Baldi *et al. EURASIP Journal on Wireless Communications and Networking* (2016) 2016:272
DOI 10.1186/s13638-016-0769-z

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

On the use of ordered statistics decoders for low-density parity-check codes in space telecommand links

Marco Baldi[1,2*], Nicola Maturo[1,2], Enrico Paolini[3] and Franco Chiaraluce[1,2]

Abstract

The performance of short low-density parity-check (LDPC) codes that will be included in the standard for next-generation space telecommanding is analyzed. The paper is focused on the use of a famous ordered statistics decoder known as most reliable basis (MRB) algorithm. Despite its complexity may appear prohibitive in space applications, this algorithm is shown to actually represent a realistic option for short LDPC codes, enabling significant gains over more conventional iterative algorithms. This is possible by a hybrid approach which combines the MRB decoder with an iterative decoding procedure in a sequential manner. The effect of quantization is also addressed, by considering two different quantization laws and comparing their performance. Finally, the impact of limited memory availability onboard of spacecrafts is analyzed and some solutions are proposed for efficient processing, towards a practical onboard decoder implementation.

Keywords: Low-density parity-check codes, Most reliable basis algorithm, Ordered statistics decoding, Space missions, Telecommand links

1 Introduction

Wireless communication links in space missions extensively use error-correcting codes in order to improve transmission reliability [1–4]. There are two types of space links: telemetry (TM) links from space to ground and telecommand (TC) links from ground to space. The purpose of a TM link is to reliably and transparently convey remote measurement information to users located on Earth. For such a purpose, several coding schemes are employed. In particular,

- convolutional codes,
- Reed–Solomon codes,
- parallel turbo codes,
- low-density parity-check (LDPC) codes.

A comprehensive description of all these families of codes can be found, for example, in [5].

A TC link, in turn, is used to initiate, modify, or terminate equipment functions onboard (O/B) of space objects. From the communication viewpoint, TCs have a number of distinctive features. Among them,

- the data rates (measured in bits/s) are usually very low compared with TM links,
- TCs are originated and assembled on the ground.

The first above feature implies milder requirements for the error-correcting code. Similarly, because of the second feature, fewer limits and constraints are imposed on the available transmitted power. Recommendations for TC space links are issued traditionally by two organizations: the Consultative Committee for Space Data Systems (CCSDS) and the European Cooperation for Space Standardization (ECSS). The only error-correcting code so far included in these recommendations [6, 7] is a Bose–Chaudhuri–Hocquenghem (BCH) code. This linear block code is characterized by a codeword length $n = 63$ bits, an information block length $k = 56$ bits and is typically decoded O/B via hard-decision decoding.

*Correspondence: m.baldi@univpm.it
[1]Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Ancona, Italy
[2]Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), Parma, Italy
Full list of author information is available at the end of the article

Reliable space telecommanding is of fundamental importance as the success of a mission may be compromised because of an error corrupting a TC message. This imposes strict constraints on the maximum tolerable error rates. In particular, the codeword error rate (CER) is defined as the ratio of the number of decoding failures to the total number of received codewords. Requirements are often specified in terms of average CER, and a typical value is CER $\leq 10^{-5}$.

TC links are currently evolving to include new and more challenging mission profiles, characterized by much more demanding requirements. Compliance with these new requirements imposes the use of more advanced coding techniques than the BCH(63, 56) code. In space telecommanding, it is always of fundamental importance to ensure the integrity of very short emergency commands, with a very low data rate and limited latency. Such constraints impose the adoption of *short codes*.

The mentioned BCH code is short enough, but its performance is far from meeting the requirements of next generation missions. In the following analysis, we will refer to a classical binary phase shift keying (BPSK) modulation over an additive white Gaussian noise (AWGN) channel. This model is particularly suitable to describe deep-space communications. Using the standard BCH(63, 56) code with hard-decision decoding, the signal-to-noise ratio required to achieve CER $\approx 10^{-5}$ is $E_b/N_0 \approx 9.1$ dB, where E_b is the energy per information bit and N_0 the one-sided noise power spectral density. This value of E_b/N_0 can be reduced, so improving performance, by applying soft-decision decoding. For example, by using a decoder based on the BCJR algorithm [8], a gain of about 2.1 dB with respect to the hard-decision decoder can be achieved. These E_b/N_0 values, however, remain too large for space missions of next generation. These missions shall be characterized by a significant increase of the supported data rate and/or maximum distance with respect to the present scenarios. Both these factors degrade the signal-to-noise ratio and impose to achieve the target CER over a worse channel.

In order to achieve higher coding gains, new advanced coding schemes have been designed and evaluated [9]. After a long campaign of simulations and comparisons, two short binary LDPC codes have emerged as the most attractive candidates. Most of the steps necessary for their inclusion in the standard for space TC have already been completed, so that they will certainly be employed in next-generation missions.

LDPC codes can be decoded through classical soft-decision iterative algorithms (IAs), such as the sum-product algorithm (SPA) [10] or its simplified versions, e.g., min-sum (MS) [11] or normalized min-sum (NMS) [12]. As regards SPA, throughout this paper, we will refer to its log-likelihood ratio (LLR) implementation. By using LDPC codes with IAs, the coding gain is larger than with the BCH code, although the performance remains relatively far from the theoretical limits.

More substantial improvements can result from the adoption of algorithms based on ordered statistics decoding (OSD) [13]. In this paper, in particular, we consider the well-known most reliable basis (MRB) algorithm [14]. MRB is a non-iterative soft-decision decoding algorithm able to approach the performance of a maximum likelihood (ML) decoder. The latter is optimal, in the sense of minimizing the decoding error probability. The main drawback of MRB is represented by its complexity, which makes its use problematic in TC links where decoding is performed O/B with very limited computational and memory resources. As mentioned, however, the length of the TC LDPC codes is small, namely, $n = 128$ bits and $n = 512$ bits, which makes MRB a potential candidate especially for the shorter code.

Complexity can be reduced by resorting to a hybrid solution, which combines MRB with IAs. More precisely, the hybrid approach consists of performing low-complexity decoding through an IA, at first, and invoking the MRB algorithm only when the IA is not able to find any valid codeword (detected error). The hybrid decoder has recently been used to decode also LDPC codes constructed on non-binary finite fields [15], which represent another option for space TC links [16, 17]. Due to their higher decoding complexity, however, non-binary LDPC codes are less attractive than their binary counterparts.

For the LDPC codes analyzed in this paper, the hybrid decoder outperforms the IA or the MRB algorithm when used individually. A qualitative explanation of this favorable behavior is as follows. The IA decoders here considered are not bounded-distance decoders: Therefore, they may be able to successfully decode soft-decision sequences from the channel even if they have relatively large Euclidean distances from the BPSK-modulated transmitted codeword and, at the same time, they may fail to decode soft-decision sequences at relatively small Euclidean distances from it. The MRB decoder instead works in a completely different way: Once having fixed its order i, it is able to correct all the error patterns involving i errors or less in the most reliable basis. Then, the two decoders complement one each other, this way improving the individual performances. On the other hand, it is evident that in case the IA is characterized by a high undetected codeword error rate (UCER), the MRB algorithm is not invoked in many times where it could correct and the performance of the plain MRB is better than that of the hybrid approach. This event does not occur for the considered codes that, by design, are characterized by values of UCER which are orders of magnitude lower than the CER. As an example, while the

CER value at the working point must be at least as low as 10^{-5}, the UCER value at the same point must be at least as low as 10^{-9} [18]. Consequently, the MRB decoder is not invoked in a very few number of cases where it should be, and the advantage offered by the SPA-LLR to sporadically correct a (relatively) large number of errors dominates.

While ensuring excellent error rate performance, the hybrid decoder is characterized by a significantly lower average complexity than MRB decoding used alone. Moreover, such a performance is not seriously affected by practical issues like, for example, quantization, on condition that suitable quantization rules are applied.

Motivated by the above considerations, in this paper, a thorough analysis of the performance of the MRB algorithm, used individually or in hybrid form, is developed with reference to space TC applications. While a preliminary version of this analysis has been considered in [19], the study is here deepened by addressing all technical issues in a more complete way. Among them, we investigate the impact of limited memory availability (a typically very stringent constraint in O/B processing), with the aim to discuss also the problems arising in practical implementations. In fact, while the presence of enough O/B memory would help fast convergence of the decoding algorithm, memories are usually considered not reliable in hostile environments like the one characterizing space missions, so that their usage is generally minimized. As a consequence, "on-the-fly" computations are preferred to memory accesses. Therefore, we extend the analysis in [19] by addressing both the case of a limited amount of memory and the case of total absence of memory. We also study a parallel implementation for controlling other important variables, like the decoding latency.

The organization of the paper is as follows. In Section 2, the considered error-correcting codes are described. In Section 3, the various decoding procedures are presented and expressions for computing their complexities are provided. The performance of these schemes is then assessed in Section 4, also taking into account the quantization issues. Section 5 is devoted to the analysis of the impact of limited memory and to the latency evaluation. Concluding remarks are given in Section 6.

