Dynamic Window Decoding for LDPC Convolutional Codes in Low-Latency Optical Communications

Xia, T.; Koike-Akino, T.; Millar, D.S.; Kojima, K.; Parsons, K.; Miyata, Y.; Sugihara, K.; Matsumoto, W.

TR2015-019 March 2015

Abstract

We propose a dynamic window decoding scheme for LDPC convolutional codes to reduce the latency compared to the belief propagation decoding. The BER performance of our proposed memory-efficient decoding scheme is verified by simulations.

Optical Fiber Communication Conference and Exposition (OFC)

Dynamic Window Decoding for LDPC Convolutional Codes in Low-Latency Optical Communications

Tian Xia[1,2], Toshiaki Koike-Akino[1], David S. Millar[1], Keisuke Kojima[1], Kieran Parsons[1], Yoshikuni Miyata[3], Kenya Sugihara[3], Wataru Matsumoto[3]

[1] *Mitsubishi Electric Research Labs. (MERL), 201 Broadway, Cambridge, MA 02139, USA. koike@merl.com*
[2] *School of Electrical Engineering and Computer Science, Louisiana State University, Baton Rouge, LA 70803, USA.*
[3] *Information Technology R&D Center, Mitsubishi Electric Corporation, 5-1-1, Ofuna, 247-8501, Japan.*

Abstract: We propose a dynamic window decoding scheme for LDPC convolutional codes to reduce the latency compared to the belief propagation decoding. The BER performance of our proposed memory-efficient decoding scheme is verified by simulations.

OCIS codes: (060.4510) Optical communications, (060.1660) Coherent communications, (060.4080) Modulation.

1. Introduction

Low-density parity-check convolutional codes (LDPC-CCs) (examples of *spatially-coupled* codes) are demonstrated to be strong candidates for future optical communications systems due to the threshold saturating property [1,2]. Since the non-zero entries only lie along the diagonal block in an LDPC-CC's parity-check matrix (PCM), *window decoding* (WD) techniques were proposed recently to decode a sub-block of the codeword (or stream) sequentially [3–5], thereby lowering the memory requirement and decision latency compared to the belief propagation (BP) decoding.

We focus on a terminated protograph-based (J,K,L,M) LDPC-CC when its PCM has exactly J nonzero entries in each column and K nonzero entries in each row (the top and bottom of the PCM has actually less row weight), the terminated length is L, and the graph lifting factor is M. We consider binary and nonbinary LDPC-CC since nonbinary LDPC codes [6] have recently received much attension due to its higher performance.

One key feature of the WD technique is that the *variable* nodes or *check* nodes may be covered in more than one windows. Due to this dependency, the historical edge information in previous windows should be utilized in the following window if the variable (check) nodes connected with these edges are included in both windows. However, how to utilize and propagate those historical edge information is not explicitly explained in the existing WD literature.

In this paper, we propose a dynamic window decoding scheme which mimic the BP decoding by keeping and propagating the edge information in previous windows to the following ones. The memory requirement and the computational complexity are then analyzed for our proposed dynamic WD scheme. The effectiveness of the dynamic WD scheme are demonstrated for both binary and nonbinary LDPC-CCs with comparison to the BP decoding.

2. Dynamic Window Decoding for LDPC-CC

The key parameter in the window decoding scheme is the window size, denoted as W. In this paper, we adopts the definition in [4] such that W determines the number of rows of the PCM included in the window. The illustration of WD with window size $W=4$ on a $(3,15,9,M)$ LDPC-CC is depicted in Figure 1. By assuming that each instant the window slides over M rows, the window now is at the fourth decoding instant.

The dynamic WD scheme mimics the belief propagation by keeping propagating only the *extrinsic information* when the decoding window slides. Specifically, it records the pair of extrinsic information (one is coming to check nodes, the other one is coming to variable nodes) for each edge when the window is going to slide to the next position. The historical edge information is then the ones recorded in previous windows but no longer to be updated as these edges are not included anymore in current and future windows. For the edges which are also included in the next window (due to the overlapping of consecutive windows), the corresponding edge information will be used as initialization for the next window. The historical edge information, the key part of the dynamic WD scheme, can be further divided into two parts according to the current window: the ones which are connected to the variable nodes inside the current window and the other ones which are no longer included in the current and future windows. Thus, the historical edge

