BLIND TIME-PERIOD SYNCHRONIZATION FOR LDPC CONVOLUTIONALLY CODED TRANSMISSION

Čedomir Stefanović, Dejan Vukobratović, Dragana Bajić

Dept. of Power, Elect. and Comm. Engineering
University of Novi Sad, Novi Sad, Serbia
phone: +381-21-485-2525
email:{cex,dejanv,dbajic}@uns.ac.rs

Lina Stanković, Vladimir Stanković

Dept. of Electronic and Electrical Engineering
University of Strathclyde, Glasgow, UK
phone: +44-141-548-2679
email:{vladimir,lina}.stankovic@eee.strath.ac.uk

ABSTRACT

In this paper we propose a scheme for blind time-period synchronization of Low-Density Parity-Check Convolutional Codes (LDPC CC) over the AWGN channel. Reliable time-period synchronization for LDPC CC coded transmission is a requirement for successful decoding at the receiving end. The proposed scheme exploits parity-check constraints incorporated into LDPC CC coded stream, and variants based both on the hard and soft-detected symbols are presented. We show that time-period synchronization can be acquired during the time-frame of one LDPC CC time period, which is considerably faster and less complex when compared to the block LDPC coded transmission of similar performance. Additionally, by appending the time-period synchronization preprocessors, we effectively incorporate the time-period synchronization into the iterative LDPC CC decoder "pipeline" structure. Our simulation study demonstrates flexibility and excellent performance of the proposed scheme.

1. INTRODUCTION

A rate b/c LDPC convolutional code (LDPC CC) encoder outputs a block of c channel symbols for every input block of b information symbols. The output block consists of information and parity symbols, where parity symbols are derived using information symbols from current and previous M blocks. Both encoding and decoding of LDPC CC are performed using syndrome-former matrix H^T (transpose of the parity-check matrix), which is infinite but has a periodic structure. In order to perform successful decoding of the received LDPC CC coded stream, the decoder should be time aligned with the periods of the syndrome-former matrix, i.e., time-period synchronization is required on the receiving side.

Time-period synchronization for LDPC CC codes is analogues to node synchronization of conventional convolutional codes decoded by a Viterbi decoder, treated in [1]-[5]. The node synchronization scheme presented in [1] for BPSK modulated rate $1/2$ code uses hard values of the received symbols and observes the syndrome. For the incorrect offset, content of the syndrome is random, while for the correct offset, a syndrome mostly contains zeroes (some errors could occur due to the noise). The extension of these results to other code rates and modulation formats is given in [2] and [3]. A further upgrade of the syndrome based node synchronization is given in [4], where soft information is used in order to improve the decision. Another approach is given in [5], where decoding and subsequent re-encoding is performed for one of the two possible offset positions for rate $1/2$ codes. The re-encoded stream is then compared to hard detected channel symbols using metric based on correlation with soft values of the channel symbols, after which the offset position is chosen based on the value of the metric.

On the other hand, time-period synchronization for LDPC CC codes is similar in nature to the blind (or pilotless) frame synchronization of block-coded data. Blind frame synchronization refers to the frame synchronization of coded systems where code redundancy is used in finding the start of the frame/codeword. With the introduction of capacity-achieving iteratively decodable codes, such as LDPC codes, blind frame synchronization has gained an increased attention recently [6]-[10]. In [6], the metric used to discriminate the beginning of the LDPC coded frame among the candidate positions is the mean of the absolute values of variable node Log-Likelihood Ratios (LLRs), which are outputs of the sum-product algorithm after a single iteration. Although the results obtained this way are good, the procedure is costly in terms of the number of required operations. A simplified approach to frame synchronization is to use the metric based on constraints (check nodes). In [7][8], a simple hard-decision detection of the beginning of the frame is employed in LDPC coded transmission. For every bit position in the time-frame of one whole codeword, the syndrome is formed using hard values of the received bits. The output of the algorithm is the position for which the syndrome contains the largest number of satisfied constraints. In [9], both hard-decision and soft-decision detection algorithms were analyzed. While the hard-decision variant is essentially the same as in [8], the soft-decision variant is based on the soft-values of the check node LLRs. The output of the soft-decision algorithm is the position for which the sum of the LLRs of the check nodes is largest.

