Performance Bounds for Quantized Spatially Coupled LDPC Decoders Based on Absorbing Sets

Homayoon Hatami*, David G. M. Mitchell†, Daniel J. Costello, Jr*, and Thomas Fuja*
*Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, Indiana, USA
Email: {hhatami,dcostel1,tfuja}@nd.edu
†Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, USA
Email: dgmm@nmsu.edu

***Abstract*—Absorbing sets are known to be the primary factor in the error-floor performance of low-density parity-check (LDPC) codes with message passing decoders over the additive white Gaussian noise (AWGN) channel. Besides showing excellent waterfall performance, spatially coupled LDPC (SC-LDPC) codes that are constructed by an edge spreading technique are known to have fewer cycles and absorbing sets than their block code counterparts, and therefore to exhibit better error-floor performance. Based on our previously obtained results for quantized LDPC block decoders, we derive lower bounds on the performance of quantized SC-LDPC decoders, including both a flooding schedule decoder and a sliding window decoder. Numerical simulation results confirm the accuracy of the obtained bounds and show that, for quantized decoders, properly designed SC-LDPC codes have better error-floor performance than their underlying LDPC block codes.**

I. Introduction

Low-density parity-check (LDPC) codes [1] are a powerful class of capacity approaching channel codes. One major benefit of these codes is that they can be decoded with an iterative belief propagation (BP) message passing decoder whose complexity grows linearly with block length, which is a significant advantage over maximum-likelihood decoders in term of complexity. Spatial coupling (connecting) L copies of an LDPC block code (LDPC-BC) Tanner graph creates a coupled chain with a convolutional structure, called a spatially coupled LDPC code (SC-LDPCC), with improved performance [2]–[4]. Iterative BP decoding using the standard flooding schedule over the entire code graph can be used to achieve capacity approaching performance with SC-LDPCCs, but the decoding latency and complexity is large. Therefore, in [5], a sliding *window decoder* was proposed to allow low latency operation with minimal loss in performance.

One important aspect of any decoder implementation is finite precision or quantization, which impacts both the *waterfall* and *error-floor* performance of message passing decoders. The main factor affecting the error-floor performance of message passing decoders for LDPC codes was identified as *trapping sets* in [6]. A special class of particularly harmful trapping sets are called *absorbing sets*, which are combinatorially-defined structures existing in the Tanner graph of an LDPC code [7]. In [8], we introduced a *code-independent* lower bound on the frame error rate (FER) of a quantized decoder for an LDPC-BC based on absorbing sets, which can predict the effect that each absorbing set has on code performance. Typically, there is one type of absorbing set that dominates the error-floor performance. If the code-independent bound is calculated for this absorbing set and multiplied by the absorbing set multiplicity in the code graph, a good estimate of the error-floor performance is obtained.

Array based LDPC codes are a class of (γ, p)-regular quasi-cyclic LDPC-BCs defined by the parameter pair (γ, p), where p is a prime and $p \geq \gamma$, with attractive implementation properties [9]. Also, thanks to the quasi-cyclic structure of these codes, it is possible to completely characterize their absorbing sets [7], [10]. Several examples of designing array-based SC-LDPCCs with the aim of decreasing the average number of absorbing sets per symbol are contained in [10]–[13]. Spatial coupling via edge spreading [14] is known to break cycles and absorbing sets in the code graph [10], [11]. Hence, the error-floor is expected to be better for SC-LDPCCs than for their underlying LDPC-BCs.

In this paper, we consider two different BP decoding schedules for SC-LDPCCs: the standard flooding schedule[1] used for LDPC-BCs and a sliding window decoder used for low latency applications. We extend the code-independent bound of [8] to bound both the FER of the standard decoder and the block error rate (BLER) of the window decoder. We examine the proposed lower bounds for array-based SC-LDPCCs with quantized sum-product algorithm (SPA) and quantized min-sum algorithm (MSA) decoders.[2] Finally, we compare the lower bounds of quantized SC-LDPCC decoders with the lower bounds of their underlying quantized LDPC-BC decoders and provide simulation results to verify the accuracy of the bounds.

