Performance Comparison between Sum-Product and Bit-Flipping decoding Algorithms of Gallager's Construction for regular LDPC Convolutional Codes

Oulfa LAOUAR
Laboratoire de Recherche en Electronique de Skikda(LRES)
Université de 20 Aout 1955
Skikda, Algeria
o.laouar@univ-skikda.dz

Nadir DEROUICHE
Laboratoire de Recherche en Electronique de Skikda(LRES)
Université de 20 Aout 1955
Skikda, Algeria
n.derouiche@univ-skikda.dz

Imed AMAMRA
Laboratoire de Recherche en Electronique de Skikda(LRES)
Université de 20 Aout 1955
Skikda, Algeria
i.amamra@univ-skikda.dz

***Abstract*—In this research, we present a performance evaluation and comparison, in terms of complexity and performance, of two decoding algorithms for regular LDPC convolutional codes with Gallager's construction. We define some characteristics of LDPC Convolutional ensembles that enable them to perform better with the Windowed decoder. The Belief Propagation decoding is used to discuss both Bit-Flipping and Sum-Product algorithms. We will focus at how these ensembles and codes perform over an AWGN channel using the BPSK modulation. Our simulation results using the sliding window decoding procedure shows that the Sum-product decoding algorithm outperforms the Bit-Flipping one.**

***Index Terms*— Belief Propagation (BP), Bit Error Rate (BER), Bit-Flipping (BF), Gallager's construction, LDPC Convolutional codes (LDPC-CCs), Sum-product (SP).**

I. Introduction

In the early 1960's Gallager [1] introduced Low-Density Parity-Check (LDPC) codes, which are one of the most important Forward Error Correcting (FEC) codes, which have been showing to perform well on a variety of channels.

Felstrom and Zigangarov [2] proposed the convolutional counterparts of LDPC block codes (LDPC-BC), namely LDPC Convolutional Codes (LDPC-CC). After that, a lot of attention has been attracted to these codes due to many additional advantages against the LDPC codes. LDPC-CC have a very flexible code length, comparable performance to LDPC-BC, simple encoding, more excellent error correction performance for the start bits of a frame, which contain some important information [3]. Like LDPC-BC, LDPC-CC are constructed by sparse parity-check matrices and are decoded iteratively, but on the other hand, a sliding window-based strategy can be used with the iterative message passing algorithms, as Belief Propagation (BP), which appeared to be very promising in terms of achieving low-error probabilities at high data rates. Iterative decoding with its different algorithms has attracted a lot of interest in recent years. The Message-Passing Algorithm (MPA) iteratively determines the A-Posteriori Probability (APP) of bits in the transmitted codeword provided the received word using the BP algorithm [4]. The concept behind BP decoding is to exchange uncertainties between correlated bits according to the encoder's or parity-check matrices (PCM) requirements.

In this paper, we present a review of Sum-Product algorithm (SP) [5], which is a common algorithm to understand a large number of widely varying algorithms, constructed in the fields of computer science and engineering. We compared its Bit Error Rate (BER) performance under several assumptions (equal decoding memory requirements and equal decoding code period) with Bit-Flipping algorithm (BF) [6] based BP which is proposed for decreasing the computational complexity.

The rest of the paper is organized as follows: We begin in section II by briefly introducing the LDP-CCs and their main parameters with a presentation of Gallager's construction method. In the next section III, we focus firstly on encoding steps of LDPC-CC, and then on the two different decoding BP algorithm methods (Sum-Product and Bit-Flipping) using the sliding-window strategy. The main part of the paper is section IV, where comparisons between the SP and BF decoding algorithms are presented, based on several criteria for Gallager's construction matrices. Finally, section V concluded the paper.

