Sign-Preserving Min-Sum Decoders

F Cochachin, Emmanuel Boutillon, D Declercq

Sign-Preserving Min-Sum Decoders

F. Cochachin*†, E. Boutillon* *Senior Member, IEEE* and D. Declercq† *Senior Member, IEEE*

*Lab-STICC, UMR 6285, Université Bretagne Sud, Centre de Recherche BP 92116, Lorient 56321, France

†ETIS UMR 8051, CY Cergy Paris Université, ENSEA, CNRS, Cergy, France

Abstract

This paper proposes a new finite precision iterative decoder for low-density parity-check (LDPC) codes. The proposed decoder, named Sign-Preserving Min-Sum (SP-MS), significantly improves the decoding performance compared to the classical Offset Min-Sum (OMS) decoder when messages are quantized on $q = 2$, 3, or 4 bits. The particularity of the SP-MS decoder is that messages cannot take the 0 value, and can fully benefit from the q bits of precision. The optimization of the SP-MS decoder is investigated in the asymptotic limit of the code length using density evolution (DE). Our study shows that 3-bit SP-MS decoders can achieve the same error-correcting performance as 5-bit OMS decoders, and 2-bit SP-MS decoders outperform 3-bit OMS decoders. The finite-length simulations confirm the conclusions of the DE analysis for several LDPC codes. Our SP-MS decoder shows a signal-to-noise ratio (SNR) gain up to 0.43 dB, with a memory/wire reduction of up to 40%, compared to the OMS decoder. Moreover, the SP-MS decoder converges faster and uses fewer iterations than the OMS decoder, with an improvement of up to 83.3% of the average decoding throughput. On an FPGA, the SP-MS decoder reduces resource utilization by up to 56% compared to the OMS decoder.

Index Terms

Error correction, Low-Density Parity-Check (LDPC) codes, Density Evolution, Sign-Preserving Min-Sum (SP-MS) decoders.

I. Introduction

Low-Density Parity-Check (LDPC) codes [1]–[3], first introduced by Gallager in 1963, are widely used in communication standards and storage applications [4]–[8] because they provide

an exceptional error correction capability. LDPC codes can be efficiently decoded by Message-Passing (MP) algorithms that use a Tanner graph [9] representation. One of the best MP algorithm is the Sum-Product algorithm also called Belief-Propagation (BP) algorithm [10]. The BP decoder has excellent decoding performance in the waterfall region but at the cost of high computational complexity. It is worth noting that large length irregular LDPC codes decoded by the BP decoder can approach the Shannon limit [11], [12].

The effect of quantization on the messages of MP decoders has been extensively examined over the past 20 years. In [13], the authors show that for a quantized BP decoder, at least 6 bits of precision should be used for the Binary Input Additive White Gaussian Noise (BI-AWGN) channel to obtain performance similar to infinite precision decoders. The Min-Sum (MS) and Offset-corrected Min-Sum (OMS) decoders [14], [15], derived from the BP decoder, reduce computational complexity at the cost of a small performance degradation in the waterfall region, especially when they are implemented in finite precision. For quantized MS and OMS decoders, the effect of clipping and quantization for the BI-AWGN channel are examined in [15], [16]. Thanks to their reduced complexity, the hardware implementation of the OMS decoder [17] and other variants of the MS decoder [18]–[20] show a good trade-off between complexity and decoding performance.

Recently, finite-alphabet LDPC decoders [21] with reduced precision (*e.g.*, from 4 bits to 3 bits), and good error correction performance have been proposed [22], [23]. Reducing the precision of messages is a natural strategy to reduce hardware complexity and increase the throughput of LDPC decoders. It is worth mentioning that there is a non-negligible performance loss when the number of precision bits becomes too small, which is the case of 2-bit MS-based decoders [24]–[26].

However, many LDPC decoders that use 1 bit of precision for messages are discussed in the literature. These 1-bit LDPC decoders are not quantized versions of either the BP decoder or the MS decoder; we can cite for example stochastic-based decoders [27], [28], bit-flipping-based decoders [29], [30], and binary message-passing decoders [31], [32]. The advantage of 1-bit LDPC decoders is their very low hardware complexity compared to higher precision decoders, but they usually require a much larger number of decoding iterations, and their performance degradation is significant.

In this paper, we propose a new finite precision iterative decoder for low complexity hardware implementation, named Sign-Preserving Min-Sum (SP-MS) decoder, which always preserves the

sign of messages. The SP-MS decoder forbids the 0 value in its message alphabet during the iterative decoding, meaning that a message cannot be erased. Also, unlike traditional quantized decoders, the proposed decoder uses all the possible values of the alphabet generated for a given precision and uses the sign of the incoming messages to increase the reliability of the outgoing messages at variable nodes. We investigate SP-MS decoders defined over small message alphabets constructed from $q = 2$, 3, or 4 bits of precision. We also quantize the Log-Likelihood Ratios (LLRs) on small alphabets constructed from $q_{ch} = 3$ or 4 bits of precision.

In [33], the Sign-Preserving Noise-Aided Min-Sum (SP-NA-MS) decoders were examined for the case of regular LDPC codes considering the same precision (3 and 4 bits) for both the messages and the LLRs. In this paper, we modify and extend the work of [33] from regular to irregular LDPC codes. Moreover, we study the performance of SP-MS decoders in which the messages are quantized with one bit less than the LLRs. We also propose an offset model that depends on the magnitude of unsaturated variable-to-check messages, and optimize the offset values with Density Evolution (DE). Finally, we propose a method to improve the performance of SP-MS decoders in the error floor region, when the precision of the messages is lower than the precision of the LLRs.

Our study of the SP-MS decoders with Density Evolution shows that $(q_{ch} = 4, q = 3)$-bit SP-MS decoders have the same convergence threshold as classical $(q_{ch} = 5, q = 5)$-bit OMS decoders. Also, we show that $(q_{ch} = 3, q = 2)$-bit SP-MS decoders have better error correction performance than classical $(q_{ch} = 3, q = 3)$-bit OMS decoders. These conclusions are corroborated by finite length Monte Carlo (MC) simulations, and we obtain signal-to-noise ratio (SNR) gains up to 0.43 dB compared to the MS/OMS decoders.

Comparing the hardware complexity of the SP-MS and the OMS decoders, we observe that the SP-MS decoder allows us to greatly reduce the complexity of the message update rules when the precision of the messages is reduced. When using the SP-MS, a memory/wire reduction of up to 40% is achieved. Also, the SP-MS uses fewer iterations (converges faster) with an improvement of up to 83.3% of the average decoding throughput. Comparing the synthesis results on a Xilinx FPGA of the SP-MS and the OMS decoders, we observe that the SP-MS decoder has a significant improvement in clock frequency by up to 30%, and reduces resource utilization by up to 56%. These results open new possibilities for massive parallel implementation of LDPC decoders.

The outline of the paper is as follows. Section II introduces the basic notions of quantized decoders and LDPC codes. Section III explains how to preserve the sign of the messages in

SP-MS decoders, and how to optimize the SP-MS decoders with DE. Section IV presents the results of the asymptotic analysis of regular and irregular LDPC codes. In section V, finite length performance of the SP-MS decoders and the MS/OMS decoders are presented. In section VI, we propose a method to mitigate the early appearance of the error floor of SP-MS decoders. In Section VII, we discussed the hardware complexity of the SP-MS decoders and we present the synthesis results on a Xilinx FPGA of the SP-MS decoders and the OMS decoders. In Section VIII, we compare the SP-MS decoder with other decoders proposed in the literature. Finally, Section IX concludes this paper.

II. Basic Notions of Classical Quantized Decoders and LDPC Codes

An LDPC code is a linear block code defined by a sparse parity-check matrix $H = [h_{mn}]$ composed of M rows by N columns, with $M < N$. The usual graphical representation of an LDPC code is a Tanner graph, which is a bipartite graph G composed of two types of nodes: the variable nodes (VNs) $v_n, n = 1, ..., N$ and the check nodes (CNs) $c_m, m = 1, ..., M$. A VN in the Tanner graph corresponds to a column of H, and a CN corresponds to a row of H. An edge connecting CN c_m to VN v_n exists if and only if $h_{mn} \neq 0$.

Let $\mathcal{V}(v)$ denote the set of neighbors of a VN v, and $\mathcal{V}(c)$ denote the set of neighbors of a CN c. The degree of a node is the number of its neighbors in G. A code is said to have a regular column-weight d_v if all VNs v have the same degree d_v. Similarly, if all CNs c have the same degree d_c, a code is said to have a regular row-weight d_c. In case of irregular LDPC codes, the nodes can have different connection degrees, defining an irregularity distribution, which is usually characterized by the two polynomials $\lambda(x) = \sum_{i=1}^{d_{v,max}} \lambda_i x^{i-1}$, and $\rho(x) = \sum_{j=1}^{d_{c,max}} \rho_j x^{j-1}$. The parameters λ_i (respectively ρ_j) indicate the fraction of edges connected to degree i VNs (respectively degree j CNs) [34]. For regular codes, the polynomials are reduced to monomials, $\lambda(x) = x^{d_v-1}$ and $\rho(x) = x^{d_c-1}$.

Let $\boldsymbol{x} = (x_1, ..., x_N) \in \{0, 1\}^N$ be a codeword that satisfies $H\boldsymbol{x}^T = \mathbf{0}$. In this paper, $\boldsymbol{x}$ is mapped by the Binary Phase-Shift Keying (BPSK) modulation and transmitted over the BI-AWGN channel with noise variance σ^2. The channel output $\boldsymbol{y} = (y_1, ..., y_N)$ is modeled by $y_n = (1 - 2x_n) + z_n$ for $n = 1, \ldots, N$, where z_n is a sequence of independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance σ^2. The decoder produces the vector $\hat{\boldsymbol{x}} = (\hat{x}_1, ..., \hat{x}_N) \in \{0, 1\}^N$ that is an estimation of $\boldsymbol{x}$. To check whether $\hat{\boldsymbol{x}}$ is a valid codeword, we verify that the syndrome vector is all-zero, *i.e.* $H\hat{\boldsymbol{x}}^T = \mathbf{0}$.

The LLRs that can be computed at the channel output are equal to:

$$LLR(y_n) = \log\left(\frac{\Pr(y_n \mid x_n = 0)}{\Pr(y_n \mid x_n = 1)}\right) = \frac{2y_n}{\sigma^2} \quad \forall n = 1, \ldots, N. \tag{1}$$

For quantized decoders, the LLRs have to be quantized and saturated. Let $\mathcal{A}_L$ denote the decoder input alphabet, defined as $\mathcal{A}_L = \{-N_{ch}, ..., -1, 0, +1, ..., +N_{ch}\}$, composed of $2N_{ch}+1$ states, with $N_{ch} = 2^{(q_{ch}-1)} - 1$ and where q_{ch} is the number of precision bits used to quantize the LLRs. Let the quantizer be defined by $\mathcal{Q} : \mathbb{R} \to \mathcal{A}_L$

$$\mathcal{Q}(a) = \mathcal{S}\left(\lfloor \alpha\, a + 0.5 \rfloor, N_{ch}\right), \tag{2}$$

where $\lfloor . \rfloor$ depicts the floor function and $\mathcal{S}(b, N_{ch})$ is the saturation function clipping the value of b in the interval $[-N_{ch}, N_{ch}]$, *i.e.* $\mathcal{S}(b, N_{ch}) = \min(\max(b, -N_{ch}), +N_{ch})$. The parameter α is called *channel gain factor* and is used to increase or decrease the amplitude of LLRs at the decoder input. The value of α can be seen as an extra parameter in the quantized decoder that can be analyzed and optimized for better performance. With these notations, we define the quantized version of the LLRs that initialize the MP decoder by the vector $\boldsymbol{I} = (I_1, ..., I_N) \in \mathcal{A}_L^N$, with

$$I_n = \mathcal{Q}(LLR(y_n)) \quad \forall n = 1, \ldots, N. \tag{3}$$

A MP decoder exchanges messages between VNs and CNs along the edges of the Tanner Graph. During each iteration, the VN update (VNU) and CN update (CNU) compute outgoing messages from all incoming messages.