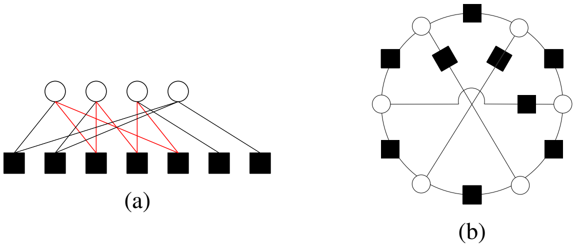
II. Preliminaries

1) Absorbing sets: Let A denote a subset of cardinality a of all the variable nodes in a Tanner graph. Let $E(A)$ and $O(A)$ represent the subsets of check nodes connected to variable nodes in A with even and odd degrees, respectively, where $|O(A)| = b$. Here, A is called an (a, b) *trapping set*. Also, A is defined to be an (a, b) *absorbing set* if each variable node in A is connected to fewer check nodes in $O(A)$ than $E(A)$. As an illustration, Fig. 1(a) shows a $(4, 2)$ absorbing set, with variable nodes and check nodes represented by ○

This material is based upon work supported by the National Science Foundation under Grant No. CCSS-1710920.

[1] Henceforth, we refer to this as the standard decoder.

[2] The proposed method is not restricted to array-based SC-LDPCCs. These codes are examined due to the known characteristics of their absorbing sets.


(a)

(b)

Fig. 1: (a) $(4,2)$ absorbing set, and (b) $(6,0)$ absorbing set.

and ■, respectively. This structure appears often in $(3,K)$-regular LDPC codes and is the dominant cause of error-floors for quantized SPA decoders. Also, Fig. 1(b) shows a $(6,0)$ absorbing set, which was shown in [15] to be the dominant cause of error-floors in $(3,K)$-regular LDPC codes for quantized MSA decoders.

2) Code-Independent Lower Bound: In [8], a lower bound on the performance of a quantized LDPC-BC decoder was proposed based on a single *dominant* absorbing set A.[3] The bound is denoted as $\lambda_R\left(E_b/N_0(\text{dB})\right)$, where R is the rate of the code containing A. Henceforth we denote this bound simply as λ.[4] A lower bound on the FER of any code containing N copies of A is then approximated by $N\lambda$.[5] The bound was obtained by considering the absorbing set A with an extra edge connected to each of its check nodes. These single edges represent all edges connected to the check nodes from outside A. It was assumed that these edges carry the largest possible quantizer outputs (maximum log-likelihood-ratios (LLRs)) from outside A at each decoding iteration. Under this assumption, all possible combinations of quantized channel LLRs were input to the variable nodes of A, and each combination was decoded iteratively until all the variable nodes of A were decoded without error (success) or until a maximum number of iterations $I_{\max}$ was reached (failure). The probability of those combinations of inputs that failed to decode correctly were then added to provide a lower bound λ on the FER of any code containing A [8].

In ongoing work, we further show that a better lower bound on the error-floor performance can be obtained if the assumption regarding the LLRs of the extra edges connected to the check nodes is modified. Specifically, if these edges have LLRs that monotonically increase with iteration number until they reach the maximum LLR level, new combinations of absorbing set LLRs beyond those that cannot be decoded under the previous assumption can sometimes be found. Computing the probability of the intersection of the combinations of absorbing set inputs that fail under both assumptions gives an improved estimate of the error-floor performance of a quantized decoder. A similar observation was made in [15] to obtain *code-dependent* lower bound. The code-independent bounds for MSA decoders presented in this paper are calculated based on this modified assumption.