II. Fundamentals of LDPC-CCs

An LDPC Convolutional code (m_s, j, k) is defined by an infinite time-varying parity-check matrix $H_{[0,\infty]}$ which is periodically time-varying in practice. The code is given as the null space of $H_{[0,\infty]}$, or as a set of sequences satisfying the equation:

$$H_{[0,\infty]} v_{[0,\infty]}^{T} = 0^{T} \quad (1)$$

Both v and H are defined over $GF(2)$ in binary codes. The parity-check matrix of these codes with a rate $R = b/c$ $(b < c)$ is constructed by coupling the parity-check matrices of LDPC-BCs as follows:

$$H_{[0,\infty]} = \begin{bmatrix} H_0(0) & & & & & \\ H_1(1) & H_0(1) & & & & \\ \vdots & \vdots & \ddots & & & \\ H_{m_s}(m_s) & H_{m_s-1}(m_s) & \cdots & H_0(m_s) & & \\ & H_{m_s}(m_s+1) & H_{m_s-1}(m_s+1) & \cdots & H_0(m_s+1) & \\ & & \ddots & & & \ddots \\ & & H_{m_s}(t) & H_{m_s-1}(t) & \cdots & H_0(t) \\ & & \ddots & & & \ddots \end{bmatrix} \quad (2)$$

Where m_s refers for the parity-check matrix's syndrome former memory. Furthermore, the $H_i(t), i = 0,1,\dots,m_s$ sub-matrices are binary $(c-b) \times c$ given by:

$$H_i(t) = \begin{bmatrix} h_i^{(1,1)}(t) & \cdots & h_i^{(1,c)}(t) \\ \vdots & & \vdots \\ h_i^{(c-b,1)}(t) & \cdots & h_i^{(c-b,c)}(t) \end{bmatrix} \quad (3)$$

The syndrome matrix $H_{[0,\infty]}$ in (2) ignores the irregularity at the beginning, if $H_i(t)$ are full rank for all time instances t. Starting from the $(m_s \cdot (c-b)+1)$-th row and $(m_s \cdot c + 1)$-th column, $H_{[0,\infty]}$ has exactly k ones in each row and j ones in each column.

- Gallager's construction matrices:

In this paper, we construct regular matrices according to Gallager's method [1], which is defined as a matrix with column and row weights respectively d_v and d_c.

$$H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{d_v} \end{bmatrix} \tag{4}$$

Gallager's method construct a parity-check matrix with the form defined in Eq.(4) that can be divided into d_v sub-blocks. The first sub-block H_1 is structured by forming a $h_{d_v} \times h_{d_c}$sub-matrix that consists of $d_c(h_{d_v} \times h_{d_c})$ with column weight 1 and row weight d_c. However, the other sub-blocks are of random column permutation version of H_1. Figures 1 and 2 present Gallager's construction matrices used in Section IV for simulations.

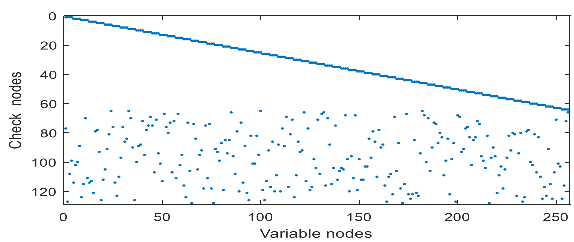

Fig. 1. Gallager's parity-check matrix of a code $(256,2,4)$.

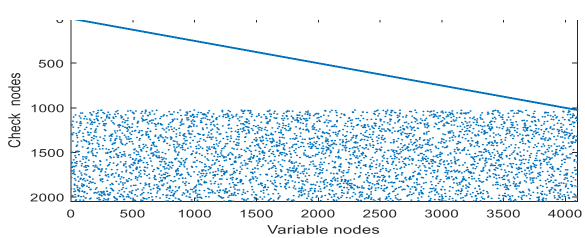

Fig. 2. Gallager's parity-check matrix of a code $(4096,2,4)$.

III. Encoder and Decoder architecture

A data frame has a start and an end in most practical systems. Therefore, the LDPC-CC encoder and decoder must comply.

A. LDPC Convolutional Codes Encoder

Even if both of LDPC-BCs and LDPC-CCs encoders add redundancy to a stream of information bits, their implementation complexity is different. The implementation of an LDPC-CC encoder is usually much simpler than the implementation of an LDPC-BC encoder with identical characteristics.