2 LDPC codes for space telecommand links

The two considered LDPC codes are described in detail in [16]. The first code has information block length $k = 64$ bits and codeword length $n = 128$ bits. The second code has information block length $k = 256$ bits and codeword length $n = 512$ bits. Hence, both codes have code rate $R_c = k/n = 1/2$. A short overview of their main features is reported in the following. In particular, in Section 2.1, their structure is described, by specifying their parity-check matrices, while in Section 2.2, their weight spectrum properties are addressed.

2.1 Parity-check and generator matrices

For the codes we consider, the parity-check matrix $\mathbf{H}$ is an array of $M \times M$ square submatrices, where $M = k/4 = n/8$. This is specified, for both codes, in Fig. 1.

In the figure, $\mathbf{I}_M$ and $\mathbf{0}_M$ denote the $M \times M$ identity and zero matrices, respectively, and $\mathbf{\Phi}$ is the first right circular shift of $\mathbf{I}_M$. This means that $\mathbf{\Phi}$ has a non-zero entry at row i and column j if and only if $j = (i+1) \bmod M$. Moreover, $\mathbf{\Phi}^2$ represents the second right circular shift of $\mathbf{I}_M$, that is, $\mathbf{\Phi}^2$ has a non-zero entry at row i and column j if and only if $j = (i+2) \bmod M$, and so on. The $\oplus$ operator indicates element-wise modulo-2 addition.

As an alternative representation, a $k \times n$ generator matrix $\mathbf{G}$ can be obtained from the parity-check matrix $\mathbf{H}$. The length-n codeword $\mathbf{c}$ corresponding to a length-k information block $\mathbf{u}$ can be then generated as $\mathbf{c} = \mathbf{u}\,\mathbf{G}$. The matrices $\mathbf{G}$ for the two considered codes are reported in [16].

2.2 Weight distribution properties

The Hamming weight of a codeword is defined as the number of its nonzero elements. The performance of a linear block code under optimum ML decoding is governed by its weight spectrum, that is the number of codewords of Hamming weight w for all integer $0 \leq w \leq n$. Unfortunately, even when the weight spectrum is perfectly known (which is a very favorable condition,

$$H_{64\times128} = \begin{bmatrix} I_M \oplus \Phi^7 & \Phi^2 & \Phi^{14} & \Phi^6 & 0_M & I_M & \Phi^{13} & I_M \\ \Phi^6 & I_M \oplus \Phi^{15} & I_M & \Phi^1 & I_M & 0_M & \Phi^0 & \Phi^7 \\ \Phi^4 & \Phi^1 & I_M \oplus \Phi^{15} & \Phi^{14} & \Phi^{11} & I_M & 0_M & \Phi^3 \\ I_M & \Phi^1 & \Phi^9 & I_M \oplus \Phi^{13} & \Phi^{14} & \Phi^1 & I_M & 0_M \end{bmatrix}$$

$$H_{256\times512} = \begin{bmatrix} I_M \oplus \Phi^{63} & \Phi^{30} & \Phi^{50} & \Phi^{25} & 0_M & \Phi^{43} & \Phi^{62} & I_M \\ \Phi^{56} & I_M \oplus \Phi^{61} & \Phi^{50} & \Phi^{23} & I_M & 0_M & \Phi^{37} & \Phi^{26} \\ \Phi^{16} & I_M & I_M \oplus \Phi^{55} & \Phi^{27} & \Phi^{56} & I_M & 0_M & \Phi^{43} \\ \Phi^{35} & \Phi^{56} & \Phi^{62} & I_M \oplus \Phi^{11} & \Phi^{58} & \Phi^3 & I_M & 0_M \end{bmatrix}$$

Fig. 1 Parity-check matrices of the considered LDPC codes

normally unverified for LDPC codes), it is not possible to find an exact analytical expression of the error rate achieved by the ML decoder, even for very simple communication channel models. Thus, it is a common practice resorting to analytical bounds, an example of which is represented by the union bound, that establishes an upper bound on the error rate of the considered code under ML decoding [20].

For a binary linear block code over the AWGN channel and BPSK modulation, let us denote by A_w the multiplicity of the weight-w codewords and by $d_{\min}$ the code minimum distance. Then, the expression of the union bound for the CER is [21]

$$\mathrm{CER}_{\mathrm{UB}} = \sum_{w=d_{\min}}^{n} \frac{1}{2} A_w \mathrm{erfc}\sqrt{w R_c \frac{E_b}{N_0}}\,. \tag{1}$$

The first term of the sum, corresponding to $w = d_{\min}$, is also known as the "error floor." For sufficiently high values of E_b/N_0, the error floor provides an excellent approximation of the performance of ML decoding. In this sense, it represents a first benchmark for any sub-optimum decoding algorithm: the closer the error rate offered by the decoding algorithm to the error floor, the smaller the gap between the sub-optimum decoder and the optimum ML decoder. As from (1), the computation of $\mathrm{CER}_{\mathrm{UB}}$ requires the knowledge of the complete weight spectrum of the code. It is known that for LDPC codes this may be a non-trivial task. For the considered codes, however, much work has been done to overcome this issue.

In particular, the first and most significant terms of the weight distribution for the LDPC(128, 64) code have been specified as

$$\begin{aligned} A(x)|_{64\times 128} = {} & 16x^{14} + 528x^{16} + 5632x^{18} + 35968x^{20} \\ & + 123888x^{22} + 364944x^{24} + \ldots \end{aligned} \tag{2}$$

where the presence of the term $A_w\, x^w$ means that there are A_w codewords with Hamming weight w. The multiplicities A_{14}, A_{16} and A_{18} are exact [22]; this part of the weight spectrum has been obtained through computer searches using a carefully tuned "error impulse" technique [23]. The other multiplicities are lower bounds on the actual values and have been obtained by using the approach proposed in [24]. The overall estimate is anyway sufficiently stable and allows to draw a reliable union bound, as will be done in Section 4.

The most accurate evaluation, at least till now, of the weight distribution of the LDPC(512, 256) code has been reported in [22], where the estimated first terms of the spectrum have been specified as

$$\begin{aligned} A(x)|_{256\times 512} = {} & 64x^{40} + 704x^{42} + 6336x^{44} \\ & + 44736x^{46} + \ldots \end{aligned} \tag{3}$$

The multiplicities A_{40} and A_{42} appear to be exact, while A_{44} and A_{46} are approximate and, in general, there is not yet sufficient confidence on the reliability of the estimate, even as regards the value of $d_{\min}$. This does not allow to draw a sufficiently reliable union bound for the LDPC(512, 256) code.

3 Decoding algorithms

As decoding algorithms we consider one OSD (the MRB algorithm) and three IAs (SPA, MS, and NMS). We investigate their performance when used alone or combined in the aforementioned hybrid decoding scheme.

3.1 MRB decoder

Let us denote by $\mathbf{1}$ the length-n vector with all coordinates equal to 1. Then, let $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ and $\mathbf{x} = (x_0, x_1, \ldots, x_{n-1}) = \mathbf{1} - 2\,\mathbf{c} \in \{-1, +1\}^n$ be the transmitted codeword and its baseband BPSK-modulated version (that is, before being modulated with the carrier frequency), respectively. Moreover, let $\mathbf{y} = (y_0, y_1, \ldots, y_{n-1}) \in \mathbb{R}^n$ be the corresponding received vector of soft values after transmission over the AWGN channel and BPSK demodulation. The MRB decoder relies on a transformation from the original generator matrix, $\mathbf{G}$, to a new matrix, $\mathbf{G}^\star$, based on the *k most reliable* independent bits corresponding to $\mathbf{y}$. Upon reception of $\mathbf{y}$, the reliability of a bit is measured in terms of the magnitude of its a priori LLR. More specifically, the a priori LLR for the ith codeword bit is defined as

$$\mathsf{L}(c_i) = \log \frac{\Pr(c_i = 0|y_i)}{\Pr(c_i = 1|y_i)} = \frac{2y_i}{\sigma^2}$$

where σ^2 is the variance of the Gaussian thermal noise. The reliability of bit c_i is then defined as $|\mathsf{L}(c_i)|$.

The MRB algorithm of order i may be summarized as follows:

- Upon receiving the signal from the AWGN channel and demodulating it into the sequence $\mathbf{y}$, find the k most reliable received bits and collect them in a length-k vector $\mathbf{v}^\star$.
- Perform Gauss-Jordan elimination on the matrix $\mathbf{G}$, with respect to the positions of the k most reliable bits, to obtain a systematic generator matrix $\mathbf{G}^\star$ corresponding to such bits.[1]
- Encode $\mathbf{v}^\star$ to obtain a *candidate codeword* $\mathbf{c}^\star = \mathbf{v}^\star\mathbf{G}^\star$.
- Consider all (or an appropriate subset of) *test error patterns* (TEPs). By definition, a TEP, noted by $\mathbf{e}$, is a binary vector of length k and Hamming weight $w \le i$.
- For each TEP $\mathbf{e}$:

- Calculate $\tilde{\mathbf{v}} = \mathbf{v}^\star + \mathbf{e}$.
- Encode $\tilde{\mathbf{v}}$ to obtain a codeword $\tilde{\mathbf{c}} = \tilde{\mathbf{v}}\,\mathbf{G}^\star$.
- If the Euclidean distance between $\tilde{\mathbf{x}} = \mathbf{1} - 2\,\tilde{\mathbf{c}}$ and $\mathbf{y}$ is smaller than the one between $\mathbf{x}^\star = \mathbf{1} - 2\,\mathbf{c}^\star$ and $\mathbf{y}$, then update the candidate codeword as $\mathbf{c}^\star \leftarrow \tilde{\mathbf{c}}$.