For classical quantized decoders, the finite message alphabet $\mathcal{A}_M$ is defined as $\mathcal{A}_M = \{-N_q, ..., -1, 0, +1, ..., +N_q\}$, and consists of $2N_q + 1$ states, with $N_q = 2^{(q-1)} - 1$ and where q is the number of quantization bits for the messages. Typically, the message alphabet is equal to the decoder input alphabet, *i.e.* $q_{ch} = q$ and $\mathcal{A}_M = \mathcal{A}_L$. Let $\ell \in \mathbb{N}$ denote the iteration number. Let also $m_{v\to c}^{(\ell)} \in \mathcal{A}_M$ denote the variable-to-check message sent from VN v to CN c in the ℓ^{th} iteration, and $m_{c\to v}^{(\ell)} \in \mathcal{A}_M$ denote the check-to-variable message sent from CN c to VN v in the ℓ^{th} iteration.

Let us briefly recall the VNU and CNU equations for the quantized MS-based decoders, before introducing the SP-MS decoders. For this purpose, let $\Psi_v : \mathcal{A}_L \times \mathcal{A}_M^{(d_v-1)} \to \mathcal{A}_M$ denote the discrete function used for the update at a VN v of degree d_v, and let $\Psi_c : \mathcal{A}_M^{(d_c-1)} \to \mathcal{A}_M$ denote the discrete function used for the update at a CN c of degree d_c.

The update rule at a CNU is given by

$$m_{c_m \to v_n}^{(\ell)} = \Psi_c \left(\left\{ m_{v \to c_m}^{(\ell)} \right\}_{v \in \mathcal{V}(c_m) \setminus \{v_n\}} \right) = \left(\prod_{v \in \mathcal{V}(c_m) \setminus \{v_n\}} \text{sign}\left(m_{v \to c_m}^{(\ell)} \right) \right) \left(\min_{v \in \mathcal{V}(c_m) \setminus \{v_n\}} \left(\left| m_{v \to c_m}^{(\ell)} \right| \right) \right). \quad (4)$$

Let $m_{v_n \to c_m}^{(\ell+1),U}$ denote the unsaturated variable-to-check message in the $(\ell+1)^{th}$ iteration, defined as

$$m_{v_n \to c_m}^{(\ell+1),U} = I_n + \sum_{c \in \mathcal{V}(v_n) \setminus \{c_m\}} m_{c \to v_n}^{(\ell)}.$$

The alphabet of the unsaturated variable-to-check message $m_{v_n \to c_m}^{(\ell+1),U}$, denoted $\mathcal{A}_U$, is defined as $\mathcal{A}_U = \{-N_q\,(d_v - 1) - N_{ch}, ..., -1, 0, +1, ..., +N_q\,(d_v - 1) + N_{ch}\}$. Then, the update rule at a VNU is expressed as

$$m_{v_n \to c_m}^{(\ell+1)} = \Psi_v \left(I_n, \left\{ m_{c \to v_n}^{(\ell)} \right\}_{c \in \mathcal{V}(v_n) \setminus \{c_m\}} \right) = \Lambda \left(m_{v_n \to c_m}^{(\ell+1),U}, \varphi_v \right), \quad (5)$$

where the function $\Lambda(.)$ is defined by $\Lambda(a, \varphi_v) = \text{sign}(a)\,\mathcal{S}\left(\max(|a| - \varphi_v, 0), N_q\right)$.

The CNU (4) and VNU (5) define the classical OMS decoder with offset value $\varphi_v \in \{+1, ..., +(N_q - 2)\}$, applied at the VN[1], where the special case of $\varphi_v = 0$ corresponds to the classical MS decoder. The discrete functions Ψ_v and Ψ_c satisfy the symmetry conditions presented in [34].

Let $\boldsymbol{\gamma}^{(\boldsymbol{\ell})} = (\gamma_1^{(\ell)}, ..., \gamma_N^{(\ell)})$ denote the *a posteriori* probabilities (APP) in the ℓ^{th} iteration with alphabet $\mathcal{A}_{app} = \{-N_q\,d_v - N_{ch}, ..., -1, 0, +1, ..., +N_q\,d_v + N_{ch}\}$. The APP $\gamma_n^{(\ell)} \in \mathcal{A}_{app}$ is associated to a VN v_n, $n = 1, 2, \ldots, N$. Let $\Psi_a : \mathcal{A}_L \times \mathcal{A}_M^{(d_v)} \to \mathcal{A}_{app}$ denote the discrete function used for the APP computation at a VN v of degree d_v. The function Ψ_a satisfies the same conditions of symmetry as the function Ψ_v. With these notations, the APP computation at a VN v_n is given by

$$\gamma_n^{(\ell)} = \Psi_a \left(I_n, \left\{ m_{c \to v_n}^{(\ell)} \right\}_{c \in \mathcal{V}(v_n)} \right) = I_n + \sum_{c \in \mathcal{V}(v_n)} m_{c \to v_n}^{(\ell)}. \quad (6)$$

From the APP, $\hat{x}_n$ can be computed as $\hat{x}_n = (1 - \text{sign}(\gamma_n^{(\ell)}))/2$ if $\gamma_n^{(\ell)} \neq 0$, otherwise, $\hat{x}_n = (1 - \text{sign}(I_n))/2$, for $n = 1, ..., N$.

At the initialization stage of MS-based decoders ($\ell = 0$), variable-to-check messages are initialized using $m_{v_n \to c_m}^{(0)} = \mathcal{S}(I_n, N_q)$ where $c_m \in \mathcal{V}(v_n)$, for $n = 1, ..., N$; the saturation function $\mathcal{S}$ is required only when $q < q_{ch}$, *i.e.*, $N_q < N_{ch}$.

[1]Note that in the literature, φ_v is often applied at the CN. Applying the offset at the VN or at the CN is equivalent only when the saturation function is not used since $\min_{i=1,...,n}(|x_i| - \varphi_v) = \min_{i=1,...,n}(|x_i|) - \varphi_v$.

Analyzing (4) and (5), we can see that the message alphabet $\mathcal{A}_M$ of classical quantized decoders uses only $2^q - 1$ levels out of a total of 2^q levels achievable with q precision bits. For example, using $q = 3$ bits of precision, only 7 levels among 8 levels are used in $\mathcal{A}_M$.

III. Sign-Preserving Min-Sum Decoders

In the classical MS-based decoders, the value of the messages can be zero. In that case, the erased message, *i.e.* $m_{v_n \to c_m}^{(\ell+1)} = 0$, does not carry any information and does not participate in the convergence of the decoder. In this paper, we propose a new type of decoder, with a modified VNU using a *sign preserving factor*, so that the VNU never generates erased messages.

A. Quantization used for SP-MS Decoders

Using the sign-and-magnitude representation, one can obtain discrete alphabets that are symmetric and composed of $2^{q_{ch}}$ states. The decoder input alphabet for SP-MS decoders is defined as $\mathcal{B}_L = \{-N_{ch}, ..., -1, -0, +0, +1, ..., +N_{ch}\}$, with $N_{ch} = 2^{(q_{ch}-1)} - 1$. Similarly, the message alphabet for SP-MS decoders denoted by $\mathcal{B}_M$ is defined as $\mathcal{B}_M = \{-N_q, ..., -1, -0, +0, +1, ..., +N_q\}$, with $N_q = 2^{(q-1)} - 1$. The sign of a message $m \in \mathcal{B}_M$ indicates the estimated bit value while the magnitude $|m|$ represents its reliability. In this paper, it is assumed that $2 \leq q \leq q_{ch}$. The alphabets $\mathcal{B}_L$ and $\mathcal{B}_M$ can be easily implemented in hardware because each value of $\mathcal{B}_L$ and $\mathcal{B}_M$ has a natural (sign, magnitude) binary representation. An example of the binary representation of $\mathcal{A}_M$ and $\mathcal{B}_M$ for $q = 3$ is shown in Table I, showing that -0 is represented by 100_{2s}, $+0$ is represented by 000_{2s}, etc., where the index "2s" indicates the (sign, magnitude) format.

In order to obtain the quantized version of the LLRs belonging to $\mathcal{B}_L$, the quantization process defined in (2) is replaced by

$$\mathcal{Q}^*(a) = (\text{sign}(a), \mathcal{S}(\lfloor \alpha \ |a| \rfloor, N_{ch})), \tag{7}$$

The quantized LLRs $I_n = \mathcal{Q}^*(LLR(y_n)) \in \mathcal{B}_L$ for $n = 1, \ldots, N$ are used to initialize the decoder. In the initialization stage, *i.e.* at $\ell = 0$, the variable-to-check messages $m_{v_n \to c_m}^{(\ell)}$ are computed as $m_{v_n \to c_m}^{(0)} = \mathcal{S}(I_n, N_q)$ where $c_m \in \mathcal{V}(v_n)$, for $n = 1, ..., N$. Note that the operation of saturation $\mathcal{S}$ is required only when $q < q_{ch}$, *i.e.*, $N_q < N_{ch}$.

Table I: Binary representation of the quantized values.

Classical Decoder		Sign-Preserving Decoder		
$m \in \mathcal{A}_M$	$q = 3$ bits	$m \in \mathcal{B}_M$	$q = 3$ bits	(sign(m), $\lvert m \rvert$)
-3	101	-3	111	($-$1,3)
-2	110	-2	110	($-$1,2)
-1	111	-1	101	($-$1,1)
unused	100	-0	100	($-$1,0)
0	000	$+0$	000	(+1,0)
$+1$	001	$+1$	001	(+1,1)
$+2$	010	$+2$	010	(+1,2)
$+3$	011	$+3$	011	(+1,3)

B. Sign-Preserving Min-Sum Decoders

Let us now define the VNU and CNU update rules for SP-MS decoders. The discrete functions Ψ_v, Ψ_c, and Ψ_a of SP-MS decoders are defined in their modified alphabets, *i.e.* $\Psi_v : \mathcal{B}_L \times \mathcal{B}_M^{(d_v-1)} \rightarrow \mathcal{B}_M$, $\Psi_c : \mathcal{B}_M^{(d_c-1)} \rightarrow \mathcal{B}_M$, and $\Psi_a : \mathcal{B}_L \times \mathcal{B}_M^{(d_v)} \rightarrow \mathcal{B}_{app}$.

The update rule at the CNU of SP-MS decoders is given by

$$m_{c_m \rightarrow v_n}^{(\ell)} = \Psi_c \left(\left\{ m_{v \rightarrow c_m}^{(\ell)} \right\}_{v \in \mathcal{V}(c_m) \backslash \{v_n\}} \right) = \left(\prod_{v \in \mathcal{V}(c_m) \backslash \{v_n\}} \text{sign} \left(m_{v \rightarrow c_m}^{(\ell)} \right), \min_{v \in \mathcal{V}(c_m) \backslash \{v_n\}} \left(\left| m_{v \rightarrow c_m}^{(\ell)} \right| \right) \right). \quad (8)$$

The CNU computes outgoing messages that always belong to $\mathcal{B}_M$ based on all the incoming messages that also belong to $\mathcal{B}_M$. It is to be noted that (8) is identical to (4).

In the case of the VNU, the update rule of classical MS decoders (5) does not guarantee that the outgoing message belongs to $\mathcal{B}_M$. We therefore modify the VNU rule to ensure that the outgoing message will always belong to $\mathcal{B}_M$. Let us denote by $\mu_{v_n \rightarrow c_m}^{(\ell)}$ the *sign-preserving factor* for the variable-to-check message from VN v_n to CN c_m, defined as

$$\mu_{v_n \rightarrow c_m}^{(\ell)} = \xi \, \text{sign}(I_n) + \sum_{c \in \mathcal{V}(v_n) \backslash \{c_m\}} \text{sign} \left(m_{c \rightarrow v_n}^{(\ell)} \right), \quad (9)$$

where the value of ξ depends on the value of the column-weight d_v of a VN v_n:

$$\xi = \begin{cases} 0, & \text{if} \quad d_v = 2, \\ 1, & \text{if} \quad d_v > 2 \text{ and } d_v \text{ is odd}, \\ 2, & \text{if} \quad d_v > 2 \text{ and } d_v \text{ is even}. \end{cases} \quad (10)$$

$\mu_{v_n \rightarrow c_m}^{(\ell)}$ takes its values in $\{-1, +1\}$ for $d_v = 2$, in $\{-d_v, -d_v + 2, ..., -1, +1, ..., +d_v\}$ for d_v odd and $d_v > 2$, and in $\{-d_v - 1, -d_v + 1, \ldots, -1, +1, ..., +d_v + 1\}$ for d_v even and $d_v > 2$. Thus, $\mu_{v_n \rightarrow c_m}^{(\ell)}$ is always an odd number.