3) Spatially Coupled LDPC Codes: Given an underlying LDPC-BC with a $\mu \times \nu$ parity-check matrix $\mathbf{H}_{\text{BC}}$, a spatially coupled code with parity-check matrix $\mathbf{H}_{\text{SC}}$ and syndrome former memory m can be formed by partitioning $\mathbf{H}_{\text{BC}}$ into m component matrices $\mathbf{H}_i, i = 0, 1, \ldots m$, each of size $\mu \times \nu$, such that $\mathbf{H}_{\text{BC}} = \sum_0^m \mathbf{H}_i$, and arranging them as

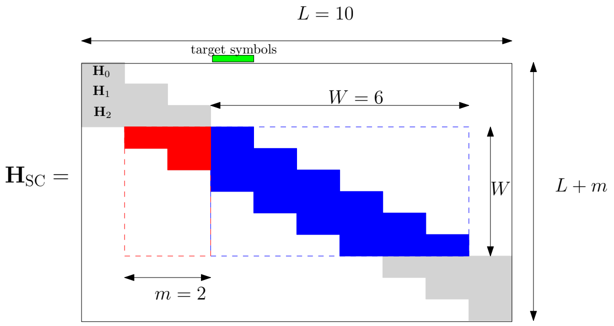


Fig. 2: Window decoder for an SC-LDPCC operating on the parity-check matrix $\mathbf{H}_{\text{SC}}$.

$$\mathbf{H}_{\text{SC}} = \begin{bmatrix} \mathbf{H}_0 & & & \\ \mathbf{H}_1 & \mathbf{H}_0 & & \\ \vdots & \mathbf{H}_1 & \ddots & \\ \mathbf{H}_m & \vdots & \ddots & \mathbf{H}_0 \\ & \mathbf{H}_m & \ddots & \mathbf{H}_1 \\ & & \ddots & \vdots \\ & & & \mathbf{H}_m \end{bmatrix}. \tag{1}$$

In (1), each column including $\mathbf{H}_0, \mathbf{H}_1, \ldots, \mathbf{H}_m$ is referred to as a *block*, and L denotes the number of block columns in $\mathbf{H}_{\text{SC}}$.

4) Window Decoding: A window decoder was introduced in [5] to address the large latency and complexity requirements of decoding SC-LDPCCs with a standard decoder. Fig. 2, adopted from [5], shows a window decoder with window size $W = 6$ operating on the parity-check matrix $\mathbf{H}_{\text{SC}}$ of an SC-LDPCC with $m = 2$ and $L = 10$. All the variable nodes and check nodes included in the window (the blue area) are updated using a message passing algorithm that has access to previously decoded symbols (the red area). The goal is to decode the variable nodes in the first block of the window, called the *target symbols*. The message passing algorithm updates the nodes in the window for some maximum number of iterations $I_{\max}$, after which the first block of target symbols is decoded. Then the window slides one block (ν bits) to the right and one block down (μ bits) to decode the second block, and the process continues until the last block is decoded.

5) Array-Based Spatially Coupled Codes: An array-based SC-LDPCC [10] is defined by the parameters γ, p, L, m and the partitioning used to form $\mathbf{H}_i, i = 0, 1, \ldots, m$, from the parity-check matrix $\mathbf{H}_{\text{BC}}$ of an array-based LDPC-BC. In [10], an $m = 1$ partitioning is described by a *cutting vector* $\boldsymbol{\xi}$, where $\boldsymbol{\xi} = [\xi_0, \xi_1, \ldots, \xi_{\gamma-1}]$ is a vector of non-decreasing non-negative integers $(0 \leq \xi_0 \leq \xi_1 \leq \ldots \leq \xi_{\gamma-1} \leq p)$ that determines the partition of $\mathbf{H}_{\text{BC}}$ of size $\gamma p \times p$ into two components $\mathbf{H}_0$ and $\mathbf{H}_1$ of the same size. Fig. 3 shows an example of an array-based parity-check matrix $\mathbf{H}_{\text{BC}}$ being partitioned into $\mathbf{H}'_0$ and $\mathbf{H}'_1$. In this case, $\mathbf{H}_0$ is obtained by taking $\mathbf{H}'_0$ and setting the $\mathbf{H}'_1$ area with all zeros. Similarly, $\mathbf{H}_1$ is obtained by taking $\mathbf{H}'_1$ and setting the $\mathbf{H}'_0$ area with all zeros. The resulting SC-LDPCC is denoted $\mathcal{C}(\gamma, p, L, \boldsymbol{\xi})$, and

[3] If the decoding errors are mostly caused by different copies of A, we call A the dominant absorbing set.