In this paper, we propose a scheme for blind time-period synchronization for LDPC CC coded transmission over the additive white Gaussian noise (AWGN) channel. Although the proposed approach is similar to the one used for LDPC block codes [7]-[10], it clearly demonstrates the "synchronization" advantages of LDPC CC codes. We show that the time-period synchronization for LDPC CC codes, which is equivalent task as frame/codeword synchronization of LDPC codes, can be acquired during the time frame of one LDPC CC time period. It is faster and less complex as compared to the block LDPC coded transmission of similar performance. We incorporate the time-period synchronization procedure as the set of preprocessors preceding the iterative processors of the decoder's "pipeline" structure. Simulation results demonstrate that a time-period synchronization scheme based on LDPC CC is a flexible and efficient solution.

2. LDPC CONVOLUTIONAL CODES

In this section, we review LDPC CC following their exposition and the notation used in [11]. A rate $R = b/c$ binary LDPC CC with syndrome-former memory M is defined using its syndrome-former matrix H^T given by:

$$H^T = \begin{bmatrix} H_0^{(0)} & H_1^{(1)} & \dots & H_t^{(M)} & & \\ & H_1^{(0)} & \dots & H_t^{(M-1)} & H_{t+1}^{(M)} & \\ & & \ddots & \vdots & \vdots & \ddots \end{bmatrix}.$$

H^T consists of $c \times (c-b)$ binary submatrices $H_t^{(m)}$, where $t = 0, 1, \dots$, and $m = 1, 2, \dots, M$, which fill the "diagonal stripe" of width equal to $M+1$ submatrices. Each $c \times (c-b)$ submatrix $H_t^{(m)}$ is of the form:

$$H_t^{(m)} = \begin{bmatrix} h_{1,1}^{t,m} & \dots & h_{1,c-b}^{t,m} \\ \vdots & \ddots & \vdots \\ h_{c,1}^{t,m} & \dots & h_{c,c-b}^{t,m} \end{bmatrix}.$$

The submatrices $H_t^{(m)}$ of a time-varying LDPC CC differ for different time instants t. When $H_t^{(m)} = H_{t+T}^{(m)}$ for any t and m, the LDPC CC is periodic with period T.

The syndrome-former matrix H^T defines the corresponding LDPC CC as a set of all encoded sequences (codewords):

$$x_{[0,\infty]} = [x_0, x_1, \dots, x_t, \dots], \tag{1}$$

where $x_t = [x_t^{(1)}, x_t^{(2)}, \dots, x_t^{(c)}]$, such that they satisfy the following parity-check equation:

$$x_{[0,\infty]} H^T = 0. \tag{2}$$

The LDPC CC codeword $x_{[0,\infty]}$ is obtained after LDPC CC encoding of the input information sequence:

$$u_{[0,\infty]} = [u_0, u_1, \dots, u_t, \dots], \tag{3}$$

where $u_t = [u_t^{(1)}, u_t^{(2)}, \dots, u_t^{(b)}]$ is a b-bit sequence. The input information sequence is of arbitrary length, which is important flexibility offered by LDPC CC codes when applied to variable length frame transmission. After transmission over a noisy communication channel, the LDPC CC codeword is transformed into the received sequence:

$$y_{[0,\infty]} = [y_0, y_1, \dots, y_t, \dots], \tag{4}$$

where $y_t = [y_t^{(1)}, y_t^{(2)}, \dots, y_t^{(c)}]$. The received sequence $y_{[0,\infty]}$ is passed to the iterative LDPC CC decoder.

The "diagonal stripe" structure of the LDPC CC parity-check matrices introduces additional, temporal, dimension in their code (Tanner) graph. In this representation, each symbol or check node in the code graph is connected to its neighbors which are in the neighborhood of maximum M time instants away. This enables application of the "pipelined" iterative Belief-Propagation (BP) decoder that consists of a sequence of I consecutive "iteration processors" sliding along the code graph structure. The details on the LDPC CC pipeline decoder can be found in [11].

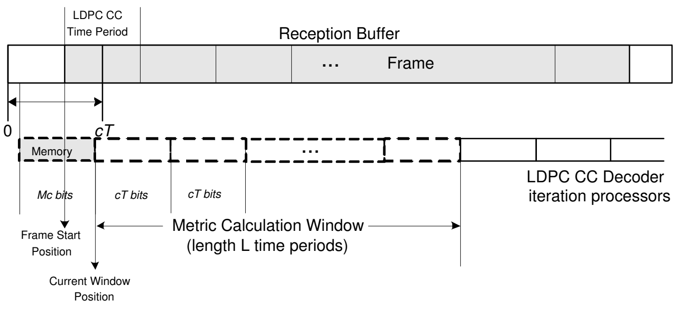

Figure 1: Time-period synchronization using LDPC CC codes.

