Semi-LDPC Convolutional Codes: Construction and Low-Latency Windowed List Decoding

Qianfan Wang, Suihua Cai, Li Chen, Xiao Ma

***Abstract*—This paper presents a new coding scheme called semi-low-density parity-check convolutional code (semi-LDPC-CC), whose parity-check matrix consists of both sparse and dense sub-matrices, a feature distinguished from the conventional LDPC-CCs. We propose sliding-window list (SWL) decoding algorithms with a fixed window size of two, resulting in a low decoding latency but a competitive error-correcting performance. The performance can be predicted by upper bounds derived from the first event error probability and by genie-aided (GA) lower bounds estimated from the underlying LDPC block codes (LDPC-BCs), while the complexity can be reduced by truncating the list with a threshold on the difference between the soft metrics in the serial decoding implementation. Numerical results are presented to validate our analysis and demonstrate the performance advantage of the semi-LDPC-CCs over the conventional LDPC-CCs.**

***Keywords*—low-density parity-check convolutional codes (LDPC-CCs), spatially coupled LDPC (SC-LDPC) codes, sliding-window list (SWL) decoding**

I. Introduction

Low-density parity-check convolutional codes (LDPC-CCs), also known as spatially coupled LDPC (SC-LDPC) codes, were first introduced in Ref. [1], where the parity-check matrix of the LDPC-CC is constructed from the parity-check matrix of the LDPC block code (LDPC-BC) by a matrix-based unwrapping procedure. In Ref. [2], the construction exploits similarities between quasi-cyclic block codes and time-invariant convolutional codes. These aforementioned two constructions were shown to be tightly connected via (proto)graph-cover construction in Ref. [3] and the protograph-based construction was subsequently investigated in Ref. [4], in which L disjoint, or uncoupled LDPC-BCs of length n are coupled into a single chain. Recently, several constructions of the LDPC-CCs have been proposed, e.g., a systematic protograph-based construction with a girth of eight[5], the replicate-and-mask construction with improved performance[6], and a "hardware-reusable" construction via partial superposition[7].

As a result of these careful constructions, there are several remarkable features of LDPC-CCs. One remarkable feature is that the LDPC-CCs have asymptotically capacity-achieving performance over binary memoryless symmetric (BMS) channels under the iterative belief propagation (BP) decoding[8]. This is due to the threshold saturation, discovered numerically in Ref. [9] and established analytically in Ref. [10], where the BP decoding performance of LDPC-CCs can approach the maximum a posteriori (MAP) decoding performance of the underlying LDPC-BCs. However, to obtain the promised good error-correcting performance, one needs sufficiently large length n and coupling length L[11]. This in turn leads to a large decoding latency. Another remarkable feature is that the LDPC-CCs can be decoded by a sliding-window (SW) decoder[9,11], in which the trade-off between the error-correcting performance and the decoding latency can be achieved with a tunable window size. In this case, to obtain good error-correcting performance, a sufficiently large window size is usually required. In Ref. [11], the authors proved that the window thresholds can approach the BP thresholds if the window size d exceeds some designated window size d_{min}. This has been experimentally verified in Ref. [12] that near optimal performance can be maintained in the high signal-to-noise ratio (SNR) region as long as $d \geqslant 6(m+1)$, where m is the coupling width, also referred to as encoding memory. This large window size usually implies a large decoding latency.

In some scenarios, such as the ultra-reliable and low-

Manuscript received May 31, 2021; revised Sep. 09, 2021; accepted Sep. 28, 2021. This work was supported by the National Key R&D Program of China under Grant 2020YFB1807100, the NSF of China under Grant 61971454 and Grant 62071498 and Guangdong Basic and Applied Basic Research Foundation under Grant 2020A1515010687. The associate editor coordinating the review of this paper and approving it for publication was L. Bai.

Q. F. Wang. School of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: wangqf6@mail2.sysu.edu.cn).

S. H. Cai, X. Ma. School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China. Guangdong Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: caish23@mail.sysu.edu.cn; maxiao@mail.sysu.edu.cn).

L. Chen. School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: chenli55@mail.sysu.edu.cn).

latency communication (URLLC) scenario of 5G, both latency and reliability are the key performance indicators to support new real-time applications, like virtual reality and tactile Internet[13]. In particular, the authors in Ref. [13] mentioned that the LDPC-CC is a potential solution in physical layer to realize the real-time applications. In Ref. [14], a short-length semi-LDPC-CC with the low-latency decoding algorithm was presented for these real-time applications.

In this paper, inspired by Ref. [14], we present a new construction of the semi-LDPC-CCs. Distinguished from the conventional LDPC-CCs, the parity-check matrices of semi-LDPC-CCs contain dense sub-matrices. Because of this special structure, we present sliding-window list (SWL) decoding algorithms, which integrate the ideas of list decoding[15,16] and the conventional SW decoding[9,11]. The SWL algorithm has a fixed window size of two or three and hence results in a low decoding latency. Distinguished from Ref. [14], we have made the following new contributions in this paper.

- As for construction, we present to use the progressive-edge-growth (PEG) algorithm[17] to construct the sparse sub-matrices with a systematic structure.
- As for decoding, we present the low-complexity re-encoding list decoding algorithm, which is embedded in the windowed decoding algorithm.
- As for performance analysis, we present upper and genie-aided (GA) lower bounds. The upper bounds are derived from the performance of the first sub-frame, while the GA lower bounds are estimated from the performance of the underlying LDPC-BCs.
- To reduce the complexity, we present a serial implementation with early stopping criterion for the proposed SWL decoding algorithms.
- Numerical results show that semi-LDPC-CCs with the SWL decoding algorithm can yield an extra coding gain of up to 1.4 dB over the conventional LDPC-CCs with the SW decoding algorithm under the same decoding latency. Numerical results also exhibit the effectiveness of the serial implementation and validate our analysis of the performance bounds.

The rest of this paper is organized as follows. In section II, we present the algebraic description and the encoding algorithm of semi-LDPC-CCs. In section III, we present the decoding algorithms, the performance bounds, the serial implementation, and the complexity analysis. Numerical results are presented in section IV. Section V concludes this paper.

II. Semi-LDPC Convolutional Codes

A. Preliminaries

In this paper, we use an upper-case letter, say X, to denote a random variable. The realization of the random variable is denoted by the corresponding lower-case letter $x \in \mathscr{X}$. We use $P_X(x), x \in \mathscr{X}$ (or simply $P(x)$) to denote the probability density function (PDF) of a continuous random variable or the probability mass function (PMF) of a discrete random variable.

Let $\mathbb{F}_2 = \{0,1\}$ be the binary field. Let $\boldsymbol{u} = (\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)}, \cdots, \boldsymbol{u}^{(L-1)})$ be L blocks of information sequences to be transmitted, where $\boldsymbol{u}^{(t)} = (u_0^{(t)}, u_1^{(t)}, \cdots, u_{k-1}^{(t)}) \in \mathbb{F}_2^k$ for $0 \leqslant t \leqslant L-1$. At time slot t, the encoder takes as input the information sequence $\boldsymbol{u}^{(t)}$ and delivers as output the coded sequence $\boldsymbol{c}^{(t)} \in \mathbb{F}_2^n$. Assume that $\boldsymbol{c}^{(t)}$ is modulated and transmitted over the additive white Gaussian noise (AWGN) channels, resulting in a received vector $\boldsymbol{y}^{(t)}$. Because of the windowed decoding mechanism with window size d, the decoder delivers an estimation $\hat{\boldsymbol{u}}^{(t)}$ of $\boldsymbol{u}^{(t)}$ upon receiving $(\boldsymbol{y}^{(t)}, \boldsymbol{y}^{(t+1)}, \cdots, \boldsymbol{y}^{(t+d-1)})$.