At the end of the process, the algorithm delivers the codeword $\mathbf{c}^\star$ which minimizes the Euclidean distance from $\mathbf{y}$, limited to the codeword set associated with the considered TEP set.

Hereafter, we denote by MRB(i) an instance of the algorithm with order i. Clearly, MRB(k) coincides with ML decoding, which is intractable in practice. Actually, the maximum number of TEPs to be tested, for each received sequence $\mathbf{y}$, is equal to $N_{\mathrm{TEP}}^{\max} = \sum_{w=0}^{i} \binom{k}{w}$, and this number tends to become very large even for $i \ll k$. So, in practice, only very small values of i can be considered.

There exist several possible strategies to reduce the complexity of the MRB(i) decoder. Among them, our simulations confirm the effectiveness of the following sub-optimal strategies to stop the TEP analysis before it reaches $N_{\mathrm{TEP}}^{\max}$:

1. The number of patterns to be tested is reduced by properly choosing a reliability threshold A such that, if the distance between the current candidate codeword and $\mathbf{y}$ is lower than A, then the decoding algorithm outputs the candidate codeword without testing the remaining TEPs.
2. The TEPs are commonly generated in ascending weight order. However, it is possible to precompute a list of most likely TEPs [25], that is, a list containing the TEPs ordered according to the probability to yield codewords at small distances from $\mathbf{y}$, regardless of their weight $w \leq i$. The threshold criterion at the previous point can then be applied to the ordered list.

One or both approaches allow to test an average number of patterns, denoted by N_{TEP} in the following, significantly smaller than $N_{\mathrm{TEP}}^{\max}$. As we will see in Section 3.3, this may have a significant impact on the decoding complexity.

There are also other tricks that may be used to reduce the MRB decoder implementation complexity. A very effective one is based on the observation that a large number of TEPs have overlapping supports, where the support of a TEP is defined as the set of its nonzero coordinates. Due to the code linearity, we can compute the XOR of two TEPs and encode it through $\mathbf{G}^\star$. The resulting vector can then be added to the test codeword corresponding to the first TEP, in order to obtain the codeword corresponding to the second TEP. Since, for the small values of the order i we consider, the XOR of the two TEPs has a small Hamming weight, computing the corresponding codeword requires to sum a small number of rows of $\mathbf{G}^\star$, thus reducing the computational burden. The procedure can be, obviously, iterated.

3.2 Hybrid decoder

The complexity of the MRB(i) algorithm, with an order i that allows to achieve good performance, may result too high for practical implementation. In this case, it is possible to resort to a hybrid approach, which consists of applying the MRB decoder downstream the iterative decoder, by invoking it only when the iterative decoder terminates reporting a failure, as it was not able to find any codeword [26]. The procedure is summarized by the flow-chart in Fig. 2. We note that two possible error events characterize an iterative LDPC decoding algorithm, namely, detected error and undetected error. A detected error occurs when the decoder reaches the maximum number of iterations without converging to any valid codeword. On the other hand, an undetected error occurs when the decoder converges to a codeword different from the transmitted one. Note also that every failure of the MRB decoder yields an undetected error, since a codeword is always found by this decoder (i.e., the MRB decoder is *complete*, as opposed to LDPC iterative decoders which are *incomplete*).

As from Fig. 2, contrary to similar proposals appeared in previous literature, the MRB decoder is applied on the received sequence $\mathbf{y}$ and not on the output of the IA decoder. This permits us to circumvent the effects of the deteriorated soft values after a failure of the IA.

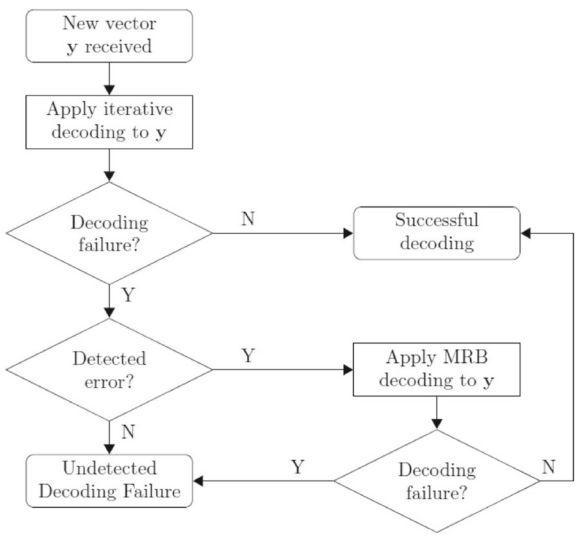

Fig. 2 Flow-chart for the hybrid algorithm (Y means "yes" and N means "no")

3.3 Decoding complexity

As mentioned in the previous sections, besides the error rate, another fundamental parameter for measuring the efficiency of a decoding algorithm is complexity. In case of the MRB algorithm, it is even the factor conditioning the applicability of the method.

A possible definition of complexity is in terms of the average number of binary operations required for decoding a single codeword. More practical issues, like chip area requirements, power consumption, number of look-up tables, number of flip-flops, and memory occupation, should be considered as well. These factors, however, are strongly dependent on the hardware architecture and design choices, while we aim at performing a more general complexity assessment, for which the number of operations turns out to be a suitable metric.

The average number of binary operations required by the MRB algorithm can be expressed by the following formula

$$\mathrm{C_{MRB}} = q\, n \log_2 n + \left(\frac{k}{2}\right)^3 + N_{\mathrm{TEP}} \left(2qi + q\frac{n-k}{2} + \frac{nk}{2}\right). \tag{4}$$

In (4), we have taken into account that some of the operations are performed on real values; so, q is the number of quantization bits used to represent a real number. As we can see, the third term on the right-hand side (r.h.s.) is proportional to the number of TEPs; therefore, using N_{TEP} instead of $N_{\mathrm{TEP}}^{\max}$, as permitted by the application of the speed-up procedures described in Section 3.1, can yield a significant advantage.

Equation (4) results from the evaluation of the computational effort required by the steps described in Section 3.1. In detail, the basic MRB decoding algorithm needs

- To order n real values.
- To perform Gauss-Jordan elimination on the $k \times n$ original generator matrix $\mathbf{G}$ to obtain the generator matrix $\mathbf{G}^\star$ in systematic form, in which the information bits coincide with $\mathbf{v}^\star$.
- To perform the vector-matrix multiplication $\mathbf{v}^\star\mathbf{G}^\star$.
- To generate, on average, N_{TEP} TEPs and perform the relevant calculations.

Next, the average complexity of the hybrid approach results in

$$\mathrm{C_{Hybrid}} = \mathrm{C_{IA}} + \alpha\, \mathrm{C_{MRB}} \tag{5}$$

where $\mathrm{C_{IA}}$ is the complexity of the IA preceding the (possible) MRB decoding attempt, while α represents the fraction of decoding instances in which MRB is invoked, i.e., the rate of the calls to MRB. This is because, in the hybrid approach, the MRB algorithm is invoked only when the IA fails. To be more precise, looking at the flow-chart of the hybrid decoder in Fig. 2, it should be noted that the rate of calls to the MRB decoder equals the *detected error rate* of the IA. In fact, when an undetected error occurs for the IA, decoding terminates unsuccessfully without any call to MRB. So, in principle, α is different from the CER, as the latter captures both detected and undetected errors. However, since the undetected error rate of an IA is usually orders of magnitude smaller than the detected one (unless the minimum distance of the code is very poor—this is not the case for the codes considered in this paper), we can assume $\alpha \approx \mathrm{CER}$, this way making a negligible error.

The expression of $\mathrm{C_{IA}}$ in (5) depends on the adopted IA and can easily be obtained from the algorithm specification (details are omitted for saving space). For SPA, MS, and NMS, we have

$$\mathrm{C_{SPA}} = I_{\mathrm{ave}} n \left[q\,(8d_v + 12R_c - 11) + d_v\right] \tag{6}$$

$$\mathrm{C_{MS}} = I_{\mathrm{ave}} n \left[q\,(3d_v + 2R_c) + 2d_v - 1 + R_c\right] \tag{7}$$

$$\mathrm{C_{NMS}} = \mathrm{C_{MS}} + I_{\mathrm{ave}} nq\,(2d_v + 1) \tag{8}$$

respectively, where I_{ave} is the average number of iterations and d_v is the average column weight of the parity-check matrix. The considered codes are both characterized by $d_v = 4$. Finally, it must be noted that, starting from the previous expressions, the corresponding decoding complexities per information bit can be obtained by dividing the r.h.s. of (4)–(8) by k.