Let us now redefine the unsaturated variable-to-check message $m_{v_n\to c_m}^{(\ell+1),U}$ as

$$m_{v_n\to c_m}^{(\ell+1),U} = \frac{\mu_{v_n\to c_m}^{(\ell)}}{2} + I_n + \sum_{c\in\mathcal{V}(v_n)\setminus\{c_m\}} m_{c\to v_n}^{(\ell)}. \tag{11}$$

The alphabet of $m_{v_n\to c_m}^{(\ell+1),U}$ is given by $\mathcal{B}_U = \{-N_q\,(d_v-1) - N_{ch} - (d_v-1+\xi)/2, ..., -1.5, -0.5, +0.5, +1.5, ..., +N_q\,(d_v-1)+N_{ch}+(d_v-1+\xi)/2\}$. We note that $(\mu_{v_n\to c_m}^{(\ell)})/2$ can be written as $(\mu_{v_n\to c_m}^{(\ell)})/2 = \lfloor(\mu_{v_n\to c_m}^{(\ell)})/2\rfloor + 0.5$. The fractional part of $(\mu_{v_n\to c_m}^{(\ell)})/2$, which is 0.5, prevents $m_{v_n\to c_m}^{(\ell+1),U}$ from being zero and guarantees that a sign is always assigned to the variable-to-check message. The integer part of $(\mu_{v_n\to c_m}^{(\ell)})/2$ increases the reliability of the message $m_{v_n\to c_m}^{(\ell+1),U}$ and thus helps the decoder to converge faster.

Then, the update rule at a VNU of the SP-MS decoder with offset value φ_v is given by

$$m_{v_n\to c_m}^{(\ell+1)} = \Psi_v\left(I_n, \left\{m_{c\to v_n}^{(\ell)}\right\}_{c\in\mathcal{V}(v_n)\setminus\{c_m\}}\right) = \left(\text{sign}\left(m_{v_n\to c_m}^{(\ell+1),U}\right), \mathcal{S}\left(\max\left(\left\lfloor\left|m_{v_n\to c_m}^{(\ell+1),U}\right|\right\rfloor - \varphi_v, 0\right), N_q\right)\right). \tag{12}$$

The APP update at a VN v_n of the SP-MS decoder is defined as follows

$$\gamma_n^{(\ell)} = \Psi_a\left(I_n, \left\{m_{c\to v_n}^{(\ell)}\right\}_{c\in\mathcal{V}(v_n)}\right) = I_n + \frac{1}{2}\,\xi\,\text{sign}(I_n) + \sum_{c\in\mathcal{V}(v_n)}\left(m_{c\to v_n}^{(\ell)} + \frac{1}{2}\,\text{sign}\left(m_{c\to v_n}^{(\ell)}\right)\right). \tag{13}$$

The alphabet of APPs for SP-MS decoders is given by $\mathcal{B}_{app} = \{-N_q\,d_v - N_{ch} - (d_v+\xi)/2), ..., -1, 0, +1, ..., +N_q\,d_v + N_{ch} + (d_v+\xi)/2)\}$. From the APP, $\hat{x}_n$ can be computed as $\hat{x}_n = (1-\text{sign}(I_n))/2$ if $\gamma_n^{(\ell)} = 0$, otherwise, $\hat{x}_n = (1-\text{sign}(\gamma_n^{(\ell)}))/2$ for $n = 1, ..., N$.

C. Optimization of Sign-Preserving Min-Sum Decoders

In order to optimize Sign-Preserving Min-Sum decoders, we analyze the offset value φ_v. In (12), we can see that the offset value φ_v is applied to $m_{v_n\to c_m}^{(\ell+1),U}$, and the value of the offset is the same for any value of the message. We can further improve the performance by replacing the constant value φ_v by an offset whose value depends on $\left|m_{v_n\to c_m}^{(\ell+1),U}\right|$. To simplify the notations in this section, we use m_u to denote any $m_{v_n\to c_m}^{(\ell+1),U} \in \mathcal{B}_U$ and m to denote any $m_{v_n\to c_m}^{(\ell+1)} \in \mathcal{B}_M$.

To perform the optimization, we consider three offset values that are denoted by $\boldsymbol{\varphi} = (\varphi_s, \varphi_a, \varphi_0)$, where φ_s is applied to $|m_u| = N_q + 0.5$, φ_a is applied to $|m_u| \in \{2.5, ..., N_q - 0.5\}$, and φ_0 is applied to $|m_u| = 1.5$. It is not necessary to consider an offset value for the values $|m_u| > N_q+0.5$ because those values will be saturated to $|N_q|$. Let $\Upsilon : \boldsymbol{\varphi} = (\varphi_s, \varphi_a, \varphi_0)$ be an offset model used at VNs. The offset corrected message is obtained as:

$$m = (\text{sign}(m_u), \mathcal{S}\left(\max\left(\lfloor|m_u|\rfloor - b, 0\right), N_q\right)), \tag{14}$$

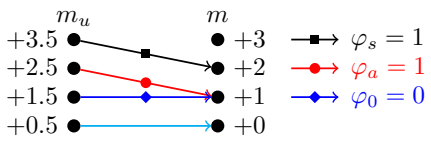


Figure 1: The offset model Υ for $(\varphi_s = 1, \varphi_a = 1, \varphi_0 = 0)$.

where the offset $b \in \{\varphi_s, \varphi_a, \varphi_0\}$ depends on the magnitude of m_u.

As an example, $\Upsilon : \boldsymbol{\varphi} = (\varphi_s = 1, \varphi_a = 1, \varphi_0 = 0)$ is depicted in Figure 1 for $(q = 3, N_q = 3)$.

Let us discuss the case of $\varphi_s = \varphi_a = \varphi_0$. The SP-MS decoder with offset $\varphi_v = 1$ can be obtained as a special case of Υ setting $\boldsymbol{\varphi} = (1, 1, 1)$; similarly, the SP-MS without offset, *i.e.* $\varphi_v = 0$, can be obtained with the setting $\boldsymbol{\varphi} = (0, 0, 0)$.

The offset values φ_s and φ_0 are applied to the extreme values of the message alphabet $\mathcal{B}_M$. Therefore, φ_s and φ_0 have to be analyzed differently from φ_a.

In the case of very low message precision $q = 2$, Υ is defined only by one offset value $\boldsymbol{\varphi} = (\varphi_s)$. For this special case, the message alphabet, which is only composed of four different values, is given by $\mathcal{B}_M = \{-1, -0, +0, +1\}$, with $N_q = 1$.

D. Density Evolution for Sign-Preserving Min-Sum Decoders

The goal of DE [34]–[36] is to recursively compute the probability mass function (PMF) of the messages in the Tanner graph along the iterations. DE enables us to predict whether an ensemble of LDPC codes, parametrized by its degree distribution, decoded with a given MP decoder, converges to zero error probability in the limit of infinite block length.

For the BI-AWGN channel, the DE threshold δ is the maximum value of the standard deviation σ or the minimum SNR for which the decoder converges to a zero error probability. As in [33], [37], δ can be expressed as $\delta_{db} = 10 \log_{10} \left(\frac{1}{2 R \sigma^{*2}} \right)$, where $\sigma^* = \delta$ and R is the rate of the code. In this paper, the details of the DE equations are not presented, and we refer to [38] for a complete presentation. The DE threshold δ depends on the degree distribution $(\lambda(x), \rho(x))$ of the code, but also on the quantized decoder parameters: the number of precision bits (q_{ch}, q), the channel gain factor α, and the offset values $(\varphi_s, \varphi_a, \varphi_0)$. We use DE to jointly optimize the offset values $(\varphi_s, \varphi_a, \varphi_0)$ and the channel gain factor α for a fixed precision (q_{ch}, q) and a fixed degree distribution $(\lambda(x), \rho(x))$ as follows

$$(\varphi_s^*, \varphi_a^*, \varphi_0^*, \alpha^*) = \arg \min_{(\varphi_s, \varphi_a, \varphi_0, \alpha)} \left\{ \delta_{db} \left(\lambda(x), \rho(x), q_{ch}, q, \alpha, \varphi_s, \varphi_a, \varphi_0 \right) \right\}. \tag{15}$$

The optimization of the offset values of the model Υ and the channel gain factor α is made using a greedy algorithm that computes a local maximum of the DE threshold. For MS and OMS decoders, the optimization (15) is reduced to the optimum channel gain factor α^* that is computed by performing a grid-search.

IV. Asymptotic Analysis of Sign-Preserving Min-Sum Decoders

In this section, we present the asymptotic analysis of SP-MS decoders for regular and irregular LDPC codes with the values of the sign preserving factor ξ defined in (10).

A. Regular LDPC codes

For all results presented in this section, we consider four ensembles of (d_v, d_c)-regular LDPC codes with the following parameters: *(i)* $(d_v = 3, d_c = 6)$ and code rate $R = 0.5$, *(ii)* $(d_v = 4, d_c = 64)$ and $R = 0.94$, *(iii)* $(d_v = 5, d_c = 20)$ and $R = 0.75$, and *(iv)* $(d_v = 6, d_c = 32)$ and $R = 0.8413$. We also consider quantized decoders with precision $(q_{ch}, q) \in \{(5,5), (4,4), (4,3), (3,3), (3,2)\}$.

DE thresholds of the classical MS and OMS decoders are given in Table II, which shows that the OMS is almost always superior to the MS for the cases involved, except when $q_{ch} = q = 3$ for regular $(d_v = 3, d_c = 6)$ LDPC codes and regular $(d_v = 4, d_c = 64)$ LDPC codes.

Table II: DE thresholds of classical MS ($\varphi_v = 0$) and OMS ($\varphi_v = 1$) decoders.

		$(d_v=3, d_c=6)$		$(d_v=4, d_c=64)$		$(d_v=5, d_c=20)$		$(d_v=6, d_c=32)$	
(q_{ch}, q)	φ_v	α^*	δ_{db}	α^*	δ_{db}	α^*	δ_{db}	α^*	δ_{db}
$(3,3)$	0	0.9375	**1.789**	0.50	**4.760**	0.56	3.645	0.455	4.081
	1	1.0625	2.204	0.69	4.904	0.92	**3.231**	0.84	**3.593**
$(4,4)$	0	2.0	1.644	1.06	4.572	1.40	3.392	1.035	3.815
	1	1.875	**1.348**	1.15	**4.421**	1.39	**2.762**	1.28	**3.169**
$(5,5)$	0	4.0	1.613	2.20	4.534	2.47	3.337	1.985	3.751
	1	2.625	**1.215**	1.73	**4.338**	1.61	**2.724**	1.45	**3.140**

In Table III, we indicate the optimal channel gain factor α^*, the optimal offset values $\boldsymbol{\varphi}^* = (\varphi_s^*, \varphi_a^*, \varphi_0^*)$, and the DE thresholds δ_{db} obtained with (15) for optimized SP-MS decoders. We also show the DE gain/loss compared to the best DE thresholds of MS/OMS decoders indicated in bold in Table II, for the same precision of the LLRs (q_{ch}). We report in Table IV the DE losses of the $(4,3)$-bit SP-MS decoder with respect to the $(5,5)$-bit OMS decoder. Several conclusions can be derived from this analysis.

Table III: SP-MS decoders for the (d_v, d_c)-regular LDPC codes.

(d_v, d_c)	(q_{ch}, q)	α^*	$(\varphi_s^*, \varphi_a^*, \varphi_0^*)$	δ_{db}	DE (dB) gain/loss	(d_v, d_c)	(q_{ch}, q)	α^*	$(\varphi_s^*, \varphi_a^*, \varphi_0^*)$	δ_{db}	DE (dB) gain/loss
$(3, 6)$	$(3, 3)$	0.95	(1, 1, 0)	**1.510**	0.279	$(5, 20)$	$(3, 3)$	0.89	(1, 1, 1)	**3.014**	0.217
	$(3, 2)$	0.48	$(0, -, -)$	**1.932**	$-0.143^{\perp}$		$(3, 2)$	0.88	$(1, -, -)$	**3.022**	$0.209^{\perp}$
	$(4, 4)$	1.79	(1, 1, 0)	**1.269**	0.079		$(4, 4)$	1.39	(1, 1, 1)	**2.741**	0.021
	$(4, 3)$	1.16		**1.391**	$-0.043^{\perp}$		$(4, 3)$	1.42		**2.738**	$0.024^{\perp}$
$(4, 64)$	$(3, 3)$	0.57	(1, 1, 1)	**4.612**	0.148	$(6, 32)$	$(3, 3)$	0.74	(1, 1, 1)	**3.396**	0.197
	$(3, 2)$	0.57	$(0, -, -)$	**4.624**	$0.136^{\perp}$		$(3, 2)$	0.74	$(1, -, -)$	**3.398**	$0.195^{\perp}$
	$(4, 4)$	1.04	(1, 1, 1)	**4.379**	0.042		$(4, 4)$	1.18	(1, 1, 1)	**3.179**	-0.010
	$(4, 3)$	1.05		**4.379**	$0.042^{\perp}$		$(4, 3)$	1.22		**3.174**	$-0.005^{\perp}$

$^{\perp}$ DE gains are obtained by comparing the $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS and the $(q_{ch}, q = q_{ch})$-bit MS/OMS.

