NB-LDPC check node with pre-sorted input

Cédric Marchand, Emmanuel Boutillon

NB-LDPC check node with pre-sorted input

Cédric Marchand, Emmanuel Boutillon

Université de Bretagne Sud, Lab-STICC (UMR 6285), Lorient, France.

Email: name.surname@univ-ubs.fr

Abstract—**Non-binary low-density parity-check codes have better communication performance compared to their binary counterparts but they suffer from higher complexity, especially for the check node processing. In this paper a sorting of the input vectors based on a reliability criteria is performed prior to the check node processing. This presorting process allows the Extended Min-Sum (EMS) check node process to focus its effort mainly on the weakest inputs. Proof is given for a check node of degree 12 in GF(64) for the syndrome based algorithm with a number of computed syndromes reduced by a factor of four which directly impacts the check node complexity without performance degradation.**

Index Terms—**NB-LDPC, Check Node, syndrome-based, EMS.**

Fig. 1. Message notation on a CN

I. INTRODUCTION

Low-Density Parity-Check (LDPC) codes where first proposed by R.G. Gallager in 1963 [1] and rediscovered by D. Mackay et al in 1996 [2]. Today many commercial standards (WiMAX, WiFi, DVB-C2, DVB-S2X, DVB-T2) make use of LDPC codes. Very long binary LDPC codes have been proven to perform close to the Shannon limit. However when considering short blocks of only some hundred bits of length for low latency applications, these codes suffer from performance degradation. The extension of binary LDPC codes to Galois Fields of order q (GF(q) with $q > 2$) is a promising approach to solve this problem. Moreover the symbols of high-order modulation schemes can be directly mapped to the decoder input symbols. Thus an additional gain in communication performance is observed for systems combining high-order modulation and Non-Binary Low-Density Parity-Check (NB-LDPC) codes [3]. This gain is obtained at the cost of a significant increase of the decoding complexity. The decoding can be performed by message passing algorithms like Belief Propagation (BP). However a straightforward implementation of the BP algorithm has a complexity of $\mathcal{O}(q^2)$ [4]. In recent years several approaches have been proposed to reduce the decoding complexity without sacrificing the communication performance.

An extension of the well-known binary Min-Sum algorithm to the non-binary domain, called the Extended Min-Sum (EMS) algorithm [5], [6] gives a good compromise between hardware complexity and communication performance. The largest complexity in the EMS algorithm is the computation of the Check Node (CN). Three categories of simplified EMS emerge frome state of the art. The Forward-Backward Check Node (FB-CN) algorithm [4] is efficiently implemented in [7] and [8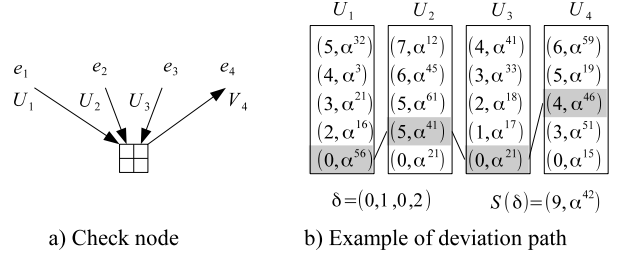] but can be limited in throughput due to its serial processing. The trellis based EMS [9] algorithm allows parrallel processing and efficient emplementation.

The Syndrome-Based Check Node (SB-CN) algorithm, recently presented in [10], [11], allows an efficient parallel computation of the CN function for high order Galois fields ($q \geq 16$). However, the complexity of the SB-CN algorithm is dominated by the number z of syndromes to compute. The value of z is in the order of $\mathcal{O}(d_c^2)$, which limits the interest of the SB-CN algorithm for high coding rates with high values of d_c.

In this paper we show that permuting the input messages of a CN based on reliability criteria significantly reduces the search space to explore during the check node processing. This reduction of search space can be exploited to simplify the hardware complexity of the SB-CN processor with practically no performance degradation.