B. Algebraic Description of Semi-LDPC-CCs

A terminated semi-LDPC-CC with encoding memory $m = 1$ can be treated as a linear block code with dimension kL and length $n(L+1)$, whose parity-check matrix is a banded block matrix,

$$\boldsymbol{H}_{semi\text{-LDPC-CC}} = \begin{bmatrix} \boldsymbol{H}_0 & & & & \\ \boldsymbol{R} & \boldsymbol{H}_0 & & & \\ & \boldsymbol{R} & \boldsymbol{H}_0 & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{R} & \boldsymbol{H}_0 \end{bmatrix},$$

where $\boldsymbol{H}_0$ is a sparse matrix of size $(n-k) \times n$ and $\boldsymbol{R}$ is a dense matrix of size $(n-k) \times n$. For the sparse sub-matrix, to simplify the encoding process, we consider in this paper only the form of $\boldsymbol{H}_0 = [\boldsymbol{P}, \boldsymbol{I}]$. By optimization with computer search, the row weight of $\boldsymbol{P}$ is $w_r = 2$ and column weight of $\boldsymbol{P}$ is $w_c = 2$ in this paper. This sparse sub-matrix can be constructed by the PEG algorithm[17]. The main difference from the conventional LDPC-CCs lies in the dense sub-matrix $\boldsymbol{R}$, which can be randomly generated (but fixed) by sampling from a Bernoulli process with success probability $1/2$. By this construction, roughly one-half elements of the dense sub-matrix $\boldsymbol{R}$ are nonzero.

C. Encoding Algorithm

Given $\boldsymbol{u} = (\boldsymbol{u}^{(0)}, \boldsymbol{u}^{(1)}, \cdots, \boldsymbol{u}^{(L-1)})$ with $\boldsymbol{u}^{(t)} = (u_0^{(t)}, u_1^{(t)}, \cdots, u_{k-1}^{(t)}) \in \mathbb{F}_2^k$, at time slot t, the encoder takes as input the information sequence $\boldsymbol{u}^{(t)}$ and delivers as output the coded sequence of the form $\boldsymbol{c}^{(t)} = (\boldsymbol{u}^{(t)}, \boldsymbol{v}^{(t)}) \in \mathbb{F}_2^n$, where $\boldsymbol{v}^{(t)} \in \mathbb{F}_2^{n-k}$ denotes the parity-check bits.

Similar to Ref. [18], the encoding structure of a semi-LDPC-CC with $m = 1$ is shown in Fig. 1, in which the register $\boldsymbol{D}$ of length n is initialized to $\boldsymbol{0}$. Given the information sequences $\boldsymbol{u}$ as the input, the encoding algorithm has the same

Algorithm 1 Encoding of the semi-LDPC-CCs

1. **Initialization**: Set $\boldsymbol{c}^{(-1)} = \boldsymbol{0} \in \mathbb{F}_2^n$.
2. **Recursion**: For $t = 0, 1, \cdots, L-1$,
   (a) Compute $\boldsymbol{z}^{(t)} = \boldsymbol{u}^{(t)}\boldsymbol{P}^{\mathrm{T}} \in \mathbb{F}_2^{n-k}$, where $\boldsymbol{P}$ is the sub-matrix of $\boldsymbol{H}_0 = [\boldsymbol{P}, \boldsymbol{I}]$.
   (b) Compute $\boldsymbol{s}^{(t-1)} = \boldsymbol{c}^{(t-1)}\boldsymbol{R}^{\mathrm{T}} \in \mathbb{F}_2^{n-k}$, where $\boldsymbol{R}$ is the dense matrix.
   (c) Compute $\boldsymbol{v}^{(t)} = \boldsymbol{z}^{(t)} + \boldsymbol{s}^{(t-1)} \in \mathbb{F}_2^{n-k}$, resulting in $\boldsymbol{c}^{(t)} = (\boldsymbol{u}^{(t)}, \boldsymbol{v}^{(t)}) \in \mathbb{F}_2^n$, which is taken as the tth sub-frame of transmission.
3. **Termination**: Set $\boldsymbol{u}^{(L)} = \boldsymbol{0} \in \mathbb{F}_2^k$ in Step Recursion to obtain the Lth sub-frame $\boldsymbol{c}^{(L)} = (\boldsymbol{0},\ \boldsymbol{c}^{(L-1)}\boldsymbol{R}^{\mathrm{T}}) \in \mathbb{F}_2^n$ for transmission.

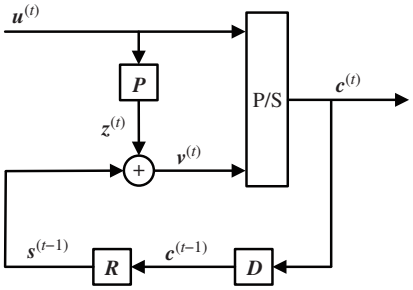

Fig. 1 Encoding structure of a semi-LDPC-CC with $m = 1$. Here, $\boldsymbol{P}$ is the sparse matrix, $\boldsymbol{R}$ is the dense matrix and $\boldsymbol{D}$ is the register. The tth sub-frame is $\boldsymbol{c}^{(t)} = (\boldsymbol{u}^{(t)}, \boldsymbol{v}^{(t)})$, where $\boldsymbol{u}^{(t)}$ is the input information bits at time slot t and $\boldsymbol{v}^{(t)}$ is the parity-check bits calculated by $\boldsymbol{v}^{(t)} = \boldsymbol{u}^{(t)}\boldsymbol{P}^{\mathrm{T}} + \boldsymbol{c}^{(t-1)}\boldsymbol{R}^{\mathrm{T}}$

framework as Ref. [14] and is described in Algorithm 1 for completeness, see Fig. 1 for illustration.

The real code rate[1] of the semi-LDPC-CCs with $m = 1$ is $\frac{kL}{n(L+1)} \approx k/n$ for large L. Similar to conventional LDPC-CCs, semi-LDPC-CCs also inherit streaming properties. That is, the encoding can be executed in a streaming manner without waiting for the whole block of data, while the decoding can be implemented by a sliding-window algorithm. The trade-off between the performance and the latency can be achieved by tuning the window size d, corresponding to a decoding latency dn in terms of bits.

III. Decoding Algorithms and Performance Analysis

A. Sliding-Window List Decoding Algorithms

For simplicity, we assume that $\boldsymbol{c}^{(t)} \in \mathbb{F}_2^n$ is modulated using binary phase-shift keying (BPSK) signaling and transmitted over an AWGN channel, resulting in a received vector $\boldsymbol{y}^{(t)} \in \mathbb{R}^n$ at the receiver. Owing to the existence of the dense sub-matrix, the conventional SW decoding algorithm is not applicable here. Instead, we propose the SWL decoding algorithms for the semi-LDPC-CCs, in which the list decoding algorithms are performed after a BP decoding algorithm performed over the partial Tanner graph corresponding to the sparse sub-matrix. The decoding window size is fixed to two or three to achieve a low-latency decoding. In this subsection, we focus on the SWL with window size of two for simplicity. That is, for each time slot t, we recover $\boldsymbol{u}^{(t)}$ from $(\boldsymbol{y}^{(t)}, \boldsymbol{y}^{(t+1)})$. The details are presented as follows and can be adapted to the window size of three.