The LDPC-CC encoder needs to know previous information bits as well as previous parity bits in order to generate the parity-bit $v(t)$. There is no need to implement a method of control to divide the input sequence into fixed-length blocks with LDPC-CCs encoders. At time t, the encoder output $v(t)$ is a function of the current input $u(t)$and even the encoder state.

The encoded sequence for a rate $1/2$ systematic LDPC-CC can be determined as follows:

$$v_1(t) = u(t) \tag{5}$$

$$v_2(t) = \sum_{i=0}^{m_s} h_1^{(i)}(t)\, u(t-i) + \sum_{i=1}^{m_s} h_2^{(i)}(t)\, v_2(t-i) \tag{6}$$

Figure 3 illustrates a regular LDPC-CC encoder of a rate $1/2$. The parameters α and β change with the period $T = m_s + 1$. A $'0'$ for α or β indicates disassociation, while a $'1'$ signifies association.

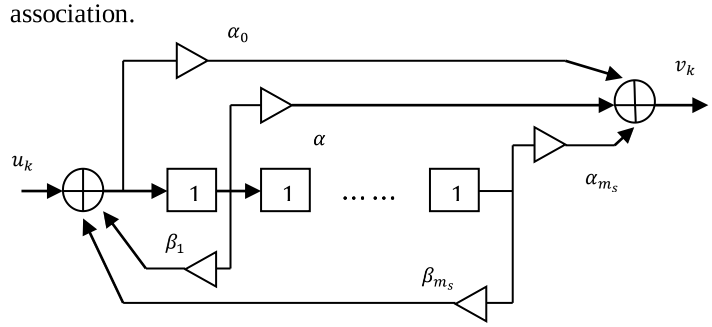

Fig. 3. LDPC-CC encoder for rate $R = 1/2$.

The figure shows that there are two outputs for each input u_k. The first output is systematic, which is u_k, while the second is the parity, which is v_k.

For a rate $R = 1/2$, if the last row $(c-b)$ of the sub-matrices $H_0(t)$ is linearly independent, and the first b symbols of $v(t)$ coincide with the information symbols, the LDPC-CC encoder is systematic [7] as follows:

$$v_t^{(1)} = u_t^{(1)}$$

B. LDPC Convolutional Codes Decoder

A soft-decision, iterative message-passing decoding algorithm is used by LDPC Convolutional codes. Messages containing probabilistic information are passed from one decoder processor to the next in a chain of similar processors using the message passing algorithm (MPA) due to the large values of the syndrome memory (m_s) that make the trellis representation of the LDPC-CC impossible which leads to the use of the iteration message passing decoding.

The Belief Propagation (BP) iterative algorithm, first introduced in [2], is usually used in the LDPC-CC decoding algorithm. The LDPC-CC decoding attempts to reconstruct the transmitted codeword from the possibly distorted received word by using the parity-check matrix H, which can be

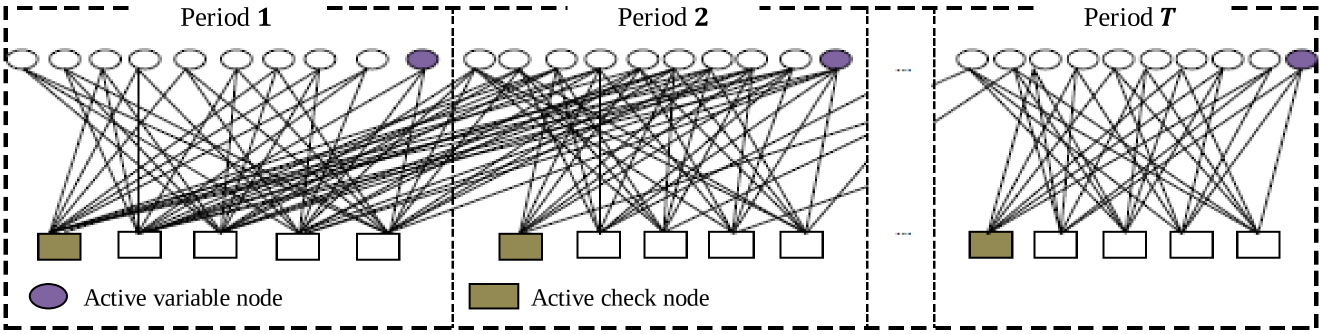

Fig. 4. Continuous decoding of LDPC-CC with T periods.