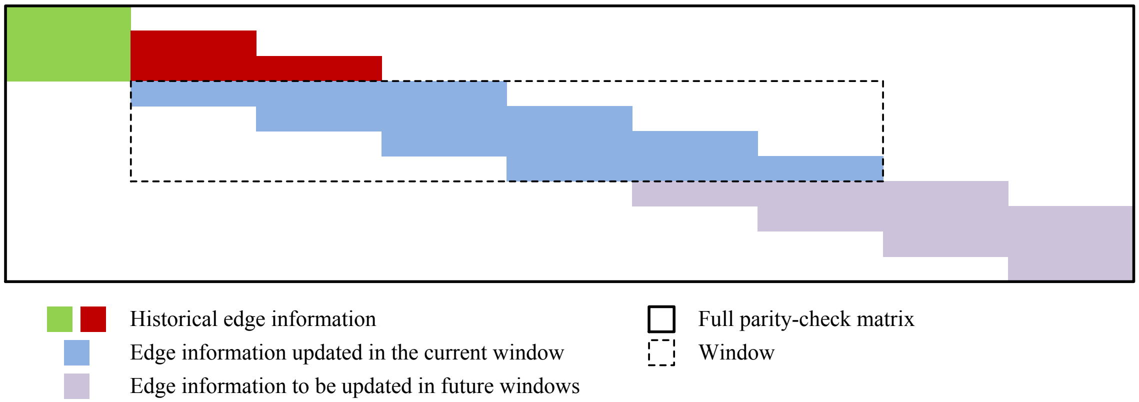

Fig. 1. $(3,15,9,M)$ LDPC-CC's PCM and window decoding with window size $W=4$ at the 4-th decoding instant.

information in red is still propagated to the current window, and the historical edge information in green can be used to decode the corresponding variable nodes.

The memory requirement for our dynamic WD scheme is analyzed with comparison to the conventional BP decoding technique. Specifically, given a (J,K,L,M) LDPC-CC's PCM, the total number of edges in the PCM is $(m_s+1)bML$, where m_s is syndrome former memory, and b is the number of permutation matrix for each time instant of the LDPC-CC's PCM. For the dynamic WD with window size W, the number of edges within the window is then at most $(m_s+1)bMW$. In addition, the historical extrinsic information over $(m_s+1)m_sbM/2$ edges from previous m_s windows is needed for the current window in dynamic WD scheme. Thus, the number of edges involved in each window is at most $(m_s+1)(W+m_s/2)bM$. As the memory needed in iterative decoding is linear with the number of edges, the memory requirement of the dynamic WD can be reduced to $(W+m_s/2)/L$ of the BP decoding. The computational complexity for the dynamic WD scheme is also of interest. In our dynamic WD scheme, there is neither syndrome check nor early termination mechanism. Thus, in each window, the dynamic WD will use up all available iterations, denoted as N_{iter}. As an edge is at most included in W windows, the iteration number per edge for dynamic WD is at most WN_{iter}.

3. Simulation Results

The bit-error-rate (BER) performances of our proposed dynamic WD scheme using different window size $W=6,10$ and different iteration number $N_{\text{iter}}=5,10,20,100$ are shown in Figure 2. The conventional BP performance is also evaluated for comparison, where the maximum iteration number N_{iter} is 100. For the upper two sub-figures, a girth-8 $(3,15,20,384)$ binary LDPC-CC with BPSK modulation is investigated. For the lower two sub-figures, a girth-8 $(3,15,20,192)$ nonbinary LDPC-CC over $\mathbb{GF}(4)$ with 4-QAM modulation is taken as examples for illustration. These two codes are generated by optimizing the index of circulant permutation matrix using the technique in [7]. The codeword length is 38400 and 19200, respectively. Both codes have the same code rate 0.78.

In is shown in each sub-figure that with a ten times complexity reduction, the performance degradation for the dynamic WD using $N_{\text{iter}}=10$ is within 0.1 dB compared to it using $N_{\text{iter}}=100$. The larger the window size, the better performance the dynamic WD scheme. With a sacrifice of less than 0.3 dB for a BER of 10^{-4}, $N_{\text{iter}}=5$ can be employed if the latency is of high priority. Note that the phenomenon that the dynamic WD can outperform BP in low SNR regions is not observed in the exiting WD references.

4. Conclusions

In this paper, we propose a dynamic WD scheme for decoding LDPC-CCs. The BER simulation results demonstrate the effectiveness of our proposed WD schemes. Due to the low-latency and memory efficient natures, our proposed WD schemes can provide a possible solution for low-latency optical communication systems.