3. TIME-PERIOD SYNCHRONIZATION SCHEME

We assume that the binary rate b/c LDPC CC is employed in coded transmission, where the symbols of the coded stream are BPSK modulated and sent over AWGN channel with zero mean and variance $N_0/2$ (N_0 is one-sided power spectral density of the AWGN). We also assume that ideal bit synchronization is achieved at the receiving end, where the received data is stored in the reception buffer (Fig. 1). As described in Section 2, LDPC CC are time periodic and their graph structure repeats every $(c-b)T$ check nodes (cT bit nodes).

The aim of the time-period synchronization is to align the decoder with the periodicity of the LDPC CC. Without reliable synchronization with the LDPC CC coded stream, the receiver cannot successfuly apply the iterative decoding procedure. As LDPC CC codewords are usually set to be whole number of the time periods (multiple of cT bits), the time-period synchronization may be considered equivalent to the frame synchronization for LDPC CC coded transmission.

To achieve time-period synchronization, the receiver uses a simplified and suboptimal hypothesis testing algorithm. Starting from each position within the first cT symbols of the reception buffer, the receiver examines the block containing L subblocks of cT consecutive received bits (metric calculation window, Fig. 1), taking into account the memory preamble of length Mc bits. For the μ-th shift of the metric calculation window, its content is denoted as $y_L(\mu) = (y_\mu, y_{\mu+1}, \dots, y_{LcT+\mu-1})$. The received bits in the metric calculation window are used to test the hypothesis that the window is aligned with the set of L consecutive time-periods of LDPC CC and for each position μ, a metric related to the probability of the hypothesis is calculated. The metric is based on the parity-check constraints that the received bits within the block should satisfy if the metric calculation window is aligned with L consecutive time-periods. The algorithm outputs the position $\hat{\mu}$ which results in the largest metric. In Fig. 1, the "memory preamble" is shown only for the first subblock, for the sake of figure clarity (memory preambles of the following subblocks overlap with the preceding subblocks).

3.1 Hard-Decision Detection

The hard-decision detection algorithm uses the hard-values of the received bits for the metric calculation. For every examined block the number of satisfied parity-check constraints is calculated. The output of the algorithm is the position for which the number of satisfied constraints is largest.

As shown in Section 2, there are $L(c-b)T$ parity checks in L periods of the LDPC CC code. Denoting as $c_i(\mu)$ the indicator function that the i-th check applied to the μ-th shift of metric calculation window is satisfied, the metric is:

$$C_H(\mu) = \sum_{i=1}^{L(c-b)T} c_i(\mu). \tag{5}$$

The output of the algorithm is:

$$\hat{\mu} = \arg \max_{0\le\mu\le cT-1} C_H(\mu) = \arg \max_{0\le\mu\le cT-1} \sum_{i=1}^{L(c-b)T} c_i(\mu). \tag{6}$$

The exact analysis of the above rule is complicated due to the dependencies between the parity-checks. However, if an interleaver is used (as in [8] and [9] for LDPC block code case), it can be assumed that the parity-checks are randomly satisfied/unsatisfied for the incorrect offset μ. In this case, the above sum behaves as a binomial random variable whose mean and variance are $\frac{L(c-b)T}{2}$ and $\frac{L(c-b)T}{4}$, respectively. For the correct offset position, an important phenomenon confirmed by the simulation results is that by increasing SNR, the expected number of satisfied parity-checks increases, but its variance increases as well, due to the dependencies between the parity-checks.

3.2 Soft-Decision Detection

In this case soft values of the received bits are used for calculating the probability that the current window position is aligned with L consecutive LDPC CC time-periods. This probability is computed as the probability that all the corresponding parity-checks are satisfied:

$$C_S(\mu) = P\big(c_i(\mu) = 1, \forall i \in \{1,\dots,L(c-b)T\} | y_L(\mu)\big). \tag{7}$$

Again, due to the parity-check inter-dependences, the above probability is analytically intractable. If we assume that the parity-checks are independent, we obtain:

$$C_S(\mu) \approx \prod_{i=1}^{L(c-b)T} P\big(c_i(\mu) = 1 | y_L(\mu)\big) \tag{8}$$

