Windowed Erasure Decoding
of LDPC Convolutional Codes

Marco Papaleo*, Aravind R. Iyengar[†], Paul H. Siegel[†], Jack K. Wolf[†],
Giovanni E. Corazza*, Alessandro Vanelli-Coralli*

*University of Bologna, DEIS-ARCES, Viale Risorgimento, 2 - 40136 Bologna, Italy
email: {mpapaleo,gecorazza,avanelli}@arces.unibo.it
[†] University of California, San Diego, La Jolla CA 92093 email: {aravind,psiegel,jwolf}@ucsd.edu

Abstract—**We consider windowed decoding of LDPC Convolutional Codes on the Binary Erasure Channel (BEC) to study the trade-off between the decoding latency and the code performance. We make some key observations about regular LDPC Convolutional code ensembles under windowed decoding and give modified constructions of these codes that allow us to efficiently trade-off performance for gains in latency.**

I. INTRODUCTION

The good performance of LDPC codes [1] often is seen when the blocklength of the codes is large. An immediate consequence of this, especially in the context of packet-level coding [2] wherein each codeword symbol consists of several hundreds of bits, is the associated large latency for decoding. In this paper, we propose a windowed decoder for LDPC Convolutional Codes [3] over the BEC and explore the trade-off between latency and performance. We also give code constructions which perform well, not only with the usual Belief Propagation (BP) decoder, but also with the proposed windowed erasure decoder.

This paper is organized as follows. In Section II we introduce the LDPC Convolutional code terminology and recall various code constructions. We describe a windowed erasure decoder, referred to henceforth as the windowed decoder, in Section III. In Section IV, we show that regular LDPC Convolutional Codes [4] do not allow us to trade-off performance for latency efficiently. We then give a few codes that perform well with both the BP and the windowed decoders, and evaluate their performance experimentally in Section V. We summarize our findings in Section VI.

II. LDPC CONVOLUTIONAL CODES

LDPC Convolutional Codes (LDPCCC) were first introduced in [3]. In the following we will focus on terminated LDPCCC following the notation introduced in [5].

A. Definition

A rate $R = b/c$ binary, time-varying LDPC convolutional code is defined as the sequences $\mathbf{v}_{[0,\infty]}$, satisfying $\mathbf{v}_{[0,\infty]}\mathbf{H}_{[0,\infty]}^{\mathrm{T}} = \mathbf{0}$, where $\mathbf{H}_{[0,\infty]}^{\mathrm{T}}$ is the *syndrome former matrix* given in (1). The elements $\mathbf{H}_i(t)$, $i = 0, 1, ..., m_s$ in (1) are binary matrices of size $(c - b) \times c$ that satisfy the following properties [6]:

- $\mathbf{H}_i(t) = \mathbf{0}$, for $i < 0$ and $i > m_s$, $\forall\, t$
- $\exists\, t$ such that $\mathbf{H}_{m_s}(t) \neq \mathbf{0}$
- $\mathbf{H}_0(t)$ has full rank $\forall\, t$

The parameter m_s is called the syndrome former *memory* and $\nu_s = (m_s + 1)c$ is referred to as the *decoding constraint length*. In order to get sparse graph codes, the Hamming weight of

each row, $\mathbf{h}$, of $\mathbf{H}_{[0,\infty]}^{\mathrm{T}}$ must be very low, i.e., $w_h(\mathbf{h}) \ll \nu_s$. We will consider terminated LDPCCC that have a finite syndrome former $\mathbf{H}^{\mathrm{T}}$ as in (1). Such a code is said to be J-K regular if $\mathbf{H}^{\mathrm{T}}$ has exactly J ones in every row and K ones in every column, excluding the first and the last m_s columns[1]. It follows that for a given J, the syndrome former can be made sparse by increasing c or m_s or both, leading to different code constructions [7]. In this paper, we will consider LDPCCC characterized by large c and small m_s. As in [5], we will focus on LDPCCC which can be constructed from a *protograph*.

