IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 31, 2017, accepted October 24, 2017, date of publication November 8, 2017, date of current version March 13, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2771375

Improving Windowed Decoding of SC LDPC Codes by Effective Decoding Termination, Message Reuse, and Amplification

INAYAT ALI[1], JONG-HWAN KIM[1], SANG-HYO KIM[1], (Member, IEEE), HEEYOUL KWAK[2], AND JONG-SEON NO[2], (Fellow, IEEE)

[1]College of Information and Communication Engineering, Sungkyunkwan University, Suwon 16419, South Korea
[2] INMC, Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea

Corresponding author: Sang-Hyo Kim (iamshkim@skku.edu)

This research was supported in part by the MSIT, Korea, under the ITRC support program (IITP-2016-000309-002) supervised by the IITP and by the Basic Science Research Program through the NRF of Korea funded by the Ministry of Education (NRF-2015R1D1A1A01058975).

ABSTRACT In this paper, we address a number of weaknesses of the windowed decoding of spatially coupled low-density parity-check (SC LDPC) codes and propose three modifications that simultaneously improve its performance, complexity, and latency. An effective termination method of the windowed decoding and the reuse of edge messages of previous target symbols provide a good performance-latency tradeoff when compared with the conventional windowed decoder. Also, we propose a scheme that lowers the error floor, in which the amplified edge messages of the previous window are used in the present window. The proposed windowed decoding, consisting of the three schemes, provides a significant performance gain with smaller latency. The validity of the new windowed decoding is verified by the evaluation with codes from different SC LDPC ensembles.

INDEX TERMS Spatially coupled LDPC codes, windowed decoder, density evolution, decoding termination.

I. INTRODUCTION

The low-density parity-check (LDPC) codes have long been forgotten since they were first introduced by Gallager [1]. The rediscovery of the LDPC codes [2], [3] has ignited one of most influential series of studies in coding theory and today's communication systems. Spielman [2] and Mackay and Neal [3] independently found that regular LDPC codes are very good under belief propagation (BP) decoding. Luby *et al.* [4] and Richardson *et al.* [5] verified that irregular codes can perform better and even approach channel capacity when their degree distributions are well designed. For the design and optimization of LDPC ensembles, the density evolution technique was used to evaluate the noise threshold or the capacity of LDPC codes [5], [6]. On top of the ensemble design, graph construction techniques such as the progressive edge growth (PEG) algorithm [7], [8] can be used for the construction of finite length block codes. Well-designed LDPC block codes have been successfully adopted in various wireless communication systems [9].

Convolutional codes defined based on a low-density parity-check matrix were first proposed in [10]. It was shown that the convolutional gain in the performance of LDPC convolutional codes over block codes was considerable [11]. Later, a significant threshold improvement was observed from the termination of the LDPC convolutional codes [12]. It has recently been proved that the belief propagation (BP) thresholds of terminated LDPC convolution codes are actually equal to their maximum a posteriori (MAP) thresholds over the binary erasure channel (BEC) [13] by using the MAP threshold evaluation method [14]. These codes were renamed as 'spatially coupled (SC) LDPC codes' since they reflect the corresponding graph structure [13]. The same phenomenon was also observed and conjectured by Lentmaier *et al.* [15]. In [16], it was proved that SC LDPC codes can universally achieve the channel capacity under BP decoding. While these codes attain a good BP threshold, they are subjected to rate loss due to termination. However, the rate loss is mitigated if the coupling length is large [13].

Decoding long codes may require large memory and lead to high latency. In [17], the windowed decoder (WD) was proposed as a solution to this problem. The diagonal stair-like structure of parity check matrices (PCM) provides the condition in which a sub-block of variable nodes is only connected with a local-group of check nodes. Hence, the BP decoder can be operated inside the constrained dimension of a window. The window slides after the iterative decoding of a set of target symbols. A trade-off exists between the decoding performance and the latency in terms of the window size W. The decoding performance improves with an increase in W, but at the cost of latency. Note that if W is set sufficiently large, the performance loss becomes marginal.

To improve the performance of the WD, ensemble design rules have been suggested [17], [18]. With an effective ensemble structure, the WD threshold increases rapidly as a function of the window size. In [19] and [21], it was shown that the decoding complexity of WD can be significantly reduced without performance degradation by employing special scheduling techniques. On the other hand, this paper focuses on improving the performance of WD. We propose an improved windowed decoder (iWD) for the SC LDPC codes that incorporates three modifications to the conventional WD. The first technique of the iWD is a new termination method in which the decoding stops earlier than the conventional WD by expanding the group of target symbols. The second technique is the reuse of edge messages of previous target symbols instead of their output log-likelihood-ratios (LLRs). Finally, we propose an error floor lowering technique by message amplification. A significant performance gain over the conventional WD is obtained.

The organization of this paper is as follows. The construction of protograph-based SC LDPC codes is introduced in Section II. The preliminary for SC LDPC codes and the asymptotic analysis of LDPC codes are also discussed. Section III includes the main contribution of this paper. The iWD of SC LDPC codes is detailed and compared with the conventional WD. The latency and asymptotic analysis are also given. In Section IV, numerical results demonstrate the effectiveness of the proposed iWD. Finally, Section V concludes the paper.

II. SC LDPC CODES CONSTRUCTED FROM PROTOGRAPHS

This section discusses the protograph-based SC LDPC ensembles and the finite-length code construction procedure. We also explain the asymptotic analysis of protograph-based LDPC codes, which is useful for analyzing WD.

A. PROTOGRAPH & LDPC CODES

A *protograph* is a small bipartite graph from which a long LDPC code can be obtained by a procedure known as ‘copy and permute’ [22]. The protograph is copied M times so as to form the size M bundles of edges, variable nodes (VNs), and check nodes (CNs). A permutation is applied to the edges within each bundle connecting VNs to CNs, yielding an LDPC graph with block length M times as large as a single protograph. Let n_P and m_P be the number of VNs and CNs in a protograph, respectively. The derived graph will then have $n = M \times n_P$ VNs and $m = M \times m_P$ CNs. Let (J, K) be the degree pair of a regular protograph, where J and K are the degree of edges connected to VNs and CNs in a protograph, respectively. We can represent a protograph by its $m_P \times n_P$ bi-adjacency matrix $\mathbf{B}$, called the base matrix.

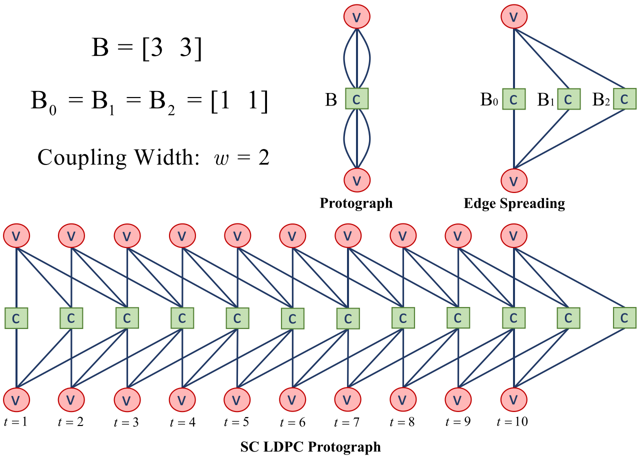


FIGURE 1. **SC LDPC protograph construction using 10 LDPC protographs with component base matrices of ensemble** A.