4 Error rate versus complexity tradeoff evaluation

In this section, the CER/complexity tradeoff offered by each of the decoding algorithms considered in Section 3 is assessed, when used to decode the two LDPC codes described in Section 2. As previously stated, the modulation format is BPSK. Moreover, a maximum number of iterations $I_{\max} = 100$ has been used for each iterative algorithm. This value has been determined, from simulations, for the considered codes, as capable to obtain the best possible performance from the adopted IAs. This means that no significant extra gain is achievable by using a value of $I_{\max}$ larger than 100. An explicit example will be presented in Section 4.2. On the other hand, by using (6), (7), and (8), but with $I_{\max}$ in place of I_{ave}, we see that the complexity in the worst case (that is, when the decoder reaches the maximum number of iterations) grows linearly with $I_{\max}$. As we will show in Section 5, the worst case complexity determines the worst case latency, the latter quantity being another design parameter that must be properly limited. As another argument to justify the limited gain induced by an increase in the maximum number of iterations, we can observe that in the region of medium/low error rates, I_{ave} is usually much lower than $I_{\max}$, thus confirming that the maximum number of iterations is rarely reached (but it

determines the maximum latency in the sense specified above).

4.1 LDPC (*128, 64*) code

The CER curves for the LDPC(128, 64) code are shown in Fig. 3 in the ideal case in which all involved soft reliability values are unquantized, i.e., are represented with a very high precision (e.g., 32-bit floating point for each reliability value). The best performance is exhibited by the MRB algorithm with order $i = 4$, either employed alone or in hybrid form. For the latter, we observe that performance is practically independent of the adopted IA; in fact, the CER curves by using SPA + MRB(4) or NMS + MRB(4) are superposed in the region of interest. The use of MRB allows to achieve a gain in the order of 1.6 dB at CER $= 10^{-5}$, over the IAs. Moreover, the gap to the union bound is very limited, in the order of 0.5 dB. The union bound has been plotted making use of (1) and (2) (i.e., including the terms up to weight 24).

To confirm the goodness of the hybrid algorithm, we have realized a further experiment, by simulating the performance of a mixed (purely theoretical and impractical) decoder that feeds the received sequence to the inputs of both decoders but keeps only the output of the successful decoder. We have verified that the CER performance of this "ideal" decoder is practically coincident with that of the "real" hybrid decoder over the whole range of error rate values of interest.

To analyze the impact of quantization on the performance curves shown in Fig. 3, two quantization rules have been adopted, namely, a linear rule and a logarithmic rule. The input/output characteristics of these two quantizers are shown, with an example, in Fig. 4a, b, respectively.

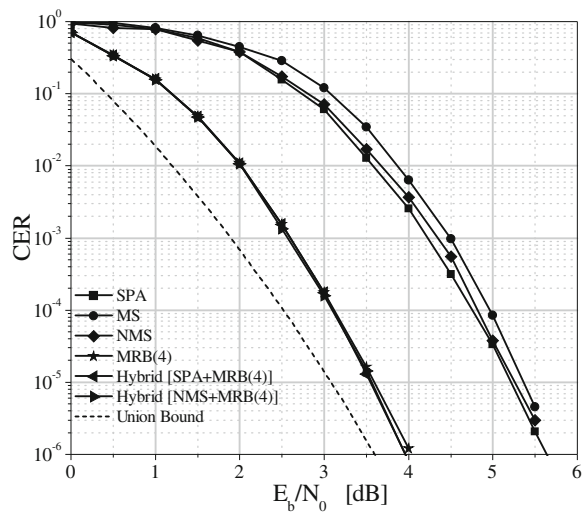

Fig. 3 Performance comparison among the considered decoding algorithms for the LDPC(128, 64) code: unquantized case

While the linear quantization law is obvious, some explanations and comments about the logarithmic one are reported in the Appendix. Further details can be found in [27]. Due to the large number of possible combinations of IA, quantization law, and number of quantization bits, we do not report the simulation curves for all considered decoding algorithms.

In Fig. 5, the impact of quantization is shown for $q \in \{4, 5, 6\}$ when using the hybrid algorithm, based on the SPA + MRB combination. The corresponding curves for the NMS algorithm are reported in Fig. 6. Looking at the figures, it is possible to conclude that $q = 6$ is sufficient to reach practically the same CER of the ideal (i.e., unquantized) case. If $q < 6$, the NMS algorithm exhibits a slightly lower sensitivity to quantization than the hybrid algorithm. This reflects into a smaller performance degradation: looking at Fig. 6 only the linear law with $q = 4$ shows a significant loss while, for example, assuming $q = 5$ with the linear law or $q = 4$ with the logarithmic law does not cause any loss. In Fig. 5, in contrast, the same choices yield a limited, yet visible, loss (in the order of 0.15 dB). We can conclude that adopting the logarithmic law instead of the linear one allows, for hybrid decoding, to achieve the same CER but saving one quantization bit.

As mentioned previously, another fundamental issue to be addressed is complexity. Figure 7 shows the average number of operations required by the various decoding algorithms by assuming $q = 6$. These curves have been obtained by using the expressions reported in Section 3.3. For better evidence, the figure illustrates the dependence on both the signal-to-noise ratio E_b/N_0 and the CER. As expected, the MS algorithm exhibits the lowest complexity, followed by the NMS algorithm and the SPA. On the opposite, the MRB algorithm is characterized by the highest complexity.

The hybrid algorithm, which invokes MRB only upon a detected IA failure, has a complexity comparable with that of the plain MRB algorithm in the low E_b/N_0 regime (or, equivalently, in the high CER regime). By increasing E_b/N_0 (or, equivalently, by decreasing the CER), its complexity is gradually reduced, until it becomes comparable with that of the IA alone, in the high E_b/N_0 regime. At CER = 10^{-5}, the average number of operations required by the hybrid algorithm is still about one order of magnitude larger than that of the corresponding IA alone. Such a higher complexity, however, is compensated by the extra-gain ensured by the hybrid approach.

4.2 LDPC (*512, 256*) code

The CER curves for the LDPC(512, 256) code, for unquantized soft reliability values, are reported in Fig. 8. Coherently with the rest of the paper, for the IA decoding algorithms $I_{\max} = 100$ has been set. This is because

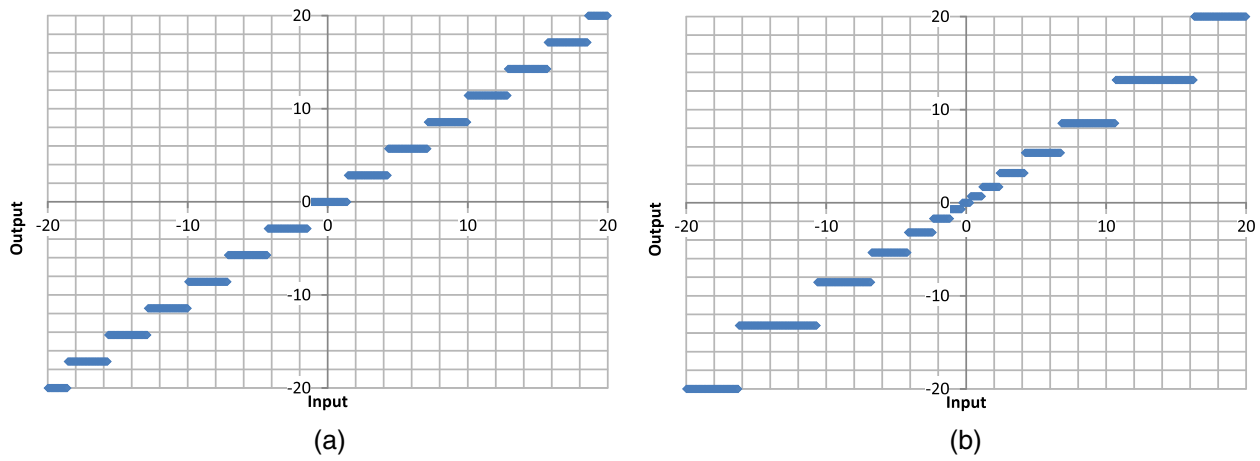

Fig. 4 Examples of **a** linear and **b** logarithmic quantization laws; $q = 4$

such a value gives the best trade-off between performance and complexity. Actually, Fig. 9 shows the impact of this parameter for some values of $I_{\max} \in [100, 500]$, by assuming the SPA for decoding. From the figure, we see that a very limited addional gain (less than 0.1 dB at CER = 10^{-5}) is achieved by moving from $I_{\max} = 100$ to $I_{\max} = 200$. No significant further gain results instead from the adoption of $I_{\max} > 200$. On the other hand, as mentioned above, doubling the maximum number of iterations the latency in the worst case is doubled as well and, taking into account typical design constraints (see Section 5 for details), the disadvantage is more significant than the advantage.

As regards the MRB algorithm, the simulations have been performed with an order $i = 3$, when the algorithm is used alone, as a larger order becomes too complex to manage for $k = 256$ and $n = 512$. This limited MRB order has a negative impact on the error rate performance, which turns out to be unsatisfactory. On the contrary, when the MRB algorithm is employed in the framework of the hybrid decoder, it is possible to increase its order up to $i = 4$ with beneficial effects in terms of CER. Consequently, the performance of the hybrid decoder (here implemented with the SPA) reveals to be the best one, although the gain over the SPA or the NMS algorithms used alone is much less remarkable than for the short code—being now in the order of 0.15 dB.