1) BP Decoding: The log-likelihood ratios (LLRs) $\boldsymbol{\Lambda}^{(t)}$ associated with $\boldsymbol{c}^{(t)}$ are defined as

$$\Lambda_i^{(t)} = \ln \frac{P_{Y|C}(y_i^{(t)}|0)}{P_{Y|C}(y_i^{(t)}|1)}, \text{ for } 0 \leqslant i \leqslant n-1. \tag{1}$$

Taking $\boldsymbol{\Lambda}^{(t)}$ as input, if $\boldsymbol{c}^{(t-1)}$ is known, we can perform an iterative BP algorithm over a partial Tanner graph specified by the constraint as

$$\boldsymbol{c}^{(t)}\boldsymbol{H}_0^{\mathrm{T}} = \boldsymbol{c}^{(t-1)}\boldsymbol{R}^{\mathrm{T}}, \tag{2}$$

to obtain a temporary estimate $\hat{\boldsymbol{c}}^{(t)}$ and the a posteriori LLRs associated with $\boldsymbol{c}^{(t)}$. This is almost the same as the iterative BP algorithm, i.e., sum-product algorithm (SPA) of a conventional LDPC-BC except that the constraint is given by (2) instead of $\boldsymbol{c}^{(t)}\boldsymbol{H}_0^{\mathrm{T}} = \boldsymbol{0}$. In practice, $\boldsymbol{c}^{(t-1)}$ of (2) is replaced by its estimate $\hat{\boldsymbol{c}}^{(t-1)}$.

2) List Decoding: Given $\hat{\boldsymbol{c}}^{(t)}$ and the a posteriori LLRs associated with $\boldsymbol{c}^{(t)}$, a list of candidate codewords for $\boldsymbol{c}^{(t)}$, denoted as $\hat{\boldsymbol{c}}^{(t,\ell)} (1 \leqslant \ell \leqslant \ell_{max})$ can be obtained, where ℓ_{max} denotes the maximum list size. In this paper, we consider two different list decoding algorithms with different complexities. The first one is the re-encoding list (REL) algorithm, in which the temporary estimated information bits of $\hat{\boldsymbol{c}}^{(t)}$ are flipped and then re-encoded to generate candidate codewords $\hat{\boldsymbol{c}}^{(t,\ell)}$ such that $\hat{\boldsymbol{c}}^{(t,\ell)}\boldsymbol{H}_0^{\mathrm{T}} = \hat{\boldsymbol{c}}^{(t-1)}\boldsymbol{R}^{\mathrm{T}}$, $1 \leqslant \ell \leqslant \ell_{max}$. The second one is the ordered re-encoding list (OREL) algorithm, in which the temporary estimated coded bits $\hat{\boldsymbol{c}}^{(t)}$ are ordered according to their a posteriori LLRs and the most reliable basis (MRB) that contains k most reliable and linearly independent bits is formed. By flipping the ordered bits in the MRB and re-encoding using the flipped MRB, we can also generate candidate codewords $\hat{\boldsymbol{c}}^{(t,\ell)}$ such that $\hat{\boldsymbol{c}}^{(t,\ell)}\boldsymbol{H}_0^{\mathrm{T}} = \hat{\boldsymbol{c}}^{(t-1)}\boldsymbol{R}^{\mathrm{T}}$, $1 \leqslant \ell \leqslant \ell_{max}$. As observed from our simulations, given the maximum list size ℓ_{max}, the order of list matters. Hence, we consider the flipping pattern tree (FPT) algorithm[19], which arranges the flipping patterns with an ordered rooted tree according to their soft weights. The FPT algorithm is embedded in the flipping procedure of both the REL algorithm and the OREL algorithm, resulting in the FPT-REL algorithm and the FPT-OREL[2] algorithm, respectively.

[1]The code length can be reduced to $n(L+1)-k$ by the simple termination process. Nevertheless, we will treat the code length as $n(L+1)$ for notational convenience in the rest of this paper.

[2]The FPT-OREL algorithm is similar to the ordered statistics decoding (OSD) algorithm except that the flipping patterns are ordered by their

Algorithm 2 Sliding-window list decoding of the semi-LDPC-CCs

1. **Initialization**: Set $\hat{\boldsymbol{c}}^{(-1)}=\boldsymbol{0}\in\mathbb{F}_2^n$. Suppose that $\boldsymbol{y}^{(0)}$ has been received. Calculate LLRs $\boldsymbol{\Lambda}^{(0)}$ according to (1).
2. **Sliding-window decoding**: For $t=0,\ 1,\ \cdots,\ L-1$, after receiving $\boldsymbol{y}^{(t+1)}$ and obtaining the estimate $\hat{\boldsymbol{c}}^{(t-1)}$ of the $(t-1)$th sub-frame, do the following steps.
   (a) ***BP decoding for*** $\hat{\boldsymbol{c}}^{(t)}$: Given $\boldsymbol{\Lambda}^{(t)}$ and $\hat{\boldsymbol{c}}^{(t-1)}$, the temporary estimate $\hat{\boldsymbol{c}}^{(t)}$ and the a posteriori LLRs are obtained by performing the iterative BP algorithm.
   (b) ***List decoding***: Given their a posteriori LLRs, $\hat{\boldsymbol{c}}^{(t-1)}$ and the temporary estimate $\hat{\boldsymbol{c}}^{(t)}$, a list of candidate codewords $\hat{\boldsymbol{c}}^{(t,\ell)}(1\leqslant\ell\leqslant\ell_{max})$ can be generated by performing the FPT-REL algorithm or the FPT-OREL algorithm.
   (c) ***BP decoding for*** $\hat{\boldsymbol{c}}^{(t+1,\ell)}$: Calculate LLR $\boldsymbol{\Lambda}^{(t+1)}$ similar to (1). Given $\boldsymbol{\Lambda}^{(t+1)}$ and each $\hat{\boldsymbol{c}}^{(t,\ell)}$, the corresponding estimate of the $(t+1)$th sub-frame can be determined as $\hat{\boldsymbol{c}}^{(t+1,\ell)}$ by performing the iterative BP algorithm under the constraint of (3).
   (d) ***Decision***: Select a candidate $\hat{\boldsymbol{c}}^{(t,\ell)}$ that maximizes $\Gamma_{\mathrm{SWL}}(\hat{\boldsymbol{c}}^{(t,\ell)})$ as (4) from the list of candidate codewords and output the corresponding $\hat{\boldsymbol{u}}^{(t)}$.