[4] We simplify the notation by dropping the rate subscript for convenience. The bound can be adjusted for rate using $\lambda_R\left(E_b/N_0(\text{dB})\right) = \lambda_1\left(E_b/N_0(\text{dB}) + 10\log(R)\right)$.

[5] The approximation of the lower bound is referred to as a bound for convenience in this paper.

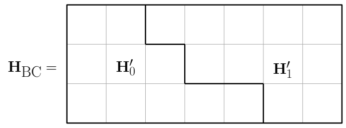


Fig. 3: Array-based code with $\gamma = 3$ and $p = 7$, partitioned by a cutting vector $\boldsymbol{\xi} = [2, 3, 5]$ to construct two component matrices based on $\mathbf{H}'_0$ and $\mathbf{H}'_1$. (Each square represents a $p \times p$ circulant permutation matrix.)

the code rate is given by $R_{\text{SC}} = 1 - \frac{(L+1)\gamma p - \gamma + 1}{Lp^2}$, where the underlying LDPC-BC has rate $R_{\text{BC}} = 1 - \frac{\gamma p - \gamma + 1}{p^2}$.

III. Lower bounds for SC-LDPCCs

The code-independent bound is based on the multiplicity of a given absorbing set. We categorize the absorbing sets in an SC-LDPCC by how many consecutive blocks are spanned by the variable nodes of the absorbing set. Depending on the memory size, the spatial coupling method, and the absorbing set structure, the number of blocks spanned may vary. For example, in the graph of the $\mathcal{C}(3, 61, L, [15, 30, 45])$ SC-LDPCC, the $(4, 2)$ and $(6, 0)$ absorbing sets of Fig. 1 are either contained completely in one block or span two blocks. We denote the multiplicity of absorbing set A that spans i blocks by N_i. It follows that the total number of absorbing sets A in a graph with L blocks is

$$N \triangleq \sum_{i=1}^{L} N_i(L - i + 1). \tag{2}$$

Therefore, the average number of absorbing sets A per block is

$$\overline{N} \triangleq \frac{1}{L} \sum_{i=1}^{L} N_i(L - i + 1). \tag{3}$$

Example 1. Consider the $\mathcal{C}(3, 61, 100, [15, 30, 45])$ SC-LDPCC. Table I shows the resulting multiplicities for the $(4, 2)$ and $(6, 0)$ absorbing sets, which are counted using the methods described in [16].

	N_1	N_2	$N_i, i \geq 3$	N	$\overline{N}$
$(4, 2)$	37,088	29,097	0	6,589,403	65,894.03
$(6, 0)$	48,129	211,975	0	25,721,100	257,211

TABLE I: Multiplicities of absorbing sets in the $\mathcal{C}(3, 61, 100, [15, 30, 45])$ SC-LDPCC.

A. Lower Bound for Standard Decoding of SC-LDPCCs

Since the standard decoder operates on the entire graph, thus treating an SC-LDPCC as a large block code, our previously developed code-independent lower bound can be directly applied. To evaluate the approximate lower bound on FER for a single dominant absorbing set A, the multiplicity N is used to obtain

$$\text{FER} \geq N\lambda, \tag{4}$$

where λ is the lower bound for A. To compare with the underlying LDPC-BC, which operates on a shorter graph. we normalize the FER by L, to obtain a normalized FER per block. denoted $\widehat{\text{FER}} \triangleq \text{FER}/L$. Consequently, the normalized (approximated) lower bound is given by

$$\widehat{\text{FER}} \geq \frac{N\lambda}{L} = \overline{N}\lambda. \tag{5}$$

B. Lower Bounds for Window Decoding of SC-LDPCCs

For window decoding, we define the block error rate at window position $j \in [L] \triangleq \{1, 2, \ldots L\}$ by $\text{BLER}^{(j)}$. By taking the average over all the L window positions, the average BLER can be written as

$$\overline{\text{BLER}} \triangleq \frac{1}{L} \sum_{j=1}^{L} \text{BLER}^{(j)}. \tag{6}$$

Since $\overline{\text{BLER}}$ gives the average error rate for blocks of length ν with window decoding, it is appropriate to compare this to the FER of the underlying LDPC-BC or to $\widehat{\text{FER}}$ for standard decoding of the SC-LDPCC.