The probability that the check $c_i(\mu)$ is satisfied is:

$$P\big(c_i(\mu) = 1 | y_L(\mu)\big) = \frac{1 + \prod_{k\in X(i)}(1 - 2p_{k+\mu})}{2} \tag{9}$$

where $X(i)$ is the set of all indices of channel input bits x_j constrained by the i-th parity-check, and $p_{k+\mu}$ is:

$$p_{k+\mu} = P(x_{k+\mu} = 1 | y_{k+\mu}). \tag{10}$$

It can be shown that:

$$P\big(c_i(\mu) = 1 | y_L(\mu)\big) = \frac{1 + \prod_{k\in X(i)} \tanh \frac{LLR(y_{k+\mu})}{2}}{2}, \tag{11}$$

where, for the AWGN channel, $LLR(y_{k+\mu}) = 2\frac{y_{k+\mu}}{\sigma^2}$. If we take the logarithm of the right hand side of equation (8), and make use of equation (11), we obtain:

$$C_S(\mu) = \sum_{i=1}^{L(c-b)T} \ln \frac{1 + \prod_{k\in X(i)} \tanh \frac{LLR(y_{k+\mu})}{2}}{2}. \tag{12}$$

Due to monotonicity of ln function, a simplified version of the soft-decision metric can be obtained as:

$$C_S(\mu) \approx \sum_{i=1}^{L(c-b)T} \prod_{k\in X(i)} \tanh \frac{LLR(y_{k+\mu})}{2}. \tag{13}$$

Furthermore, due to monotonicity of $tanh$ function, the metric can be further simplified as:

$$C_S(\mu) \approx \sum_{i=1}^{L(c-b)T} \prod_{k\in X(i)} \tanh LLR(y_{k+\mu}). \tag{14}$$

Finally, the output of the soft-decision algorithm is:

$$\begin{aligned}\hat{\mu} &= \arg \max_{0\le\mu\le cT-1} C_S(\mu) = \\ &= \arg \max_{0\le\mu\le cT-1} \sum_{i=1}^{L(c-b)T} \prod_{k\in X(i)} \tanh LLR(y_{k+\mu}).\end{aligned} \tag{15}$$

As stated before, due to the structure of LDPC CC, the parity-check constraints in one LDPC CC time period depend on the codeword bits in the same time period, as well as on the Mc previous codeword bits, where M is the LDPC CC memory. Therefore, in order to calculate the (hard or soft) values of parity-check constraints in each of L subblocks of the metric calculation window, the Mc bits prior to each subblock are needed. For positions $\mu < Mc$ of the metric calculation window, the missing $Mc - \mu$ bits of the memory preamble are padded with zeroes, as in LDPC CC decoding scheme.

The time-period synchronization metric calculation window can be effectively incorporated into the structure of the iterative "pipeline" decoder for LDPC CC. It can be seen as the set of simple synchronization preprocessors that precede the iterative LDPC CC decoder iteration processors (Fig. 1). Each synchronization preprocessor contains the parity-checks of one LDPC CC time period. In the particularly important case of the LDPC CC code design, where the size of LDPC CC code memory is aligned with its time period, i.e., $T = M + 1$, the synchronization preprocessors and the iterative decoder processors all represent subblocks of the same size, equal $cT = (M+1)c$ bits.

4. SIMULATION RESULTS

To evaluate the LDPC CC time-period synchronization scheme we perform simulations and present Synchronization Error Rate (SER) results. We use rate $1/2$ regular LDPC CC, introduced by Feltstrom and Zigangirov in [11]. The parameters of these codes are $(M, J = 3(2), K = 5)$, $b = 1$, and $c = 2$, where J and K are symbol and check node degree, respectively. The LDPC CC memory is set to $M = 129, 257$ and 513, and the corresponding time period is equal to $T = M - 1$. A random interleaver of length equal to one LDPC CC code period (i.e., cT bits) is used.

We assume transmission of a randomly generated coded stream over an AWGN channel. At the receiver side, noisy bit values are stored in the reception buffer. The time-period borders are not aligned with the beginning of the buffer, i.e., the first "time-period start" is at the μ_S-th bit position of the buffer. For the time-period start position μ_S, we assume that the following holds: $1 \le \mu_S \le cT$ (Fig. 1).