B. PROTOGRAPH-BASED SC LDPC ENSEMBLES AND CODE CONSTRUCTION

Spatially coupled LDPC codes can be derived from spatial coupling of protographs. For constructing an SC LDPC protograph, L replicas of an LDPC protograph are coupled by a procedure called edge spreading, where L is defined as the coupling length of an SC LDPC protograph. The L protographs are indexed by t, and the coupling is performed by spreading the edges from VNs at t and connecting the other end to CNs at $t + z$, $z = 0, 1, \ldots, w$, where $w > 0$ is the coupling width of the SC LDPC protograph [23]. Due to the spreading of the edges, the base matrix of a protograph is partitioned into component base matrices $\mathbf{B}_i$, $i = 0, 1, \ldots, w$. Hence, $\mathbf{B}_i$ exhibits the edge connectivity between adjacent protographs in the coupled graph. As an example, Fig. 1 shows the construction method for constructing an SC LDPC ensemble with component base matrices $\mathbf{B}_0 = \mathbf{B}_1 = \mathbf{B}_2 = [1 \quad 1]$ and $L = 10$. The base matrix of an SC LDPC protograph with a diagonal stair-like structure is given as

$$\mathbf{B}_{[1,L]} = \begin{bmatrix} \mathbf{B}_0 & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & \\ \vdots & \mathbf{B}_1 & \ddots & \\ \mathbf{B}_w & \vdots & \ddots & \mathbf{B}_0 \\ & \mathbf{B}_w & \ddots & \mathbf{B}_1 \\ & & \ddots & \vdots \\ & & & \mathbf{B}_w \end{bmatrix}_{[(L+w)J_g \times LK_g]}, \tag{1}$$

TABLE 1. SC LDPC (3, 6) ensembles.

Ensemble	Coupling Width (w)	Component Base Matrices
Ensemble A	2	$\mathbf{B}_0=[1\ \ 1]$, $\mathbf{B}_1=[1\ \ 1]$, $\mathbf{B}_2=[1\ \ 1]$
Ensemble B	1	$\mathbf{B}_0=[2\ \ 2]$, $\mathbf{B}_1=[1\ \ 1]$
Ensemble C	1	$\mathbf{B}_0=[1\ \ 1]$, $\mathbf{B}_1=[2\ \ 2]$

where the size of component base matrix $\mathbf{B}_i$ is $J_g \times K_g$ (here, $J_g = \frac{J}{gcd(J,K)}$ and $K_g = \frac{K}{gcd(J,K)}$). For a protograph based SC LDPC code of lifting factor M, we call each group of $K_g \times M$ VNs a 'VN sub-block'. Similarly, each group of $J_g \times M$ CNs is called a 'CN sub-block'.

The spreading of edges gives the coupled protograph w overhanging CNs. These extra CNs result in rate loss compared to the rate of an uncoupled LDPC protograph. The rate of the SC LDPC code is given by

$$R_L = 1 - \frac{J_g(L+w)}{K_g L} = 1 - \left(1+\frac{w}{L}\right)\left(\frac{J_g}{K_g}\right) \\ = 1 - \left(1+\frac{w}{L}\right)(1-R), \tag{2}$$

where $R = 1 - (J_g/K_g)$ is the rate of the underlying (J, K) protograph. It can be seen that the factor $(1 + w/L)$ delineates the rate loss, and its effect vanishes at a speed $1/L$ [13]. This means that when $L \to \infty$, the code rate $R_L \to R$.

We define SC LDPC ensembles with a set of component base matrices from which SC LDPC protographs of length L can be constructed. The entries of $\mathbf{B}_i$, which are non-negative integers, define the edge connectivity with the neighboring w protographs. In this paper, we use codes constructed from the three ensembles defined in Table 1 to validate our proposed methods.

Ensemble A is a classical (3, 6) SC LDPC ensemble [18] defined by $\mathbf{B}_0 = \mathbf{B}_1 = \mathbf{B}_2 = [1\ \ 1]$. In the construction of this ensemble, an individual protograph is at most connected to two neighboring protographs; therefore, the coupling width is $w = 2$. Ensembles B and C with component base matrices $\mathbf{B}_0 = [2\ \ 2]$, $\mathbf{B}_1 = [1\ \ 1]$ and $\mathbf{B}_0 = [1\ \ 1]$, $\mathbf{B}_1 = [2\ \ 2]$, respectively, have a smaller coupling width of $w = 1$. For the construction of finite length codes from these ensembles, we use the progressive edge growth (PEG) algorithm to generate random [24] and quasi-cyclic (QC) SC LDPC codes [8], [25].

C. ASYMPTOTIC ANALYSIS

The density evolution (DE) [5], [6] is a technique used for finding the iterative decoding threshold of LDPC ensembles by tracking the probability density function (pdf) of edge messages in the BP decoding. For BEC, a single parameter density (ε) is passed over the graph, whereas for the binary-input additive white Gaussian noise (BI-AWGN) channel, the densities are a continuous function, which can only be approximated by a vector. Consequently, the threshold calculation for BI-AWGN can become rather complex. Therefore, the reciprocal channel approximation (RCA), which is an approximation of the exact DE, was proposed, with which the calculation of threshold is less complex [26].

For employing RCA for the BI-AWGN channel, the single scalar parameter is the signal-to-noise ratio (SNR) denoted by p_σ, and its reciprocal q_σ is defined such that $C(p_\sigma) + C(q_\sigma) = 1$, where $C(\cdot)$ is the capacity function of the BI-AWGN channel. The reciprocal channel function $\Omega(x) = C^{-1}(1 - C(x))$ transforms p_σ into q_σ and vice versa, i.e., $p_\sigma = \Omega(q_\sigma)$ and $q_\sigma = \Omega(p_\sigma)$. For protograph ensembles, we label the edges from both the VN and CN perspectives. Let $e[v_i, r]$ be the r^{th} edge originating from VN v_i; similarly, let $e[c_j, s]$ be the s^{th} edge originating from CN c_j of the protograph. If v_i and c_j are connected, then it follows that $e[v_i, r]$ and $e[c_j, s]$ represent the same edge. At each edge, $\Omega(x)$ is used for transformation between p_σ and q_σ. At all VNs, incoming $p_\sigma^{e[v_i,r']}$ are added to the initial channel value p_σ^{ch} to determine the outgoing message;

$$p_\sigma^{e[v_i,r]} = p_\sigma^{ch} + \sum_{r'\neq r} p_\sigma^{e[v_i,r']}. \tag{3}$$

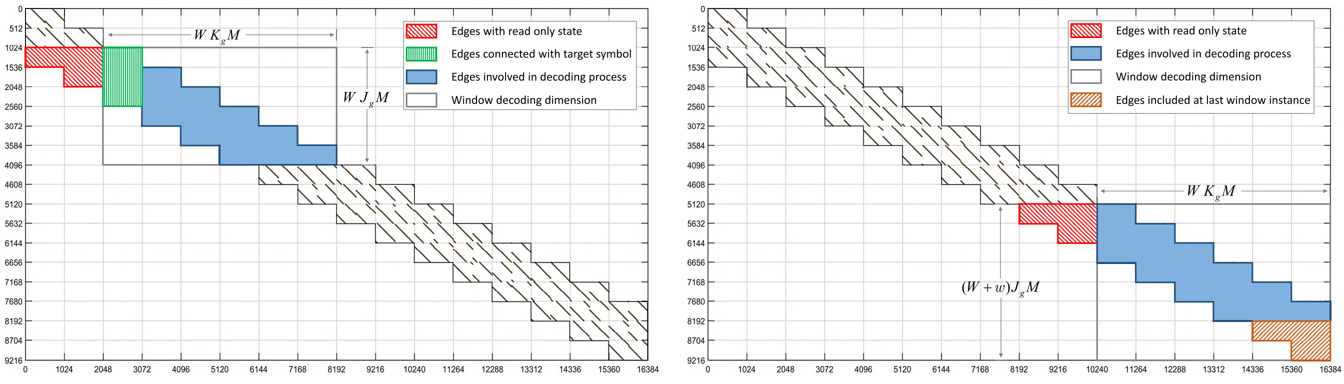
Similarly, at all CNs, incoming $q_\sigma^{e[c_j,s']}$ are added to determine the corresponding outgoing message;

$$q_\sigma^{e[c_j,s]} = \sum_{s'\neq s} q_\sigma^{e[c_j,s']}. \tag{4}$$

This message passing process continues for infinite iterations (in the actual implementation, a sufficiently large number of iterations is considered). The iterative decoding threshold is determined by the smallest value of p_σ^{ch}, such that the unbounded growth of all messages $p_\sigma^{e[v_i,r]}$ become achievable.