represented by an infinite Tanner graph, with columns corresponding to variable nodes (VNs) and rows corresponding to check nodes (CNs) as Figure 5 indicates:

$$\boldsymbol{H} = \overbrace{\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \cdots & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & & \mathbf{0} \\ \vdots & & & & & \vdots \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \end{bmatrix}}^{VNs} \Bigg\} \; CNs$$

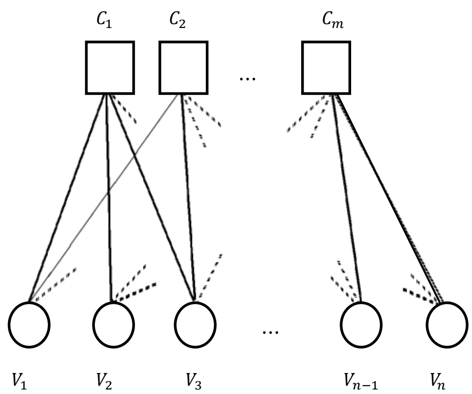

Fig. 5. General Tanner graph representation of the parity-check matrix H.

Continuous decoding of LDPC-CC with T periods, based on Tanner graph is presented in Figure 4.

The types of passed messages or those of operations performed at the nodes are used to classify different MPA [8]:

- BF decoding Algorithm:

For LDPC-CC, the BF algorithm is a hard decision MP algorithm that also works for LDPC-BC. Each check node sends a message to each linked bit node, declaring what value the bit is based on the check node's information. If the modulo-2 sum of equation (1) is zero, the check node determines that this equation is satisfied.

If the majority of received messages vary from the received values, the current value of the bit node is flipped. The BF algorithm's basic principle is that a codeword bit that appears in a large number of incorrect check equations is likely to be incorrect as well. Between the advantages of using this approach is that once a solution is found, additional iterations are avoided, and the inability to converge to a codeword is often detected.

Paper [6] proposed a new class of iterative BF decoding algorithm based on LDPC Convolutional codes, which improves Gallager's original BF algorithm [1] in terms of both coding gain and error correction speed. The BF algorithm works on the assumption that the greater the number of unsatisfied parity checks equations involving a specific bit, the more possible that bit is incorrect.

- SP decoding Algorithm:

The Tanner graph can be used to define an iterative decoder for LDPC-CCs. Messages are exchanged between the symbol nodes and the check nodes during all decoding iterations [5]. In this paper, an algorithm that is similar to Gallager's probabilistic iterative decoding algorithm known as the Sum-Product algorithm is presented.

The SP algorithm is a probabilistic soft decision message passing algorithm in which the messages represent each decision. The sum-Product algorithm is similar to the BF algorithm, except that the messages which represent each decision (whether the bit value is 0 or 1) are probabilities instead of bits. Since the input bit probabilities for the received bits were known in advance before running the LDPC-CC decoder, they are referred to as a priori probabilities [8], and the bit probabilities returned by the decoder are referred to as a posteriori probabilities. These probabilities are expressed as log-likelihood ratios (LLR) in sum product decoding as follows:

$$LLR(x) = \log\left(\frac{p(x=0)}{p(x=1)}\right) \qquad (7)$$

For AWGN channel, $y_i = x_i + n_i$ is the $i\,th$ received sample, where n_i are independent and normally distributed as ${}^{N_0}/_{2}$, and N_0 is the noise density which its estimated value is necessary in practice.

- Windowed Decoding (WD) for BF and SP algorithms:

For LDPC Convolutional codes, Belief Propagation decoding algorithms are usually used. However, like block code decoding, these algorithms' iterations cannot continuously use the messages of recently joined variable nodes within the next iteration windows, limiting decoding performance. The windowed scheme is proposed for LDPC-CC decoding [9], and it is verified that it can achieve good performance. The fusion of two different iterative decoders is considered: the conventional BP decoder (Bit-Flipping and Sum-Product) and the variant Windowed Decoder.