3) Metrics Calculation: Given the list of candidate codewords $\hat{\boldsymbol{c}}^{(t,\ell)}(1\leqslant\ell\leqslant\ell_{max})$, we present two different soft metrics, both of which are likelihood metric to choose the output from the list. Different soft metrics lead to different decoding algorithms. These decoding algorithms can work because of the effectiveness of the likelihood metric in the list decoding. The first soft metric is calculated from $(\hat{\boldsymbol{c}}^{(t,\ell)},\hat{\boldsymbol{c}}^{(t+1,\ell)})$, where $\hat{\boldsymbol{c}}^{(t+1,\ell)}$ is the output from the BP decoder with the constraint as

$$\hat{\boldsymbol{c}}^{(t+1,\ell)}\boldsymbol{H}_0^{\mathrm{T}}=\hat{\boldsymbol{c}}^{(t,\ell)}\boldsymbol{R}^{\mathrm{T}}. \tag{3}$$

The soft metric of each candidate codeword $\hat{\boldsymbol{c}}^{(t,\ell)}$ can be calculated as

$$\Gamma_{\mathrm{SWL}}(\hat{\boldsymbol{c}}^{(t,\ell)})=\sum_{i=0}^{n-1}(-1)^{\hat{c}_i^{(t,\ell)}}\Lambda_i^{(t)}+\sum_{i=0}^{n-1}(-1)^{\hat{c}_i^{(t+1,\ell)}}\Lambda_i^{(t+1)}, \tag{4}$$

where $\Lambda_i^{(t+1)}$ denotes the LLR associated with $c_i^{(t+1)}$ and can be calculated similar to (1). The first soft metric $\Gamma_{\mathrm{SWL}}(\hat{\boldsymbol{c}}^{(t,\ell)})$ is a likelihood metric and the candidate codeword $\hat{\boldsymbol{c}}^{(t,\ell)}$ that maximizes this soft metric will be selected as the decoded codeword. Using the soft metric as (4), the SWL decoding algorithm is described in Algorithm 2.

With the soft metric $\Gamma_{\mathrm{SWL}}(\hat{\boldsymbol{c}}^{(t,\ell)})$, we need to perform the BP decoder for each given $\hat{\boldsymbol{c}}^{(t,\ell)}$ to obtain the corresponding $\hat{\boldsymbol{c}}^{(t+1,\ell)}$, as described in Algorithm 2. In order to reduce the decoding complexity, we present a simplified soft metric. Under the constraint as

$$\boldsymbol{c}^{(t+1)}\boldsymbol{H}_0^{\mathrm{T}}=\boldsymbol{c}^{(t)}\boldsymbol{R}^{\mathrm{T}}=\boldsymbol{s}^{(t)}, \tag{5}$$

the LLRs $\boldsymbol{\lambda}^{(t)}$ associated with the syndrome bits $\boldsymbol{s}^{(t)}$ can be estimated from $\boldsymbol{c}^{(t+1)}\boldsymbol{H}_0^{\mathrm{T}}$ and $\boldsymbol{\Lambda}^{(t+1)}$. That is

$$\lambda_i^{(t)}=\log\frac{P_{S_i^{(t)}}(0)}{P_{S_i^{(t)}}(1)}=2\tanh^{-1}\left(\prod_{j:\boldsymbol{H}_0(i,j)=1}\tanh\left(\frac{1}{2}\Lambda_j^{(t+1)}\right)\right), \tag{6}$$

for $0\leqslant i\leqslant n-k-1$, where $\boldsymbol{H}_0(i,j)$ is the element at the ith row and jth column of $\boldsymbol{H}_0$. The simplified soft metric of each candidate codeword $\hat{\boldsymbol{c}}^{(t,\ell)}$ can be calculated as

$$\Gamma_{\mathrm{SSWL}}(\hat{\boldsymbol{c}}^{(t,\ell)})=\sum_{i=0}^{n-1}(-1)^{\hat{c}_i^{(t,\ell)}}\Lambda_i^{(t)}+\sum_{i=0}^{n-k-1}(-1)^{\hat{s}_i^{(t,\ell)}}\lambda_i^{(t)}, \tag{7}$$

where $\hat{s}_i^{(t,\ell)}$ is the ith element of $\hat{\boldsymbol{s}}^{(t,\ell)}=\hat{\boldsymbol{c}}^{(t,\ell)}\boldsymbol{R}^{\mathrm{T}}$. The second soft metric $\Gamma_{\mathrm{SSWL}}(\hat{\boldsymbol{c}}^{(t,\ell)})$ is a simplified likelihood metric and the candidate codeword $\hat{\boldsymbol{c}}^{(t,\ell)}$ that maximizes this soft metric will be selected as the decoded codeword. Using the soft metric as (7), the simplified SWL (SSWL) decoding algorithm is described in Algorithm 3.

The above SWL and SSWL decoding algorithms are designed with window size of $d=2$, which can be extended to the setup of $d=3$. To recover $\boldsymbol{u}^{(t)}$ from $(\boldsymbol{y}^{(t)},\boldsymbol{y}^{(t+1)},\boldsymbol{y}^{(t+2)})$, we need generalize the soft metrics by including one more blocks of LLRs. The details are omitted here.

Algorithm 3 Simplified sliding-window list decoding of the semi-LDPC-CCs

1. **Initialization**: Set $\hat{\boldsymbol{c}}^{(-1)}=\boldsymbol{0}\in\mathbb{F}_2^n$. Suppose that $\boldsymbol{y}^{(0)}$ has been received. Calculate LLR $\boldsymbol{\Lambda}^{(0)}$ according to (1).
2. **Sliding-window decoding**: For $t=0,\ 1,\ \cdots,\ L-1$, after receiving $\boldsymbol{y}^{(t+1)}$ and obtaining the estimate $\hat{\boldsymbol{c}}^{(t-1)}$ of the $(t-1)$th sub-frame, do the following steps.
   (a) ***BP decoding and list decoding***: Perform Step BP decoding for $\hat{\boldsymbol{c}}^{(t)}$ and list decoding in Algorithm 2.
   (b) ***LLR calculations***: Calculate the LLRs $\boldsymbol{\lambda}^{(t)}$ associated with the syndrome bits according to (6).
   (c) ***Decision***: Select a candidate $\hat{\boldsymbol{c}}^{(t,\ell)}$ that maximizes $\Gamma_{\mathrm{SSWL}}(\hat{\boldsymbol{c}}^{(t,\ell)})$ as (7) from the list of candidate codewords and output the corresponding $\hat{\boldsymbol{u}}^{(t)}$.

B. Upper Bound and GA Lower Bound

Denote by fER_t $(0\leqslant t\leqslant L-1)$ the probability that the decoding result $\hat{\boldsymbol{u}}^{(t)}$ is not equal to the information sequence $\boldsymbol{u}^{(t)}$ and by frame error rate (FER) for the probability that the decoding result $\hat{\boldsymbol{u}}$ is not equal to $\boldsymbol{u}$. It has been proved in Ref. [20] that

$$fER_0\leqslant\max_t fER_t\leqslant FER\leqslant\sum_{t=0}^{L-1}fER_t. \tag{8}$$

For integrity, we rederive these bounds and present a simple estimated GA lower bound in this subsection. Similar to Refs. [20,21], we define

$$fER=\frac{1}{L}\sum_{t=0}^{L-1}fER_t, \tag{9}$$

a posteriori LLRs, which is crucial in the case of the limited list size.

which is used as the performance measure in this paper and can be evaluated in practice by

$$fER = \frac{\text{number of erroneous decoded sub-frames}}{\text{total number of transmitted sub-frames}}. \quad (10)$$