III. WINDOWED DECODING OF SC LDPC CODES

Let us first call the decoder with full flooding schedule over the entire code the 'BP' decoder. The band diagonal structure in the parity check matrix of the SC LDPC codes makes it possible to run BP decoding within a window of dimension W ($W \ll L$) [17]. WD utilizes this innate characteristic of the SC LDPC codes by decoding a set of bits in the bitstream progressively until the entire frame is decoded. Localizing the BP decoder inside a window effectively reduces the latency, decoding complexity, and memory requirements, whereas the BP decoding of the entire frame becomes infeasible under practical circumstances, especially when L is large. Consequently, a penalty must be paid in the decoding performance due to restricting the BP decoder inside a window.


W K_g M
W J_g M
Edges with read only state
Edges connected with target symbol
Edges involved in decoding process
Window decoding dimension
W K_g M
(W + w)J_g M
Edges with read only state
Edges involved in decoding process
Window decoding dimension
Edges included at last window instance
0
512
1024
1536
2048
2560
3072
3584
4096
4608
5120
5632
6144
6656
7168
7680
8192
8704
9216
0
1024
2048
3072
4096
5120
6144
7168
8192
9216
10240
11264
12288
13312
14336
15360
16384

(a)

(b)

FIGURE 2. **WD and termination of iWD with $W = 6$ for an SC LDPC code from ensemble $\mathcal{A}$ with $L = 16$ and $M = 512$. (a) WD at the third decoding instance (b) Proposed termination of iWD at $L - W + 1$ (the last) decoding instance.**

A. CONVENTIONAL WINDOWED DECODER (WD)

We consider a PCM $\mathbf{H}$ that is made by lifting a base matrix $\mathbf{B}$ with lifting factor M. The window size W determines the number of CN sub-blocks of size $(J_g \times M)$ inside a window; therefore, the number of CNs inside a window will be WJ_gM. Similarly, the number of VNs inside a window will be WK_gM. Figure 2(a) shows WD that is operational at the third decoding instance. First, WD decodes the leftmost VN sub-block, called *target symbols* (green-highlighted vertical hatched area in Fig. 2(a)) in the window. The WD then slides right and down by a sub-block to decode consecutive target symbols. Let us define the window position p as the position of the target symbols of the current window. Note that since the target symbols have a direct connectivity with other symbols as far as w VN sub-blocks, the edges connected with the previously decoded symbols (red-highlighted forward hatched area in Fig. 2(a)) will pass the output LLRs[1] of these symbols inside the window [20], [21]. The BP decoding is carried out within the window until the target symbols are decoded or the maximum number of iterations is reached. When the window slides down in $\mathbf{H}$ for decoding the next target symbols at p, the edges involved in the decoding of the previous target symbols at $(p - 1)$ maintain the calculated LLRs in the edge memory instead of initializing them again with the received channel LLRs [20]. The number of iterations is greatly reduced by keeping the edge information stored in the edge memory; consequently, the overall decoding complexity is also reduced.

1) LATENCY OF WD

For decoding a frame of LK_gM bits, the WD attempts to decode L target symbols, i.e., the decoding is terminated when WD slides L instances to decode the overall frame [17]. For decoding the target symbols at a particular position p, the latency can be given by $\tau_{\text{WD}}^p = T_R^p(W) + T_D^p(W)$, where $T_R^p(W)$ is the time taken to receive WK_gM bits at p and $T_D^p(W)$ is the time taken to decode the target symbols. Note that if a proper stopping criterion for decoding is applied, i.e., stopping when the target symbol's parity checks are satisfied, then τ_{WD}^p varies with p.

To simplify the analysis, we assume the decoder runs for a fixed number of iterations and the same processing power is used. The latency calculated under this assumption is an upper bound to the actual latency of the decoder with a stopping rule. Let T_F and T_D be the time taken to receive all symbols in a frame and the time taken to decode the frame using the BP decoder, respectively

$$T_R^p(W) = \begin{cases} \dfrac{WK_gM}{LK_gM}T_F = \dfrac{W}{L}T_F, & \text{for } 1 \le p \le L-W+1 \\ \dfrac{(L-p+1)K_gM}{LK_gM}T_F = \Big(\dfrac{L-p+1}{L}\Big)T_F, & \\ \qquad\qquad \text{for } L-W+1 < p \le L. \end{cases} \tag{5}$$

Similarly,

$$T_D^p(W) = \begin{cases} \dfrac{W}{L}T_D, & \text{for } 1 \le p \le L-W+1 \\ \Big(\dfrac{L-p+1}{L}\Big)T_D, & \text{for } L-W+1 < p \le L. \end{cases} \tag{6}$$

Under the same processing power, the latency of the WD is related to that of the BP decoder as

$$\tau_{\text{WD}}^p = \begin{cases} \dfrac{W}{L}\tau_{\text{BP}}, & \text{for } 1 \le p \le L-W+1 \\ \Big(\dfrac{L-p+1}{L}\Big)\tau_{\text{BP}}, & \text{for } L-W+1 < p \le L. \end{cases} \tag{7}$$

The latency for WD is reduced by a factor of W/L compared with the BP decoder, up to the target symbols at $p = L - W + 1$. For the following target symbols, the factor is changed to $(L - p + 1)/L$. The memory requirement is reduced by the factor W/L for WD.

Let us now consider the overall latency for the entire frame. For simplicity, we assume the reception of a sub-block and the

[1] The sum of received channel LLR, and the LLRs from each connected CN.

decoding for the corresponding window do not concurrently occur. We then have

$$\tau_{\mathrm{WD}}^{F} = T_F + \sum_{p=1}^{L-W+1} \frac{W}{L} T_D + \sum_{i_p=1}^{W-1} \left(\frac{W-i_p}{L}\right) T_D, \quad (8)$$

where τ_{WD}^{F} is the latency for decoding an entire frame by WD.

B. IMPROVED WINDOWED DECODER (iWD)

In this section, an improved windowed decoder (iWD) is proposed. The proposed iWD consists of an early termination method and edge message reuse of decoded target symbols. Another scheme for mitigating error floor is also introduced in this paper, in which we amplify the edge messages. For independent evaluation of the schemes, we separately call the combined decoder 'iWD-M' (here, M means 'modified').