B. Protograph-based terminated LDPCCC

A protograph [8] is a relatively small bipartite graph from which a larger graph can be obtained by a copy-and-permute procedure: the protograph is copied M times, and then the edges of the individual replicas are permuted among the M replicas to obtain a single, large bipartite graph. Suppose the protograph possesses N_P variable nodes and M_P constraint nodes, with degrees $J_j, j = 1, ..., N_p$, and $K_i, i = 1, ..., M_p$, respectively. Then the derived graph will consist of $n_v = N_P M$ variable nodes and $m_c = M_P M$ constraint nodes. The nodes of the protograph are labeled, so that if the Variable Node (VN) V_j is connected to the Check Node (CN) C_i in the protograph, then V_j in a replica can only connect to one of the M replicated C_i's. Doing so preserves the decoding threshold properties of the protograph [9].

Protographs can be represented by means of an $M_P \times N_P$ bi-adjacency matrix $\mathbf{B}$, called the *base matrix* of the protograph where the entry $\mathbf{B}_{m,n}$ represents the number of edges between CN C_m and VN V_n (a non-negative integer, i.e., multiple parallel edges, multiedges, are permitted.). The copy-and-permute operation is realized by replacing each edge (or multiedge) in the base matrix $\mathbf{B}$ with a size M permutation matrix (or the sum of as many distinct size M permutation matrices as the number of multiedges, respectively). For different values of M, different blocklengths of the derived Tanner graph can be achieved, keeping the original graph structure imposed by the protograph. This means that the density evolution analysis can be performed within the protograph, instead of on the derived Tanner graph. Furthermore, protographs impose a structure on the derived graph, which facilitates the design of fast decoders and efficient encoders, as well as a refined control on the derived graph edge connections.

Analogous to LPDC block codes, LDPCCC can also be derived by a protograph expansion. The syndrome formers of

---

[1]Ignoring the terminated portion of the code. Note that all the rows, however, have J ones.

$$\mathbf{H}_{[0,\infty]}^{\mathrm{T}} = \begin{bmatrix} \ddots & & \ddots & & & \\ & \mathbf{H}_0^{\mathrm{T}}(0) & \cdot & \mathbf{H}_{m_s}^{\mathrm{T}}(m_s) & & \\ & & \ddots & & \ddots & \\ & & & \mathbf{H}_0^{\mathrm{T}}(t) & \cdot & \mathbf{H}_{m_s}^{\mathrm{T}}(t+m_s) \\ & & & & \ddots & & \ddots \end{bmatrix} \qquad \mathbf{H}^{\mathrm{T}} = \begin{bmatrix} \mathbf{H}_0^{\mathrm{T}}(0) & \cdot & \mathbf{H}_{m_s}^{\mathrm{T}}(m_s) & & \\ & \ddots & & \ddots & \\ & & \mathbf{H}_0^{\mathrm{T}}(L-1) & \cdot & \mathbf{H}_{m_s}^{\mathrm{T}}(L+m_s-1) \end{bmatrix}$$

$$(1)$$

these codes are composed of blocks of size M permutation matrices. Let a be the greatest common divisor (gcd) of J and K. Then there exist positive integers J' and K' such that $J = aJ'$, $K = aK'$, and $\gcd(J', K') = 1$. Starting from these parameters, it is possible to define the ensemble of LDPCCC $\mathcal{C}_P(J, K, M)$, characterized by $b = (K' - J')M$, and $c = K'M$. If we start the convolutional code at time instant $t = 1$ and terminate it after L instants, we obtain a block code, described by the base matrix[2] $\mathbf{B}_{[1,L]}$ as in (2). The protograph of the terminated code has $N_p = LK'$ variable nodes and $M_p = (L + m_s)J'$ check nodes.

$$\mathbf{B}_{[1,L]} = \begin{bmatrix} \mathbf{B}_0(1) & & & \\ \mathbf{B}_1(1) & \mathbf{B}_0(2) & & \\ \vdots & \mathbf{B}_1(2) & \ddots & \\ \mathbf{B}_{m_s}(1) & \vdots & \ddots & \mathbf{B}_0(L) \\ & \mathbf{B}_{m_s}(2) & \ddots & \mathbf{B}_1(L) \\ & & \ddots & \vdots \\ & & & \mathbf{B}_{m_s}(L) \end{bmatrix} \qquad (2)$$

The rate of the block code is therefore

$$R_L = 1 - \left(\frac{L + m_s}{L}\right)\frac{J'}{K'} = 1 - \left(1 + \frac{m_s}{L}\right)(1 - R) \qquad (3)$$

where $R = \frac{b}{c} = 1 - \frac{J'}{K'}$ is the rate of the non-terminated code. Note that $R_L \to R$ and the LDPCCC has a regular degree distribution [5] when $L \to \infty$.

