window covers only two constraint lengths, which is too small for SW decoding to be effective.

V. Conclusions

This paper presented a protograph-based systematic design of QC-SC-LDPC codes, resulting in a girth $g \geq 8$ for the lifted codes. The SC base matrix is decomposed into a set of blocks to guide the design of edge spreading. Excluded patterns were utilized to initialize the submatrices, while the constituent block was utilized to modify them. Unless the coupling width ω is too small, an SC base matrix with $g \geq 6$ can be obtained. For large enough lifting factor M, and a systematic QC lifting, an SC parity-check matrix with $g \geq 8$ can be obtained. Simulation results show our design leads to significantly improved performance, particularly in the error floor.

Acknowledgment

This is supported by the National Natural Science Foundation of China (NSFC) under grants 61372079 and 61671486.

References

[1] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," Jet Propulsion Laboratory, Pasadena, CA, INP Progress Report 42-154, Aug. 2003.

[2] D. Divsalar, S. Dolinar. C. Jones and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876-888, Aug. 2009.

[3] M. 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.

[4] D. Mitchell, M. Lentmaier and D. Costello, Jr., "Spatically coupled LDPC codes constructed from protographs," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866-4889, Sept. 2015.

[5] J. Li, S. Lin, K. Abdel-Ghaffar, W. Ryan and D. Costello, Jr., *LDPC Code Designs, Constructions and Unification*, Cambridge Press, 2016.

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

[7] M. Zhang, Z. Wang, Q. Huang and S. Wang, "Time-invariant quasi-cyclic spatially coupled LDPC codes based on packings," *IEEE Trans. Commun.*, vol. 64, no. 12, pp. 4936-4945, Dec. 2016.

[8] J. Cho and L. Schmalen, "Construction of protographs for large-girth structured LDPC convolutional codes," *Proc. IEEE ICC*, London, UK, Jun. 2015.

[9] A. Pusane, R. Smarandache, P. Vontobel and D. Costello, Jr., "Deriving good LDPC convolutional codes from LDPC block codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 835-857, Feb. 2011.

[10] M. Lentmaier, A. Sridharan, D. Costello, Jr. and K. Sh. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274-5289, Oct. 2010.

[11] A. Iyengar, M. Papaleo, P. Siegel, J. Wolf, A. Vanelli-Coralli and G. 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.