To proceed, it is necessary to identify the absorbing sets that can cause decoding errors in one block of target symbols so that the code-independent bound can be applied to evaluate the lower bound on BLER for that block.

Definition 1. Absorbing set A is a *window absorbing set* if all the variable nodes and connected check nodes are in the window (the blue area in Fig. 2).

Consider the window decoder illustrated in Fig. 2. Assuming BP decoding is performed on all W blocks (the blue area), the code-independent bound can be applied to all the window absorbing sets to estimate the probability of error in the whole window.[6] However, we are interested in the probability of error only in the target symbols. Hence, only window absorbing sets with variable nodes in the target symbols should be considered. To this end we propose the following definition.

Definition 2. *Target symbol absorbing set* A is a window absorbing set with at least one variable node in the set of target symbols.

For each window position $j \in [L]$, we denote the multiplicity of target symbol absorbing sets by $T^{(j)}$. To evaluate the BLER of the j^{th} block of target symbols, the code-independent bound with multiplicity $T^{(j)}$ can be applied. Defining T_i as the number of target symbol absorbing sets that span $i \in [W]$ consecutive blocks, we can write

$$T^{(j)} = \sum_{i=1}^{\min(W, L-j+1)} T_i. \tag{7}$$

Applying the code-independent bound, the probability that there is at least one error in the j^{th} block of target symbols can now be (approximately) bounded below as

$$\text{BLER}^{(j)} \geq T^{(j)}\lambda, \quad j \in [L]. \tag{8}$$

Finally, using (7) and (8) in (6), we obtain

$$\overline{\text{BLER}} \geq \frac{\lambda}{L} \sum_{j=1}^{L} \sum_{i=1}^{\min(W, L-j+1)} T_i. \tag{9}$$

[6] In Fig. 2, the LLRs from previous window positions (the red area) are also used. However, we exclude any absorbing set with at least one node in the red area for two reasons. First, the LLRs in the red area are not being updated, so these absorbing sets do not satisfy the bound assumption that all variable node combinations are considered. Second, at high SNR, it is assumed that these LLRs have large values and therefore are less likely to cause an absorbing set to get trapped. Essentially, by excluding these absorbing sets, we are obtaining a lower bound.

By examining Fig. 2, we can see that $\sum_{j=1}^{L}\sum_{i=1}^{\min(W,L-j+1)} T_i = \sum_{i=1}^{W} T_i(L-i+1)$, and therefore

$$\overline{\text{BLER}} \geq \frac{\lambda}{L}\sum_{i=1}^{W} T_i(L-i+1). \tag{10}$$

Remark 1. Any absorbing set A of the SC-LDPCC with all its variable nodes located within the first $W-m$ blocks of the window is a window absorbing set. Therefore, for the first $W-m$ blocks in the window, the absorbing set multiplicity is counted in the same way as for the entire SC-LDPCCs, *i.e.*,

$$T_i = N_i \quad \text{for} \quad i \in [W-m]. \tag{11}$$

Remark 2. Any absorbing set A of the SC-LDPCC that shares variable nodes with the last m blocks in the window is not necessarily a window absorbing set A since some check nodes of the SC-LDPCC code are not contained in the window for these blocks. Therefore, for the last m blocks, the multiplicity of absorbing sets must be counted separately.

Using (11) in (10) it follows that

$$\begin{aligned}\overline{\text{BLER}} \geq \frac{\lambda}{L}&\sum_{i=1}^{W-m} N_i(L-i+1)\\ &+\frac{\lambda}{L}\sum_{i=W-m+1}^{W} T_i(L-i+1).\end{aligned} \tag{12}$$

If the target symbol absorbing sets are all confined to $W-m$ positions, *i.e.* $N_i = 0,\ i \geq W-m+1$, it can be concluded that right hand side of (5) and (12) are the same, *i.e.*, the $\widehat{\text{FER}}$ of the standard decoder and the $\overline{\text{BLER}}$ of the window decoder have the same lower bound.