III. WINDOWED ERASURE DECODING

LDPCCC are characterized by a very large value of constraint length $\nu_s = (m_s + 1)K'M$, and this makes the Viterbi decoder unfeasible for this kind of code. Nevertheless, the sparsity of the syndrome former matrix can be exploited and a message passing algorithm can be adopted. For the case of a terminated LDPCCC, decoding can be performed similar to the case of an LDPC block code, meaning that each frame carrying a codeword obtained through the termination can be decoded with the classical Belief Propagation (BP) algorithm. For the case of non-terminated LDPC convolutional codes, the size of the corresponding Tanner graph is potentially infinite. Nevertheless, the convolutional structure of the code imposes a maximum distance constraint on the variable nodes connected to the same parity check equations – two variable nodes that are at least $(m_s + 1)$ time units apart cannot be involved in the same equation. This characteristic can be exploited in order to perform continuous decoding of the received stream through a *window* that slides along the bit sequence[3] [6]. Moreover, this

structure allows for the possiblity of parallelizing the iterations of the message passing decoder, through several processors working in different regions of the Tanner graph. A pipeline decoder based on this idea was proposed in [3]. In this paper we analyze the performance achievable by the terminated LDPCCC presented in Section II-B. More specifically, we are interested in determining the trade-off between decoder performance and latency. To this end, we propose a different decoder architecture, which is still reminiscent of the sliding window approach described in [6] for the unterminated LDPCCC.

Consider a terminated parity check matrix $\mathbf{H}$ built from a base protograph $\mathbf{B}$ and a sliding window consisting of $J_W = WJ'M = W(c - b)$ rows and $K_W = WK'M = Wc$ columns, $m_s + 1 \le W \le L + m_s$, i.e., the sliding window covers[4] WJ' rows and WK' columns of the protograph $\mathbf{B}$. At the first time instant, $t = 1$, the decoder performs message passing iterations over the equations within the sliding window with the aim of decoding all of the first c symbols in the window, called the *targeted symbols*. The sliding window shifts down $(c - b)$ rows and right c columns (J' rows and K' columns in $\mathbf{B}$) after a maximum number of message passing iterations have been performed, or when all targeted symbols within the window have been recovered, and continues decoding at the new position at the second time instant, i.e., $t = 2$.
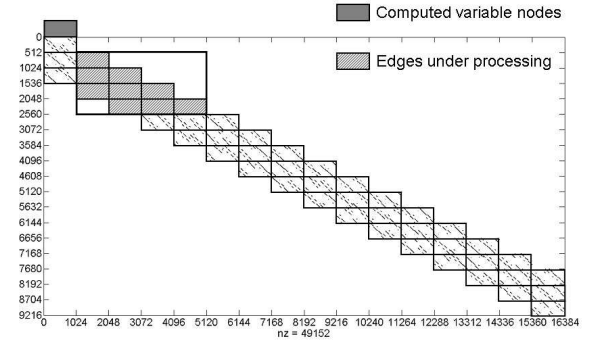

Fig. 1. Windowed decoder for sliding window size $W = 4$, for a $(3, 6, 512)$ LDPCCC with $L = 16$ at time instant $t = 2$.