1) EARLY TERMINATION OF WINDOWED DECODING

Note that in the conventional WD, the decoding window slides down in $\mathbf{H}$ by decoding target symbols consecutively until it processes the last sub-block as a target. When $p = L - W + 1$, the right side of the window meets the right boundary of the PCM. Subsequently, (i.e., for $p > L-W+1$), the decoding window slides out from the PCM, then the effective dimension of the window (the size of the processed VNs and CNs) reduces until the decoding of the last group of symbols as the target finishes.

On the other hand, we propose to stop sliding the window when it reaches $p = (L - W + 1)$ and we process all the symbols inside the window as target symbols. Here, the vertical dimension of the window is extended such that the remaining w CNs are included in the window. Figure 2(b) illustrates the decoding termination technique of iWD. Since we know that the low degree CNs at the terminated[2] side of the graph are the basis for better performance of the SC LDPC codes, at the last window position iWD attempts to decode all symbols inside the window. The graph inside the last window position of iWD can be viewed as an SC LDPC code with $L = W$, where both ends of the graph can be viewed as terminated (if we assume that perfect decoded information is fed at the left end).

Algorithm 1 explains the decoding termination setting of iWD. The vertical dimension of the last window position is changed to $(W + w)J_gM$, whereas the horizontal dimension remains unchanged. Significant latency improvement can be attained from iWD in terms of decoding an entire frame since the window does not slide any further from $p = L - W + 1$;

$$\tau_{\mathrm{iWD}}^{F} = T_F + \sum_{p=1}^{L-W+1} \frac{W}{L} T_D. \quad (9)$$

Analysis in Asymptotic Settings: We now analyze the effect of the decoding termination method of iWD in asymptotic settings. We assume the transmission over the BI-AWGN channel with unity symbol power and noise variance σ^2. The RCA-based density evolution is used for the analysis. The noise threshold σ_p^* is calculated at each window position p. For the target symbols, we pre-decide a threshold δ, the target SNR of RCA DE; if all $p_\sigma^{e[v_i,r]}$ at the target symbols exceed δ, the RCA DE iteration is stopped. When the window slides, the edges associated with the previous target symbols are preset to δ. The WD threshold σ_p^* at p is defined as the maximum σ for which the RCA DE stopping condition is satisfied.

Algorithm 1 iWDTermSetting($p, L, W, M, (J_g, K_g)$)

```
Inputs: $p, L, W, M, (J_g, K_g)$
    Horizontal Window Dimension ← $WK_gM$
if $p < (L - W + 1)$ then
    /* if window as not reached the boundary of PCM */
    Vertical Window Dimension ← $WJ_gM$
    Target Symbols ← $K_gM$
else
    /* if window reached the boundary of PCM */
    Vertical Window Dimension ← $(W + w)J_gM$
    Target Symbols ← $WK_gM$
end if
```

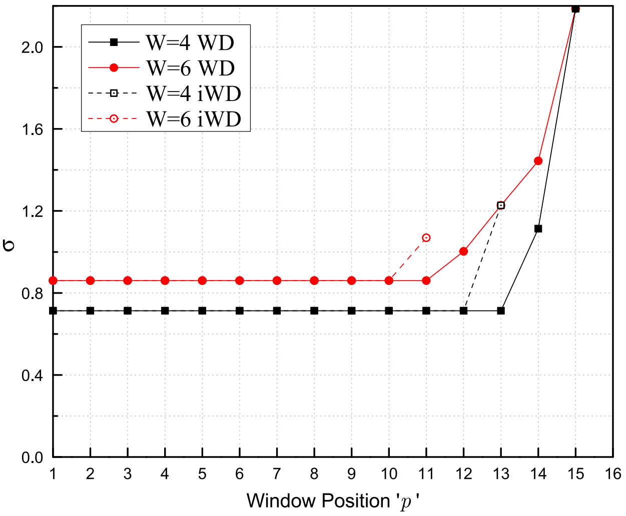


FIGURE 3. Threshold (σ_p^*) comparison at each window position between conventional WD and iWD for ensemble A.

Figure 3 shows the WD thresholds (σ_p^*) for the target symbols at each p of the SC LDPC ensemble of $L = 16$ formed from ensemble A. We preset $\delta = 15$ and find WD thresholds for two window sizes: $W = 4$ and 6. The threshold σ_p^* is constant before the window position p reaches $L - W + 1$ where the decoding enters the termination phase. In the conventional WD, for the subsequent window positions, the effective vertical and horizontal dimensions of the window shrink and the lower degree CNs at the terminated side of the graph are involved in WD; therefore, the WD threshold increases

[2] Additional wK_g CNs with lower degree at both ends of the SC LDPC graph can be viewed as graph termination.

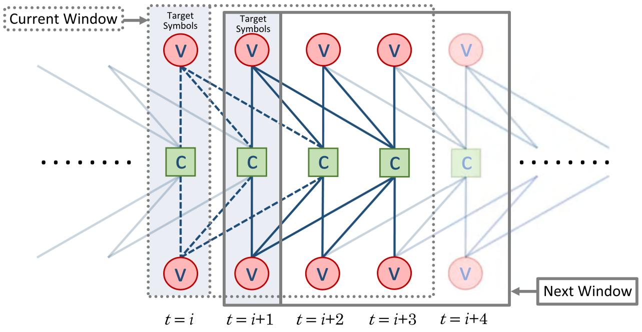


FIGURE 4. Decoding window sliding from target position $p = i$ to $i + 1$ over the ensemble A protograph.

gradually as the window moves out. In the case of iWD, it can be seen that at the window position $(L - W + 1)$, which is the last position, the vertical dimension of the window is extended by w to include all of the remaining CNs; thus, the WD threshold of the last window increases by processing all remaining symbols as target symbols. Even though the threshold of the last group of symbols in the conventional WD is higher than that of iWD, the earlier threshold increase at $p = L - W + 1$ for iWD is more beneficial to bit error rate (BER) performance, as shown in Section IV.

2) MESSAGE REUSE

In [17], the WD of the SC LDPC codes was proposed to provide flexibility in decoding latency over BEC. Since the WD was first introduced for BEC, it is natural that when WD moves to the next p, the edges of the previous target symbols connected to the symbols in current window will pass the decoding decisions: erasure or recovered binary values. For the BI-AWGN channel, the messages are LLRs for an LLR-based BP decoder; therefore, it was considered that the output LLR of symbols are passed to the current window through the connected edges; i.e., the edges of the previous target symbols are initialized with the output LLRs of these symbols [20], [21]. If the previous target symbols are erroneous, the errors propagates to the current window and interferes with the decoding of the target symbols. For a small W, the effect of error propagation becomes severe.

In this section, we show that for LLRs from the previous target symbols, the extrinsic LLRs (LLRs calculated at line 7 of Algorithm 2) retained in the edges are better than the output LLRs. Figure 4 depicts the sliding window of size $W = 4$ moving from $p = i$ to $i + 1$ over the SC LDPC protograph of ensemble A. The target symbols are connected to $w + 1$ CNs, as indicated by the dashed edges in Fig. 4. The rightmost CNs in the window are connected to the VNs that have a lower degree, i.e., 1 and 2 in Fig. 4, because of the truncation. It is shown that the BER of the rightmost VNs in the window is poorer [21]. Therefore, less reliable information can propagate from right to left in the decoding window. The output LLRs of the previous target symbols include less reliable messages from the right neighbor check node. Passing the output LLRs of the previous target symbols to the next window also breaks the message independence; using the extrinsic LLRs is thus better choice than the output LLRs.