The event that the decoding result $\hat{\boldsymbol{u}}^{(0)}$ is not equal to the transmitted vector $\boldsymbol{u}^{(0)}$ is referred to as the first error event $\mathscr{E}_0$. From definition, we have

$$fER_0 = \Pr\{\mathscr{E}_0\}. \quad (11)$$

Generally, we denote by $\mathscr{E}_t$ the event that the first error event occurs at time slot t. That is, $\mathscr{E}_t$ represents $\hat{\boldsymbol{u}}^{(i)} = \boldsymbol{u}^{(i)}$ for all $i < t$ but $\hat{\boldsymbol{u}}^{(t)} \neq \boldsymbol{u}^{(t)}$. The probability that the first error event occurs at time t can be bounded by

$$\Pr\{\mathscr{E}_t\} \leqslant fER_0, \quad (12)$$

since with $\hat{\boldsymbol{u}}^{(t-1)}$ being correct, the performance of the tth sub-frame will not be worse than that of the first sub-frame. Therefore, the fER_t can be bounded by

$$fER_t = \sum_{i=0}^{t} \Pr\{\mathscr{E}_i\}\Pr\{\hat{\boldsymbol{u}}^{(t)} \neq \boldsymbol{u}^{(t)} | \mathscr{E}_i\} \leqslant \sum_{i=0}^{t} \Pr\{\mathscr{E}_i\} \leqslant (t+1) fER_0. \quad (13)$$

Therefore, the fER can be upper bounded by

$$fER \leqslant \frac{1}{L}\sum_{t=0}^{L-1}(t+1) fER_0 = \frac{L+1}{2} \cdot fER_0. \quad (14)$$

The above upper bound can also be understood by noticing that the event $\mathscr{E}_t$ usually causes catastrophic error propagation. Typically, in the case when event $\mathscr{E}_t$ occurs, a random-like sequence $(\hat{\boldsymbol{c}}^{(t)} - \boldsymbol{c}^{(t)})\boldsymbol{R}^{\mathrm{T}}$ is superimposed onto the syndromes $\boldsymbol{s}^{(t)}$ due to the existence of $\boldsymbol{R}$, which will lead to occurrence of event $\{\hat{\boldsymbol{U}}^{(t+1)} \neq \boldsymbol{U}^{(t+1)}\}$. This fact can also be used to derive an approximated lower bound as described below.

Obviously, the fER_0 can be lower bounded by

$$fER_0 \geqslant fER_0^{GA}, \quad (15)$$

where fER_0^{GA} denotes the error performance of the first sub-frame with a GA decoder that outputs the transmitted $\boldsymbol{u}^{(0)}$ if it is in the list. In particular, fER_0^{GA} can be obtained by simulating with a GA decoder for the LDPC-BC specified by the parity-check matrix $\boldsymbol{H}_0$. Given that expected fact of the catastrophic error propagation, we present the estimated GA (EGA) lower bound for the semi-LDPC-CCs from the underlying LDPC-BCs. That is,

$$fER \gtrsim \frac{L+1}{2} \cdot fER_0^{GA}. \quad (16)$$

Notice that both the upper and the lower bounds can be evaluated from the performance of the first sub-frame of a semi-LDPC-CC by simulating only the first decoding window. This is helpful in constructing good codes by computer search.

C. Serial Implementation

The presented SWL or SSWL decoding algorithm can be implemented in a serial manner, whereby the complexity can be reduced by introducing an early stopping criterion. Different from Ref. [22], we use a more efficient early stopping criterion in this paper. A minimum list size ℓ_{min} is set to guarantee the performance, while a maximum list size ℓ_{max} is set to control the worst-case complexity. The early stopping criterion is designed based on the difference between the maximum soft metric and second maximum soft metric of the current candidate list. The basic rule for setting parameters is to control the type I error. That is, in the case when the null hypothesis that the transmitted codeword is not in the list is true, the rejection (early stopping) probability should be small. This is illustrated by the following example.

Example 1: Consider the sub-frame with $n = 64$ and $k = 32$. The matrix $\boldsymbol{H}_0 = [\boldsymbol{P}, \boldsymbol{I}]$ is constructed by the PEG algorithm[17], where the sub-matrix $\boldsymbol{P}$ has a column weight $w_c = 2$ and a row weight $w_r = 2$. The matrix $\boldsymbol{R}$ is randomly generated but fixed, where the elements are generated independently according to the Bernoulli distribution with success probability $1/2$. We set $\ell_{max} = 529$ and $SNR = 4$ dB for simulation.

By performing the proposed SSWL decoding with FPTOREL algorithm, the rank of the transmitted codeword in the list can be viewed as a random variable and its PMF can be approximated by the histograms shown in Fig. 2. We see that, to guarantee the performance obtained by the original SSWL decoding, say $fER \approx 4 \times 10^{-3}$, it is necessary to set a minimum list size $\ell_{min} = 25$. After generating ℓ_{min} candidates, we would like to choose a threshold T such that the probability of "too-early"[3] stopping is small in the case when the transmitted codeword is not in the list. To this end, we have estimated by simulation the density distribution of the difference between the maximum soft metric and second maximum soft metric conditional on that the transmitted codeword is not in the list or in the list. The histogram results are shown in Fig. 3, from which we may choose $T = 20$, for example, to achieve a probability less than 0.04 for too-early stopping. With these two observations, we present the following two-round serial decoding implementation.

Let ℓ_{min} be the number of candidate codewords in the first round and δ be the difference between the maximum soft metric and the second maximum soft metric of $\hat{\boldsymbol{c}}^{(t,\ell)}$. Let T be a preset threshold of the soft metric difference. The serial implementation for the proposed decoding algorithms is described in Algorithm 4.

[3]The ideal stopping time is the time once when the transmitted codeword is included in the list, which is obviously impractical. We refer a stopping time to as a "too-early" stopping time if at which the probability of the transmitted codeword not in the list is high.

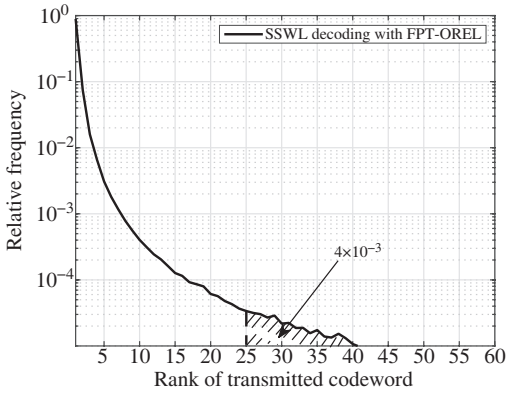

Fig. 2 Histograms of the rank of transmitted codeword in the list

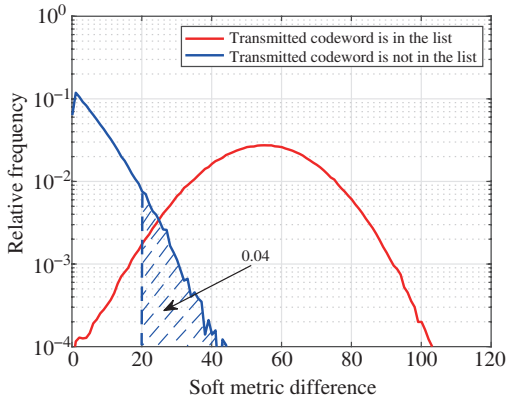

Fig. 3 Histograms of the difference between the maximum soft metric and second maximum soft metric

Algorithm 4 Two-round serial decoding implementation

First round: for $\ell = 1, 2, \cdots, \ell_{min}$ **do**

Generate $\hat{\boldsymbol{c}}^{(t,\ell)}$ using the proposed list decoding algorithms.