The latency of the sliding window decoder is therefore given by $\Delta_W = WcT_s$, where T_s is the intersymbol duration, which amounts to a fraction $w = \frac{W}{L}$ of the latency in comparison with the BP decoder[5]. Figure 1 shows a schematic representation of the windowed decoder for $W = 4$. The light gray blocks represent the edges which are currently involved in the message passing decoding, while the dark gray blocks represent the VNs already processed by the decoder. As illustrated in the figure, the first block of VNs are still involved in the check equations in

---

[2]For a time-invariant terminated LDPCCC, $\mathbf{B}_i(t) = \mathbf{B}_i \ \forall \ t = 1, 2, \cdots, L$, and $i = 0, 1, \cdots, m_s$.

[3]This implementation is motivationally analogous to the Viterbi decoder with a finite backsearch length.

[4]When the sliding window goes beyond the protograph, it is assumed to be bounded by the boundaries of the protograph.

[5]By BP decoder, we mean the decoder which uses the entire parity-check matrix.

the sliding window at $t = 2$. However, on successful decoding at $t = 1$, these VNs can be removed from the corresponding equations, so that the edges under processing are the ones within the highlighted sliding window.

It is clear that the performance achievable by the windowed decoder improves with the window size, and therefore the decoding latency. In the following we will analyze the probability that the windowed decoder fails to decode uncorrelated erasures introduced due to transmission over a BEC.

IV. ANALYSIS OF THE WINDOWED DECODER

We now analyze the performance of the windowed decoder described in the previous section. In the limit of infinite blocklength, each term in the base protograph $\mathbf{B}$ is replaced by a permutation matrix of infinite size, and therefore the latency of any window size is infinite, apparently defeating the purpose of the windowed decoder. Our interest in the asymptotic performance, however, is justified as it allows us to establish upper bounds on the performance of the practical windowed decoder over finite-length codes. We are able to give ensembles which have little change in asymptotic performance for the entire range of possible window sizes.

The main distinction in the analysis here from the code performance analysis for protograph codes is the definition of decoding success. While in the case of a protograph code, the decoding is considered a success only when, for any symbol in the codeword, the probability of failing to decode the symbol goes to zero, here the decoding is successful as long as the probability of failing to decode the targeted symbols, i.e., the first c symbols in the decoding window, goes to zero.

For the BEC, we define the *windowed threshold* ε_W^* of a code with a decoder of window size W as the supremum of the set of channel erasure rates for which the decoding is successful. Note that it is sufficient to evaluate the windowed thresholds for a single window, as when the decoding is successful for the window in its present layout, the situation is exactly the same as when the window moves onto the next set of targeted symbols and will therefore be successful for any subsequent window layout, as was noted in [4], where a similar windowed decoding was considered.

The success or failure of the windowed decoder is evaluated using the Protograph-EXIT method developed in [10]. This method is similar to the standard EXIT analysis in that it tracks the mutual information between the message on an edge and the bit value corresponding to the VN to which the edge is incident maintaining the graph structure dictated by the protograph. The decoder is assumed to be successful when the a-posteriori mutual information at all the targeted VNs of the protograph converges to 1.

The processing at a CN of degree d_C results in an updating of the mutual information as

$$I_{\text{out}} = \mathscr{C}\left(I_{\text{in},1}, \cdots, I_{\text{in},d_C-1}\right) = \prod_{i=1}^{d_C-1} I_{\text{in},i} \quad (4)$$

and the corresponding update at a VN of degree d_V gives

$$I_{\text{out}} = \mathscr{V}\left(I_{\text{ch}}, I_{\text{in},1}, \cdots, I_{\text{in},d_V-1}\right) = 1 - \varepsilon \prod_{i=1}^{d_V-1}\left(1 - I_{\text{in},i}\right) \quad (5)$$

where $I_{\text{ch}} = 1 - \varepsilon$ is the mutual information obtained from the channel. Note that the edge multiplicities are included in the above check and variable node computations. We now give two lemmas which will be made use of in the analysis. The proofs have been omitted due to space constraints.