Algorithm 2 iWD($\mathbf{H}, L, W, M, I_{max}, (J_g, K_g)$)

```
    Inputs: $\mathbf{H}, L, W, M, I_{max}, (J_g, K_g)$
 1: Set $\mathcal{I}(0) \leftarrow \varnothing$
 2: for $p \leftarrow 1$ to $(L - W + 1)$ do
        /* window dimensions at each p */
 3:     iWDTermSetting($p, L, W, M, (J_g, K_g)$)
        /* initialization of edge messages */
 4:     iWDInit($p, \mathcal{I}(p-1), W, M, (J_g, K_g)$)
        /* BP iterations start here */
 5:     for $I \leftarrow 1$ to $I_{max}$ do
 6:         Check Nodes Processing:
              $l(e[c_j, y]) = 2\tanh^{-1}(\prod_{y' \neq y} \tanh(\frac{1}{2} l(e[c_j, y'])))$
 7:         Variable Nodes Processing:
              $l(e[v_i, x]) = l(P_{ch}^{v_i}) + \sum_{x' \neq x} l(e[v_i, x'])$
 8:         Hard Decision:
              $l(Q_{out}^{v_i}) = l(P_{ch}^{v_i}) + \sum_i l(e[v_i, x])$
              $\hat{c_{v_i}} = \begin{cases} 1 & \text{for } l(Q_{out}^{v_i}) > 0, \\ 0 & \text{for } l(Q_{out}^{v_i}) \leq 0. \end{cases}$
 9:         if $I = I_{max}$ or $\hat{\mathbf{c}}_p^T \tilde{\mathbf{H}}_p = 0$ then
              Break;
10:         end if
11:     end for
12:     UpdateSet($\mathcal{I}(p)$)
13: end for
```

Algorithm 2 describes the decoding process of iWD. Except for initialization, the same BP decoding runs inside a window until the target symbols are decoded. Let us denote '$\mathcal{I}(p)$' as the set of VN indices which are included in the window at p, 'l' as an LLR and '$e[\cdot, \cdot]$' as the edge index. Algorithm 3 describes the initialization step of the decoder. The edges associated with the VNs which were not included in $\mathcal{I}(p-1)$ are initialized by the received channel LLRs. Let $\mathcal{I}(0)$ be an empty set. Therefore, all VNs are initialized by the received channel LLRs for WD at the first position $p = 1$. At subsequent window positions, only the edges associated with the newly included rightmost VNs are initialized by the channel LLRs. The decoder makes decisions only for target symbols at the last step of the decoding iteration, i.e., CNs connected with the target symbols are checked for syndrome satisfaction. Note that at line 9 of Algorithm 2, $\tilde{\mathbf{H}}_p$ is a submatrix showing the connectivity between target symbols and the associated CNs at p. At $p = L - W + 1$, all symbols inside the window are target symbols; therefore, $\tilde{\mathbf{H}}_p$ has the same dimension as that of the window at this position. Before incrementing p, $\mathcal{I}(p)$ is updated at line 12 by the function UpdateSet($\mathcal{I}(p)$).

Algorithm 3 iWDInit($p, \mathcal{I}(p-1), W, M, (J_g, K_g)$)

```
Inputs: $p, \mathcal{I}(p-1), W, M, (J_g, K_g)$
for $i \leftarrow (p-1)K_gM$ to $((W+p-1)K_gM)-1$ do
    if $i \notin \mathcal{I}(p-1)$ then
        $L(e[v_i, x]) = L(P_{ch}^{v_i})$
    end if
end for
```

Algorithm 4 iWDAmpInit($p, \mathcal{I}(p-1), W, M, (J_g, K_g)$)

```
Inputs: $p, \mathcal{I}(p-1), W, M, (J_g, K_g)$
for $i \leftarrow (p-1)K_gM$ to $((W+p-1)K_gM)-1$ do
    if $i \notin \mathcal{I}(p-1)$ then
        $L(e[v_i, x]) = L(P_{ch}^{v_i})$
    else
        if $\hat{\mathbf{c}}_{p-1}^T \tilde{\mathbf{H}}_{p-1} = 0$ then
            $L(e[v_i, x]) = L(e[v_i, x]) \times \alpha$
        end if
    end if
end for
```

3) MESSAGE AMPLIFICATION FOR ERROR FLOOR MITIGATION

In both WD and iWD, the structure of an ensemble not only affects the performance in the waterfall region, but also in the error floor region. For instance, a code from ensemble C performs better than a code from ensemble B in the waterfall region; but worse in the error floor region under WD. Because only codes with large girth (e.g., 10) are used, we can conclude that the high error floor of the code from ensemble C is mainly due to the combination of the ensemble structure and the WD.

We propose an error floor lowering technique that uses an amplified version of the edge messages of the previous window. Algorithm 4 describes the modified initialization step of the decoder where α is the amplifying factor. When the parity checks are satisfied for the target symbols, the edge messages at that specific p are considered reliable. Before the window shifts, we amplify the messages by the factor of α for all edges that are involved again in the next window. The lower degree VNs in the right side of the window contribute largely to the error floor, as shown in the next section. The effect of the lower degree VNs is mitigated by the amplification method. We call iWD with amplification method as 'iWD-M' (here, M denotes 'modified') for independent evaluation of both schemes. In iWD-M, line 4 of Algorithm 2 is replaced by Algorithm 4.

IV. NUMERICAL RESULTS: PERFORMANCE EVALUATION OF iWD

To demonstrate the effectiveness of the iWD, the finite length performance and the corresponding decoding complexity are analyzed for both WD and iWD for a code taken from the classical SC LDPC ensemble (i.e., ensemble A). The code used in the Monte Carlo simulation is a QC SC LDPC code made using the method in [25]. The code was constructed with parameters $L = 16$ and $M = 512$; therefore, the code length $n = L \times M \times K_g = 16384$ and the code rate $R = 0.4375$ are obtained after lifting the graph. The code was constructed such that its girth is larger or equal to 10. The BP threshold for this ensemble is $\sigma^* = 0.951730$ ($E_b/N_o{}^* = 1.009644$ dB), evaluated by the RCA-based DE. The maximum number of iterations, I_{max}, is set as 100. Note that iWD refers to Algorithm 2 with the initialization of Algorithm 3 and iWD-M refers to Algorithm 2 with the initialization of Algorithm 4.

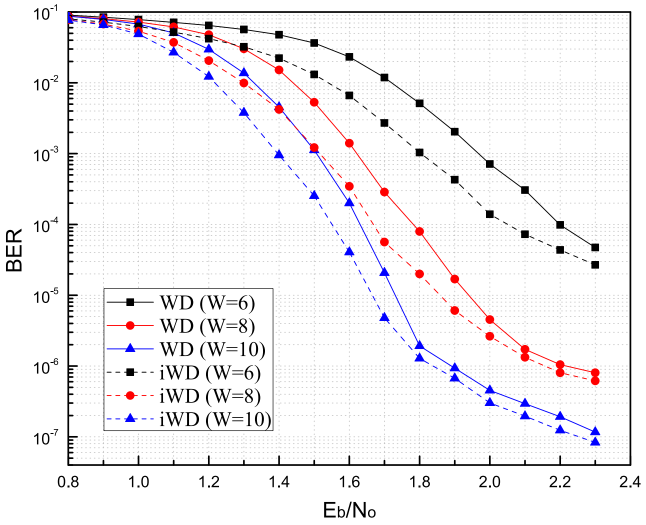


FIGURE 5. BER comparison between iWD and WD.