Example 2. For the code of Example 1, where $m = 1$ and $N_i = 0,\ i \geq 3$, any window size $W \geq 3$ gives the same lower bound for the $\widehat{\text{FER}}$ of the standard decoder and the $\overline{\text{BLER}}$ of the window decoder. This result suggests that the choice of window size depends primarily on how it affects the performance in the waterfall, not in the error-floor.[7]

C. Lower Bound Comparison: Array-Based SC-LDPCCs vs. the Underlying LDPC-BCs

In this subsection, we compare the lower bounds on $\widehat{\text{FER}}$ and $\overline{\text{BLER}}$ of array-based SC-LDPCCs to the lower bounds on FER of the underlying LDPC-BCs.[8] In particular, we compare the (γ, p) array-based LDPC-BC with the corresponding array-based SC-LDPCC with the same (γ, p). Two factors must be considered: first, the respective multiplicities N_{BC} and $\overline{N}_{\text{SC}}$, and second, the respective rates of R_{BC} and R_{SC} of the LDPC-BC and SC-LDPCC being compared. In [10], [12] it is shown that $\overline{N}_{\text{SC}}$ for SC-LDPCCs is less than N_{BC} for the underlying LDPC-BC, so we define $\alpha = N_{\text{BC}}/\overline{N}_{\text{SC}}$. For the code of Example 1, for the $(4,2)$ and $(6,0)$ absorbing sets, N_{BC} can be found in [7], [17], $\overline{N}_{\text{SC}}$ is given in Table I, and we find that $\alpha = 5.08$ and 8.535, respectively. Regarding the rates, we make the assumption that the difference between the rate R_{BC} of the block code and the rate R_{SC} of the SC-LDPCC is negligible for large L.[9] For code of Example 1, with $L = 100$, $R_{\text{SC}} = 0.9503$ and $R_{\text{BC}} = 0.9514$, so in this case we can ignore the rate difference. Therefore the SC-LDPCC standard decoder $\widehat{\text{FER}}$ and window decoder $\overline{\text{BLER}}$ lower bounds are α times better than the underlying LDPC-BC FER lower bound.

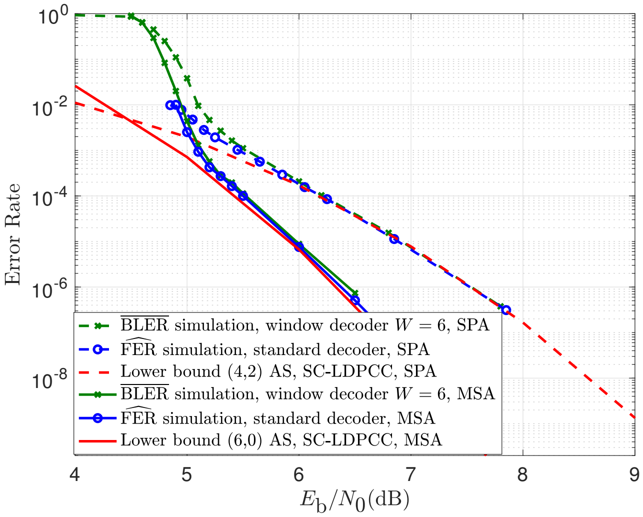


Fig. 4: Proposed lower bound versus simulation results for standard decoding and window decoding of $\mathcal{C}(3, 61, 100, [15, 30, 45])$, with both quantized MSA and quantized SPA.