Similarly to Figs. 5 and 6, in Figs. 10 and 11, the CER curves for the LDPC(512, 256) code are shown for a finite number of quantization bits. Specifically, Fig. 10 is relevant to the hybrid algorithm and Fig. 11 to the NMS algorithm, respectively. In both figures, results are shown

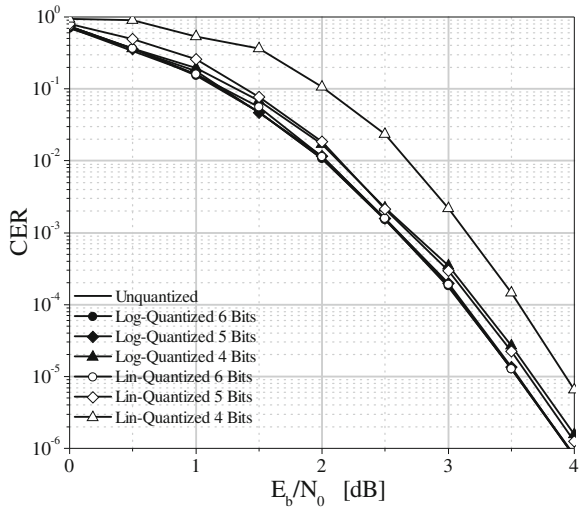

Fig. 5 Impact of quantization on the performance of the hybrid algorithm (SPA + MRB(4)) for the LDPC(128, 64) code

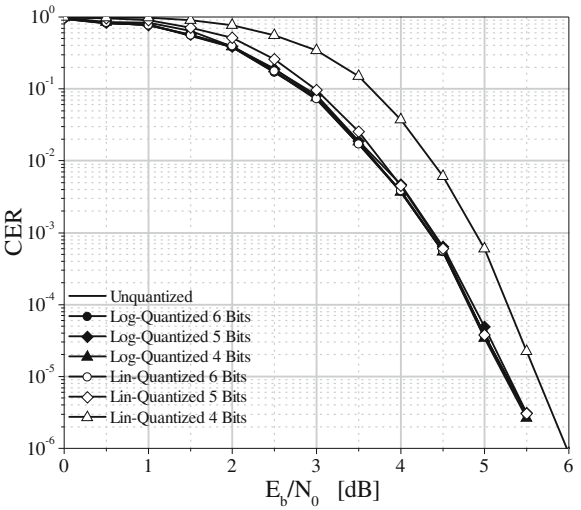

Fig. 6 Impact of quantization on the performance of the NMS algorithm for the LDPC(128, 64) code

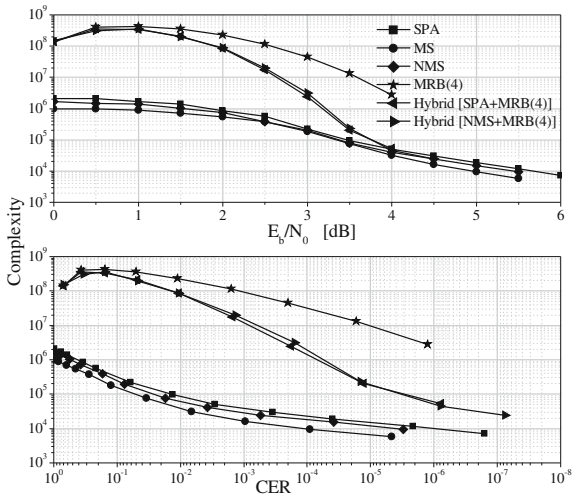

Fig. 7 Average number of binary operations per decoding attempt for the LDPC(128, 64) code in the case $q = 6$

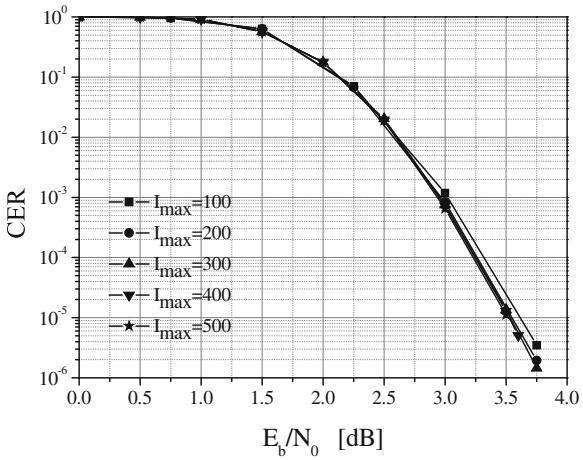

Fig. 9 CER for the LDPC(512, 256) code, with SPA decoding in the unquantized case, by assuming different values of the maximum number of iterations I_{max}

for the two considered quantization laws with $q \in \{4, 5, 6\}$. The conclusions we can draw from this analysis are similar to those valid for the LDPC(128, 64) code. In this case, however, the loss resulting from the adoption of the linear law with $q = 4$ is more pronounced (in the order of 0.8 dB). Hence, using $q = 4$ is not advisable in this case, while $q \geq 5$ is enough to ensure a negligible loss. We also note that the NMS algorithm exhibits a sensitivity on the number of quantization bits that is slightly more evident than for the LDPC(128, 64) code.

The complexity curves for the LDPC(512, 256) code, by assuming $q = 6$, are illustrated in Fig. 12. The gap between the hybrid algorithm and the IAs obviously exist also in this case, but it is relatively less evident. In particular, at CER = 10^{-5}, the hybrid algorithm and the SPA need almost the same average number of binary operations. At low E_b/N_0 (or, equivalently, high CER), the MRB algorithm used alone is less complex than the hybrid algorithm. This is because, for the LDPC(512, 256) code, the plain MRB decoder uses order 3 while the hybrid decoder, for which the MRB algorithm is invoked more intensively when the channel quality is poor, uses order 4.

5 Impact of limited memory

The results presented in Section 4 have been obtained under the assumption that no constraints are put on the

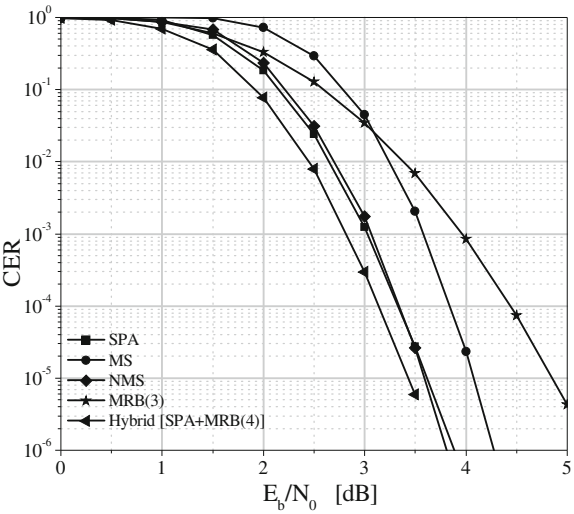

Fig. 8 Performance comparison among the considered decoding algorithms for the LDPC(512, 256) code: unquantized case

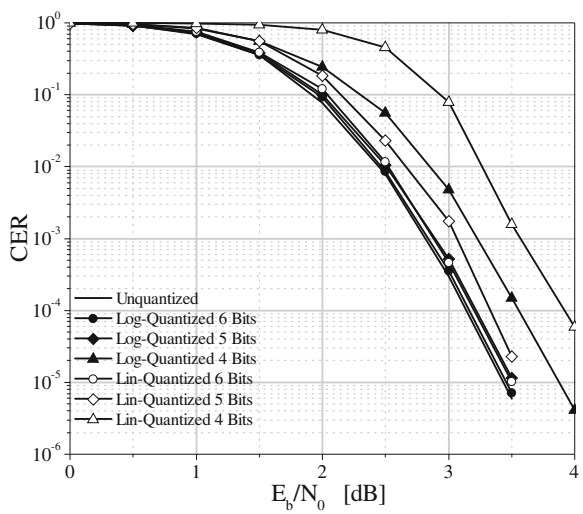

Fig. 10 Impact of quantization on the performance of the hybrid algorithm (SPA + MRB(4)) for the LDPC(512, 256) code

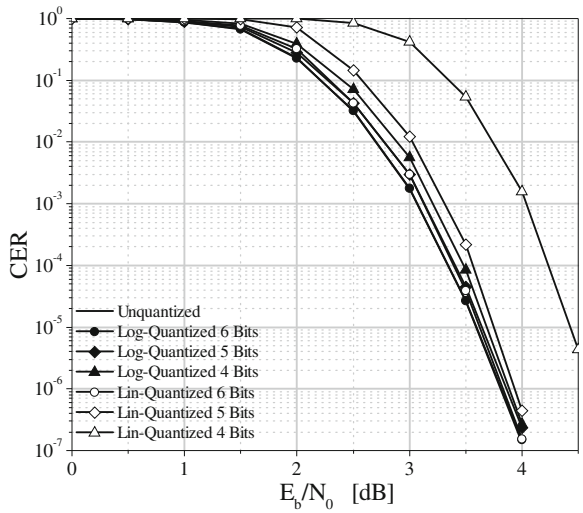

Fig. 11 Impact of quantization on the performance of the NMS algorithm for the LDPC(512, 256) code

O/B available memory. Actually, in the optimal implementation of the MRB algorithm, an ordered TEP list is required. The size of this list can be very large, as it depends on the maximum number of TEPs that for the LDPC(128, 64) code and $i = 4$, for example, results in $N_{\text{TEP}}^{\max} = 679,121$. Such a large number of TEPs requires the availability of more than 2.7 MB of memory (thanks to the sparse character of the TEPs, it is convenient to store only the positions of the set bits). This value may be significantly larger than the memory available for decoding O/B of a spacecraft: looking at recent missions, a typical size of the O/B memory is in fact 0.5 MB. For this reason, it is of paramount importance to investigate the performance degradation resulting from:

- the adoption of a non-ordered list (that occurs when the TEP list is generated on-the-fly, rather than storing it in a memory);
- the adoption of an incomplete list (that occurs when only part of the list can be stored).