A. BER PERFORMANCE COMPARISON

Figure 5 shows the performance curves of WD and iWD for window sizes $W = 6$, 8, and 10. As can be seen, the performance gap is largest when W is small. For $W = 6$ and a bit error rate (BER) of 10^{-3}, the gap between the performance curves is about 0.17 dB. For $W = 8$ and 10, the performance gap is reduced to about 0.11 dB. One of the benefits of using iWD is the performance improvement with small W. The BERs at each VN sub-block indexed with t are shown in Fig. 6 to examine the separate effects of the new termination (Algorithm 1) and the message reuse (Algorithm 3) on the performance of WD at $E_b/N_o = 1.9$ dB. We can observe significant improvement in BER at all sub-blocks when Algorithm 3 is applied over the conventional WD. As the window moves down in $\mathbf{H}$, the BER improvement of iWD increases because, instead of the output LLRs, the extrinsic LLRs stored in the edge memory are fed to consecutive windows. For the conventional WD operated with the new termination setting defined in Algorithm 1, BER improvement is observed for $t \geq 11$. This is because, at the last window position $p = 11$, the CNs with lower degrees have a significant impact on performance because they pass more reliable information inside the window.

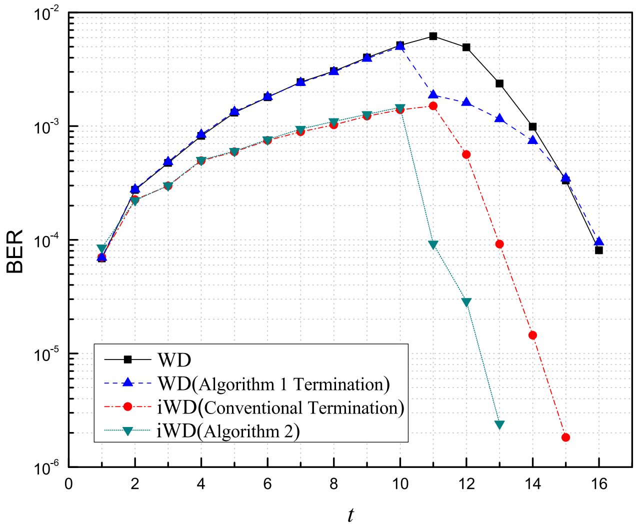


FIGURE 6. BER at each VN sub-block position 't' with $W = 6$ (at $E_b/N_o = 1.9$ dB).

B. COMPLEXITY COMPARISON

For the same simulation settings, we analyzed the operational complexity of WD and iWD with two metrics: the normalized average VN processing per frame and the average number of iterations.

For a single frame, we normalized the number of VN processed with the length of the codeword; consequently, for multiple frames, the normalized average number of VN processed per frame is given by

$$\Lambda_{avg} = \frac{1}{f}\sum^{f}\frac{\sum_{i=1}^{n}\Lambda_i}{n}, \tag{10}$$

where Λ_i is the number of times the i^{th} VN was processed during the decoding of an entire frame by WD, f is the total number of frames received, and Λ_{avg} is the normalized average VN processed per frame. In Fig. 7 the normalized average VN processing per frame is plotted with respect to E_b/N_o for $W = 6$, 8, and 10. In the waterfall region of the performance curves for each W, a notable difference is observed in the normalized average number of VNs processed. For $W = 6$ and $E_b/N_o = 1.8$ dB, about 127 × n (here, n=16384) more VNs are processed by WD than by iWD. The iWD has 42.5% less complexity compared to WD. A similar trend is observed for $W = 8$ and 10. Figure 8 shows the average number of iterations taken by the decoder at each window position to decode the target symbols (at $E_b/N_o = 1.6$ dB). Until $p = L-W+1$, iWD takes fewer iterations than WD as the window slides down in $\mathbf{H}$. This indicates that, besides improving the performance, iWD also improves the convergence behavior of the decoder. Since iWD stops earlier at $p = L - W + 1$, the complexity is significantly reduced for decoding a frame.

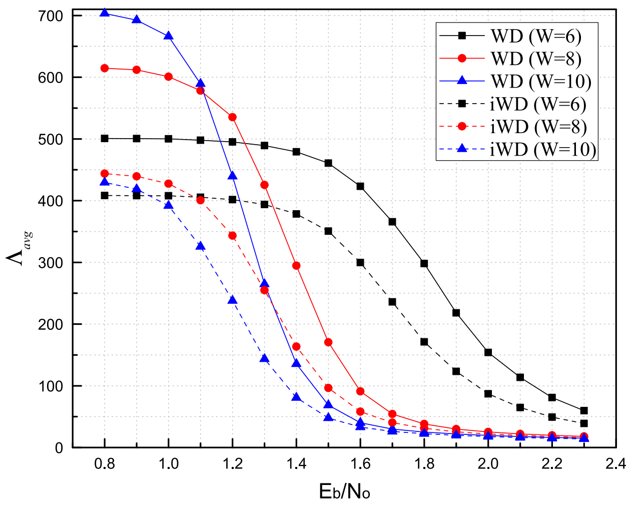


FIGURE 7. Average number of VN processings per frame for iWD and WD.

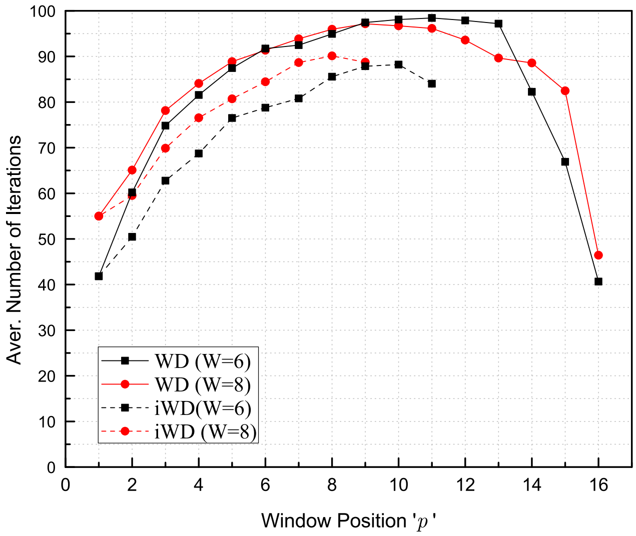


FIGURE 8. Number of iterations at each window position (at $E_b/N_o = 1.6$ dB).

Similar results in terms of performance and complexity were observed when iWD was applied to other SC LDPC codes constructed with random edge permutation using the PEG algorithm for large girth ≥ 10. In order to show the generality of iWD over any code construction technique, the simulations in the next sub-section are performed with codes constructed using the PEG algorithm.