The VNs connected to the same parity-check equations are constrained by the code's convolutional structure. This property can be used to perform continuous decoding of the received stream using a "Window" that slides along the bit sequence. This structure allows parallelization of the MP decoder's iterations across multiple processors operating in different Tanner graph regions.

Long LDPC-C codes decoded with BP can have a high latency and require a large memory. The advantage of limiting BP decoding to a window is that it reduces decoding latency, complexity and memory use. The LDPC-CC parity matrix has a nonzero diagonal band structure represented with the gray area, as seen in Figure 6. Therefore the BP decoding can be constrained in a window, and the condition [10]:

$$(w+1) \leq W \leq L \qquad (8)$$

must be satisfied for the window size W with L copies of an LDPC-CC and coupling width w.

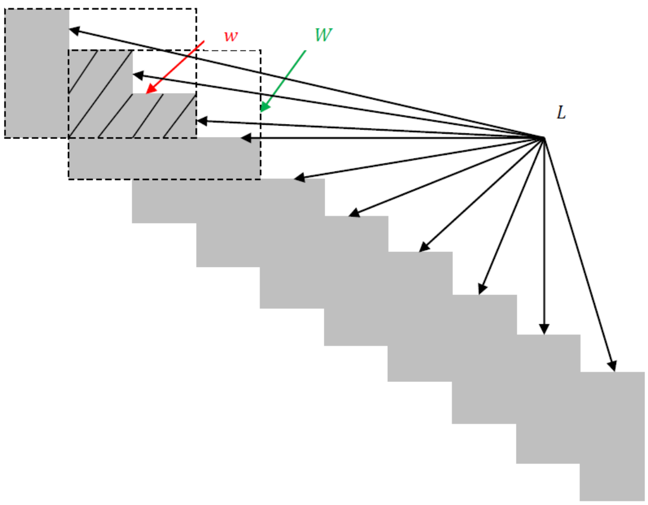

Fig. 6. WD of LDPC-CC where $L = 10, w = 2$ and $W = 3$.

The procedure is then repeated until the entire codeword is decoded.

Although an LDPC-CC's Tanner graph has an infinite number of nodes, the distance between two variable nodes connected to the same check node is limited by the code's memory. This allows continuous decoding using a decoder that slides a finite window along the received sequence.

In this paper, after receiving the partial codeword, the WD starts BP decoding through a partial Tanner graph of the codeword and slides along the diagonal band with an overlap between the windows.

Example: Considering the regular full-rank parity-check matrix:

$$H = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

With size $M \times N = 5 \times 10$, of a rate-1/2 block code. As indicated above, we cut the parity-check matrix into two parts (red and black), where each cutting pattern is such that we repeatedly move $c = 2$ columns to the right and $b = 1$ row down. After achieving this diagonal cut, we copy and paste the two parts ($L = 3$) following the unwrapping method presented in [11] to obtain the matrix shown in Figure 7:

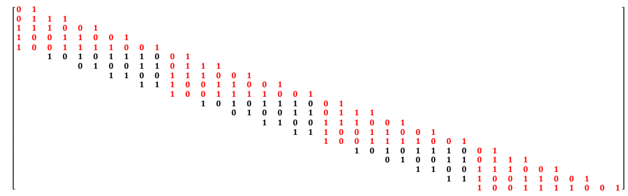

Fig. 7. Deriving a rate $R = 1/2$ periodically time-varying convolutional code from a rate-1/2 block code of length 10.