Lemma 1 (Monotonicity of $\mathscr{C}$). *The CN operation in* (4) *is monotonic in its arguments, i.e.,*

$$x \le x' \Rightarrow \mathscr{C}(x,y) \le \mathscr{C}(x',y) \ \forall \ y \in [0,1].$$

Lemma 2 (Monotonicity of $\mathscr{V}$). *The VN operation in* (5) *is monotonic in its arguments, i.e.,*

$$x \le x' \Rightarrow \mathscr{V}(x,y) \le \mathscr{V}(x',y) \ \forall \ y \in [0,1].$$

A. Zero windowed threshold LDPCC codes

Consider the windowed decoding of a $3 - 6$ regular LDPCC code with $m_s = 2$ using a window of size $W = 3$, so that the decoding performance depends on the protograph

$$\mathbf{B}_{W3} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (6)$$

Let us label the CNs as A, B, C and the VNs as 1, 2, 3, 4, 5 and 6. Since the targeted VNs[6] are the VNs 1 and 2, Figure 2 shows the computation tree for the message passed from the VN 1 to the CN A. Looking for the fixed point mutual information I on
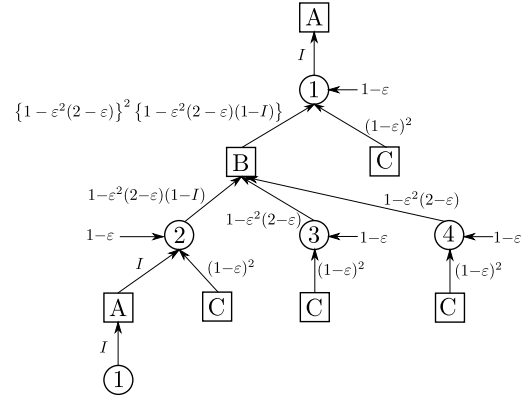

Fig. 2. Computation tree for the edge $1 \rightarrow A$ in the protograph (6)

the edge $1 \rightarrow A$, we upper bound the mutual information on any edge from the CN C by $(1-\varepsilon)^2$ as the CN C is connected to two VNs (5 and 6) that are connected only to the channel. From Figure 2, we can write

$$I \le \frac{1 - \varepsilon^2(2-\varepsilon)\left[1 - \left\{1 - \varepsilon^2(2-\varepsilon)\right\}^3\right]}{1 - \varepsilon^4(2-\varepsilon)^2\left\{1 - \varepsilon^2(2-\varepsilon)\right\}^2}.$$

The fixed point mutual information on the edge $A \rightarrow 1$ is the same as that on the edge $2 \rightarrow A$ which, due to symmetry[7], has the same fixed point I. Hence, the fixed point a-posteriori

---

[6]Note that the threshold for decoding all the VNs is clearly zero, as two VNs of degree 1 (5 and 6) are connected to the same CN C.

[7]The symmetry is seen by noting that the degrees and the neighborhoods of the VNs 1 and 2 are exactly the same.

mutual information (denoted $I_{\text{APP, 1}}$ or, more simply, $\mathcal{I}$) at VN 1 satisfies[8]

$$\mathcal{I} = 1 - (1 - I_{1 \to A})(1 - I_{A \to 1})$$
$$= 1 - (1 - I)^2 \leq 1 - 64\varepsilon^8 + O(\varepsilon^9) < 1 \; \forall \; \varepsilon > 0 \quad (7)$$

implying that the windowed threshold $\varepsilon_{W3}^* = 0$. The error probability (probability of failure) of the decoder is lower bounded for small ε by $\mathbb{P}_e^{\mathbf{B}_{W3}} \geq 64\varepsilon^8$. Using similar arguments based on Lemmas 1 and 2, we can show the following for the $3 - 6$ regular LDPCCC with windowed decoding (proofs omitted).

Lemma 3 (Upper bound for $\mathcal{I}$). *For a window of size $W \geq 3$*

$$\mathcal{I} \leq 1 - \kappa_W^2,$$

where $\kappa_i = \varepsilon \left(1 - (1 - \kappa_{i-1})^2\right)\left(1 - (1 - \kappa_{i-2})^2\right)$, $i = 3, 4, \cdots, W$, $\kappa_2 = \varepsilon\left(1 - (1 - \varepsilon)^2\right)$ *and* $\kappa_1 = \varepsilon$.