The paper is structured as follows: Section II reviews the decoding of NB-LDPC codes making use of the EMS algorithm and its hardware implementation with the Syndrome Based CN algorithm. The presorting of input messages and its effect on FB-CN algorithm is presented in section III. Section IV presents the simulations results and section V gives the conclusions.

II. CHECK NODE PROCESSING

In this section we first review the original EMS decoding algorithm and the SB-CN decoding algorithm. For simplicity, the algorithms are described only at the CN level.

Let us define a CN equation of degree d_c in GF(q) as $e_1 \oplus e_2 \oplus e_3 \oplus \ldots \oplus e_{d_c} = 0$. Each input e_i can take q values. The a priori information about variable e is the discrete probability distribution $P(e = x)$, $x \in$ GF(q). Each element of the probability distribution E associated to e can be expressed in the log domain as the Log Likelihood Ratio (LLR) $e^+(x)$ defined as

$$e^+(x) = -\log\left(\frac{P(e=x)}{P(e=\bar{x})}\right) \quad (1)$$

where $\bar{x}$ is the hard decision on e obtained by taking the most probable GF symbol, i.e. $\bar{x} = \arg\max_{x \in \text{GF}(q)} P(e=x)$. By definition of the LLR, we have: $e^+(\bar{x}) = 0$ and $\forall x \in \text{GF}(q)$, $e^+(x) \geq 0$. The distribution (or message) E associated to e is thus $E = \{e^+(x)\}_{x \in \text{GF}(q)}$. In the sequel, operator $\oplus$ is used to represent addition in GF(q). The Min-Sum algorithm defines the LLR value of the ith output for the GF symbol x as

$$v_i^+(x) = \min\left\{ \sum_{i'=1, i \neq i'}^{d_c} e_{i'}^+(x_{i'}) \mid \bigoplus_{i'=1, i' \neq i}^{d_c} x_{i'} = x \right\}, \quad (2)$$

where $x_{i'} \in \text{GF}(q)$ for $i' = 1, 2, ..., d_c, i' \neq i$. The EMS algorithm is a simplification of the Min-Sum algorithm.

A. EMS algorithm

The main characteristic of the EMS algorithm is to shrink the q values of the message E to its n_m most reliable components, with $n_m << q$. The resulting message U is composed of n_m couples coming in ascending order of the LLR (from the most reliable to the less reliable). The input U of a CN is thus a list (or vector) $\{U[j]\}_{j=0...n_m-1}$ of couples, with $U[j] = (U^+[j], U^\oplus[j])$, where $U^+[j]$ designates the jth smallest LLR value of E and $U^\oplus[j]$ designates its associated GF element, i.e., $e^+(U^\oplus[j]) = U[j]^+$. Note also that $U^+[0] = 0$, $U^\oplus[0] = \bar{x}$, and that $j \leq j' \Rightarrow U^+[j] \leq U^+[j']$. The same process is also performed for each output V of a CN.

The EMS processing is performed in two steps. First, it evaluates (2) by replacing $x_{i'} \in \text{GF}(q)$ by $x_{i'}$ in the set of available GF data, i.e., $x_{i'} \in U_i^\oplus$, with $U_i^\oplus = \{U_i^\oplus[j]\}_{j=0,1,...,n_m-1}$. The obtained equation is thus

$$v_i^+(x) = \min\left\{ \sum_{i'=1, i' \neq i}^{d_c} U_{i'}^+[j_{i'}] \mid \bigoplus_{i'=1, i' \neq i}^{d_c} U_{i'}^\oplus[j_{i'}] = x \right\}, \quad (3)$$

where $j_{i'} \in \{0, 1, ... n_m - 1\}$ for $i' = 1, 2, ..., d_c, i' \neq i$. Second, the $v_i^+(x)$ are sorted in ascending order and the first n_m smallest values are kept to generate the output vector V_i. One should note that $V_i^+[0] = 0$ (in fact, $j_{i'} = 0 \Rightarrow U_{i'}^+[j_{i'}] = 0$ and thus, at least one term in (3) is the summation of zero values). Fig. 1.a illustrates the principle of a CN processing.