The received codeword using random $u(t)$ is as follows:

$$V = [\underbrace{0\,0\,0\,0\,0\,0\,0\,0\,0\,0}_{V_0}\;\underbrace{1\,0\,1\,1\,0\,0\,1\,0\,1\,0}_{V_1}\;\underbrace{0\,0\,0\,0\,1\,1\,1\,1\,1\,1}_{V_2}\;\underbrace{0\,1\,1\,1\,1\,0\,0\,0\,1\,0}_{V_3}]$$

Applying the WD on the first part of the LDPC-CC matrix:

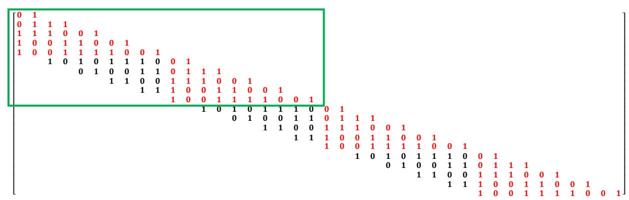

Fig. 8. WD applied on $[V_0 V_1]$.

The WD then slides to the next sub-vector to decode it as follows:

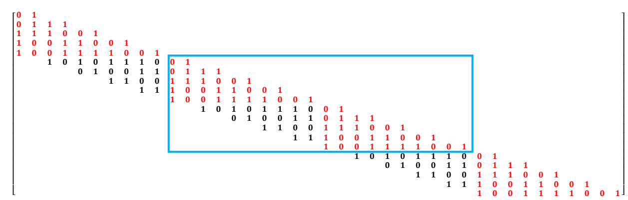

Fig. 9. WD applied on $[V_1 V_2]$.

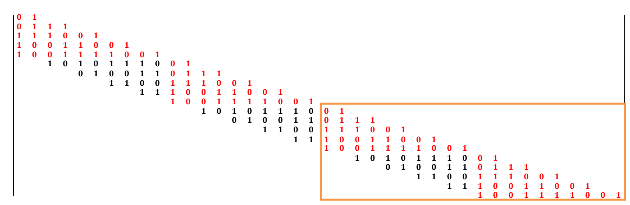

Fig. 10. WD applied on $[V_2 V_3]$.

As Figure 11 shows, the WD is then slides with overlap of M rows and N columns between windows :

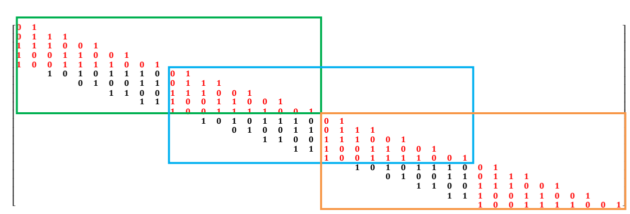

Fig. 11. WD applied on V.

The WD terminates and outputs the current sub-decoding vector's result once the BER of the sub-vector is 0 or the number of iterations achieves the limit. It is clear for each iteration; the previous sub-vector is re-decoded with the present sub-vector, which increases the decoding performance.

IV. Simulation Results

This section examines the performance of regular LDPC-CC for Gallager's construction matrices shown in Figure 1 and 2. As well as, the performance curves of WD-BF and WD-SP under AWGN channel are presented in Figure 12.

To compare BER performance of a rate$-1/2$, the corresponding LDPC-BCs with block length $N = 256, N = 4096$ are chosen respectively for constructing LDPC-CCs of periods $(T = 128, T = 2048)$ under the same processor complexity decoding, where the LDPC-CCs are decoded using a sliding-window BF and SP algorithms.

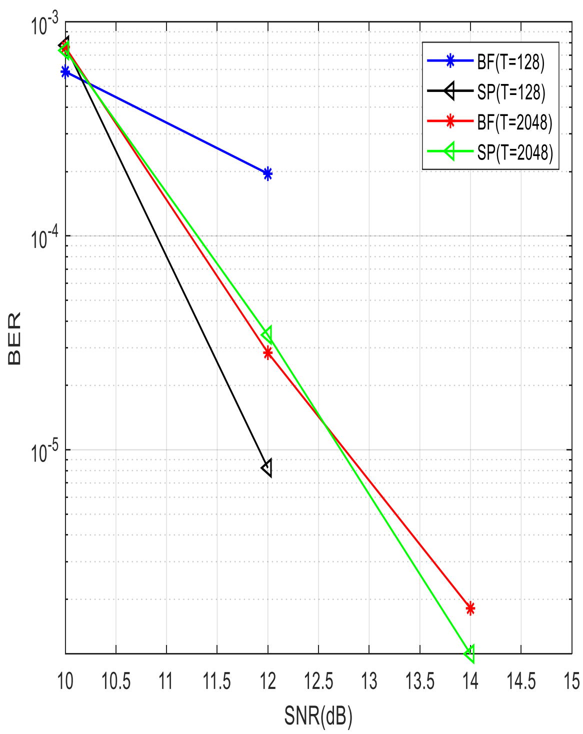

Fig. 12. Performance Comparison between WD-BF and WD-SP of Gallager's Construction.