Lemma 4 (Lower bound for $\mathcal{I}$). *For a window of size $W \geq 3$*

$$\mathcal{I} \geq 1 - \delta_W^2,$$

where $\delta_1 = 1 - (1 - \varepsilon)^5$, $\delta_i = 1 - (1 - \varepsilon)^3 \left[1 - \varepsilon \prod_{j=1}^{i-1} \delta_j\right]^2$,

$$\delta_{W-1} = \begin{cases} 1 - (1 - \varepsilon\delta_{W-2})(1 - \varepsilon\delta_{W-2}\delta_{W-3})^2 & W \geq 4 \\ 1 - (1 - \varepsilon\delta_{W-2})^3 & W = 3 \end{cases},$$

and $\delta_W = \varepsilon\delta_{W-1}\delta_{W-2}$.

It follows from the above that the terminated $3 - 6$ regular LDPCC code with $m_s = 2$ has a zero windowed threshold for all window sizes except the full window, whose BP threshold (for sufficiently large L) is $\varepsilon_{BP}^* = 0.488$. However, as the window size increases, the error probability is very close to zero. For a non-zero target error probability, certain erasure rates can still be handled – the higher the tolerable error rate, the larger the channel erasure probability that can be handled. Figure 3 plots the lower and upper bounds and the actual fixed point a-posteriori mutual information at the first VN for the $3 - 6$ regular LDPCCC with windowed decoder of sizes $W = 3, 5$ and 10. While tighter bounds are obtainable by going deeper
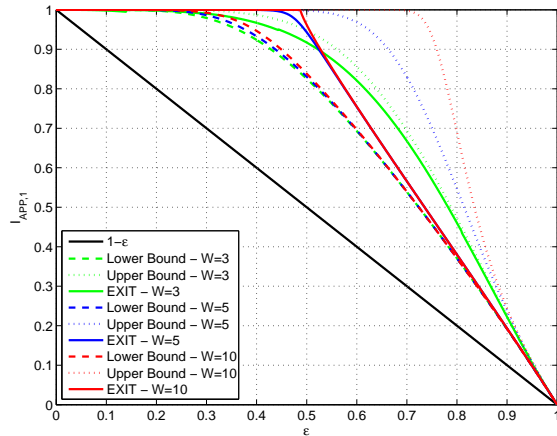

Fig. 3. Upper bound (Lemma 3), lower bound (Lemma 4), and estimated a-posteriori mutual information $\mathcal{I}$ from EXIT analysis, for window size W=3, 5, and 10.

[8]$\mathcal{I}_i = 1 - (1 - I_{i \to j})(1 - I_{j \to i})$ for every VN $i = 1, 2, \cdots, N_P$, and CN $j = 1, 2, \cdots, M_P$ [11].

into the computation tree as in Figure 2, the upper bounds given in Lemma 3 suffice to show that the thresholds are zero, and the lower bounds in Lemma 4 suffice to evaluate thresholds for non-zero target error probabilities.

It follows from the bounding technique employed here that a protograph with no multiedges that has a single CN connected to two VNs, one of degree 1 and the other of degree 2, always results in a zero threshold. For protographs with no such "bad structures" this upper bounding technique on a finite computation tree always results in the trivial upper bound $\mathcal{I} \leq 1$. Whereas the observation of the bad performance of such protograph codes was made in [10], a zero threshold was not reported. Our analysis suggests that we avoid bad structures while designing codes to ensure non-zero windowed thresholds.

B. LDPCC codes for windowed decoding

In order to find good LDPCCC ensembles suitable for windowed decoding, we need to avoid the structures that resulted in the zero thresholds in the sub-protograph identified by the decoding window. Let us consider two new code ensembles which we call Ensembles B and C. Ensemble B is still a $3 - 6$ terminated code, but it differs from the $3 - 6$ regular LDPCCC (Ensemble A) in that it has a memory $m_s = 1$ and component based matrices $\mathbf{B}_0 = [2 \quad 2]$ and $\mathbf{B}_1 = [1 \quad 1]$. Ensemble C is a $4 - 8$ terminated code characterized by $m_s = 2$, $\mathbf{B}_0 = [2 \quad 2]$, and $\mathbf{B}_1 = \mathbf{B}_2 = [1 \quad 1]$. All ensembles are terminated with $L = 50$. The BP threshold for Ensemble B is still 0.488, which is the same as that of Ensemble A, while the Ensemble C has a BP threshold of 0.496, which is the same as that of the $4 - 8$ regular LDPCCC [7]. Thus, ensemble B has the same BP threshold as A, but has a smaller memory (allows a smaller windowed decoder), whereas ensemble C has the same memory but a better BP threshold than A.