Calculation of soft metric difference: Calculate δ, the difference between the maximum soft metric and second maximum soft metric in the list.

Second round: while $\delta < T$ and $\ell \leqslant \ell_{max}$ **do**

$\ell \leftarrow \ell + 1$.

Generate one more candidate codewords, denoted by $\hat{\boldsymbol{c}}^{(t,\ell)}$, using the proposed list decoding algorithms and update δ.

Output: Output the candidate that maximizes the soft metric from $\{\hat{\boldsymbol{c}}^{(t,1)}, \hat{\boldsymbol{c}}^{(t,2)}, \cdots, \hat{\boldsymbol{c}}^{(t,\ell)}\}$.

D. Complexity Analysis

In this subsection, taking the conventional (d_v, d_c)-regular SC-LDPC codes with SW decoding algorithm as the benchmark, we analyze the complexity of the semi-LDPC-CCs under the proposed SSWL decoding algorithm.

Firstly, owing to the specific decoding process, the Tanner graph used in the SSWL decoding algorithm is small (about half or less) compared with that used in the conventional SW decoding algorithm with decoding window size $d \geqslant 2$. Secondly, the degree of the variable nodes is 1 or 2 for the semi-LDPC-CCs, while as shown in Ref. [23], for (d_v, d_c)-regular SC-LDPC codes to have good performance, it is usually required that $d_v \geqslant 3$. According to Ref. [24], in each iteration, the total computational load for a variable-node update is $2d_v + 1$ real additions and the total computational load for a check-node update consists of $3(d_c - 2)$ core operations, where each core operation can be realized using 4 real additions, 1 comparison, and 2 corrections. Because of the smaller Tanner graph and lower degree distribution, the computational load in each iteration of BP decoder for semi-LDPC-CCs is lower compared with that of the iterative decoder for conventional SC-LDPC codes. We need to point out that the maximum iteration number can be small in the proposed decoding algorithms compared with that in the conventional SW decoding algorithm. Considering the whole SSWL decoding algorithm, the main extra computational load is caused by the FPT-REL or FPT-OREL algorithm with a preset maximum list size ℓ_{max}. Given $\boldsymbol{H}_0$ and $\boldsymbol{R}$, the complexity of the list decoding is linearly increased with the list size. Indeed, the average list size can be small in the high SNR region of interest by using the two-round serial implementation, as confirmed by the following numerical results.

IV. Numerical Results

In this section, we present numerical results of the semi-LDPC-CCs. We set the sub-frame with $n = 64$ and $k = 32$. In our simulations, $\boldsymbol{H}_0 = [\boldsymbol{P}, \boldsymbol{I}]$ is constructed by the PEG algorithm[17], where $\boldsymbol{P}$ has a column weight of $w_c = 2$ and a row weight of $w_r = 2$. $\boldsymbol{R}$ is a randomly generated but fixed matrix whose elements are generated independently according to the Bernoulli distribution with success probability $1/2$. In all examples, the encoder terminates every $L = 32$ sub-frames. Codewords $\boldsymbol{c}^{(t)}$ are transmitted with BPSK modulation over AWGN channels. The SWL, SSWL decoding algorithms and the two-round serial decoding implementation are employed for the decoding. The maximum numbers of iteration for the conventional SC-LDPC codes and the semi-LDPC-CCs are 50 and 5, respectively.

A. Comparison with the SC-LDPC Code

The following example is provided to compare the performance of a semi-LDPC-CC and a (3, 6)-regular SC-LDPC code under the equal decoding latency constraint.

Example 2 [Comparison with the (3, 6)-Regular SC-LDPC Code]: Consider a semi-LDPC-CC with sub-frame of $[n, k] = [64, 32]$ and $L = 32$. The SWL and SSWL decoding algorithms with FPT-OREL algorithm are employed for the decoding. The decoding window size of the semi-LDPC-CC is $d = 2$ and hence the decoding latency is 128 bits. For comparison, we choose the (3, 6)-regular SC-LDPC code with an equal coupling width of one and an equal coupling length $L = 32$. The sub-matrices $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$ of the SC-LDPC code are constructed by lifting the base matrices $\boldsymbol{B}_0 = [2, 1]$

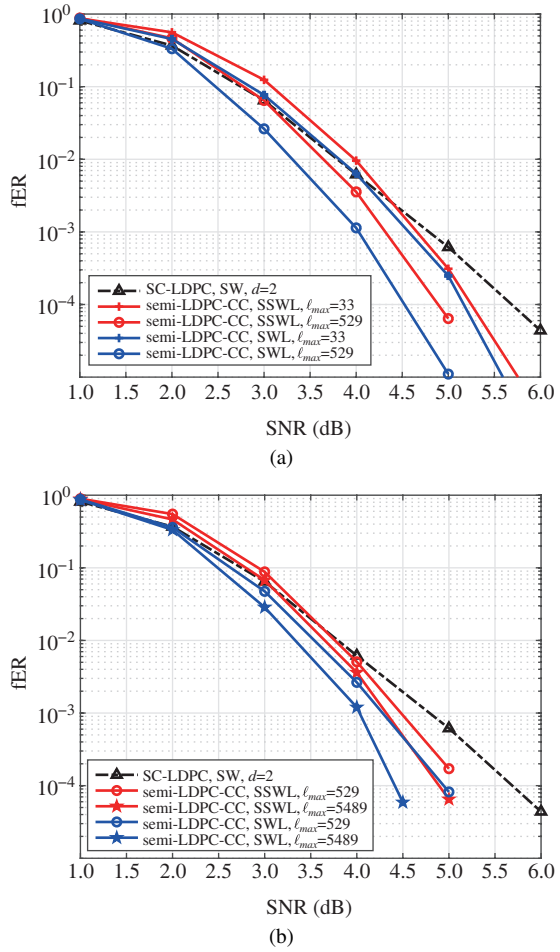

Fig. 4 The fER performance of a semi-LDPC-CC with a real code rate of about 0.4848 and a (3, 6)-regular SC-LDPC code with a real code rate of about 0.4844. The FPT-OREL and FPT-REL algorithms are used in the SWL and SSWL decoding algorithms, respectively: (a) FPT-OREL; (b) FPT-REL

and $\boldsymbol{B}_1 = [1,2]$ with a lifting factor of 32, respectively. For (3, 6)-regular SC-LDPC code, the conventional SW decoding algorithm is employed for decoding, where the decoding window size is $d = 2$, resulting in an equal decoding latency of 128 bits. The fER performance comparison is shown in Fig. 4(a), where we observe that the semi-LDPC-CCs under the proposed decoding algorithm with $\ell_{max} = 33$ exhibits 0.5 dB coding gain at $fER = 10^{-4}$ over the (3, 6)-regular SC-LDPC code under the same decoding window size, which verifies the effectiveness of the list decoding. As the maximum list size enlarges to $\ell_{max} = 529$, the coding gain can increase up to about 1.2 dB. We also observe that the semi-LDPC-CCs with the SWL decoding algorithm can outperform the proposed codes with SSWL decoding algorithm.

We have made similar observations for the SWL and SSWL decoding algorithms with FPT-REL algorithm as shown in Fig. 4(b). Under the simpler FPT-REL algorithm, the coding gain can increase up to about 1.4 dB as the maximum list size enlarges to $\ell_{max} = 5\,489$.