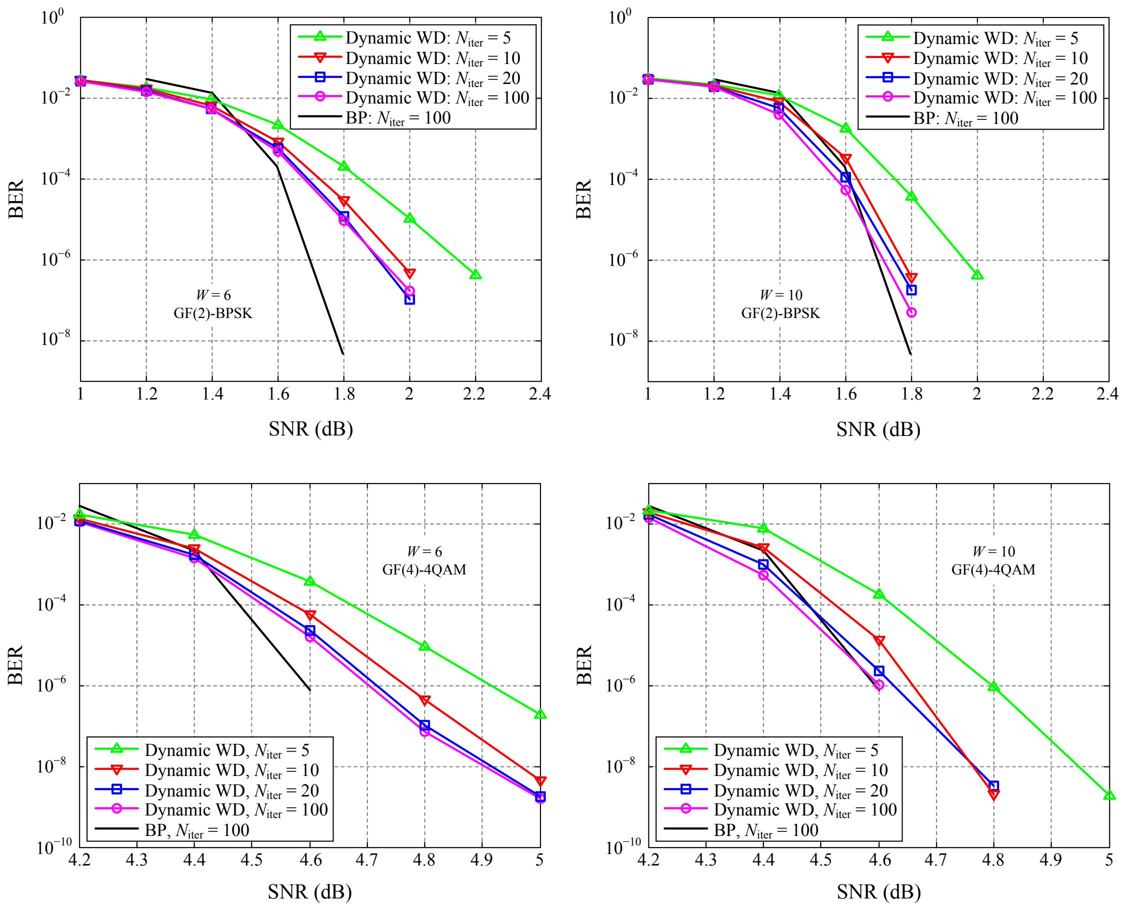

Fig. 2. The BER performances of dynamic WD scheme with window size $W = 6, 10$ for $N_{\text{iter}} = 5, 10, 20, 100$.

References

1. A. Leven and L. Schmalen, "Status and recent advances on forward error correction technologies for lightwave systems," J. Lightw. Technol. **32**, 2735–2750 (2014).
2. Deyuan Chang et al., "LDPC convolutional codes using layered decoding algorithm for high speed coherent optical transmission," OFC/NFOEC 2012 pp. 1–3 (2012).
3. M. Lentmaier, A. Sridharan, D. Costello, and K. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," IEEE Trans. Inf. Theory **56**, 5274–5289 (2010).
4. A.R. Iyengar et al., "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," IEEE Trans. Inf. Theory **58**, 2303–2320 (2012).
5. L. Schmalen, V. Aref, J. Cho, K. Mahdaviani, "Next generation error correcting codes for lightwave systems," ECOC **Th.1.3.3** (2014).
6. M. Arabaci, I. Djordjevic, R. Saunders, and R. Marcoccia, "High-rate nonbinary regular quasi-cyclic LDPC codes for optical communications," J. Lightw. Technol. **27**, 5261–5267 (2009).
7. Y. Wang, S. Draper, and J. Yedidia, "Hierarchical and high-girth QC LDPC codes," IEEE Trans. Inf. Theory **59**, 4553–4583 (2013).