In Figure 4, we show the windowed thresholds plotted against the window size for the ensembles B, and C[9]. This plot depicts
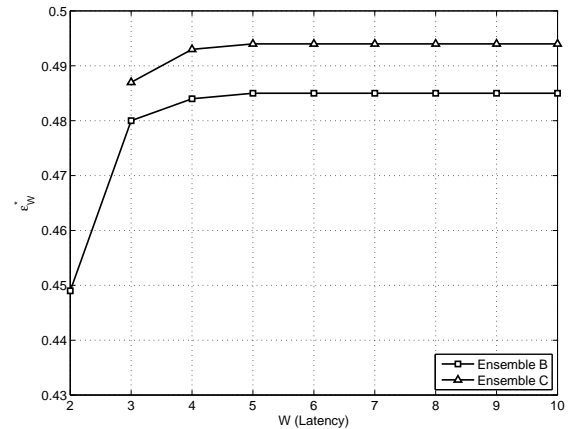

Fig. 4. Threshold as a function of the window size for ensembles B and C.

the trade-off between code performance and latency. As can be observed, even for small window sizes the two codes present good threshold performance. The ensemble C outperforms B, but B can be decoded with a window of size 2. Also note that

[9]From the previous analysis it is clear that the threshold for ensemble A is 0 for any value of window size.
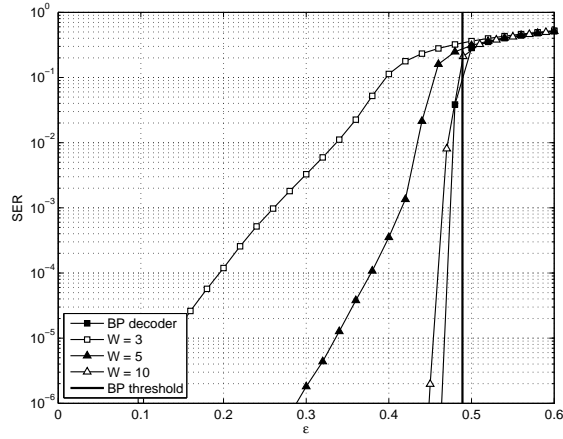
Fig. 5. Symbol Error Rate performance for several sliding window sizes. Code from Ensemble A.

V. NUMERICAL RESULTS

We have assessed the performance of the three LDPCCCs ensembles – ensembles A, B, and C. All of the protographs have been terminated after $L = 16$ time instants and expanded with random permutation matrices of size $M = 512$. The resulting code rate is $14/32$ for codes A and C, and $15/32$ for code B, while the BP thresholds are 0.489, 0.487, and 0.497 for codes A, B, and C, respectively.
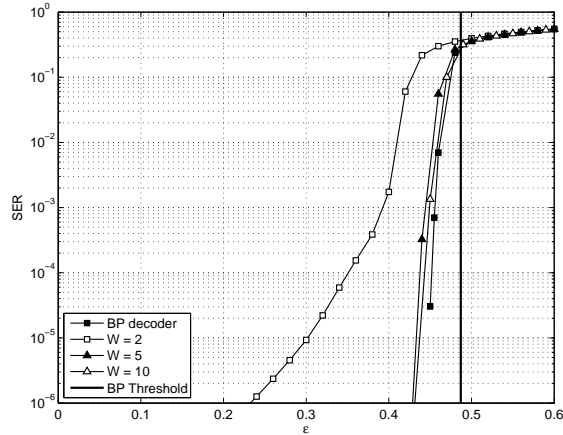

Fig. 6. Symbol Error Rate performance for several sliding window sizes. Code from Ensemble B.