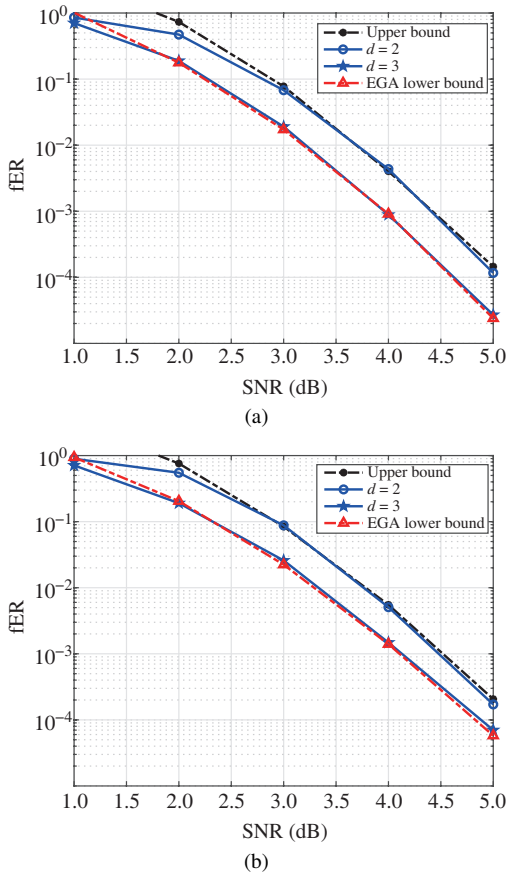

Fig. 5 The fER performance and the corresponding bounds of the semi-LDPC-CC under SSWL decoding algorithm with FPT-OREL and FPT-REL algorithms: (a) SSWL with FPT-OREL, $\ell_{max} = 100$; (b) SSWL with FPT-REL, $\ell_{max} = 529$

B. Upper Bound and GA Lower Bound

In this subsection, we present the performance of the semi-LDPC-CCs using the proposed decoding algorithms with different window sizes and the corresponding upper and GA lower bounds.

Example 3 (Performance and Bounds): Consider the semi-LDPC-CC that was used in Example 2. The SSWL decoding algorithm with FPT-OREL and FPT-REL algorithms is employed for the decoding, where the decoding window sizes are $d = 2$ and 3, respectively. The fER performance curve of the semi-LDPC-CC under the SSWL decoding algorithm with FPT-OREL algorithm is depicted in Fig. 5(a). From the figure, we see that the performance of the semi-LDPC-CCs using SSWL decoding algorithm with $d = 2$ is very close to the upper bound in the high SNR region. We also observe that the performance can be further improved to approach the EGA lower bound by increasing the window size of $d = 3$. We have made similar observations for the SSWL decoding algorithm with FPT-REL algorithm as shown in Fig. 5(b).

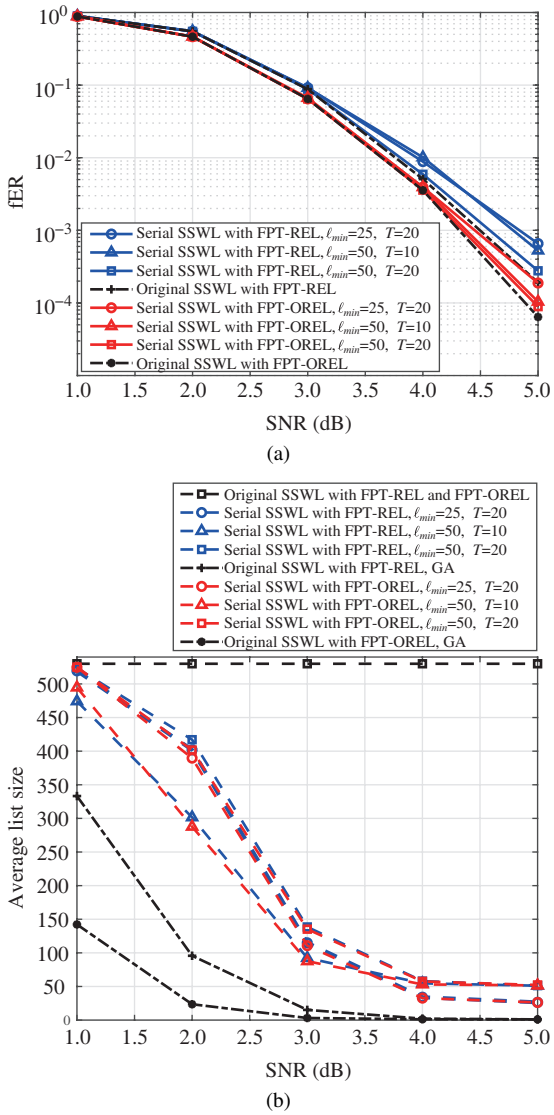

Fig. 6 The fER performance and the average list size of the semi-LDPC-CC under SSWL decoding algorithm with FPT-OREL and FPT-REL algorithms. The maximum list size is $\ell_{max} = 529$: (a) the fER performance; (b) the average list size

C. Serial Implementation of the Proposed Decoding Algorithms

The following example is provided to show the two-round serial implementation of the proposed decoding algorithms.

Example 4 (Two-Round Decoding): Consider the semi-LDPC-CC that was used in Example 2. The two-round serial implementation of the SSWL decoding algorithm with FPT-OREL and FPT-REL is employed for the decoding. The fER performance is shown in Fig. 6(a), where we observe that the performance of the two-round serial SSWL decoding with proper parameter is very close to that of the original SSWL decoding algorithm. The average list size comparison is shown in Fig. 6(b), where we observe that at the cost of negligible performance loss, the average list size can be significantly reduced by using the two-round serial SSWL decoding with properly selected parameters. For example, at $SNR = 4$ dB, the average list size can be reduced more than 10 times.

V. Conclusions

This paper has presented a new construction of LDPC-CCs, whose parity-check matrix contain dense sub-matrices. Based on this special construction, we proposed the SWL and SSWL decoding algorithms with a small window size, resulting in a low decoding latency but a competitive error-correcting performance. For performance analysis, we have presented the upper and GA lower bounds to predict the performance of the semi-LDPC-CCs. Moreover, we have presented the two-round serial implementation of the proposed decoding algorithms for the purpose of reducing complexity. Our numerical results have shown that the semi-LDPC-CCs can obtain a coding gain of up to 1.4 dB over the conventional SC-LDPC codes. They have also shown that the effectiveness of the two-round serial implementation and these performance bounds.

The proposed semi-LDPC-CCs are constructed for the potential streaming applications, especially in which the latency and reliability are both the key performance indicators. In the construction, a randomly generated but fixed dense sub-matrix is involved, which is not convenient for practical implementations. As a future work, we need to find ways to jointly optimize the sparse and the dense sub-matrices with structures.

References

[1] FELSTROM A J, ZIGANGIROV K S. Time-varying periodic convolutional codes with low-density parity-check matrix[J]. IEEE Transactions on Information Theory, 1999, 45(6): 2181-2191.

[2] TANNER R M, SRIDHARA D, SRIDHARAN A, et al. LDPC block and convolutional codes based on circulant matrices[J]. IEEE Transactions on Information Theory, 2004, 50(12): 2966-2984.