In the next section, we will derive the SB-CN algorithm [10] to process the EMS algorithm.

B. Syndrome-based CN processing

The SB-CN is based on the definition of a deviation path and its associated syndrome.

Definition of deviation path: A deviation path δ is defined as a d_c-tuple of integer values between 0 and $n_m - 1$, i.e. $\delta = (\delta(1), \delta(2), \ldots, \delta(d_c))$, with $\delta(i) \in \{0, 1, \ldots, n_m - 1\}$, $i = 1, 2, \ldots, d_c$.
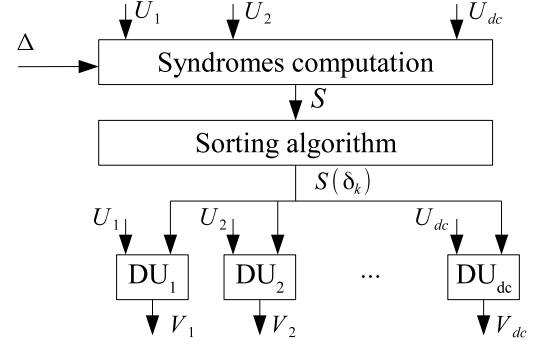

Fig. 2. Syndrome-based CN processing

Definition of syndrome: Assuming a CN with d_c input U_i, $i = 1, 2, ..., d_c$, it is possible to associate to a deviation path δ a syndrome $S(\delta)$ defined as the couple $(S^+(\delta), S^\oplus(\delta))$, with

$$S^+(\delta) = \sum_{i=1}^{d_c} U_i^+[\delta(i)], \; S^\oplus(\delta) = \bigoplus_{i=1}^{d_c} U_i^\oplus[\delta(i)]. \quad (4)$$

Fig. 1.b shows an example of a CN with $d_c = 4$ and messages of size $n_m = 5$. In this figure, the deviation path $\delta = (0, 1, 0, 2)$ is represented by a gray area in each input vector. It is also represented with straight lines linking $U_1[0]$, $U_2[1]$, $U_3[0]$ and $U_4[2]$. Assuming that the elements of GF(64) are represented by the power of the monomial α in GF(64)$[\alpha]/P[\alpha]$, with $P[\alpha] = \alpha^6 + \alpha + 1$, the syndrome associated to δ is $S(\delta) = (9, \alpha^{42})$.

Let Δ_a be the set of all possible deviation paths, i.e., $\Delta_a = \{0, \ldots, n_m - 1\}^{d_c}$. Using the syndrome associated to a deviation path, (3) can be reformulated as:

$$v_i^+(x) = \min_{\delta \in \Delta_a, S^\oplus(\delta) \oplus U_i^\oplus[\delta(i)] = x} \left\{ S^+(\delta) - U_i^+[\delta(i)] \right\}, \quad (5)$$

Lemma: Let δ and β be two deviation paths verifying $\delta(i') = \beta(i')$ for $i' \neq i$ and $\delta(i) = 0$, then

$$S^+(\beta) - U_i^+[\beta(i)] = S^+(\delta), \quad (6)$$

$$S^\oplus(\beta) - U_i^\oplus[\beta(i)] = S^\oplus(\delta) - U_i^\oplus[0]. \quad (7)$$

In others words, these two paths will lead exactly to the same $v^+(x)$ values. Thus, it is possible to further reduce (5) to

$$v_i^+(x) = \min_{\delta \in \Delta_a, \delta(i)=0, S^\oplus(\delta) \oplus U_i^\oplus[0] = x} \left\{ S^+(\delta) \right\}. \quad (8)$$

The SB-CN algorithm proposed in [10] is summarized in Algorithm 1 and the associated architecture is given in Fig. 2. Step 1, 2 and 3 of Algorithm 1 are represented by the three layers of processing in Fig. 2. In particular, decorrelation (step 3) is performed by the d_c Decorrelation Units (DU) represented in parallel to show the inherent parallelism of the SB decoder.