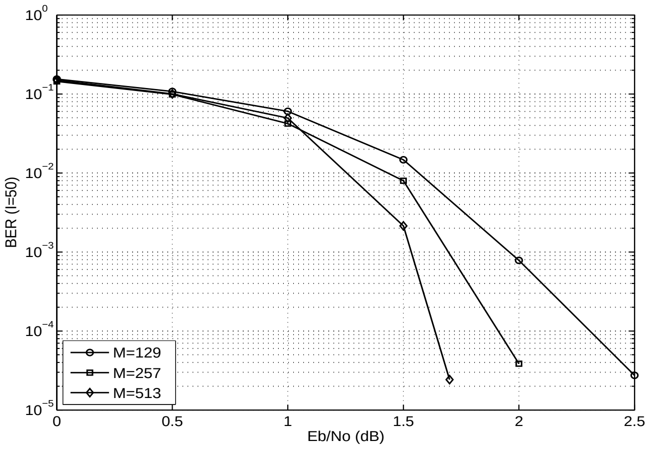

Figure 2: BER performance of LDPC CC for different memory lengths and $I = 50$ iterations of iterative BP decoder.

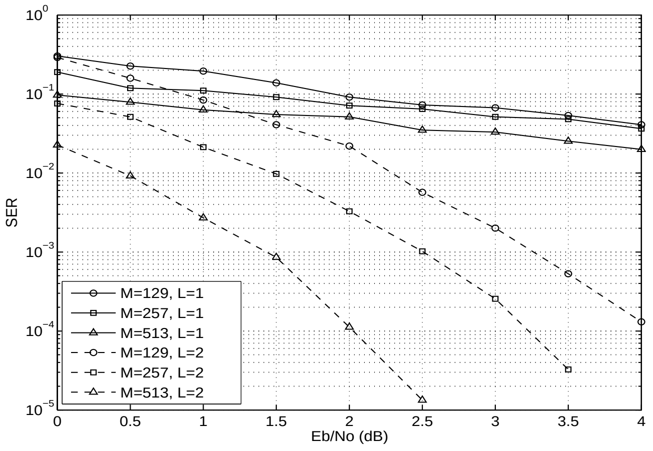

Figure 3: SER performance of LDPC CC for different memory lengths and Metric Calculation Window size equal to $L = 1$ or $L = 2$ using hard-decision detection.

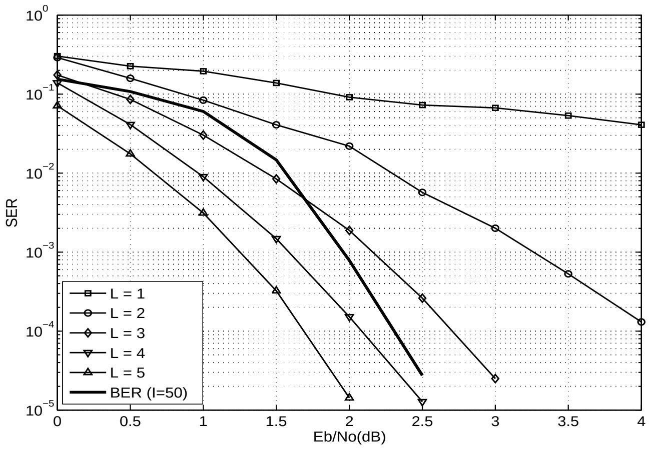

Figure 4: SER performance of LDPC CC for fixed memory $M = 129$ and varying Metric Calculation Window size using hard-decision detection.

The task of the time-period synchronization scheme is to provide an estimate $\hat{\mu}$ of the correct time-period start position μ_S. The SER performance of the scheme is estimated using the Monte Carlo method, where the time-period start estimation is repeated until a maximum number of 100 errors (i.e., incorrect time-period alignments) is encountered. In the following, we provide the simulation results of the LDPC CC coded transmission for both hard and soft-detection schemes.