[3] PUSANE A E, SMARANDACHE R, VONTOBEL P O, et al. Deriving good LDPC convolutional codes from LDPC block codes[J]. IEEE Transactions on Information Theory, 2011, 57(2): 835-857.

[4] MITCHELL D G M, LENTMAIER M, COSTELLO D J. Spatially coupled LDPC codes constructed from protographs[J]. IEEE Transactions on Information Theory, 2015, 61(9): 4866-4889.

[5] MO S, CHEN L, COSTELLO D J, et al. Designing protograph-based quasi-cyclic spatially coupled LDPC codes with large girth[J]. IEEE Transactions on Communications, 2020, 68(9): 5326-5337.

[6] LIU K, EL-KHAMY M, LEE J. Finite-length algebraic spatially-coupled quasi-cyclic LDPC codes[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(2): 329-344.

[7] WANG Q, CAI S H, LIN W C, et al. Spatially coupled LDPC codes via partial superposition and their application to HARQ[J]. IEEE Transactions on Vehicular Technology, 2021, 70(4): 3493-3504.

[8] KUDEKAR S, RICHARDSON T, URBANKE R. Spatially coupled ensembles universally achieve capacity under belief propagation[J]. IEEE Transactions on Information Theory, 2013, 59(12): 7761-7813.

[9] LENTMAIER M, SRIDHARAN A, COSTELLO D J, et al. Iterative decoding threshold analysis for LDPC convolutional codes[J]. IEEE Transactions on Information Theory, 2010, 56(10): 5274-5289.

[10] KUDEKAR S, RICHARDSON T, URBANKE R. Threshold saturation via spatial coupling: why convolutional LDPC ensembles perform so well over the BEC[J]. IEEE Transactions on Information Theory, 2011, 57(2): 803-834.

[11] IYENGAR A R, SIEGEL P H, URBANKE R L, et al. Windowed decoding of spatially coupled codes[J]. IEEE Transactions on Information Theory, 2013, 59(4): 2277-2292.

[12] HUANG K, MITCHELL D G M, WEI L, et al. Performance comparison of LDPC block and spatially coupled codes over GF (q)[J]. IEEE Transactions on Communications, 2015, 63(3): 592-604.

[13] SIMSEK M, AIJAZ A, DOHLER M, et al. 5G-enabled tactile Internet[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(3): 460-473.

[14] CHEN Z, CAI S, CHEN L, et al. Semi-LDPC convolutional codes with low-latency decoding algorithm[C]//2020 IEEE International Conference on Computer and Communications. Piscataway: IEEE Press, 2020: 105-109.

[15] ELIAS P. List decoding for noisy channels[C]//1957 IRE WESCON Convention Record. [S.l.: s.n.], 1957: 94-104.

[16] WOZENCRAF J M. List decoding[R]. Quarter Progress Report, Cambridge, 1958: 90-95.

[17] HU X Y, ELEFTHERIOU E, ARNOLD D M. Regular and irregular progressive edge-growth Tanner graphs[J]. IEEE Transactions on Information Theory, 2005, 51(1): 386-398.

[18] PUSANE A E, FELTSTROM A J, SRIDHARAN A, et al. Implementation aspects of LDPC convolutional codes[J]. IEEE Transactions on Communications, 2008, 56(7): 1060-1069.

[19] TANG S, MA X. A new Chase-type soft-decision decoding algorithm for Reed-Solomon codes[J]. ArXiv:1309.1555, 2013.

[20] LIN W, CAI S, WEI B, et al. Successive cancellation list decoding of semi-random unit memory convolutional codes[J]. ArXiv:1905.11392, 2019.

[21] LI J, CAI S, LIN W, et al. Improved block oriented unit memory convolutional codes[J]. IEEE Transactions on Communications, to be published.

[22] LIN W, WEI B, MA X. List decoding with statistical check for semi-random block-oriented convolutional code[J]. Electronic Letter, 2019, 55(10): 601-603.

[23] WEI L, MITCHELL D G M, FUJA T E, et al. Design of spatially coupled LDPC codes over GF (q) for windowed decoding[J]. IEEE Transactions on Information Theory, 2016, 62(9): 4781-4800.

[24] HU X Y, ELEFTHERIOU E, ARNOLD D M, et al. Efficient implementations of the sum-product algorithm for decoding LDPC codes[C]//2001 IEEE Global Telecommunications Conference. Piscataway: IEEE Press, 2001: 1036-1036E.

About the Authors

Qianfan Wang received the B.S. degree in Applied Physics from Henan Polytechnic University, Jiaozuo, China, in 2014, and the M.S. degree in Electronics and Communication Engineering from Sun Yat-sen University, Guangzhou, China, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include information theory, channel coding theory, and their applications to communication systems.

Suihua Cai received the B.Sc. degree in Information and Computering Science from China University of Geosciences, Wuhan, China, in 2011. He received the M.S. degree in Fundamental Mathematics in 2016 and the Ph.D. degree in Information and Communication Engineering in 2019, both from Sun Yat-sen University, Guangzhou, China. He is currently a Post-Doctoral Fellow with Sun Yat-sen University. His research interests include information theory, channel coding theory, and their applications to communication systems.

Li Chen received the B.Sc. degree in Applied Physics from Jinan University, China, in 2003, and the M.Sc. degree in Communications and Signal Processing and the Ph.D. degree in Communications Engineering from Newcastle University, U.K., in 2004 and 2008, respectively. From 2007 to 2010, he was a Research Associate with Newcastle University. In 2010, he returned to China as a Lecturer of the School of Information Science and Technology, Sun Yat-sen University, Guangzhou. From 2011 to 2012, he was a Visiting Researcher with the Institute of Network Coding, the Chinese University of Hong Kong, Hong Kong, China. From 2011 and 2016, he was an Associate Professor and a Professor of the university. Since 2013, he has been the Associate Head of the Department of Electronic and Communication Engineering (ECE). From July 2015 to October 2015, he was a Visitor of the Institute of Communications Engineering, Ulm University, Germany. From October 2015 to June 2016, he was a Visiting Associate Professor with the Department of Electrical Engineering, the University of Notre Dame, USA. From 2017 to 2020, he was the Deputy Dean of the School of Electronics and Communication Engineering. His research interests include information theory, error-correction codes, and data communications. He likes reading and photography. He is a Senior Member of the Chinese Institute of Electronics (CIE). He is a member of the IEEE Information Theory Society Board of Governors Conference Committee and External Nomination Committee, the Chair of the IEEE Information Theory Society Guangzhou Chapter, and a Committee Member of the CIE Information Theory Society. He is currently serving as an Associate Editor for IEEE Transactions on Communications. He has been involved in organizing several international conferences, including the 2018 IEEE Information Theory Workshop (ITW) at Guangzhou, for which he was the General Co-Chair.

Xiao Ma [corresponding author] received the Ph.D. degree in Communication and Information Systems from Xidian University, China, in 2000. He is a Professor of the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. From 2000 to 2002, he was a Post-Doctoral Fellow with Harvard University, Cambridge, MA. From 2002 to 2004, he was a Research Fellow with the City University of Hong Kong, Hong Kong, China. His research interests include information theory, channel coding theory, and their applications to communication systems and digital recording systems. He is a co-recipient, with A. Kavčić and N. Varnica, of the 2005 IEEE Best Paper Award in Signal Processing and Coding for Data Storage. Dr. Ma is a member of the IEEE.