As can be observed for $T = 128$, a BER of 2×10^{-4} and 8×10^{-6} at $12dB$ are obtained using WD-BF and WD-SP respectively. Also for $T = 2048$, a BER of 1.8×10^{-6} and 10^{-6} at $14\, dB$ are obtained using WD-BF and WD-SP respectively. From which we can say that WD-SP outperforms the corresponding WD-BF for the same period.

Also, it has been shown that BER values for $T = 2048$ are more reduced than those for $T = 128$, and the corresponding SNR is increased. Which means that by increasing the period, we can attend a better BER values.

V. Conclusion

Performances of both WD-BF and WD-SP Belief propagation decoding algorithms for LDPC-CC are evaluated in terms of Bit Error Rate (BER).

Sum-Product decoding of LDPC-CCs retains many of the desirable features of Bit-Flipping decoding, and LDPC-CC with large constraint length v_s can be decoded using a sliding window message passing decoder. For each iteration, the sliding-window decoder takes from the previous decoding window a sub-vector overlapping with the present window, in order to decode it for the second time. This strategy is responsible for increasing decoding performance.

Sum-Product decoders are excellent algorithms for practical communication conditions due to their excellent BER performances. Simulation results show that the WD-SP algorithm for regular LDPC-CC outperforms the WD-BF algorithm with comparable processor complexity for moderate constraint length codes.

References

[1] R. G. Gallager, "Low-Density Parity-Check Codes". Cambridge, MA: MIT Press, 1963.
[2] A. Jimenez-Felstrom¨ and K. Sh. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," IEEE Trans. Inform. Theory, vol. IT-45, pp. 2181–2191, Sept. 1999.
[3] Y. Chen, C. Zhou, Y. Huang, and X. Zeng, "An efficient multirate LDPC-CC decoder with layered decoding algorithm," in IEEE International Conference on Communications (ICC), June 2013.
[4] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," IEEE Trans. Inform. Theory, vol. 42, pp. 429–445, Mar. 1996.
[5] M. Lentmaier, A. Sridharan, K. S. Zigangirov, and D. J. Costello, Jr., "Terminated LDPC convolutional codes with thresholds close to capacity," in Proc. IEEE Int. Symp. Inf. Theory, Adelaide, Australia, Sep. 2005, pp. 1372–1376.
[6] X. S. Zhou, B. F. Cockburn and S. Bates., "Improved Iterative Bit Flipping Decoding Algorithms for LDPC Convolutional Codes", IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Canada, 2007.
[7] A. E. Pusane, K. Sh. Zigangirov, and D. J. Costello, Jr., "Construction of irregular LDPC convolutional codes with fast decoding", Proc. IEEE Int. Conf. on Communications, Istanbul, Turkey, June 2006.
[8] R. Lakshmi, G. Vijitha., "The Performance of LDPC Convolutional Codes-A Quantitative Analysis", in IJIRCCE, vol. 3, Issue 6, pp. 5402-5403, June 2015.
[9] A. R. 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, Apr. 2012.
[10] S. Mo, L. Chen, Improved sliding window decoding of spatially coupled low-density parity-check codes, in: Proc. 2017 IEEE Inf. Theory Work. (ITW), Kaohsiung, Taiwan, 2017, pp. 126–130.
[11] A. E. Pusane, R. Smarandache, P. O. Vontobel, and D. J. Costello, Jr., "Deriving good LDPC convolutional codes from LDPC block codes," IEEE Trans. Inf. Theory, vol. 57, no. 2, pp. 835–857, Feb. 2011.