The SER performance of the proposed scheme should be compared to the error-correcting bit error rate (BER) performance of the LDPC CC code, as the synchronization scheme should not compromise the error-correcting capabilities of the LDPC CC code. If we denote by P_b the BER when ideal time-period synchronization is achieved, by P_S the probability of time-period synchronization error (SER) and by P_{Tb} the total BER, the following holds:

$$P_{Tb} = (1-P_S)P_b + \frac{P_S}{2} = P_b + P_S(\frac{1}{2} - P_b) \approx P_b + \frac{P_S}{2} \quad (16)$$

where we assume that in the case of incorrect time-period synchronization the probability of bit error is $1/2$. From equation (16) it is obvious that P_S should be significantly lower than P_b, in order not to affect the error-correcting capabilities of the code. We assume that it is sufficient if P_S is lower than P_b for the order of magnitude. For the purpose of comparison, error correcting BER of rate $1/2$, regular $(M, J = 3(2), K = 5)$ LDPC CC code is presented in Fig. 2 for different memory lengths $M = 129, 257$ and 513. It is important to note the LDPC CC BER performance dependance on M, unlike on the codeword length as for LDPC block codes. It is shown in [11] that LDPC CC codes of modest memory lengths M outperform block LDPC codes of very large code lengths, for the same graph structure parameters.

4.1 Simulation Results: Hard-Decision Detection

In this subsection, we give the simulated SER performance of selected LDPC CC codes, where the hard-decision detection algorithm that employs equation (6) is applied.

Fig. 3 presents the SER performance of the $(M, J = 3(2), K = 5)$ LDPC CC. The LDPC CC memory (time period) is set to the values $129(128), 257(256)$ and $513(512)$. The metric calculation window of lengths $L = 1$ and $L = 2$ LDPC CC time periods is applied. For $L = 1$, the LDPC CC scheme shows modest performance, whereas for $L = 2$, the improvement is significant.

In Fig. 4, the SER performance of LDPC CC of memory 129 is presented for varying length of the metric calculation window. As expected, increasing the number of LDPC CC time periods contained in the metric calculation window improves the SER. We observe that, for $L = 5$, SER performance is sufficiently below the code error-correcting BER (solid line). The further increase in L does not affect the overall BER, as indicated by equation (16). Also, by increasing L it is possible to outperform LDPC CC with memory length 513 and $L = 2$ (Fig. 3), while at the same time the metric calculation complexity decreases: the LDPC CC scheme of memory 129 and $L = 5$ (Fig. 4) uses 640 parity-checks and the LDPC CC scheme of memory 513 and $L = 2$ uses 1024 parity-check. This important flexibility offered by LDPC CC enables progressive improvement in SER performance by including additional time-periods (preprocessors) in the system, for the price of a gradual increase in complexity.

4.2 Simulation Results: Soft-Decision Detection

In this subsection, the simulation results using the soft-decision detection algorithm of equation (15) are given. Fig. 5 presents the simulated SER performance of the LDPC CC codes of memory 129 and the metric calculation window lengths $L = 1$ and $L = 2$. The performance improvement of soft-decision algorithm, compared to the results obtained using the hard-decision algorithm, is significant. Sufficiently low SER performance with respect to error-correcting BER (given by solid line) is achieved already for $L = 2$, as opposed to the $L = 5$ for hard-decision detection (Fig. 4). However, this performance improvement is obtained for the price of a considerably higher metric calculation complexity.

If we denote the average parity-check weight of the LDPC CC code with $\bar{D}$, the number of operations for the hard-decision detection scheme is approximately $cT^2L(c-b)(\bar{D}-1)$ modulo-2 additions and cT comparisons, see equations (5) and (6). For the soft-decision detection scheme and if the look-up tables are used for $tanh$ function, there are $cT^2L(c-b)\bar{D}$ table look-ups, $cT^2L(c-b)(\bar{D}-1)$ real multiplications, approximately $cT^2L(c-b)$ real additions and cT comparisons, see eqns (14) and (15), which is a substantial increase in comparison with the hard-decision detection.

Finally, we compare the proposed scheme with the schemes for blind frame synchronization for LDPC block codes [7]-[10]. The LDPC CC time-period synchronization scheme is able to acquire the time-period synchronization in the interval of the order of time-period of the LDPC CC, while the block LDPC blind frame synchronization requires interval of the order of the code length. It is well known that the error-correcting performance of LDPC codes of a given code length is outperformed by the LDPC CC of considerably smaller memory and time-period lengths, affecting both the time-period synchronization time and complexity. For example, if we compare the metric calculation complexity by observing the number of parity-check constraints processed for different LDPC or LDPC CC frame synchronization schemes: the LDPC CC scheme of memory 129 and $L = 5$ (Fig. 4) uses only 640 parity-checks; the LDPC CC scheme of memory 513 and $L = 2$ (Fig. 3) uses 1024 parity-checks, and the (1944,972) LDPC scheme with $L = 2$ [7] uses 1944 parity-checks. Furthermore, the LDPC CC scheme of memory 129 and $L = 5$ demonstrates better SER (FER) performance than the (1944,972) LDPC scheme with $L = 2$ [7].