In [10], the deviation set $\Delta_{1,2}(d_1, d_2)$ is composed of the union of 3 sub-sets Δ_0, $\Delta_1(d_1)$ and $\Delta_2(d_2)$ where

<div style="border">

Pre-processing: Select $\Delta \subset \Delta_a$;
Initialisation:
for $i = 1$ *to* d_c **do**
$\quad | \quad j_i = 0;$
end
Processing:
step 1 (syndrome generation): $\forall \delta \in \Delta$, compute $S(\delta)$
step 2 (sorting process): Sort the syndrome in the ascending order of $S(\delta)^+$ to obtain an ordered list $\{S(\delta_k)\}_{k=1,2,\ldots,|\Delta|}$ of syndromes;
step 3 (decorrelation):
for $k = 1$ *to* $|\Delta|$ **do**
$\quad$ **for** $i = 1$ *to* d_c **do**
$\quad\quad$ **if** $\delta_k(i) = 0$ *and* $j_i < n_m$ **then**
$\quad\quad\quad V_i[j_i] = (S^+(\delta_k), S^{\oplus}(\delta_k) \oplus U_i^{\oplus}[0]);$
$\quad\quad\quad j_i = j_i + 1;$
$\quad\quad$ **end**
$\quad$ **end**
end

</div>

Algorithm 1: Syndrome based algorithm

<div style="border">

Input The d_c input message $\{U_i\}_{i=1,2,\ldots,d_c}$.
Step 1: Extract vector $U^1 = (U_1^+[1], U_2^+[1], \ldots, U_{d_c}^+[1])$
Sort U^1 in ascending order to generate U'^1.
return permutation $\pi = (\pi(1), \ldots, \pi(d_c))$ associated to the sorting process: $U'^1(i) = U^1(\pi(i))$, $i = 1, 2 \ldots d_c$.
Step 2: Permute input vectors using the permutation π:
for $i = 1, 2, \ldots, d_c$, $U_i' = U_{\pi(i)}$
Step 3: Perform the CN process with input vectors $\{U_i'\}_{i=1,2,\ldots,d_c}$ to generate output vectors $\{V_i'\}_{i=1,2,\ldots,d_c}$.
Step 4: Permute output vector using the inverse permutation π^{-1}:
for $i = 1, 2, \ldots, d_c$, $V_{\pi(i)} = V_i'$

</div>

Algorithm 2: Pre-sorting principle

Δ_0 contains only the deviation path δ_0 corresponding to the most reliable syndrome (all its coordinates are equal to 0), $\Delta_1(d_1)$ is the subset of deviation path deviating less than d_1 in exactly one position, and $\Delta_2(d_2)$ is the subset of deviation paths deviating less than d_2 in exactly two positions. For example, considering a CN of degree $d_c = 3$ with $d_1 = 2$ and $d_2 = 1$ then $\Delta_0 = \{(0,0,0)\}$, $\Delta_1(3) = \{(1,0,0),(2,0,0),(0,1,0),(0,2,0),(0,0,1),(0,0,2)\}$ and $\Delta_2(1) = \{(1,1,0),(1,0,1),(0,1,1)\}$. The cardinality $|\Delta_{1,2}(d_1,d_2)|$ of $\Delta_{1,2}(d_1,d_2)$ is

$$|\Delta_{1,2}(d_1,d_2)| = 1 + d_1 \binom{d_c}{1} + (d_2)^2 \binom{d_c}{2}. \qquad (9)$$

III. INPUT MESSAGE PRESORTING

The idea of the input presorting is to polarize the statistics of the input vector by sorting them according to the reliability of the hard decision input, i.e., the probability of $P(e_i = U_i^{\oplus}[0])$, $i = 1, 2, \ldots, d_c$. The main interest of this polarization process is that many deviation paths in the CN process become very unlikely to contribute to an output, and thus can be suppressed without affecting performance. The principle of polarization is first described. Then, its application to the SB algorithm is presented.