C. PERFORMANCE ANALYSIS FOR OTHER ENSEMBLES

In this sub-section, the behavior of iWD is examined for codes from SC LDPC ensembles B and C, in order to establish a link between the code structure and the performance gain achievable by iWD. The codes are constructed using the PEG algorithm with random edge connectivity. Code parameters are kept the same as the code discussed in the previous section, whereas the rate of codes from these ensembles becomes $R = 0.46875$. Let us denote the code from ensembles B and C by $\mathcal{C}_{\mathrm{B}}$ and $\mathcal{C}_{\mathrm{C}}$, respectively. Efficacy of iWD is confirmed for both codes in Fig. 9, in which the

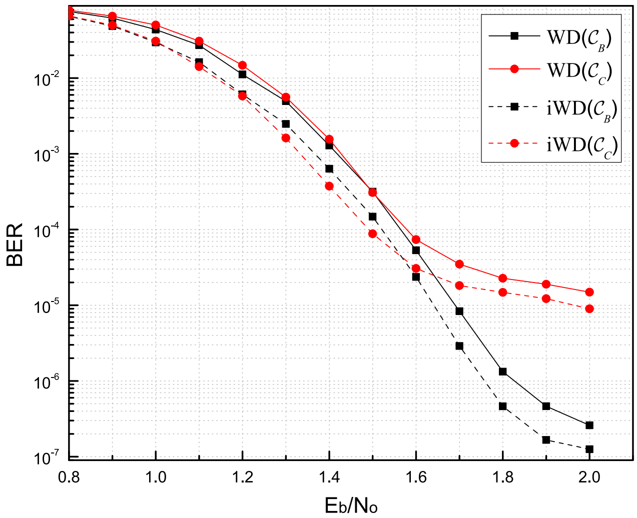


FIGURE 9. BER performance of the codes from ensemble B and C under WD and iWD ($W = 6$).

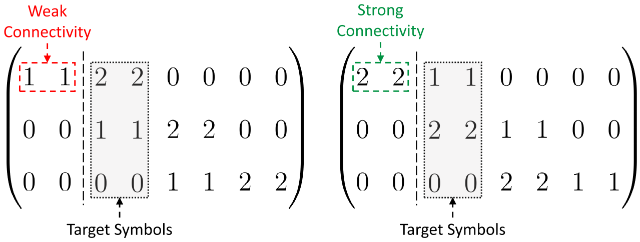


FIGURE 10. Window configurations of $W = 3$ for SC LDPC ensemble B and C.

performance curves for $\mathcal{C}_\mathrm{B}$ and $\mathcal{C}_\mathrm{C}$ are plotted for $W = 6$. In the waterfall region, it can be seen that iWD has better performance than WD for both $\mathcal{C}_\mathrm{B}$ and $\mathcal{C}_\mathrm{C}$. In Example 1, we explain the difference in performance of these codes. Let us define 'window configuration' as the arrangement of edges in a decoding window.

Example 1: Consider window configurations for SC LDPC ensembles B and C with $W = 3$ and at any p in the middle of **B** (i.e., not at terminated sides of **B**) as shown in Fig. 10. They can be viewed as protographs in conjunction with the dimension of the window. They can also be viewed as edges of codes with lifting factor $M = 1$ inside the dimension of the window of $W = 3$. It can be seen that the number of edges highlighted with a red box (ensemble B) is less than that of the edges highlighted with a green box (ensemble C); these edges correspond to the previous target symbols of the previous position of the window. We call these edges secondary edges (SEs) connected to the window configuration. Since in iWD, SEs share the extrinsic LLRs with the edges inside the window, ensemble C seems to have greater extrinsic information sharing capability due to the higher number of SEs than in ensemble B. The greater number of SEs in ensemble C indicates that iWD will have a more reliable extrinsic estimate of previous target symbols.

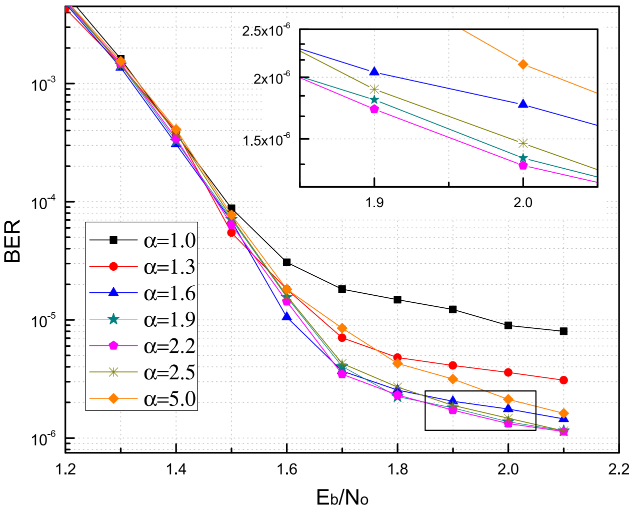


FIGURE 11. BER performance of $\mathcal{C}_\mathrm{C}$ under iWD-M for different values of α and $W = 6$.

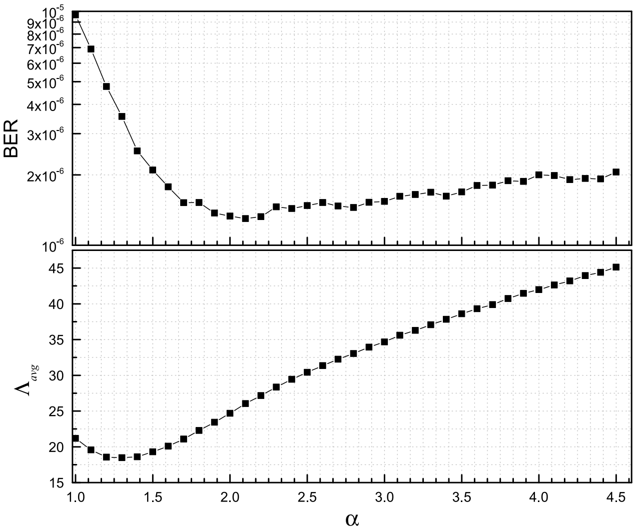


FIGURE 12. BER and average VN processing per frame versus α at $E_b/N_o = 2.0$ dB for $\mathcal{C}_\mathrm{C}$ and iWD-M ($W = 6$).

D. NUMERICAL RESULTS IN ERROR FLOOR MITIGATION

In Fig. 9, it can be seen that the performance improvement of $\mathcal{C}_\mathrm{C}$ is greater than that of $\mathcal{C}_\mathrm{B}$; however, $\mathcal{C}_\mathrm{C}$ has an apparent weakness whereby it hit the error floor at a higher error rate than $\mathcal{C}_\mathrm{B}$. As mentioned earlier, the window configuration of $\mathcal{C}_\mathrm{C}$ in Fig. 10 shows that the rightmost symbols have degree 1 and these symbols will only pass their received channel LLRs without enhancement via iterations.

Under the same simulation setup as that in the previous sub-section, Fig. 11 shows the performance curves of iWD-M with different values of α for the code from ensemble C and $W = 6$. The error floor is significantly lowered with the application of the message amplification. The error floor improvement is dependent on the value of α, which can be optimized empirically. Figure 12 shows the BER at

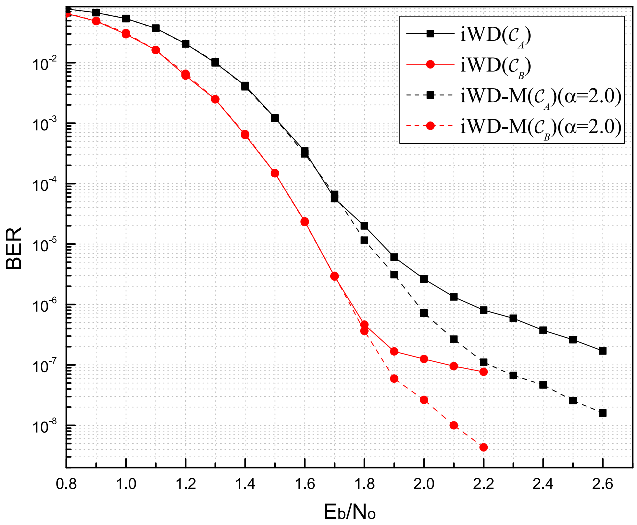


FIGURE 13. BER performance of $\mathcal{C}_\mathsf{A}$ ($W = 8$) and $\mathcal{C}_\mathsf{B}$ ($W = 6$).