Table IV: DE losses obtained by comparing the $(4, 3)$-bit SP-MS and the $(5, 5)$-bit OMS.

(d_v, d_c)	(q_{ch}, q)	α^*	$(\varphi_s^*, \varphi_a^*, \varphi_0^*)$	δ_{db}	DE loss (dB)
$(3, 6)$	$(4, 3)$	$1.16^{\dagger}$	(1, 1, 0)	**1.391**	-0.176
$(4, 64)$	$(4, 3)$	$1.05^{\dagger}$	(1, 1, 1)	**4.379**	-0.041
$(5, 20)$	$(4, 3)$	1.42	(1, 1, 1)	**2.738**	-0.014
$(6, 32)$	$(4, 3)$	1.22	(1, 1, 1)	**3.174**	-0.034

† The channel gain factor α is further optimized in section VI.

1) First, the DE thresholds of SP-MS decoders are almost always better than the DE thresholds of classical decoders when the same precision for the messages and the LLRs is used $(q_{ch} = q)$. An exception appears for the regular $(d_v = 6, d_c = 32)$ LDPC code, we observe a degradation of around 0.01 dB.
2) Second, the DE thresholds of $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS decoders are almost equal or equal to the DE thresholds of $(q_{ch}, q = q_{ch})$-bit SP-MS decoders for $d_v \in \{4, 5, 6\}$. In the case of $d_v = 3$, the DE threshold of the $(3, 2)$-bit SP-MS decoder is significantly worse than the DE threshold of the $(3, 3)$-bit SP-MS decoder.

   We can observe that the performance of the $(4, 3)$-bit SP-MS decoder can come close to the performance of the $(5, 5)$-bit OMS decoder, within 0.05 dB for $d_v \in \{4, 5, 6\}$. We also observe that the $(3, 2)$-bit SP-MS decoders show a better performance than the $(3, 3)$-bit MS and $(3, 3)$-bit OMS decoders; the only exception appears for $d_v = 3$.

   From these observations, we can conclude that the SP-MS decoders can be implemented using lower precision for the messages than for the LLRs, with $q = q_{ch} - 1$, with a negligible impact in the decoding performance.
3) Third, the DE gains of SP-MS decoders are significant for low precision $(q_{ch} = 3, q = 3)$ and very low precision $(q_{ch} = 3, q = 2)$, while the DE gains are smaller for the largest

precision $(q_{ch} = 4, q = 4)$ and $(q_{ch} = 4, q = 3)$. The largest gain obtained is around 0.279 dB for the regular $(d_v = 3, d_c = 6)$ LDPC code and precision $(q_{ch} = 3, q = 3)$.

We thus conclude that the SP-MS decoders are more effective when the decoders are implemented in low precision.

4) A final remark comes from the interpretation of the optimum $\boldsymbol{\varphi}^*$ obtained through the DE analysis. For the case of precision $q \geq 3$, we have $\boldsymbol{\varphi}^* = (\varphi_s^*, \varphi_a^*, \varphi_0^*) = (1, 1, 0)$ for regular $d_v = 3$ LDPC code. The value $\varphi_0^* = 0$ means that the offset is not applied to the message $m_u = \pm 1.5$, hence $m_u = \pm 1.5$ is mapped to $m = \pm 1$.

For SP-MS decoders, when using the precision $q \geq 3$ and regular $d_v > 3$ LDPC codes, we always have $\boldsymbol{\varphi}^* = (\varphi_s^*, \varphi_a^*, \varphi_0^*) = (1, 1, 1)$ which corresponds to the SP-MS decoder with offset $\varphi_v = 1$. In the case of very low precision $q = 2$ we obtain $\boldsymbol{\varphi}^* = (\varphi_s^*) = 1$ for regular $d_v = 5$ and $d_v = 6$ LDPC codes, and $\boldsymbol{\varphi}^* = (\varphi_s^*) = 0$ for $d_v = 3$ and $d_v = 4$.

These results indicate that the optimal offset correction depends on both the amplitudes of the messages and the VN degree.

B. Irregular LDPC codes

In the case of regular LDPC codes, we have observed that the optimum offset values $(\varphi_s^*, \varphi_a^*, \varphi_0^*)$ depend on the VN degree. Therefore, in order to optimize the SP-MS decoders for the irregular LDPC codes, we extend our optimization approach by considering an offset model Υ with different offset values for the different VN degrees. The precision considered in this section is $q_{ch} = q \in \{3, 4\}$.

Let $\Upsilon^{(2)} : \boldsymbol{\varphi}^{(2)} = \left(\varphi_s^{(2)}, \varphi_a^{(2)}, \varphi_0^{(2)}\right)$ denote the offset model for the VNs of degree $d_v = 2$, and let $\Upsilon^{(3)} : \boldsymbol{\varphi}^{(3)} = \left(\varphi_s^{(3)}, \varphi_a^{(3)}, \varphi_0^{(3)}\right)$ denote the offset model for the VNs of degree $d_v = 3$. Finally, we decide to use the same model for all other VNs with degrees $d_v \geq 4$, denoted $\Upsilon^{(\geq 4)} : \boldsymbol{\varphi}^{(\geq 4)} = \left(\varphi_s^{(\geq 4)}, \varphi_a^{(\geq 4)}, \varphi_0^{(\geq 4)}\right)$.

The optimization of the offset values for an irregular LDPC code with distribution $(\lambda(x), \rho(x))$ is performed by the maximization of the DE thresholds:

$$\left(\boldsymbol{\varphi}^{(2)*}, \boldsymbol{\varphi}^{(3)*}, \boldsymbol{\varphi}^{(\geq 4)*}, \alpha^*\right) = \arg \min_{(\boldsymbol{\varphi}^{(2)}, \boldsymbol{\varphi}^{(3)}, \boldsymbol{\varphi}^{(\geq 4)}, \alpha)} \left\{\delta_{db}\right\}. \tag{16}$$

For our analysis, we consider the ensemble of irregular LDPC codes that follows the distribution of the $R = 1/2$ LDPC code described in the WIMAX standard [6]. The degree distribution of

Table V: DE thresholds of MS and OMS decoders for the WIMAX degree distribution

R	(q_{ch}, q)	φ_v	α^*	δ_{db}
$1/2$	$(3,3)$	0	0.44	**1.8310**
		1	0.40	5.2283
	$(4,4)$	0	1.07	**1.3941**
		1	0.80	2.8140
	$(5,5)$	0	2.30	1.3013
		1	1.55	**1.1828**

the WIMAX code is $\lambda(x) = \frac{22}{76}x + \frac{24}{76}x^2 + \frac{30}{76}x^5$ and $\rho(x) = \frac{48}{76}x^5 + \frac{28}{76}x^6$. For this distribution, we indicate in Table V the DE thresholds of the MS and OMS decoders.

The DE thresholds of SP-MS decoders are summarized in Table VI, where we indicate the optimum channel gain factor α^* and the optimum offset values $\left(\boldsymbol{\varphi}^{(2)*}, \boldsymbol{\varphi}^{(3)*}, \boldsymbol{\varphi}^{(\geq 4)*}\right)$. These results confirm the conclusions of the regular LDPC codes analysis: *(i)* the DE thresholds of SP-MS decoders are better than the DE thresholds of MS and OMS decoders, *(ii)* the optimum value of φ_0^* is 0 for $d_v = 3$ VNs and for precision $q \in \{3, 4\}$, and *(iii)* the optimum value of $\boldsymbol{\varphi}^{(\geq 4)*}$ is $\boldsymbol{\varphi}^{(\geq 4)*} = (1, 1, 1)$ for precision $q \in \{3, 4\}$.

Table VI: DE thresholds of SP-MS decoders for the WIMAX degree distribution.

R	(q_{ch}, q)	α^*	d_v	$(\varphi_s^*, \varphi_a^*, \varphi_0^*)$	$\tilde{\delta}_{db}$	DE gain (dB)	SNR gain (dB) @ FER = 10^{-2}
$1/2$	$(3,3)$	0.65	2	(0,0,0)	**1.4003**	**0.4307**	**0.40**
			3	(0,0,0)			
			≥ 4	(1,1,1)			
	$(4,4)$	1.24	2	(0,0,0)	**0.9582**	**0.4359**	**0.40**
			3	(0,1,0)			
			≥ 4	(1,1,1)			

Another conclusion can be derived from this table. The DE analysis shows that the offset should not be applied on degree $d_v = 2$ VNs, since we always obtain $(\varphi_s^{(2)}, \varphi_a^{(2)}, \varphi_0^{(2)}) = (0, 0, 0)$. This observation, combined with the fact that the optimum values of $\boldsymbol{\varphi}^{(\geq 4)*}$ are always 1, leads to the conclusion that the offset in SP-MS decoders must be chosen carefully for irregular LDPC codes.

Finally, the gains of SP-MS decoders for irregular codes are larger than for the regular codes with a gain of 0.4307 dB for lower precision $q = 3$ and a gain of 0.4359 dB for the largest precision $q = 4$.

V. Finite Length Performance of Sign-Preserving Min-Sum Decoders

In this section, we present the frame error rate (FER) performance of classical MS, classical OMS, and SP-MS decoders over the BI-AWGN channel.

A. Performance of Regular LDPC codes

To corroborate the asymptotic results obtained by DE, we present the Monte Carlo simulations for *(i)* the $(d_v = 4, d_c = 64)$-regular QC-LDPC code with $N = 8960$, rate $R = 0.94$ and circulant size $L = 140$ for Flash Memory [8], and *(ii)* the $(d_v = 6, d_c = 32)$-regular LDPC codes with $R = 0.8413$ for the IEEE 802.3 ETHERNET code [5]. In addition, the PEG algorithm from [39] is used to design the following regular QC-LDPC codes: *(iii)* a $(d_v = 3, d_c = 6)$-regular QC-LDPC code with length $N = 1296$, rate $R = 1/2$ and circulant size $L = 54$, and *(iv)* a $(d_v = 5, d_c = 20)$-regular QC-LDPC code with length $N = 10240$, rate $R = 3/4$, and circulant size $L = 512$. For all codes, the Belief Propagation decoder performance is shown as a benchmark[2]. A maximum of 100 iterations have been set for $d_v = 3$ LDPC codes, while for the case of $d_v \in \{4, 5, 6\}$ LDPC codes, a maximum number of 30 iterations has been used. For each simulated SNR, the FER is estimated with at least 100 frame errors.

Figure 2 shows the FER performance comparison between the classical MS, classical OMS, and SP-MS decoders, for three precisions of messages $q \in \{2, 3, 4\}$, and for the regular $(d_v = 3, d_c = 6)$ QC-LDPC code. Figure 3 draws the same curves for the regular $(d_v = 4, d_c = 64)$ QC-LDPC code. The results show that for low precision messages $q = 3$ and for regular $d_v = 3$ LDPC codes, the MS decoder is better than the OMS decoder. This result is not surprising since the DE thresholds in Table II are better for MS than for OMS with $q = 3$ bits of precision. The same observation holds for the regular $(d_v = 4, d_c = 64)$ QC-LDPC code and precision $q = 3$. The gains/losses of the SP-MS decoders compared with the MS/OMS measured at FER $= 10^{-2}$ are reported in Table VII, which shows that the Monte Carlo simulations are congruent with the values predicted by the DE thresholds.

The FER performance curves plotted in Figure 2 and Figure 3 show that the $(4, 3)$-bit SP-MS decoders and the $(3, 2)$-bit SP-MS decoders exhibit poor performance due to an early error floor. A method to mitigate the early appearance of the error floor is proposed in section VI.