In the following, we consider the hybrid decoding algorithm, as it is more suitable (than plain MRB) in view of practical implementation. For the sake of simplicity, the numerical analysis is focused on the LDPC(128, 64) code under the assumption of using an MRB decoder of order $i = 4$. Moreover, according to the sub-optimal mechanisms described in Section 3.1, we use a stopping threshold $A = 24.5$. Such a value of A has been heuristically optimized for the considered code.

As first step of the analysis, we investigate the tradeoff between performance and complexity with and without the ordered TEP list. This is done in Fig. 13, by neglecting the impact of quantization, by considering the NMS + MRB(4) hybrid algorithm, and by assuming that all TEPs can potentially be tested. It is clear from the figure that ordering has no impact on the performance if all the TEPs can be considered as possible candidates. This is because the order of the TEPs in the ordered TEP list turns out to be not substantially different from the order in which they are generated on-the-fly. More precisely, generating the TEPs on-the-fly, they are typically organized in "ascending weight," that is from the smallest weight to the highest weight. Ordering, instead, looks at the TEPs probability to be chosen, organizing them from the highest probability

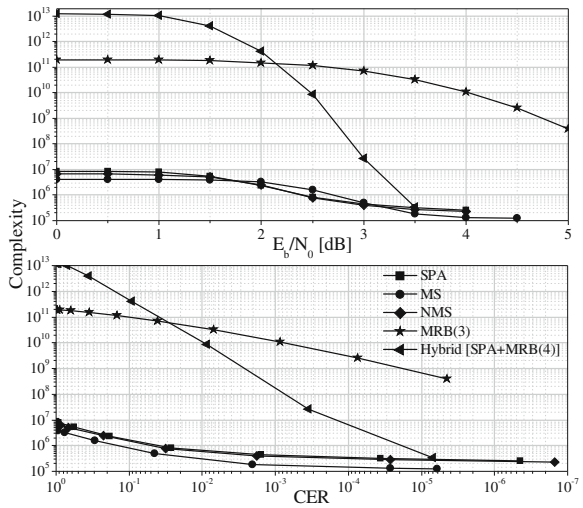

Fig. 12 Average number of binary operations per decoding attempt for the LDPC(512, 256) code in the case $q = 6$

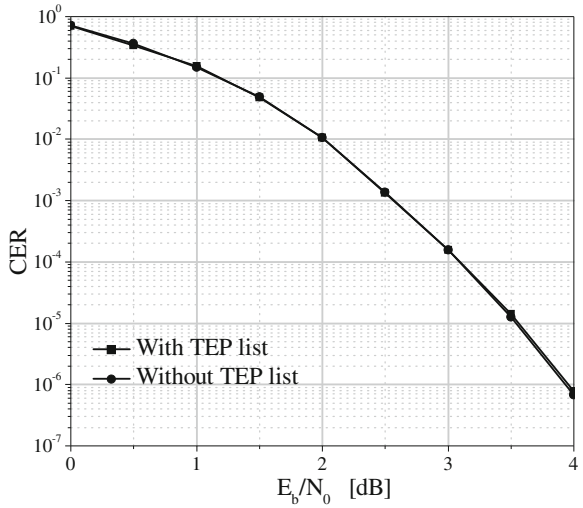

Fig. 13 CER performance of the hybrid NMS + MRB(4) algorithm for the LDPC(128, 64) code with and without an ordered complete TEP list

to the lowest probability. The procedure for doing this is reported in [25].

In practice, ordering can be seen as a perturbation of the ascending weight rule that can cause some TEPs of weight $j + z$, with z integer greater than 0, but with a higher probability, to be processed before some TEPs of weight j. An explicative example is reported below.

Example 1 *In case of $i = 4$, the ordered TEP list is a collection of vectors with length 4, whose nonzero elements represent the positions of the symbols 1 in each TEP. Let us suppose that a portion of the ordered TEP list is*

. . . .
10 0 0 0
63 64 0 0
9 0 0 0
62 64 0 0
62 63 0 0
61 64 0 0
61 63 0 0
60 64 0 0
8 0 0 0
. . . .

We see that patterns of weight 2, that is, with two nonzero coordinates, are intermingled with those of weight 1. For example, the pattern (63 64 0 0) *is considered before the pattern* (9 0 0 0). *Similarly, the weight-1 pattern* (8 0 0 0) *is considered after some weight-2 patterns.*

The result shown in Fig. 13 is not surprising since, when the complete TEP list is considered, the perturbation induced by ordering does not affect the error rate performance (the assumption is that, potentially, all TEPs can be tested both with and without ordering). However, it may have an impact on the complexity. The latter statement is confirmed in Fig. 14, where the curve without TEP list ordering exhibits complexity values higher than those in the presence of ordering.

Related with the complexity issue, it is also interesting to have a preliminary evaluation of the latency due to the MRB decoder. Latency is a measure of the time required for decoding and its value is normally subjected to restrictions, which are expected to be particularly severe when the TC link is used in emergency conditions.

To have a first estimate of the average latency, let us consider (4), which provides the average complexity for the MRB algorithm, and remove from it the parameter q (number of quantization bits). This is because we realistically suppose that, in hardware implementation, during

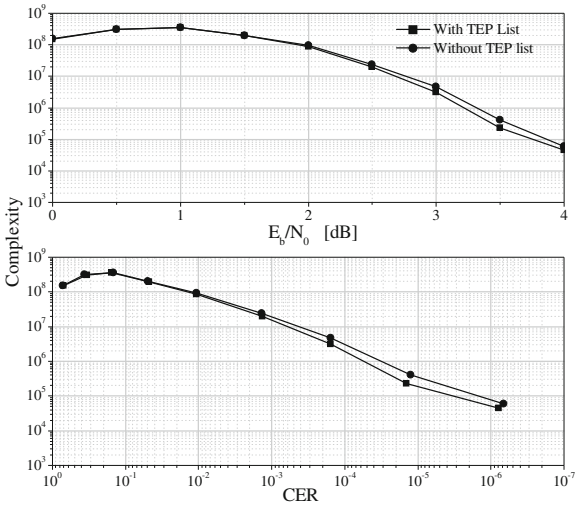

Fig. 14 Average number of binary operations per decoding attempt for the LDPC(128, 64) code in the case $q = 6$ with and without ordered TEP list; the NMS + MRB(4) algorithm is used for decoding

each clock cycle a vector of q bits is simultaneously processed. The first two terms in the resulting equation are due to the received word sorting and $\mathbf{G}^\star$ computing; as such, they are performed only once per decoding attempt. We call these terms the "fixed cost." The third term, instead, depends on the number of TEPs and may be different for each decoding operation. We call this term the "variable cost." As variable cost operations have to be performed for each TEP, they can be parallelized by assuming the availability of a number, noted by N_{Teu}, of different "TEP evaluation units." This way, the average number of vector operations to be performed at each unit results in

$$OP_{\mathrm{eu}} = n\log_2 n + \left(\frac{k}{2}\right)^3 + \left\lceil \frac{N_{\mathrm{TEP}}}{N_{\mathrm{Teu}}} \right\rceil \left(2i + \frac{n-k}{2} + \frac{nk}{2}\right) \tag{9}$$

where $\lceil\cdot\rceil$ represents the ceiling function. Denoting by f_{clock} the clock frequency, the average latency can be estimated as

$$L = \frac{OP_{\mathrm{eu}}}{f_{\mathrm{clock}}}. \tag{10}$$

Examples of average latency estimates are reported in Table 1, assuming $f_{\mathrm{clock}} = 100$ MHz, for three different values of E_b/N_0. From the last column we see that at $E_b/N_0 = 3.5$ dB (which, according to Fig. 5, is sufficient to ensure CER $\approx 10^{-5}$ for the case of $q = 6$), a clock frequency equal to 100 MHz and a number of TEP evaluation units equal to 100 yield an average latency in the order of 3 ms. Note that both these values of f_{clock} and N_{Teu} are

Table 1 Average latency (in seconds) due to the MRB decoding process, by assuming $f_{\text{clock}} = 100$ MHz

N_{Teu}	$E_b/N_0 = 1$ dB	$E_b/N_0 = 2$ dB	$E_b/N_0 = 3.5$ dB
1	4.05	2.21	$2.45 \cdot 10^{-1}$
10	$4.05 \cdot 10^{-1}$	$2.21 \cdot 10^{-1}$	$2.48 \cdot 10^{-2}$
100	$4.09 \cdot 10^{-2}$	$2.24 \cdot 10^{-2}$	$2.77 \cdot 10^{-3}$
1000	$4.41 \cdot 10^{-3}$	$2.54 \cdot 10^{-3}$	$5.60 \cdot 10^{-4}$
10,000	$7.53 \cdot 10^{-4}$	$5.81 \cdot 10^{-4}$	$3.35 \cdot 10^{-4}$

feasible in a field-programmable gate array (FPGA) implementation. In absence of parallelization, i.e., $N_{\text{Teu}} = 1$, the average latency is about two orders of magnitude larger.