5. CONCLUSION

In this paper, a blind time-period synchronization scheme for LDPC convolutional codes is presented. The time-period synchronization acquisition using LDPC CC is shown to achieve a satisfactory SER performance compared to the BER performance of the code, while having low complexity and short time-period acquisition time. Additionally, the proposed scheme is flexible as it can easily accomodate variable length frames, and due to the fact that appending more time-period synchronization "preprocessors" in the metric calculation window, we can progressively improve the SER performance of the scheme, while gradually increasing its complexity.

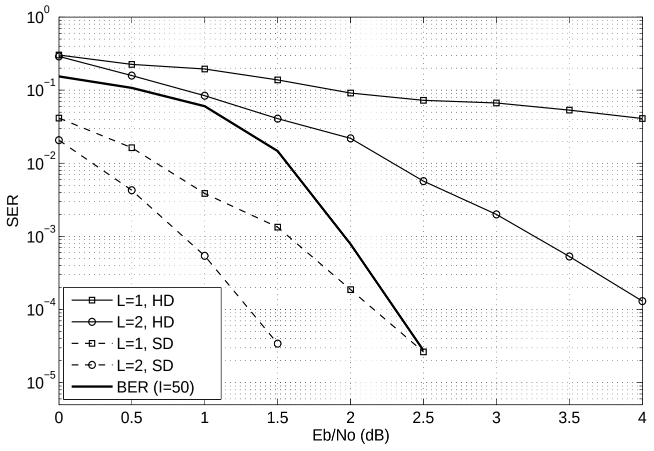

Figure 5: SER performance of LDPC CC of memory length $M = 129$ using hard and soft-decision detection algorithms.

REFERENCES

[1] G. Lorden, R. J. McEliece, and L. Swanson, "Node Synchronization for the Viterbi Decoder," *IEEE Trans. Comm.*, vol. 32, pp. 524–531, May 1984.

[2] M. Moeneclay and P. Sanders, "Syndrome-based Viterbi decoder node synchronization and out-of-lock detection," *in Proc. of GLOBECOM '90*, paper 407.4, San Diego, USA, 1990.

[3] M. L. de Mateo, "Node Synchronization techniques for any 1/n rate convolutional code," *in Proc. of ICC '91*, paper 52.1, Denver, USA, 1991.

[4] J. Sodha and D. Tait, "Soft-decision syndrome based node synchronization," *Electron. Lett.*, vol. 26, no.15, July 1990.

[5] U. Mengali, R. Pellizoni, and A. Spalvieri, "Soft-decision Based Node Synchronization for Viterbi Decoder," *IEEE Trans. Comm.*, vol. 43, pp. 2532–2539, Sep. 1995.

[6] W. Matsumoto, and H. Imai, "Blind Synchronization with Enhanced Sum-Product Algorithm for Low-Density Parity-Check Codes," *in Proc. WPMC '02* vol. 3, pp. 966–970, Honolulu, USA, 2002.

[7] D. Lee, H. Kim, C. R. Jones, and J. D. Villasenor, "Pilot-less Frame Synchronization via LDPC Code Constraint Feedback," *IEEE Comm. Letters,* vol. 11, pp. 683–685, Aug. 2007.

[8] D. Lee, H. Kim, C. R. Jones, and J. D. Villasenor, "Pilot-less Frame Synchronization for LDPC-Coded Transmission Systems," *IEEE Trans. Sign. Processing,* vol. 56, pp. 2865-2874, July 2008.

[9] R. Imad, S. Houcke, and C. Douillard, "Blind Frame Synchronization on Gaussian Channel," *in Proc. of EUSIPCO '07,* Poznan, Poland, Sep. 2007.

[10] J. Sun, and M. C. Valenti, "Optimum Frame Synchronization for Preamble-less Packet Transmission of Turbo-Codes, *in Proc. of the 38th Asilomar Conf.,* vol. 1, pp. 1126–1130, Pacific Ground, USA, 2004.

[11] A. Jimenez-Felstrom, and K. Sh. Zigangirov, "Time-Varying Periodic Convolutional Codes with Low-Density Parity-Check Matrices," *IEEE Trans. on Inform. Theory,* vol.45, pp. 2181–2191, Sept. 1999.