[2]The simulation results for the BP decoder were obtained with the open-source simulator AFF3CT: A Fast Forward Error Correction Toolbox, 2020. [Online]. Available: https://aff3ct.github.io/index.html

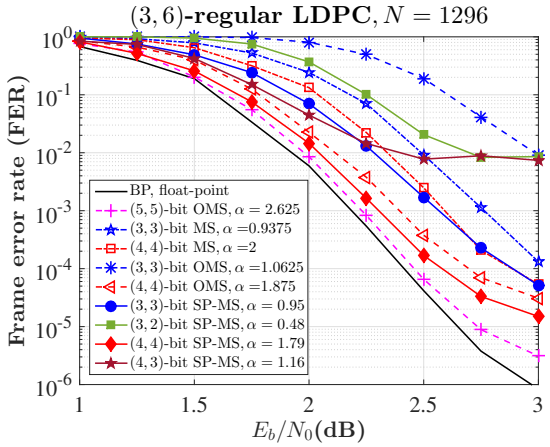


Figure 2: FER performance for $(3,6)$-regular QC-LDPC code.

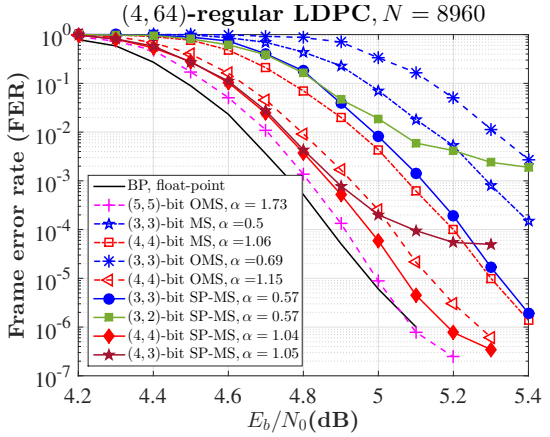


Figure 3: FER performance for $(4,64)$-regular QC-LDPC code.

Table VII: DE gains and SNR gains of SP-MS decoders for the (d_v, d_c)-regular LDPC codes.

(d_v,d_c)	(q_{ch},q)	α^*	DE gain/loss (dB)	SNR gain/loss (dB) @ FER = 10^{-2}	(d_v,d_c)	(q_{ch},q)	α^*	DE gain/loss (dB)	SNR gain/loss (dB) @ FER = 10^{-2}
$(3,6)$	$(3,3)$	0.95	0.279	0.26	$(5,20)$	$(3,3)$	0.89	0.217	0.22
	$(3,2)$	0.48†	$-0.143^{\perp}$	$-0.13^{\perp}$		$(3,2)$	0.88	0.209$^{\perp}$	0.20$^{\perp}$
	$(4,4)$	1.79	0.079	0.06		$(4,4)$	1.39	0.021	0.02
			$-0.043^{\perp}$	$0.01^{\perp}$				0.024$^{\perp}$	0.02$^{\perp}$
	$(4,3)$	1.16†	-0.176^{*}	-0.14^{*}		$(4,3)$	1.42	-0.014^{*}	-0.015^{*}
$(4,64)$	$(3,3)$	0.57	0.148	0.16	$(6,32)$	$(3,3)$	0.74	0.197	0.20
	$(3,2)$	0.57†	$0.136^{\perp}$	$0.13^{\perp}$		$(3,2)$	0.74	0.195$^{\perp}$	0.20$^{\perp}$
	$(4,4)$	1.04	0.042	0.05		$(4,4)$	1.18	-0.010	0.0
			$0.042^{\perp}$	$0.03^{\perp}$				$-0.005^{\perp}$	$0.0^{\perp}$
	$(4,3)$	1.05†	-0.041^{*}	-0.07^{*}		$(4,3)$	1.22	-0.034^{*}	-0.028^{*}

† The channel gain factor α is further optimized with Monte Carlo simulations (see section VI).

$^{\perp}$ SNR gains are obtained by comparing the $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS and the $(q_{ch}, q = q_{ch})$-bit MS/OMS.

* Results obtained when comparing the $(4,3)$-bit SP-MS decoder with the $(5,5)$-bit OMS decoder.

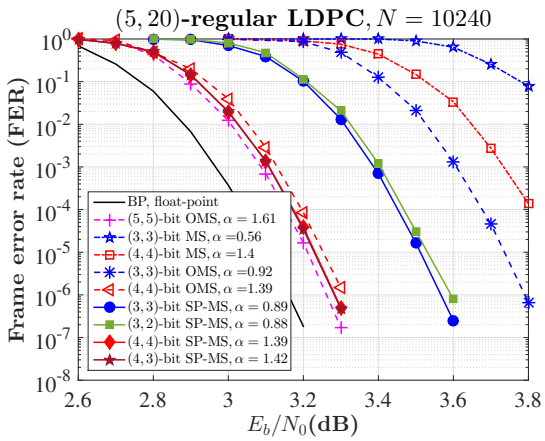


Figure 4: FER performance for $(5, 20)$-regular QC-LDPC code.

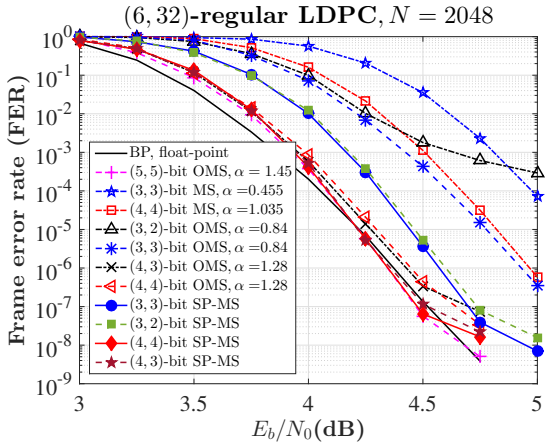


Figure 5: FER performance for the ETHERNET code.

Simulation results for the case of the $d_v = 5$ QC-LDPC code are shown in Figure 4, and in Figure 5 for the IEEE 802.3 ETHERNET code. As for the $d_v = 3$ and $d_v = 4$ LDPC codes the gains predicted by the DE correspond to the SNR gains at FER $= 10^{-2}$. The gains/losses are reported in Table VII.

From Figure 5, we can see that the $(4, 3)$-bit OMS decoder has the same performance as the $(4, 4)$-bit OMS decoder. We also observe that the $(3, 2)$-bit OMS decoder exhibits a very high error floor. In addition, the $(3, 2)$-bit SP-MS decoder mitigates the early appearance of the error floor and it greatly outperforms the $(3, 2)$-bit OMS decoder and the $(3, 3)$-bit OMS decoder, thus the SP-MS decoder is an excellent candidate for a hardware implementation.

B. Convergence Performance Analysis

FER convergence results are presented in Figure 6 for the IEEE 802.3 ETHERNET code. This figure shows that the FER curves for the SP-MS decoders decrease faster than for the MS and the OMS decoders. This figure also shows that the BP decoder converges faster than all quantized decoders during the first 6 iterations; after 7 iterations, the FER curve of the BP decoder decreases slowly. It is to be noted that from iteration 7 to iteration 20, the $(q_{ch} = 4, q)$-bit SP-MS decoders show better performance than the BP decoder.

We note that the FER convergence performance of the $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS and the $(q_{ch}, q = q_{ch})$-bit SP-MS are almost equal or equal. In addition, after 10 iterations of decoding, only marginal FER improvement is obtained for the $(q_{ch} = 4, q)$-bit SP-MS. The maximum number of decoding iterations can thus be set to 10 without significant performance degradation. Similarly, the maximum number of iterations for the $(q_{ch} = 3, q)$-bit SP-MS can be set to 15. Finally, compared to the MS/OMS decoders, the SP-MS decoder uses fewer iterations to reach the same FER. For example, at $E_b/N_0 = 4.75$ dB, the $(4, 3)$-bit SP-MS and the $(4, 4)$-bit SP-MS reach FER $= 10^{-6}$ with 6 iterations, while the $(4, 4)$-bit OMS uses 9 iterations.

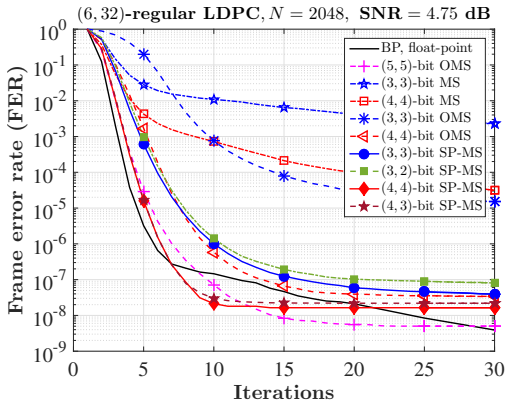


Figure 6: FER convergence comparison on the ETHERNET code at $E_b/N_0 = 4.75$ dB.

The average number of decoding iterations for the ETHERNET code is shown in Figure 7. For low precision $q_{ch} = 3$, we note that the SP-MS decoder not only has a better performance than the MS/OMS decoder, it also has the lowest average number of decoding iterations, which leads to lower latency and a higher average throughput. At $E_b/N_0 = 4.5$ dB, the average number of decoding iterations is 5.5 for the OMS whereas it is only 3 for the SP-MS. This reduction of

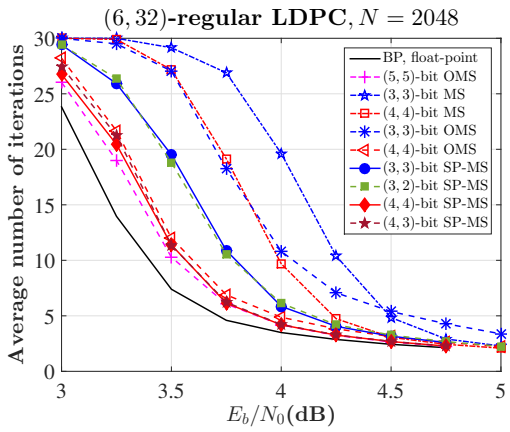


Figure 7: Average number of iterations of the ETHERNET code with different decoders.

the average number of decoding iterations translates into a 83.3%[3] improvement of the average decoding throughput.

When comparing the FER performance, the FER convergence, and the average number of iterations, the $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS decoder offers the best trade-off between performance and complexity. For applications that require low hardware complexity, the $(3, 2)$-bit SP-MS decoder is the best option. For applications where hardware complexity is not an issue and decoding performance is privileged, the $(4, 3)$-bit SP-MS decoder is the best choice.

We also performed MC simulations for the following LDPC codes: $(d_v = 4, d_c = 8, N = 1296)$, $(d_v = 5, d_c = 10, N = 1280)$, $(d_v = 3, d_c = 12, N = 1296)$, $(d_v = 4, d_c = 16, N = 1296)$, and $(d_v = 5, d_c = 20, N = 1280)$, and we obtained similar conclusions.

C. Performance of Irregular LDPC codes

Figure 8 shows the simulation results for the WIMAX rate 1/2 LDPC code, for a maximum of 100 iterations. We observe that the SNR gains in the waterfall region are congruent with the gains predicted by the DE analysis, with a 0.40 dB gain for $q = 3$ and a 0.40 dB gain for $q = 4$, at FER $= 10^{-2}$.

Additionally, the $(3, 3)$-bit SP-MS decoder has the same FER performance as the $(4, 4)$-bit MS decoder. In the waterfall region, the $(4, 4)$-bit SP-MS has the same FER performance as the $(5, 5)$-bit OMS.

[3]For a fully parallel architecture, the average decoding throughput is given by $T_{avg} = NF/L_{avg}$ (in Mbit/s), where F (in MHz) is the clock frequency and L_{avg} is the average number of decoding iterations. Note that F is constant for all decoders.

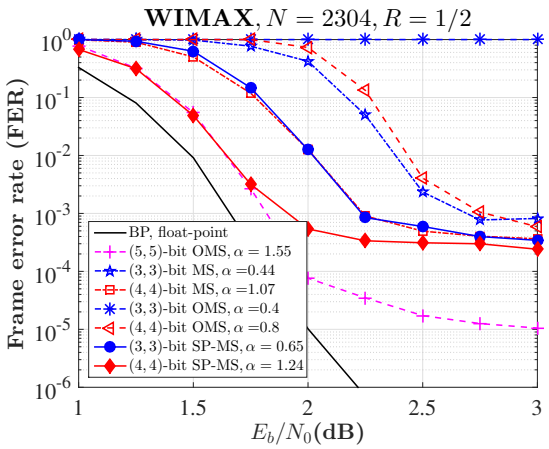


Figure 8: FER performance for the WIMAX LDPC code with $R = 1/2$.

VI. Mitigation of the Early Appearance of the Error Floor

In this section we propose a method to mitigate the early appearance of the error floor of the SP-MS decoder when the precision of the messages is lower than the precision of the LLRs, *i.e.* when $q = q_{ch} - 1$.