A. Presorting to polarize the input messages

Let us first give an approximation of $P(e_i = U_i^{\oplus}[0])$, $i = 1, 2, \ldots, d_c$. Using (1) and applying $\sum_{x \in GF} P(e_i = x) = 1$ leads to:

$$P(e_i = U_i^{\oplus}[0]) = \frac{e^{-U_i^+[0]}}{\sum_{j=0}^{q-1} e^{-U_i^+[j]}}. \qquad (10)$$

In this equation, the values of $U_i^+[j]$ for $j \geq n_m$ are equal to $U_i^+[n_m - 1] + O$, where O is a constant offset value (see [6] for a complete description of the EMS algorithm). Since
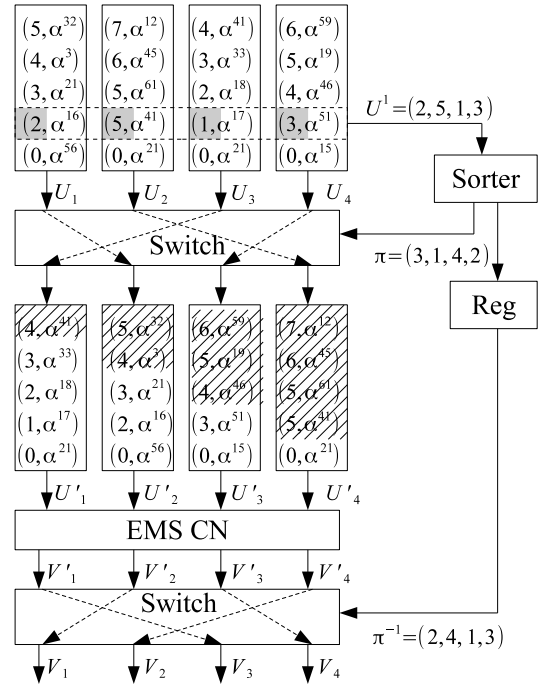

Fig. 3. Presorting principle

$U_i^+[0] = 0$ and, for $j > 2$, $U_i^+[1] \leq U_i^+[j]$, then $P(e_i = U_i^{\oplus}[0])$ can be approximated by

$$P(e_i = U_i^{\oplus}[0]) \approx \frac{1}{1 + e^{-U_i^+[1]}}. \qquad (11)$$

In other words, the higher the value of $U_i^+[1]$ is, the higher $P(e_i = U_i^{\oplus}[0])$. In this context, it is natural to sort the input vector according to the vector $U^1 = (U_1^+[1], U_2^+[1], \ldots, U_{d_c}^+[1])$ before entering the check node, as described in Algorithm 2 and illustrated in Fig. 3 where $n_m = 5$ and $d_c = 4$.

As shown in Fig. 3, compared to the standard CN, the presorting process requires extra hardware: a d_c-input vector sorter and two permutation networks (or switches). Nevertheless, it allows some simplification in the CN itself, globally leading to a complexity reduction of the whole CN processing.

The presorting can not be applied to T-EMS [9] because messages are not sorted nor truncated in the algorithm. However, the presorting can by efficiently applyed to EMS based algrithm [5], simplified EMS [12] and derived implementation derived from EMS as in FB-CN [7], [8] and SB-CN [10].

B. Presorting on syndrome-based decoder

In this section, the presorting principle is applied to the SB-CN algorithm [10]. In Fig. 3, some upper elements of the vectors U' are hashed because their associated LLR values are greater than 4 and are not contributing to the generation of the outputs. This is a particular example but, in the general case, this property remains statically true. In others word, a deviation path that implies entries from the hashed area can be pruned from the deviation set Δ. For example, if $\Delta_{1,2}(3,1)$ is used as the deviation-path set, its cardinality is reduced from 19 (see (9)) to 10, thanks to the pruning process.