IV. Results and Discussion

To verify the accuracy of the proposed lower bound, we simulated the $\mathcal{C}(3, 61, 100, [15, 30, 45])$ code for various combinations of decoders/quantizers. The results are presented in Fig 4. When an SPA decoder with a 6-bit uniform quantizer (see [8] for parameters) is used, the $(4,2)$ absorbing sets are dominant, similar to the underlying block code. When an MSA decoder with a 5-bit quasi-uniform quantizer [18] (see [15] for parameters) is used, $(6,0)$ absorbing sets are dominant, which is also the case for the underlying LDPC-BC [15]. In [18], it was also shown that MSA with a quasi-uniform quantizer outperforms the conventional quantized SPA for LDPC-BCs. We recall from Sec. III-B that the bounds for $\widehat{\text{FER}}$ of the standard decoder and $\overline{\text{BLER}}$ of the window decoder are the same for this SC-LDPCC. For standard decoding, applying the code-independent bound to these absorbing sets with the multiplicities from Table I, we see in Fig. 4 that the $\widehat{\text{FER}}$ simulation result agrees with the bound. We also see that the $\overline{\text{BLER}}$ simulation result for window size $W = 6$ agrees closely with the bound. In summary, we note that (1) the lower bounds agree with the simulation results, and (2) the error-floor performance is almost the same for standard decoding and window decoding.

[7] It is well known that small windows have poor waterfall performance, and therefore choosing $W < 3(m+1)$ is not advisable.

[8] The proposed comparison provides a tool for error-floor comparisons of SC-LDPCCs and LDPC-BCs. Waterfall comparisons are typically made via Monte-Carlo simulation.

[9] The rate R_{SC} increases monotonically with L. As $L \to \infty$, $R_{\text{SC}} \to 1-\gamma/p$ which is slightly less than R_{BC}, and the rate loss vanishes as p increases [10].

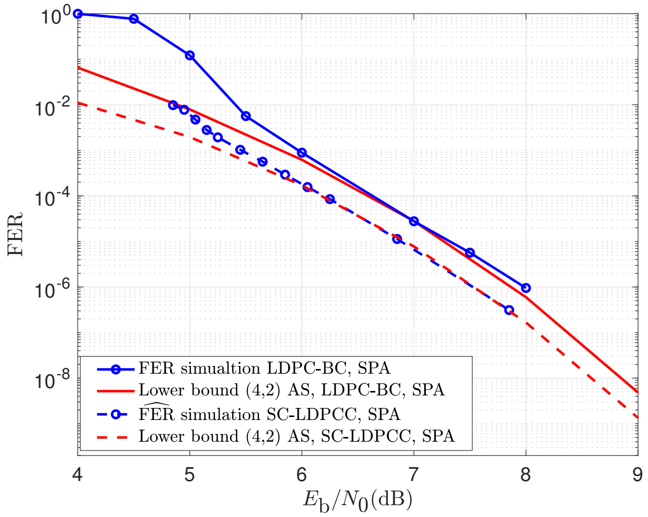


Fig. 5: $\widehat{\text{FER}}$ and FER comparison of the SC-LDPCC with standard decoding and the underlying LDPC-BC for the quantized SPA decoder.

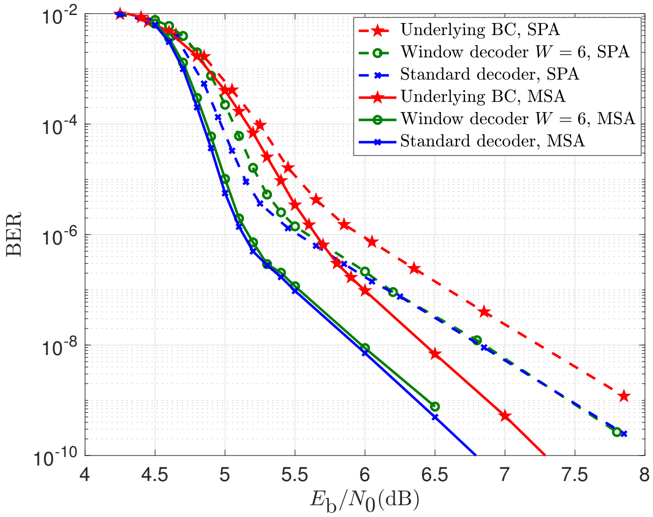


Fig. 6: BER simulation comparison of SC-LDPCCs with standard decoding and window decoding and the underlying LDPC-BC, quantized SPA vs. quantized MSA.