$E_b/N_o = 2.0$ dB versus α for the same setting as that in Fig. 11. Let α^* be the optimal value of α in terms of performance for the given SNR (here, $\alpha^* \approx 2.1$).

Figure 12 also shows the normalized average number of VNs processed per frame (Λ_{avg}) with α in order to analyze the effect of α on decoding complexity. Since Λ_{avg} increases with α for $\alpha > \alpha^*$, it is noticed that the good trade-off between performance and complexity occurs at $\alpha \leq \alpha^*$. This behavior is observed to be general, even though the specific value of α^* is dependent on the ensemble and the window size. The generality of this technique is confirmed in Fig. 13, in which the BERs of codes from ensembles A and B are shown for $\alpha = 2.0$ for iWD-M. Significant improvement in the error floor region is observed for both $\mathcal{C}_\mathsf{A}$ and $\mathcal{C}_\mathsf{B}$.

V. CONCLUSIONS

In this paper, we proposed an improved WD of SC LDPC codes. First, the proposed decoder includes a new termination method, which improves the performance and reduces the decoding latency simultaneously. The performance gain was analyzed using an RCA-based asymptotic analysis. The second technique utilizes the edge message reuse that mitigates error propagation and improves the BER performance. Lastly, we proposed internal message amplification of WD, which significantly improves the error floor performance. With the combination of the proposed techniques, iWD-M gives considerable performance improvement in both the water-fall and error floor regions. Simulations for codes from different ensembles verified the validity and generality of the proposed decoding method.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, Dept. Elect. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 1963.

[2] D. A. Spielman, "Computationally efficient error-correcting codes and holographic proofs," Ph.D. dissertation, Dept. Math., Massachusetts Inst. Technol., Cambridge, MA, USA, 1995.

[3] D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 32, no. 18, p. 1645, Aug. 1996.

[4] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[5] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

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

[7] H. Xiao and A. H. Banihashemi, "Improved progressive-edge-growth (PEG) construction of irregular LDPC codes," *IEEE Commun. Lett.*, vol. 8, no. 12, pp. 715–717, Dec. 2004.

[8] X.-Q. Jiang, H. Hai, H.-M. Wang, and M. H. Lee, "Constructing large girth QC protograph LDPC codes based on PSD-PEG algorithm," *IEEE Access*, vol. 5, pp. 13489–13500, Apr. 2017.

[9] T. Lestable and E. Zimmermann, "LDPC options for next generation wireless systems," in *Proc. 14th Wireless World Res. Forum (WWRF)*, 2005, pp. 1–10.

[10] A. J. Felström and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.

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

[12] A. Sridharan, M. Lentmaier, D. J. Costello, Jr., and K. S. Zigangirov, "Convergence analysis of a class of LDPC convolutional codes for the erasure channel," in *Proc. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Oct. 2004, pp. 953–962.

[13] S. Kudekar, T. J. Richardson, and R. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.

[14] C. Méasson, A. Montanari, and R. Urbanke, "Maxwell construction: The hidden bridge between iterative and maximum *a posteriori* decoding," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5277–5307, Dec. 2008.

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

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

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

[18] G. E. Corazza, A. R. Iyenger, M. Papaleo, P. H. Siegel, A. Vanelli-Coralli, and J. K. Wolf, "Latency constrained protograph-based LDPC convolutional codes," in *Proc. 6th Int. Symp. Turbo Codes Iterative Inf. Process.*, Brest, France, Sep. 2010, pp. 6–10.

[19] M. Lentmaier, M. M. Prenda, and G. P. Fettweis, "Efficient message passing scheduling for terminated LDPC convolutional codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Saint Petersburg, Russia, Jul./Aug. 2011, pp. 1826–1830.

[20] N. U. Hassan, M. Schlüter, and G. P. Fettweis, "Fully parallel window decoder architecture for spatially-coupled LDPC codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[21] N. U. Hassan, A. E. Pusane, M. Lentmaier, G. P. Fettweis, and D. J. Costello, "Non-uniform windowed decoding schedules for spatially coupled codes," in *Proc. IEEE Globe Commun. Conf. (GLOBECOM)*, Atlanta, GA, USA, Dec. 2013, pp. 1862–1867.

[22] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," JPL, Pasadena, CA, USA, INP Progr. Rep. 42–154, Aug. 2003.

[23] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866–4889, Sep. 2015.

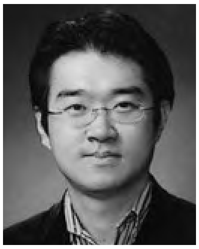
[24] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Progressive edge-growth Tanner graphs," in *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, San Antonio, TX, USA, Nov. 2001, pp. 995–1001.

[25] Z. Li and B. V. K. V. Kumar, "A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2004, pp. 1990–1994.

[26] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2000.

INAYAT ALI received the B.E. degree in electronics engineering from PAF-KIET, Karachi, Pakistan, in 2009, and the M.E. degree in telecommunication engineering from Hamdard University, Karachi, Pakistan, in 2011. He is currently pursuing the Ph.D. degree in information and communication engineering from Sungkyunkwan University, Suwon, South Korea. His research interests include LDPC codes, SC LDPC codes, modern coding theory, and information theory.

JONG-HWAN KIM received the B.S.E. degree in information and communication engineering from Sungkyunkwan University, Suwon, South Korea, in 2010, where he is currently pursuing the Ph.D. degree in information and communication engineering. His research interests include polar codes, LDPC codes, coding theory, and wireless communication systems.



SANG-HYO KIM received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Seoul National University, Seoul, South Korea, in 1998, 2000, and 2004, respectively. From 2004 to 2006, he was a Senior Engineer with Samsung Electronics. He visited the University of Southern California as a Visiting Scholar from 2006 to 2007. In 2007, he joined the College of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea, where he is currently an Associate Professor. His research interests include modern coding theory, wireless multi-terminal communications, signal design, and secure communications. He has served as an Editor for the *Transactions on Emerging Telecommunications Technologies* and the *Journal of Communications and Networks* from 2013.

HEEYOUL KWAK received the B.S. degree in electrical and computer engineering from Seoul National University, Seoul, south Korea, in 2013, where he is currently pursuing the Ph.D. degree in electrical engineering and computer science. His area of research interests includes error-correcting codes, coding theory, and coding for memory.

JONG-SEON NO (S'80–M'88–SM'10–F'12) received the B.S. and M.S.E.E. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1981 and 1984, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California at Los Angeles, CA, USA, in 1988. He was a Senior MTS with Hughes Network Systems from 1988 to 1990. He was an Associate Professor with the Department of Electronic Engineering, Konkuk University, Seoul, from 1990 to 1999. He joined the Faculty of the Department of Electrical and Computer Engineering, Seoul National University, in 1999, where he is currently a Professor. His area of research interests includes error-correcting codes, sequences, cryptography, LDPC codes, interference alignment, and wireless communication systems. He was a recipient of the IEEE Information Theory Society Chapter of the Year Award in 2007. From 1996 to 2008, he served as the Founding Chair of the Seoul Chapter of the IEEE Information Theory Society. He was the General Chair of Sequence and Their Applications 2004, Seoul. He served as the General Co-Chair of the International Symposium on Information Theory and Its Applications 2006 and the International Symposium on Information Theory 2009, Seoul. He has been a Co-Editor-in-Chief of the IEEE Journal of Communications and Networks since 2012.