Let Δ be the deviation path set associated to a SB-CN architecture. The complexity of the CN will be characterized by two parameters: the number z of deviation paths $z = |\Delta|$ and the number $\gamma(\Delta)$ of useful inputs, i.e, the number of entries reached at least by one deviation path of Δ. By definition, $\gamma(\Delta_a) = d_c \times n_m$. In the general case, the set Δ will be considered as consistent, i.e., $\forall \delta \in \Delta, \forall i \in \{1, 2, \ldots, d_c\}, \delta(i) = p > 0 \Rightarrow \exists \delta' \in \Delta \mid \delta'(i) = p - 1$. In that case, $\gamma(\Delta)$ is given by

$$\gamma(\Delta) = \sum_{i=1}^{d_c} \max_{\delta \in \Delta} \{\delta(i) + 1\}$$

The construction of the syndrome set cardinality is a trade-off between the complexity of the decoder (the smaller z and γ, the better it is) and the decoder performance (the lower z and γ, the higher the probability that a configuration path contributing to the output is missing, thus alleviating the decoder performance). Nevertheless, for a given number z, it is possible to derive the optimal deviation-path set $\Delta^o(z)$ as the set of the first z deviation paths having the highest probability to contribute to an output in a given context (the NB-LDPC code, Signal-to-Noise Ratios (SNR), number of decoding iterations). The formal derivation of $\Delta^o(z)$ is very complex, thus, we propose to use a Monte-Carlo simulation to explicitly construct $\Delta^o(z)$, as presented in Algorithm 3. T associates to each possible deviation path a counter. Note that in the pre-processing step, $|\Delta_a| = n_m^{d_c} = 2.2 \times 10^{15}$ for $n_m = 19$ and $d_c = 12$. Algorithm tricks (not described here) are used to reduce the size of the array T.

IV. SIMULATION RESULTS

In this section, we present some results of the presorting based algorithm for the SB-CN algorithm. Simulation results are based on a regular code $\mathcal{C}$ (the code can be downloaded in [13]) of coding rate 5/6, length $N = 576$ bits and $M = 480$ bits of information over GF(64). This regular code has $d_c = 12$ and a variable node degree of $d_v = 2$. The maximum number of decoding iterations is set to $I = 10$.

Input value of SNR, d_c, n_m, NB-LDPC code of size (N, M), number of decoding iterations I, $z = |\Delta(z)|$, number of codewords P.
Pre-processing Generate the set $|\Delta_a|$ from d_c and n_m. define function F as:
$F : \Delta \to \mathbb{N}, \delta \to F(\delta) = \sum_{i=1}^{d_c} \delta(i) n_m^{i-1}$.
$\forall \delta \in \Delta$, set $T(F(\delta)) = 0$
Monte-Carlo simulation
for $p = 1$ *to* P **do**
  Generate a noisy codeword C for the given code and SNR.
  Perform I decoding iterations on C
  Let $\delta(t, c, i, j)$ be the deviation path used to generate the $V_i[j]$ output of CN c ($c = 1, 2, \ldots, M$) during the tth decoding iteration ($t = 1, 2, \ldots, I$).
  $T(F(\delta(t, c, i, j))) \leftarrow T(F(\delta(t, c, i, j))) + 1;$
end
Post-Processing
Generate $\Delta^o(z)$ as the set of the deviation paths associated to the first z highest values of T.

Algorithm 3: Monte-Carlo construction of the optimal deviation-path set $\Delta^o(z)$ with z elements.

Performance is estimated in terms of Frame Error Rate (FERs) as a function of the Signal to Noise Ratio (SNR) in a Binary Phase Shift Keeing (BPSK) modulation under the Additive White Gaussian Noise (AWGN) channel.