A. Early Appearance of the Error Floor

Our MC simulations of regular LDPC codes shows that the $(q_{ch}, q = q_{ch}-1)$-bit SP-MS decoders have almost the same performance as the $(q_{ch}, q = q_{ch})$-bit SP-MS decoder for $d_v \geq 5$ LDPC codes. In the case of $d_v = 3$ and $d_v = 4$ LDPC codes, the $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS decoder exhibits a very high error floor as shown in Figure 2 and Figure 3.

The early appearance of this error floor is not due to trapping sets, but to the effect of low message precision on the VNU computation: the maximum absolute value of check-to-variable messages is too small compared to the maximum absolute value of the LLRs. This effect is due to configurations where the sum of check-to-variable messages does not exceed an erroneous high LLR value and therefore cannot correct the bit error.

Let us take the example of the $(3, 2)$-bit SP-MS decoder, with $N_{ch} = 3$, $N_q = 1$, and the case of a VN v_n connected to three check-nodes c_1, c_2, and c_3. For this example, we assume that $x_n = 0$ is the value of the bit associated with variable-node v_n, hence, the estimated bit $\hat{x}_n$ is correct ($\hat{x}_n = 0$) if $\gamma_n^{(\ell)} > 0$ or if $I_n > 0$ for $\gamma_n^{(\ell)} = 0$. In Table VIII, we show the APP $\gamma_n^{(\ell)}$ computed with the discrete function Ψ_a defined in (13) for a fixed value $m_{c_3 \to v_n}^{(\ell)} = +1$ and

Table VIII: All values that can take $\gamma_n^{(\ell)} = \Psi_a\left(I_n, m_{c_1\to v_n}^{(\ell)}, m_{c_2\to v_n}^{(\ell)}, m_{c_3\to v_n}^{(\ell)}\right)$, with $m_{c_3\to v_n}^{(\ell)} = +1$.

I_n	-3				-2				-1				-0				$+0$			
$m_{c_2\to v_n}^{(\ell)} \backslash m_{c_1\to v_n}^{(\ell)}$	-1	-0	$+0$	$+1$	-1	-0	$+0$	$+1$	-1	-0	$+0$	$+1$	-1	-0	$+0$	$+1$	-1	-0	$+0$	$+1$
-1	-5	-4	-3	-2	-4	-3	-2	-1	-3	-2	-1	0	-2	-1	0	**1**	-1	**0**	**1**	**2**
-0	-4	-3	-2	-1	-3	-2	-1	0	-2	-1	0	**1**	-1	0	**1**	**2**	**0**	**1**	**2**	**3**
$+0$	-3	-2	-1	0	-2	-1	0	**1**	-1	0	**1**	**2**	0	**1**	**2**	**3**	**1**	**2**	**3**	**4**
$+1$	-2	-1	0	**1**	-1	0	**1**	**2**	0	**1**	**2**	**3**	**1**	**2**	**3**	**4**	**2**	**3**	**4**	**5**

$I_n \in \{-3, -2, -1, -0, +0\}$. For $I_n = -3$ we obtain $\gamma_n^{(\ell)} > 0$ ($\hat{x}_n = 0$) only when $m_{c_1\to v_n}^{(\ell)} = m_{c_2\to v_n}^{(\ell)} = m_{c_3\to v_n}^{(\ell)} = +1$, *i.e.* one can estimate the correct value of the bit only if all messages have the maximum allowed value ($+N_q$). For the case of $I_n = -2$ (respectively $I_n = -1$) we can see 3 (respectively 6) configurations to correctly estimate the bit.

The effect of low precision messages is even more pronounced on the message update at the VNU. Considering $I_n = -3$, $m_{c_1\to v_n}^{(\ell)} = m_{c_2\to v_n}^{(\ell)} = +1$, and $\varphi_v = 1$ ($\varphi_s = \varphi_v$) we obtain $m_{v_n\to c_3}^{(\ell+1)} = \Psi_v(-3, +1, +1) = -0$ using (14), *i.e.* the VNU will propagate an incorrect message despite the correct values of the check-to-variable (or incoming in this case) messages. This kind of VNU error propagation has a negative impact on the decoding process and contributes to the observed high error floor. A similar behavior can be observed for other configurations such as $m_{v_n\to c_3}^{(\ell+1)} = \Psi_v\left(I_n = -2, m_{c_1\to v_n}^{(\ell)} = +1, m_{c_2\to v_n}^{(\ell)} = +0\right) = -0$ or $m_{v_n\to c_3}^{(\ell+1)} = \Psi_v\left(I_n = -2, m_{c_1\to v_n}^{(\ell)} = +0, m_{c_2\to v_n}^{(\ell)} = +0\right) = -0$.

B. Mitigation of the Early Appearance of the Error Floor

To mitigate the appearance of the early error floor, we propose a modification of the VN update that involves changing the maximum amplitude of the check-to-variable messages from N_q to $\Omega > N_q$ after a given number of iterations L_m. The amplitude N_q is used for the first $L_m - 1$ iterations, and then a larger amplitude $\Omega > N_q$ is used from iteration L_m until the end of decoding. Then the modified VNU rule is given by

$$m_{v_n\to c_m}^{(\ell+1)} = \Psi_v\left(I_n, \left\{m_{c\to v_n}^{(\ell)}\right\}_{c\in\mathcal{V}(v_n)\backslash\{c_m\}}\right) = \left(\text{sign}\left(m_{v_n\to c_m}^{(\ell+1),U}\right), \mathcal{S}\left(\max\left(\left\lfloor\left|m_{v_n\to c_m}^{(\ell+1),U}\right|\right\rfloor - b, 0\right), N_q\right)\right). \quad (17)$$

where $b \in \{\varphi_s, \varphi_a, \varphi_0\}$ is the offset and the unsaturated variable-to-check message $m_{v_n\to c_m}^{(\ell+1),U}$ is redefined as:

$$m_{v_n\to c_m}^{(\ell+1),U} = \frac{\mu_{v_n\to c_m}^{(\ell)}}{2} + I_n + \sum_{c\in\mathcal{V}(v_n)\backslash\{c_m\}} w_{c\to v_n}^{(\ell)}, \quad (18)$$

with the value of $w_{c\to v_n}^{(\ell)}$ depending on the iteration:

$$w_{c\to v_n}^{(\ell)} = \begin{cases} m_{c\to v_n}^{(\ell)}, & \text{if} \quad \ell < L_m, \\ m_{c\to v_n}^{(\ell)}, & \text{if} \quad \ell \geq L_m \text{ and } \left|m_{c\to v_n}^{(\ell)}\right| < N_q, \\ \left(\text{sign}\left(m_{c\to v_n}^{(\ell)}\right), \Omega\right), & \text{if} \quad \ell \geq L_m \text{ and } \left|m_{c\to v_n}^{(\ell)}\right| = N_q. \end{cases} \tag{19}$$

In addition, the APP computation at a VN v_n is redefined as follows

$$\gamma_n^{(\ell)} = \Psi_a\left(I_n, \left\{w_{c\to v_n}^{(\ell)}\right\}_{c\in\mathcal{V}(v_n)}\right) = I_n + \frac{1}{2}\,\xi\,\text{sign}(I_n) + \sum_{c\in\mathcal{V}(v_n)}\left(w_{c\to v_n}^{(\ell)} + \frac{1}{2}\,\text{sign}\left(w_{c\to v_n}^{(\ell)}\right)\right). \tag{20}$$

The variable-to-check messages $m_{v_n\to c_m}^{(\ell+1)} \in \mathcal{B}_M$ are computed using (17), and the CNU generates check-to-variable messages $m_{c_m\to v_n}^{(\ell+1)} \in \mathcal{B}_M$ with (8).

Table IX: Optimal values to implement the $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS decoders.

	$(d_v = 3, d_c = 6)$				$(d_v = 4, d_c = 64)$			
(q_{ch}, q)	Ω	L_m	α^*	$(\varphi_s^*, \varphi_a^*, \varphi_0^*)$	Ω	L_m	α^*	$(\varphi_s^*, \varphi_a^*, \varphi_0^*)$
$(3, 2)$	2	0	0.6	$(1, -, -)$	2	13	0.50	$(1, -, -)$
$(4, 3)$	4	0	1.16	(1, 1, 0)	5	13	0.95	(1, 1, 1)

Table X: All values that can take $\gamma_n^{(\ell)} = \Psi_a\left(I_n, w_{c_1\to v_n}^{(\ell)}, w_{c_2\to v_n}^{(\ell)}, w_{c_3\to v_n}^{(\ell)}\right)$, with $w_{c_3\to v_n}^{(\ell)} = +2$.

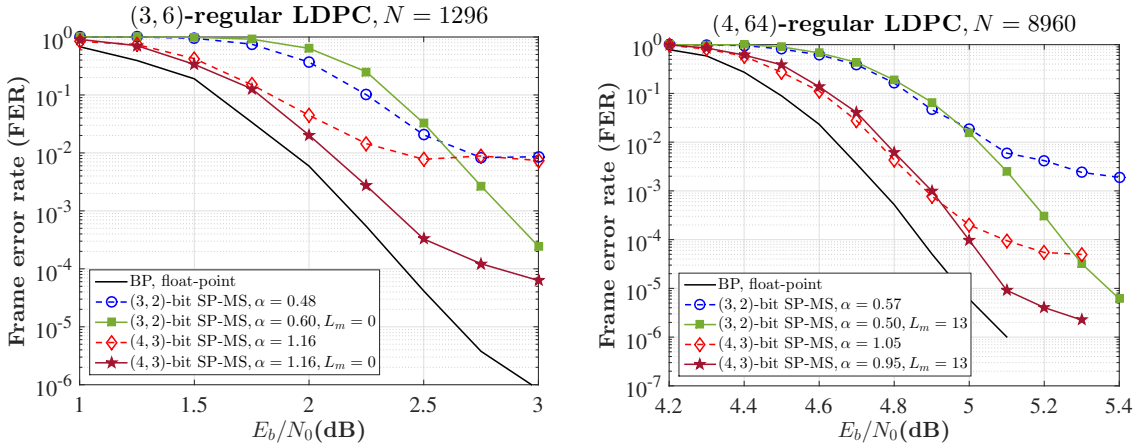
I_n	-3				-2				-1				-0				$+0$			
$w_{c_2\to v_n}^{(\ell)} \backslash w_{c_1\to v_n}^{(\ell)}$	-2	-0	$+0$	$+2$	-2	-0	$+0$	$+2$	-2	-0	$+0$	$+2$	-2	-0	$+0$	$+2$	-2	-0	$+0$	$+2$
-2	-6	-4	-3	-1	-5	-3	-2	0	-4	-2	-1	①	-3	-1	0	**2**	-2	**0**	**1**	**3**
-0	-4	-2	-1	①	-3	-1	0	②	-2	0	①	**3**	-1	①	**2**	**4**	**0**	**2**	**3**	**5**
$+0$	-3	-1	0	②	-2	0	①	**3**	-1	①	**2**	**4**	0	**2**	**3**	**5**	**1**	**3**	**4**	**6**
$+2$	-1	①	②	**4**	0	②	**3**	**5**	①	**3**	**4**	**6**	**2**	**4**	**5**	**7**	**3**	**5**	**6**	**8**

The optimal values of α^*, $\boldsymbol{\varphi}^* = (\varphi_s^*, \varphi_a^*, \varphi_0^*)$, and the new parameters Ω and L_m, computed with Monte Carlo simulations, are reported in Table IX, and those values are used to implement the $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS decoders for the $d_v = 3$ and $d_v = 4$ LDPC codes.

To illustrate how the proposed method helps the decoder, we use the example of the $(3, 2)$-bit SP-MS decoder and consider the parameters presented in Table IX. The new values of the APP $\gamma_n^{(\ell)}$ computed after iteration L_m are shown in Table X. The circled numbers correspond to the new configurations for which we can estimate the correct value of the bit with our modification. As an example for $I_n = -3$, we observe that from a single configuration (Table VIII), we now have 5 configurations (Table X) to correctly estimate the bit. Calculating the variable-to-check message $m_{v_n\to c_3}^{(\ell+1)}$ for $I_n = -3$, $m_{c_1\to v_n}^{(\ell)} = m_{c_2\to v_n}^{(\ell)} = +1$, and $\varphi_v = 1$ ($\varphi_s^* = \varphi_v$),

using the modified VNU, the output message now propagates the correct sign, with $m_{v_n \to c_3}^{(\ell+1)} = \Psi_v \left(I_n = -3, w_{c_1 \to v_n}^{(\ell)} = +2, w_{c_2 \to v_n}^{(\ell)} = +2 \right) = +0$.