Figures 5, 6 and 7 show the Symbol Error Rate (SER) performance achievable by codes from ensembles A, B, and C, under both the BP decoder and the windowed decoders. The results are in agreement with the analysis. Due to the presence of check nodes connected to both degree one and two variable nodes, code A presents the worst performance. Nevertheless, it is worth noticing that although the windowed threshold is zero for this ensemble, the decoder still shows good performance for moderate sliding window size, as pointed out earlier. The codes from ensembles B and C clearly outperform the code from ensemble A. Furthermore, for these two codes the results agree with the theoretical results presented in Figure 4, showing that
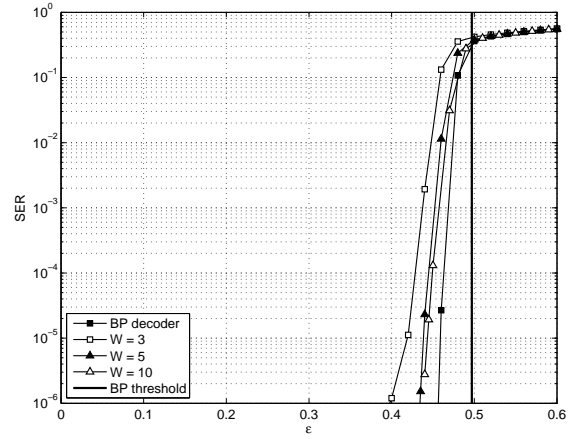
for sliding windows of size larger than 5 the windowed decoder performance is almost identical to BP decoding.

VI. CONCLUSION

In this paper we have analyzed the performance of protograph based terminated LDPCCC under windowed message passing decoder. Our analysis shows how the $3 - 6$ regular LDPCCC suffers when a windowed decoding is adopted due to the protograph structure. We have analyzed further regular LDPCCC ensembles which show good SER performance even for small decoding window size. Numerical results obtained through computer simulations agree with the theoretical analysis.

ACKNOWLEDGMENT

REFERENCES

[1] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, Massachusetts: MIT Press, 1963.
[2] M. Papaleo, R. Firrincieli, G. E. Corazza, and A. Vanelli-Coralli, "Packet coding performance with correlated fading and shadowing," in *Proc. IEEE Globecom 2009 (to appear)*, (Honolulu, Hawaii), 2009.
[3] A. J. Felstrom and K. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inform. Theory*, vol. 45, pp. 2181–2191, Sept. 1999.
[4] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inform. Theory (Submitted)*, 2009.
[5] M. Lentmaier, G. P. Fettweis, K. S. Zigangirov, and J. D. J. Costello, "Approaching capacity with asymptotically regular LDPC codes," in *Information Theory and Applications 2009*, (San Diego, California), 2009.
[6] A. Pusane, A. J. Feltstrom, A. Sridharan, M. Lentmaier, K. Zigangirov, and D. Costello, "Implementation aspects of LDPC convolutional codes," *IEEE Trans. Commmun.*, vol. 56, pp. 1060–1069, July 2008.
[7] A. Sridharan, *Design and Analysis of LDPC Convolutional Codes*. PhD thesis, University of Notre Dame, Notre Dame, Indiana, 2005.
[8] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," tech. rep., JPL INP, Tech. Rep., Aug. 2003.
[9] G. Liva, M. Papaleo, C. P. Niebla, S. Cioni, S. Scalise, A. Vanelli-Coralli, G. Corazza, P. Kim, and H.-J. Lee, "The very short frame of mobile DVB-RCS: code design and QoS performance," *Wiley International Journal on Satellite Communications (special issue on DVB-RCS+M)*, 2009.
[10] G. Liva and M. Chiani, "Protograph LDPC codes design based on EXIT analysis," in *Proc. IEEE Globecom 2007*, (Washington, D.C.), pp. 3250–3254, 2007.
[11] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

Fig. 7. Symbol Error Rate performance for several sliding window sizes. Code from Ensemble C.

with a window of size at least 5, both codes perform very close to the asymptotic full window (BP) performance.