The first step of the proposed method is to construct the optimal deviation path set $\Delta^o(z)$ as a function of z. To do so, $P = 10^6$ codewords have been simulated with/without presorting at a SNR of 4 dB. Since Algorithm 3 gives a score to each deviation path of the EMS decoder and selects the first z having the maximum score, one can derive the probability that a given deviation path δ found in the EMS decoder belongs to the set $\Delta^o(z)$. Fig. 4 shows this probability $P(\delta \in \Delta^o(z))$ as a function of z for the presorted and the original (i.e. unsorted) SB-CN decoder. As expected, presorting allows to significantly reduce the number of configurations required to obtain a given value of $P(\delta \in \Delta^o(z))$. For example, $z_0 = 47$ is required to obtained $P(\delta \in \Delta^o(z)) = 0.9$ for the presorted SB-CN algorithm while $z_1 = 257$ to obtain the same probability for the unsorted SB-CN algorithm. In this figure, the functions $\gamma(\Delta^o(z))$ are also presented for the presorted and the unsorted SB-CN decoder. Those two curves are more or less similar and we can note that $\gamma(\Delta^o(47)) = 24$ for the presorted SB-CN algorithm while $\gamma(\Delta^o(257)) = 75$ for the unsorted SB-CN algorithm. The impact of the value of z, for an SNR of 4 dB, is given in Fig. 5 for presorted and unsorted SB decoder. Optimal performance (i.e., EMS algorithm) is obtained for $z = 65$ in case of presorting, while $z = 271$ is required when no presorting is processed. This reduction of configurations by a factor of 4 translates directly in hardware complexity. Finally, Fig. 6 gives the performance of the decoder in several configurations: presorted SB-CN decoder with $z = 35, 65$, non-sorted SB-CN decoder with $z = 103, 271$.
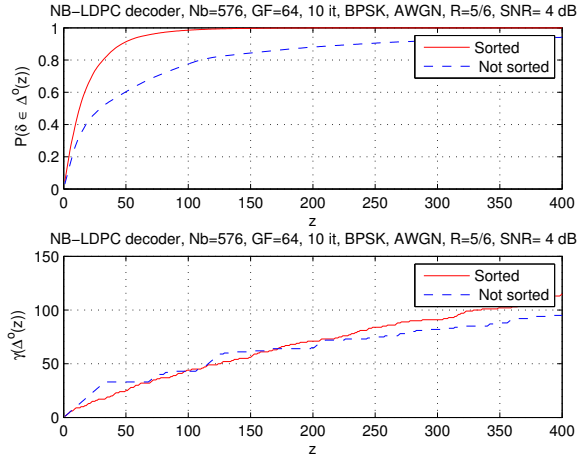
Fig. 4. $P(\delta \in \Delta^o(z))$ et $\gamma(\Delta^o(z))$ as a function of z for presorted and unsorted syndrome based decoder.
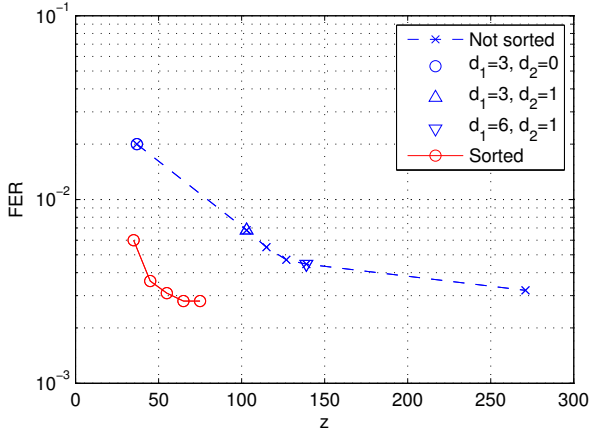