Fig. 5 shows the bound and simulation results for the FER of the underlying LDPC-BC versus the bound and simulation results for the $\widehat{\text{FER}}$ of the SC-LDPCC with SPA decoding and a uniform quantizer. It was noted earlier that $\alpha = 5.08$ in this case, which means that the bound for the LDPC-BC code is 5.08 times higher than the bound for the SC-LDPCC, and we see that the simulation results agree with this prediction. In other words, *the convolutional gain factor* $\alpha = 5.08$ of the proposed lower bounds tracks the convolutional gain exhibited by the simulation results.

In Fig. 6, another interesting comparison is made between the simulated bit error rate (BER) performance of quantized SPA and quantized MSA decoders, where we see that the trends observed in our analysis are also reflected in the BER simulation results. In particular, for both standard and window decoding, the SC-LDPCC with the quantized MSA decoder outperforms the SC-LDPCC with the quantized SPA decoder as well as the underlying LDPC-BC decoded with either the SPA or the MSA.

V. Conclusion

In this paper, we derived lower bounds on the performance of quantized SC-LDPCC decoders, for both standard flooding schedule decoders and sliding window decoders. We showed that properly designed SC-LDPCCs have better lower bounds than their underlying LDPC-BC counterparts, and this advantage was confirmed via computer simulation. We also showed that MSA decoders with quasi-uniform quantization outperform SPA decoders with uniform quantization for SC-LDPCCs, with either standard or window decoding.

References

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. IT-8, Jan. 1962.

[2] A. Jimenez Felstrom and K. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sept. 1999.

[3] S. Kudekar, T. 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.

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

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

[6] T. J. Richardson, "Error floors of LDPC codes," in *Proc. 2003 Allerton Conf. Communications, Control and Computing*, Nov. 2003, pp. 1426–1435.

[7] L. Dolecek, Z. Zhang, V. Anantharam, M. Wainwright, and B. Nikolic, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 181–201, Jan. 2010.

[8] H. Hatami, D. G. M. Mitchell, D. J. Costello, and T. Fuja, "Performance bounds for quantized LDPC decoders based on absorbing sets," in *2016 IEEE Int. Symp. Inf. Theory*, July 2016, pp. 2539–2543.

[9] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc. 2nd Int. Symp. Turbo Codes*, Sept. 2000, pp. 545–546.

[10] D. G. M. Mitchell, L. Dolecek, and D. J. Costello, "Absorbing set characterization of array-based spatially coupled LDPC codes," in *Proc. 2014 IEEE Int. Symp. Inf. Theory*, Jun. 2014, pp. 886–890.

[11] D. G. M. Mitchell and E. Rosnes, "Edge spreading design of high rate array-based SC-LDPC codes," in *Proc. 2017 IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2940–2944.

[12] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "Spatially coupled codes optimized for magnetic recording applications," *IEEE Trans. Magnetics*, vol. 53, no. 2, pp. 1–11, Feb. 2017.

[13] A. Beemer and C. A. Kelley, "Avoiding trapping sets in SC-LDPC codes under windowed decoding," in *Proc. 2016 Int. Symp. Inf. Theory and Its Applications. (ISITA)*, Oct. 2016, pp. 206–210.

[14] 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, Sept. 2015.

[15] H. Hatami, D. G. M. Mitchell, D. J. Costello, and T. Fuja, "Lower bounds for quantized LDPC min-sum decoders based on absorbing sets," in *Proc. 2017 55th Annual Allerton Conf. Communi, Control, and Computing*, Oct. 2017.

[16] Y. Hashemi and A. H. Banihashemi, "New characterization and efficient exhaustive search algorithm for leafless elementary trapping sets of variable-regular LDPC codes," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6713–6736, Dec 2016.

[17] H. Liu, L. Ma, and J. Chen, "On the number of minimum stopping sets and minimum codewords of array LDPC codes," *IEEE Comm. Letters*, vol. 14, no. 7, pp. 670–672, July. 2010.

[18] X. Zhang and P. Siegel, "Quantized iterative message passing decoders with low error floor for LDPC codes," *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 1–14, Jan. 2014.