As previously pointed out, another case that deserves investigation is the one where we have a reduced number of TEPs. Even in this case, it is interesting to distinguish the case when we use an ordered list stored in memory from the case when the TEPs are progressively generated, starting from the ones with the lowest weight. Figure 15 shows the CER, under hybrid decoding (NMS + MRB(4)), when exploiting the ordering; Fig. 16 illustrates the corresponding results in the absence of ordering. The number of TEPs is assumed to be variable between 10,000 and $N_{\text{TEP}}^{\max} = 679,121$. As expected, while the performance is independent of ordering when the complete list is considered (as shown in Fig. 13) this is no longer true for a reduced list size. More precisely, when an ordered TEP list is adopted, considering 200,000 TEPs is enough to ensure that practically no loss occurs. On the contrary, if the list is non-ordered, the same result is achieved by using 400,000 TEPs or more, whereas if the maximum number of TEPs is set equal to 200,000, there is a loss in the order of 0.25 dB.

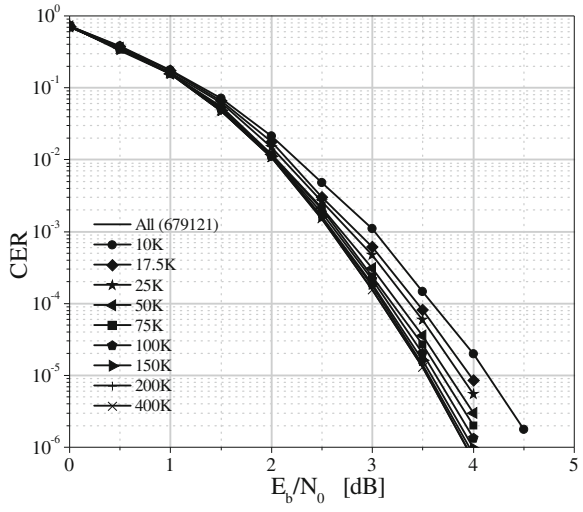

Fig. 15 CER performance of the LDPC(128, 64) code with hybrid decoding for different sizes of the ordered TEP list ($K = 10^3$)

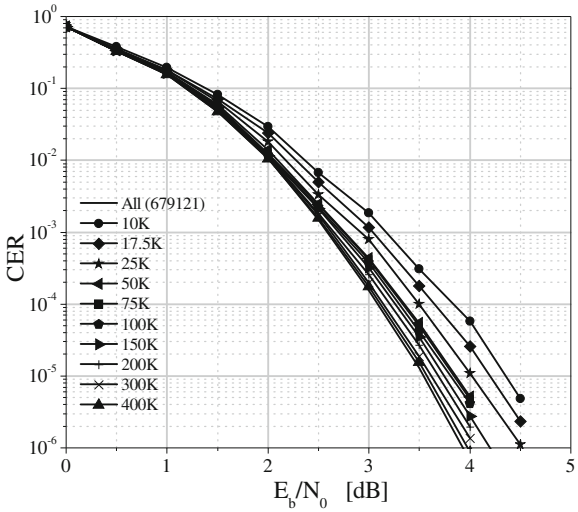

Fig. 16 CER performance of the LDPC(128, 64) code with hybrid decoding for different numbers of non-ordered TEPs ($K = 10^3$)

The maximum number of TEPs can be used, in turn, to estimate the maximum latency (i.e., the latency in the worst case), according to the method described before. In this case, there is no dependence on the E_b/N_0 value, as the worst-case latency occurs when all TEPs are needed for a single decoding operation. Table 2 reports the maximum latency, considering $f_{\text{clock}} = 100$ MHz, for both the cases with and without ordering, by assuming 200,000 TEPs and 400,000 TEPs respectively.

For $N_{\text{Teu}} = 100$, when the list is ordered the latency is at most about 84 ms while, when the list is non-ordered, it is at most about 170 ms. By tolerating a maximum latency of this size, it is possible to generate the TEPs on-the-fly with a minimum impact on the performance.

In Table 2, the worst-case latency has been determined by considering the MRB algorithm only. In the hybrid approach, the latency due to the IA must be taken into account as well. For fixing the ideas, in the following we suppose to apply the SPA but, obviously, the analysis can be repeated for any other IA.

Table 2 Worst-case latency (in seconds) due to the MRB decoding process, with and without ordered list, by assuming $f_{\text{clock}} = 100$ MHz and a reduced number of TEPs

N_{Teu}	Ordered list - 200,000 TEPs	Non-ordered list - 400,000 TEPs
1	8.40	16.79
10	$8.40 \cdot 10^{-1}$	1.68
100	$8.43 \cdot 10^{-2}$	$1.68 \cdot 10^{-1}$
1000	$8.73 \cdot 10^{-3}$	$1.71 \cdot 10^{-2}$
10,000	$1.17 \cdot 10^{-3}$	$2.01 \cdot 10^{-3}$

Combining (9) with (6), where a generic number of iterations I (i.e., not necessarily the average value) is considered and the number q of quantization bits is omitted, for the reasons explained above, the total latency results in

$$L_{\mathrm{TOT}} = L_{\mathrm{IA}} + L_{\mathrm{MRB}} = \frac{1}{f_{\mathrm{clock}}} \{[In\,(9d_v + 12R_c - 11)] + \left[n \log_2 n + \left(\frac{k}{2}\right)^3 + \left\lceil \frac{N_{\mathrm{TEP}}}{N_{\mathrm{Teu}}} \right\rceil \left(2i + \frac{n-k}{2} + \frac{nk}{2}\right) \right] \} \quad (11)$$

where L_{IA} represents the contribution due to the IA and L_{MRB} the contribution due to the MRB algorithm.

Let us denote the worst-case latency by L^{w}_{TOT} and let us suppose that, because of mission constraints, it is forced to not exceed a maximum value $L_{\max}$, i.e.,

$$L^{w}_{\mathrm{TOT}} \leq L_{\max}. \quad (12)$$

When SPA is used alone, (12) is satisfied by assuming the maximum admissible number of iterations, even larger than $I_{\max} = 100$, although in Section 4.2 we have shown this may yield a negligible improvement. For the hybrid algorithm, instead, in order to satisfy (12), besides I we can choose the value of N_{TEP} as well as the parallelization parameter N_{Teu}. A numerical example is reported next.

Example 2 *Let us assume* $L_{\max} = 10^{-2}$ *seconds and* $N_{Teu} = 100$*. Figure 17 shows the trade-off between I and* N_{TEP} *which allows to have* $L^{w}_{TOT} \approx L_{\max}$*. The maximum admissible number of iterations, for the assigned value of* $L_{\max}$*, is* $I = 244$ *which corresponds to apply only the IA, since no margin exists for invoking MRB. According to* (11)*, Fig. 17 can be used also for different* N_{Teu}*'s by properly scaling the values of* N_{TEP}*.*

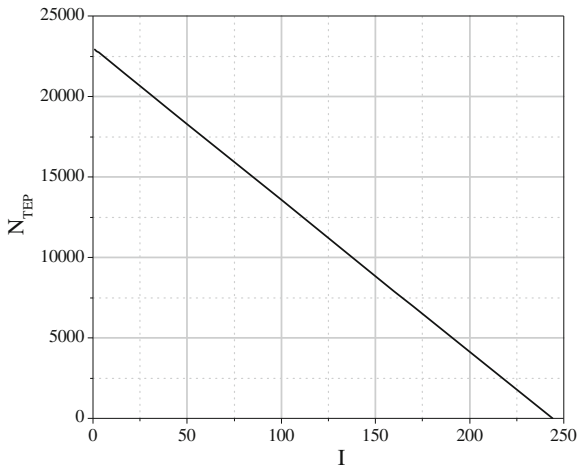

Fig. 17 Combinations $I - N_{TEP}$ which allows to have $L^{w}_{\mathrm{TOT}} \approx 10^{-2}$ seconds

The degrees of freedom offered by the choice of I and N_{TEP}, for a given value of N_{Teu}, can be used to optimize the error rate performance. In other words, we can search for the combination which allows obtaining the minimum value of E_b/N_0 for a given target CER. An example is reported next.