Figure 9 shows the simulation results of the $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS decoders for $d_v \in \{3, 4\}$ LDPC codes. We can see that the proposed modification helps greatly to lower the error floor.



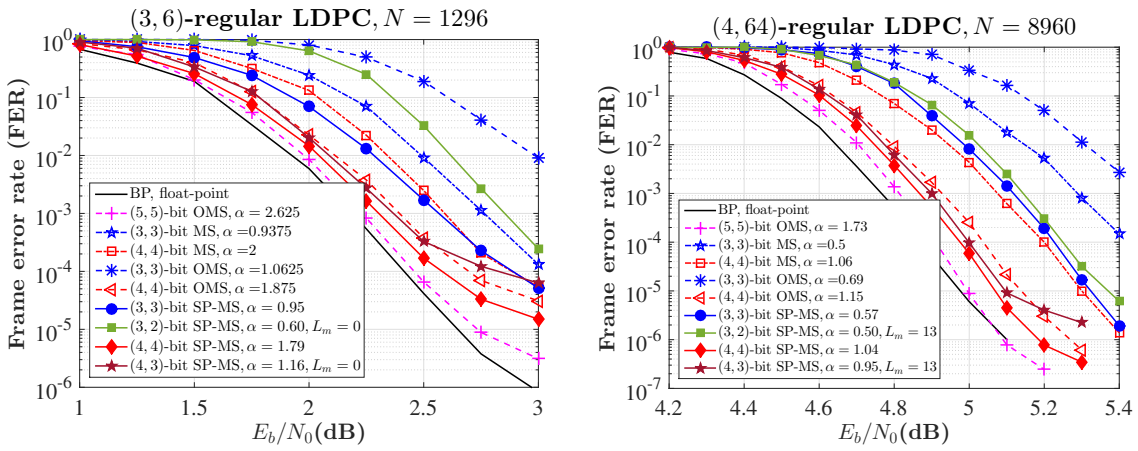
(a) Error floor for $d_v = 3$ LDPC codes.

(b) Error floor for $d_v = 4$ LDPC codes.

Figure 9: Lowering error floors for $d_v = 3$ and $d_v = 4$ LDPC codes.

The FER performance curves obtained using the optimal values presented in Table IX, plotted in Figure 10, show that the $(4, 3)$-bit SP-MS decoder is slightly better than the $(4, 4)$-bit OMS decoder. In addition, the $(3, 2)$-bit SP-MS decoder is better than the $(3, 3)$-bit MS/OMS decoder for the $d_v = 4$ LDPC code. In the case of the $d_v = 3$ LDPC code, we observe a loss of performance of the $(3, 2)$-bit SP-MS compared with the $(3, 3)$-bit MS.

The proposed method to mitigate the appearance of the early error floor can be easily adapted to the case of MS/OMS decoders. First the unsaturated variable-to-check message is redefined as $m_{v_n \to c_m}^{(\ell+1),U} = I_n + \sum_{c \in \mathcal{V}(v_n) \setminus \{c_m\}} w_{c \to v_n}^{(\ell)}$, where $w_{c \to v_n}^{(\ell)} = \Omega \operatorname{sign}\left(m_{c \to v_n}^{(\ell)} \right)$ if $\ell \geq L_m$ and $\left| m_{c \to v_n}^{(\ell)} \right| = N_q$, otherwise, $w_{c \to v_n}^{(\ell)} = m_{c \to v_n}^{(\ell)}$. Then the variable-to-check message is computed with $m_{v_n \to c_m}^{(\ell+1)} = \Lambda \left(m_{v_n \to c_m}^{(\ell+1),U}, \varphi_v \right)$. Finally, the APP computation is redefined as $\gamma_n^{(\ell)} = I_n + \sum_{c \in \mathcal{V}(v_n)} w_{c \to v_n}^{(\ell)}$. The check-to-variable message is computed using (4). For example, we obtain with MC simulation that $(\Omega = 2, L_m = 12, \alpha^* = 0.68, \varphi_v = 1)$ optimizes the FER performance of the $(3, 2)$-bit OMS decoder for the IEEE 802.3 ETHERNET code; we have observed that the $(3, 2)$-bit OMS decoder has almost the same performance as the $(3, 3)$-bit OMS decoder, and we have not observed the appearance of the error floor at $E_b/N_0 = 5$ dB.



(a) $d_v = 3$ LDPC codes, a maximum of 100 iterations. (b) $d_v = 4$ LDPC codes, a maximum of 30 iterations.

Figure 10: Comparison of the performance of $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS decoders with the MS and OMS decoders after lowering the error floor.

VII. Hardware Complexity of Sign-Preserving Min-Sum Decoders

A. Reduction of Wires and Memory

The decoding performance results indicate that the $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS decoder is a good candidate for hardware implementation. The memory and the number of wires used in an implementation is the same for the MS, OMS, and SP-MS decoders when the same precision is used $(q_{ch} = q)$. Considering a fully parallel architecture, when the precision q goes from 3 bits to 2 bits, the reduction of wires is 33.33%, whereas when the precision q goes from 4 bits to 3 bits, a reduction of 25% of wires is obtained. For a layered architecture, a memory reduction of up to 33.33% can be obtained when check-to-variable messages are stored. In the case of storing the variable-to-check messages in the *compressed format*[4]of [40] that requires the storage of the signs, first and second minima, and the index of the first minima, a memory reduction of up to 15.38% can be achieved for the regular $(d_v = 3, d_c = 6)$ LDPC code.

The FER results show that the $(4,3)$-bit SP-MS can achieve the performance of the $(5,5)$-bit OMS, thus promoting a significant reduction of the wires (by 40%), with negligible performance degradation (0.028 dB for $d_v = 6$, see Table VII for other values). In addition, a reduction of

[4]The size of memory required to store the check-to-variable messages of a CN of degree d_c is: $d_c + \lfloor \log_2(d_c) \rfloor + 1 + 2(q-1)$.

up to 40% in the size of the memory can be achieved by using $q = 3$ bits message precision instead of $q = 5$ bits.

B. Synthesis results on FPGA

In this section, we present the synthesis result of a fully parallel architecture on the Xilinx XC7V2000T-1FLG1925 FPGA chip for the IEEE 802.3 ETHERNET code. The main available resources of the FPGA are: *(i)* 2,443,200 Slice Registers, *(ii)* 1,221,600 Slice LUTs, and *(iii)* 351,321 LUT-FF pairs.

The FPGA resource utilization of the OMS and SP-MS decoders is listed in Table XI. In addition, the maximum clock frequency that each decoder can reach is listed. From the results obtained, we can see that the maximum clock frequency of $(q_{ch}, q = q_{ch}-1)$-bit SP-MS is higher compared to the maximum clock frequency of the $(q_{ch}, q = q_{ch})$-bit OMS, we observe an increase in the clock frequency of up to 30%. Reduced precision for the messages entails reduced complexity of the VNUs and CNUs and reduced wires in implementation, and thus a reduced critical path, which promotes a higher clock frequency (*i.e.* the throughput is increased).

When comparing the use of FPGA resources, we can clearly see that the $(q_{ch}, q = q_{ch} - 1)$-bit SP-MS decoders use less resources than the $(q_{ch}, q = q_{ch})$-bit OMS decoders. A large savings of FPGA resources can be observed: around 27% of slice registers and 56% of slice LUTs for low precision $(q_{ch} = 3, q = 2)$ compared to precision $(q_{ch} = 3, q = 3)$, and 20% of slice registers and 48% of slice LUTs for precision $(q_{ch} = 4, q = 3)$ compared to precision $(q_{ch} = 4, q = 4)$. These results confirm that the complexity of update rules is reduced when the precision of messages is reduced.

Table XI: Synthesis results on FPGA for the IEEE 802.3 ETHERNET code.

Decoder	(q_{ch}, q)	Maximum Clock Frequency (MHz)		Number of slice registers		Number of slice LUTs		Number of fully used LUT-FF pairs		Wires reduction
OMS	$(3, 3)$	111.100	(0.0%)	45061	(0.0%)	463739	(0.0%)	38917	(0.0%)	0.0%
SP-MS	$(3, 2)$	133.582	$(+20.24\%)$	32773	(-27.27%)	204091	(-55.99%)	32773	(-15.78%)	-33.33%
OMS	$(4, 4)$	87.790	(0.0%)	59397	(0.0%)	666300	(0.0%)	51205	(0.0%)	0.0%
SP-MS	$(4, 3)$	113.814	$(+29.64\%)$	47109	(-20.68%)	345913	(-48.08%)	47109	(-7.99%)	-25.0%

From all these results, we can conclude that the SP-MS decoder is better overall than the MS/OMS decoder. Not only does the SP-MS decoder have better performance, it also reduces the complexity of the VN and CN processing, and finally it has a better convergence speed.

Table XII: Comparison of the SP-MS decoder with other decoders reported in the literature.

	This work		[23]	[17]	[18]	[20]	[19]	[28]	[32]	[27]	[25]
Decoder	SP-MS		Finite Alphabet	OMS with post-proc.	Split-row-16 MS	Normalized Prob. MS	Reduced Complexity MS	Delayed Stochastic	Improved Differential Binary	MTFM-based Stochastic	APP based MS
(q_{ch}, q)	(3, 2)	(4, 3)	(4, 3)	(4, 4)	(5, 5)	(4, 3)	(6, 6)	(5, 1)	(6, 1)	(6, 1)	(2, 2)
Maximum Iterations	20	14	5	8 + 6 post-proc.	11	9	30	600 (with post-proc)	315	400 (with post-proc)	48
E_b/N_0 (dB)[†]	4.5	4.33	4.95	4.37	4.55	4.45	4.35	4.7	4.5	4.45	4.86
Architecture	full-parallel		unrolled full-parallel	partial-parallel	full-parallel	full-parallel	layer-parallel	full-parallel	full-parallel	full-parallel	full-parallel

[†] At a BER level of 10^{-7}.

VIII. Comparison with Other Studies

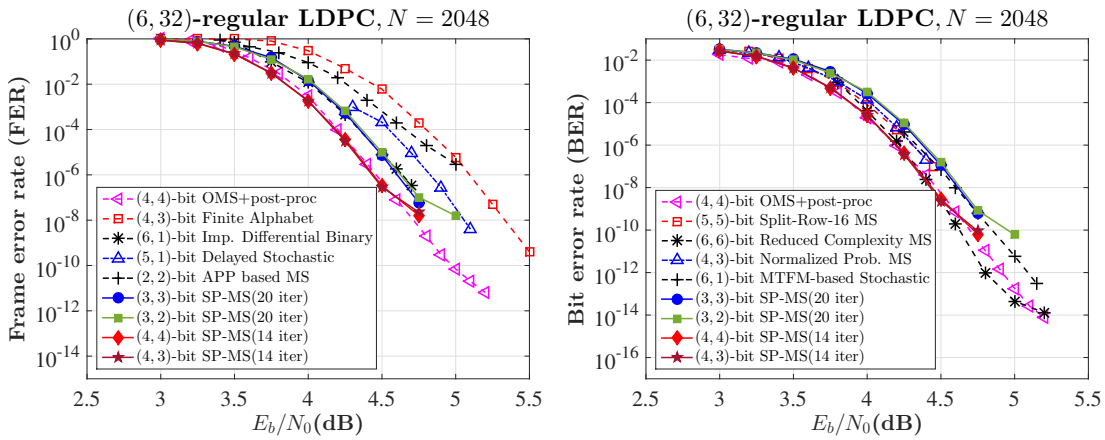
Many decoders have been proposed in the literature for the IEEE 802.3 ETHERNET code, and some of them are listed in Table XII. In Figure 11, we compare the error correction performance of the SP-MS decoders and the decoders listed in Table XII. The BER/FER curves of the decoders listed in Table XII were taken directly from the cited papers. All SP-MS decoders exhibit better BER/FER performance than the $(4, 3)$-bit finite alphabet decoder [23], the $(2, 2)$-bit APP based MS decoder [25], and the $(5, 1)$-bit delayed stochastic decoder [28].

Additionally, the $(3, 2)$-bit SP-MS decoder has almost the same performance as the $(5, 5)$-bit split-row-16 MS decoder [18] and the $(4, 3)$-bit normalized probabilistic MS decoder [20] with a much lower hardware complexity. Comparing the $(3, 2)$-bit SP-MS decoder with the $(6, 1)$-bit improved differential binary decoder [32] and the $(6, 1)$-bit MTFM-based stochastic decoder [27], we observe that all three decoders have almost the same performance. It is worth mentioning that the $(3, 2)$-bit SP-MS decoder uses only 20 iterations whereas the 1-bit LDPC decoders [27], [28], [32] use more than 300 iterations.