Fig. 5. FER as a function of z for presorted and unsorted syndrome based decoder for an SNR of 4 dB.
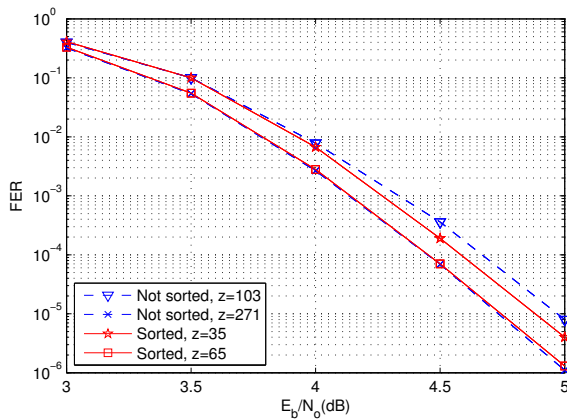

Fig. 6. Simulation results of NB-LDPC decoding algorithms for (576, 480) code over GF(64) under AWGN channel and BPSK modulation

V. Conclusion

In this paper, we have shown that input message presorting based on the reliability of the hard decision create a polarization of the space of configuration of inputs used to generate the CN outputs. In other words, some input configurations become very unlikely and the CN processor can just ignore them, leading to potential reduced hardware complexity. This algorithm simplification has been validated on a rate-5/6 NB-LDPC code with $d_c = 12$. It has been shown that the number of configurations can be reduced by a factor of 4, without performance degradation. In a near future, we will extend the presorting of the input vector to the Forward-Backward algorithm. We will also perform the hardware implementation of the SB-CN processor and extend those promising preliminary results to several coding rates (i.e. different d_c values) and to Galois Fields of higher order.

References

[1] R. Gallager, "Low-density parity-check codes," Ph.D. dissertation, Cambridge, 1963.

[2] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, no. 6, pp. 457–458, Mar 1997.

[3] S. Pfletschinger and D. Declercq, "Getting closer to mimo capacity with non-binary codes and spatial multiplexing," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, Dec 2010, pp. 1–5.

[4] M. C. Davey and D. J. C. MacKay, "Low density parity check codes over gf(q)," in *Information Theory Workshop, 1998*, Jun 1998, pp. 70–71.

[5] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary ldpc codes over gf (q)," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 633–643, April 2007.

[6] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity decoding for non-binary ldpc codes in high order fields," *IEEE Transactions on Communications*, vol. 58, no. 5, pp. 1365–1375, May 2010.

[7] O. Abassi, L. Conde-Canencia, A. Ghouwayel, and E. Boutillon, "A novel architecture for elementary check node processing in non-binary ldpc decoders," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. PP, no. 99, pp. 1–1, 2016.

[8] Y. S. Park, Y. Tao, and Z. Zhang, "A fully parallel nonbinary ldpc decoder with fine-grained dynamic clock gating," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 2, pp. 464–475, Feb 2015.

[9] E. Li, D. Declercq, and K. Gunnam, "Trellis-based extended min-sum algorithm for non-binary ldpc codes and its hardware structure," *IEEE Transactions on Communications*, vol. 61, no. 7, pp. 2600–2611, July 2013.

[10] P. Schläfer, N. Wehn, M. Alles, T. Lehnigk-Emden, and E. Boutillon, "Syndrome based check node processing of high order nb-ldpc decoders," in *Telecommunications (ICT), 2015 22nd International Conference on*, April 2015, pp. 156–162.

[11] P. Schlafer, V. Rybalkin, N. Wehn, M. Alles, T. Lehnigk-Emden, and E. Boutillon, "A new architecture for high throughput, low latency nb-ldpc check node processing," in *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015 IEEE 26th Annual International Symposium on*, Aug 2015, pp. 1392–1397.

[12] C. L. Wang, X. Chen, Z. Li, and S. Yang, "A simplified min-sum decoding algorithm for non-binary ldpc codes," *IEEE Transactions on Communications*, vol. 61, no. 1, pp. 24–32, January 2013.

[13] (2016) Web site on non-binary ldpc codes. [Online]. Available: http://www-labsticc.univ-ubs.fr/nb_ldpc/