As a last remark, we observe that the SP-MS decoders exhibit an error floor at a FER level of 10^{-8} due to trapping sets.

IX. Conclusion

In this paper, we have proposed a new message-passing iterative LDPC decoder which uses a sign-preserving factor that helps the decoder keeping the sign information of extrinsic messages during the VNU processing. The sign-preserving factor also helps increasing the reliability of variable-to-check messages and thus helps the finite precision iterative decoder improving the error-correcting performance. We have also proposed an offset model that depends on the magnitude of unsaturated variable-to-check messages. Density Evolution was used to optimize



(a) FER performance.

(b) BER performance.

Figure 11: Performance comparison of the SP-MS decoders with other decoders reported in the literature for the IEEE 802.3 ETHERNET code.

our SP-MS decoder performance. The analysis conducted with DE has shown that the sign preservation of messages is always beneficial for low precision SP-MS decoders. The DE threshold results for SP-MS decoders have also shown that the precision of messages can be reduced by one bit while maintaining the same error-correcting performance. The finite-length Monte Carlo simulations have corroborated the DE analysis.

For the regular $d_v = 3$ and $d_v = 4$ LDPC codes, reducing the precision of the messages creates an early error floor. To mitigate the appearance of the early error floor, we have proposed a method that involves increasing the maximum amplitude of the check-to-variable messages during the VNU processing.

In this study, we have demonstrated that the $(4, 3)$-bit SP-MS decoders can achieve the same error-correcting performance as the $(5, 5)$-bit MS/OMS decoders, and the $(3, 2)$-bit SP-MS decoders outperform the $(3, 3)$-bit MS/OMS decoders, with a SNR gain of up to 0.43 dB. We have also shown that the SP-MS decoders converge faster than the MS/OMS decoders. With the synthesis results on an FPGA, we have also demonstrated the low hardware complexity required by the SP-MS decoders.

ACKNOWLEDGEMENT

This study has been funded by the French National Research Agency (ANR) under grant number ANR-15-CE25-0006-01 (NAND project).

REFERENCES

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. PhD thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, 1963.

[2] R. Gallager, "Low-Density Parity-Check Codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, January 1962.

[3] D. Declercq, M. Fossorier, and E. Biglieri, *Channel Coding: Theory, Algorithms, and Applications*. Academic Press Library in Mobile and Wireless Communications. ISBN:978-0-12-396499-1, 2014.

[4] ETSI, "ETSI EN 302 307-2 V1.1.1 (2014-10) Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 2: DVB-S2 Extensions (DVB-S2X)," 2014.

[5] "IEEE Standard for Ethernet," *IEEE Std 802.3-2015 (Revision of IEEE Std 802.3-2012)*, pp. 1–4017, March 2016.

[6] "IEEE Standard for Local and metropolitan area networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1," 2005.

[7] G. Dong, N. Xie, and T. Zhang, "On the Use of Soft-Decision Error-Correction Codes in nand Flash Memory," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, pp. 429–439, Feb 2011.

[8] "IEEE Approved Draft Standard for Error Correction Coding of Flash Memory Using Low-Density Parity Check Codes," *IEEE P1890/D2, September 2017*, pp. 1–56, Jan 2018.

[9] R. Tanner, "A Recursive Approach to Low Complexity Codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, September 1981.

[10] F. R. Kschischang, B. J. Frey, and H. . Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb 2001.

[11] 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, March 1997.

[12] S.-Y. Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, pp. 58–60, Feb 2001.

[13] L. Ping and W. K. Leung, "Decoding Low-Density Parity-Check Codes with Finite Quantization Bits," *IEEE Communications Letters*, vol. 4, no. 2, pp. 62–64, Feb 2000.

[14] J. Chen and M. P. C. Fossorier, "Density Evolution for Two Improved BP-Based Decoding Algorithms of LDPC Codes," *IEEE Communications Letters*, vol. 6, no. 5, pp. 208–210, May 2002.

[15] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-Complexity Decoding of LDPC Codes," *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, Aug 2005.

[16] J. Zhao, F. Zarkeshvari, and A. H. Banihashemi, "On Implementation of Min-Sum Algorithm and its Modifications for Decoding Low-Density Parity-Check (LDPC) Codes," *IEEE Transactions on Communications*, vol. 53, no. 4, pp. 549–554, April 2005.

[17] Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "An Efficient 10GBASE-T Ethernet LDPC Decoder Design With Low Error Floors," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 843–855, April 2010.

[18] T. Mohsenin, D. N. Truong, and B. M. Baas, "A Low-Complexity Message-Passing Algorithm for Reduced Routing Congestion in LDPC Decoders," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 5, pp. 1048–1061, May 2010.

[19] F. Angarita, J. Valls, V. Almenar, and V. Torres, "Reduced-Complexity Min-Sum Algorithm for Decoding LDPC Codes With Low Error-Floor," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 7, pp. 2150–2158, July 2014.

[20] C. Cheng, J. Yang, H. Lee, C. Yang, and Y. Ueng, "A Fully Parallel LDPC Decoder Architecture Using Probabilistic Min-Sum Algorithm for High-Throughput Applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 9, pp. 2738–2746, Sept 2014.

[21] S. K. Planjery, D. Declercq, L. Danjean, and B. Vasic, "Finite Alphabet Iterative Decoders—Part I: Decoding Beyond Belief Propagation on the Binary Symmetric Channel," *IEEE Transactions on Communications*, vol. 61, pp. 4033–4045, October 2013.

[22] T. T. Nguyen-Ly, V. Savin, K. Le, D. Declercq, F. Ghaffari, and O. Boncalo, "Analysis and Design of Cost-Effective, High-Throughput LDPC Decoders," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 3, pp. 508–521, March 2018.

[23] R. Ghanaatian, A. Balatsoukas-Stimming, T. C. Müller, M. Meidlinger, G. Matz, A. Teman, and A. Burg, "A 588-Gb/s LDPC Decoder Based on Finite-Alphabet Message Passing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 2, pp. 329–340, Feb 2018.

[24] F. Zarkeshvari and A. H. Banihashemi, "On Implementation of Min-Sum Algorithm for Decoding Low-Density Parity-Check (LDPC) Codes," in *Global TeleCommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 2, pp. 1349–1353 vol.2, Nov 2002.

[25] Z. Cui and Z. Wang, "Improved Low-Complexity Low-Density Parity-Check Decoding," *IET Communications*, vol. 2, pp. 1061–1068, Sep. 2008.

[26] V. A. Chandrasetty and S. M. Aziz, "FPGA Implementation of High Performance LDPC Decoder Using Modified 2-Bit Min-Sum Algorithm," in *2010 Second International Conference on Computer Research and Development*, pp. 881–885, May 2010.

[27] S. Sharifi Tehrani, A. Naderi, G. Kamendje, S. Hemati, S. Mannor, and W. J. Gross, "Majority-Based Tracking Forecast Memories for Stochastic LDPC Decoding," *IEEE Transactions on Signal Processing*, vol. 58, pp. 4883–4896, Sep. 2010.

[28] A. Naderi, S. Mannor, M. Sawan, and W. J. Gross, "Delayed Stochastic Decoding of LDPC Codes," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5617–5626, Nov 2011.

[29] T. Wadayama, K. Nakamura, M. Yagita, Y. Funahashi, S. Usami, and I. Takumi, "Gradient Descent Bit Flipping Algorithms for Decoding LDPC Codes," *IEEE Transactions on Communications*, vol. 58, pp. 1610–1614, June 2010.

[30] G. Sundararajan, C. Winstead, and E. Boutillon, "Noisy Gradient Descent Bit-Flip Decoding for LDPC Codes," *IEEE Transactions on Communications*, vol. 62, no. 10, pp. 3385–3400, Oct 2014.

[31] C. Winstead and E. Boutillon, "Decoding LDPC Codes With Locally Maximum-Likelihood Binary Messages," *IEEE Communications Letters*, vol. 18, no. 12, pp. 2085–2088, 2014.

[32] K. Cushon, S. Hemati, C. Leroux, S. Mannor, and W. J. Gross, "High-Throughput Energy-Efficient LDPC Decoders Using Differential Binary Message Passing," *IEEE Transactions on Signal Processing*, vol. 62, pp. 619–631, Feb 2014.

[33] F. Cochachin, E. Boutillon, and D. Declercq, "Optimization of Sign-Preserving Noise-Aided Min-Sum Decoders with Density Evolution," in *2018 IEEE 10th International Symposium on Turbo Codes Iterative Information Processing (ISTC)*, pp. 1–5, Dec 2018.

[34] T. J. Richardson and R. L. Urbanke, "The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, Feb 2001.

[35] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, Feb 2001.

[36] C. K. Ngassa, V. Savin, E. Dupraz, and D. Declercq, "Density Evolution and Functional Threshold for the Noisy Min-Sum Decoder," *IEEE Transactions on Communications*, vol. 63, no. 5, pp. 1497–1509, May 2015.

[37] F. Cochachin, D. Declercq, E. Boutillon, and L. Kessal, "Density Evolution Thresholds for Noise-Against-Noise Min-Sum Decoders," *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–7, Oct 2017.

[38] F. Cochachin, *Noise-Against-Noise Decoder*. PhD thesis, Université Bretagne Sud, Lorient, France, 2019.

[39] A. Venkiah, D. Declercq, and C. Poulliat, "Design of Cages with a Randomized Progressive Edge Growth Algorithm," *IEEE Communications Letters*, vol. Vol. 12, no. 4, pp. 301–303, April 2008.

[40] Z. Wang and Z. Cui, "A Memory Efficient Partially Parallel Decoder Architecture for Quasi-Cyclic LDPC Codes," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, pp. 483–488, April 2007.

Franklin Cochachin received his Bachelor degree in Telecommunications Engineering from the National University of Engineering (Universidad Nacional de Ingeniería (UNI)), Lima, Peru, in 2014. From 2016 to 2019, he was a Ph.D. student at ETIS ENSEA/UCP/CNRS UMR-8051, Cergy-Pontoise, France, and Lab-STICC UMR-6285/UBS BP-92116, Lorient, France. He obtained his Ph.D. degree in Telecommunications in 2019, from Université Bretagne Sud, Lorient, France. His research interests include error-correction coding theory, the design and implementation of iterative error-correcting decoders on FPGA/ASIC platform, and digital communications.

Emmanuel Boutillon was born in Chatou, France, in 1966. He received his Diploma degree in engineering from Télécom ParisTech, Paris, France, in 1990, and his Ph.D. degree in 1995. In 1991, he was an Assistant Professor with the Ecole Multinationale Supérieure des Télécommunications, Dakar, Senegal. In 1992, he joined Télécom ParisTech as a Research Engineer, where he conducted research in the field of VLSI for digital communications. In 1998, he spent a sabbatical year at the University of Toronto, Toronto, ON, Canada. In 2000, he joined Université Bretagne Sud as a Professor. He ran the LESTER Lab from 2005 to 2007 and the CACS Department from 2008 to 2015. He is currently a Scientific Advisor with the Lab-STICC. In 2011, he spent his sabbatical year at INICTEL-UNI, Lima, Peru. His research interests include the interactions between algorithm and architecture in the field of wireless communications and high-speed signal processing. In particular, he works on turbo codes and low-density parity-check decoders.

David Declercq was born in June 1971. He obtained his PhD in Statistical Signal Processing in 1998, from the University of Cergy-Pontoise, France. He is a Senior member of the IEEE and held a junior position at the "Institut Universitaire de France" from 2009 to 2014. His research interests lie in digital communications and error-correction coding theory. He worked several years on the particular family of LDPC codes, from both the code and decoder design aspects. Since 2003, he has developed a strong expertise in non-binary LDPC codes and decoders in high order Galois fields GF(q). A large part of his research project is related to non-binary LDPC codes. He investigates mainly two aspects: (i) the design of GF(q) LDPC codes for short and moderate lengths, and (ii) the simplification of the iterative decoders for GF(q) LDPC codes with complexity/performance tradeoff constraints. David Declercq has published over 55 papers in major journals (IEEE-Trans. Commun., IEEE-Trans. Inf. Theo., Commun. Letters, IEEE-Trans. Circuits and Systems, etc.) and presented over 150 papers in major conferences in Information Theory and Signal Processing.