Example 3 *As in Example 2, let us assume* $L_{\max} = 10^{-2}$ *seconds. Moreover, we fix CER* $= 10^{-5}$*. Table 3 reports the minimum values of* E_b/N_0 *we have found, through a numerical search, for the case of* $N_{Teu} = 1$ *and* $N_{Teu} = 100$*, respectively, together with the corresponding optimal choice of I and* N_{TEP}*. The choice of using SPA alone is also reported for the sake of reference.*

From the table, we see that the usage of the hybrid algorithm is advantageous with respect to the SPA used alone for both the considered cases, with a gain in the order of 0.35 dB when $N_{Teu} = 1$ *and more than 1 dB when* $N_{Teu} = 100$*.*

6 Conclusions

We have investigated the advantages resulting from the application of the MRB algorithm, used alone or in hybrid form, to the new short LDPC codes for TC space links. We have discussed the impact of quantization, showing that rather small numbers of quantization bits are necessary to obtain performance very close to that of the ideal, unquantized case.

Special attention has been devoted to the evaluation of the complexity, expressed as the average number of binary operations required per each decoded codeword. Closed form expressions have been adopted for such a purpose. Our investigation has revealed that, as opposed to the common belief, the hybrid algorithm is a realistic and appealing option, and that it allows achieving, for both codes, the best known performance with an acceptable complexity.

An optimal implementation of the MRB algorithm would require the availability of an ordered TEP list and a rather large memory, which may be unavailable O/B. Therefore, we have also investigated the implications of using a non-ordered list and/or an incomplete list. We have shown that an "on-the-fly" implementation is even possible with limited loss and estimated the required decoding latency.

Table 3 Values of *Eb*/*N*0 required to achieve CER = 10^{-5} for different decoder configurations

	$N_{\mathrm{Teu}} = 1$	$N_{\mathrm{Teu}} = 100$	SPA used alone
E_b/N_0 [dB]	4.75	4.09	5.10
I	170	20	244
N_{TEP}	70	21,125	-

Although the proposed analysis refers to a very specific application, that is, space TC links, many findings are usual in a wider sense, e.g., for the application of short codes to machine-to-machine communication or for ultra-reliable communication with short codes.

Endnote

[1] Since $\mathbf{G}$ is a full rank matrix by definition, its k rows being linearly independent vectors of length $n > k$, the Gauss-Jordan elimination always succeeds. However, in case the k columns of $\mathbf{G}$ corresponding to the initial k most reliable positions are not linearly independent, it may be necessary to replace some of the most reliable bits in $\mathbf{v}^{\star}$ with some other bit outside the initial set.

Appendix

Given a real number x, its logarithmic quantization is performed by using the following rule

$$x_{\mathrm{nuq}} = x_s \frac{\exp\left[Q_u(x', q, T')\right] - 1}{F} \tag{13}$$

where F is the so-called log-quantization factor. In our simulations, we have set $F = 0.67$. Moreover

$$\begin{aligned} x_s &= \mathrm{sign}(x) \\ x' &= \ln(1 + Fx) \\ T' &= \ln(1 + FT) \\ Q_u(x, q, T) &= \begin{cases} -T & \text{if } x \le -T \\ \lfloor \frac{x}{d} + \frac{1}{2} \rfloor d & \text{if } -T < x < -T \\ T & \text{if } x \ge T \end{cases} \end{aligned} \tag{14}$$

being T the clipping threshold (i.e., the maximum value admitted) and d the quantization step. Both T and d are related to the number of quantization bits, q. The main advantage of the logarithmic rule, against the uniform one, is in the fact that the quantization levels are denser for small input values. Since the LLR values close to 0 are responsible for maximum decoder uncertainty, it is evident that reducing the quantization error on these values allows to reduce the decoding errors. Further details can be found in [27].

Acknowledgements

The authors wish to thank Dr. Kenneth Andrews for helpful discussion on the weight spectrum estimation. They also wish to thank Prof. Ioannis Chatzigeorgiou for having suggested the test with the ideal decoder.

This work was supported in part by the European Space Agency (ESA/ ESTEC) under the contract 4000111690/14/NL/FE "Next Generation of Uplink Coding Techniques – NEXCODE". The final goal of the NEXCODE Project is the hardware and software implementation of LDPC decoding schemes for TC.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Ancona, Italy. [2] Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), Parma, Italy. [3] Department of Electrical, Electronic, and Information Engineering "G. Marconi," University of Bologna, Cesena, Italy.

Received: 16 April 2016 Accepted: 9 November 2016
Published online: 29 November 2016

References

1. JL Massey, in *Advanced Methods for Satellite and Deep Space Communications (Ed. J. Hagenauer), Lecture Notes in Control and Information Science*, ed. by J Hagenauer. Deep-space communications and coding: a marriage made in heaven, vol. 182 (Springer, Heidelberg and New York, 1992), pp. 1–17
2. GP Calzolari, M Chiani, F Chiaraluce, R Garello, E Paolini, Channel coding for future space missions: new requirements and trends. Proc. IEEE. **95**(11), 2157–2170 (2007)
3. T de Cola, E Paolini, G Liva, GP Calzolari, Reliability options for data communications in the future deep-space missions. Proc. IEEE. **99**(11), 2056–2074 (2011)
4. F Chiaraluce, in *Proc. 22nd Conference on Software, Telecommunications and Computer Networks (SoftCOM 2014)*. Error correcting codes in telecommand and telemetry for European Space Agency missions: an overview and new perspectives, (Split, 2014)
5. TK Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. (Wiley, Hoboken, 2005)
6. CCSDS, TC synchronization and channel coding. Blue Book. CCSDS 231.0-B-2 (2010)
7. ECSS, Space data links—telecommand protocols, synchronization and channel coding. ECSS-E-ST-50-04C (2008)
8. LR Bahl, J Cocke, F Jelinek, J Raviv, Optimal decoding of linear codes for minimizing symbol error rate. IEEE Trans. Inf. Theory. **IT-20**(2), 284–287 (1974)
9. M Baldi, F Chiaraluce, R Garello, N Maturo, I Aguilar Sanchez, S Cioni, Analysis and performance evaluation of new coding options for space telecommand links—Part I: AWGN channels. Int. J. Sat. Commun. Netw. **33**(6), 509–525 (2015)
10. J Hagenauer, E Offer, L Papke, Iterative decoding of binary block and convolutional codes. IEEE Trans. Inf. Theory. **42**(2), 429–445 (1996)
11. M Fossorier, M Mihaljevic, H Imai, Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. IEEE Trans. Commun. **47**(5), 673–680 (1999)
12. J Chen, MP Fossorier, Near optimum universal belief propagation based decoding of low-density parity check codes. IEEE Trans. Commun. **50**(3), 406–414 (2002)
13. M Fossorier, S Lin, Soft-decision decoding of linear block codes based on ordered statistics. IEEE Trans. Inf. Theory. **41**(5), 1379–1396 (1995)
14. Y Wu, CN Hadjicostis, Soft-decision decoding using ordered recodings on the most reliable basis. IEEE Trans. Inf. Theory. **53**(2), 829–836 (2007)
15. M Baldi, F Chiaraluce, N Maturo, G Liva, E Paolini, A hybrid decoding scheme for short non-binary LDPC codes. IEEE Commun. Lett. **18**(12), 2093–2096 (2014)
16. CCSDS, Short block length LDPC codes for TC synchronization and channel coding. Orange Book CCSDS 231.1–O–1 (2015)
17. M Baldi, M Bianchi, F Chiaraluce, R Garello, I Aguilar Sanchez, S Cioni, in *Proc. IEEE 78th Vehicular Technology Conference (VTC Fall 2013)*. Advanced channel coding for space mission telecommand links, (Las Vegas, 2013)
18. M Baldi, N Maturo, G Ricciutelli, F Chiaraluce, in *IEEE 21st IEEE Symposium on Computer and Communications (ISCC 2016)*. On the error detection capability of combined LDPC and CRC codes for space telecommand transmissions, (Messina, 2016), pp. 1105–1112
19. M Baldi, N Maturo, F Chiaraluce, E Paolini, in *Proc. 6th International Conference on Information and Communication Systems (ICICS 2015)*. On the applicability of the most reliable basis algorithm for LDPC decoding in telecommand links, (Amman, 2015)
20. I Sason, S Shamai, Performance analysis of linear codes under maximum-likelihood decoding: a tutorial. Found. Trends Commun. Inf. Theory. **3**(1–2), 1–225 (2006)

21. GC Clark, JB Cain, *Error Correction Coding for Digital Communications*. (Plenum Press, New York, 1981)
22. K Andrews, Weight enumerators for LDPC codes. CCSDS 2015 spring meetings presentation, Pasadena (2015)
23. D Declercq, M Fossorier, in *Proc. IEEE International Symposium on Information Theory (ISIT 2008)*. Improved impulse method to evaluate the low weight profile of sparse binary linear codes, (Toronto, 2008), pp. 1963–1967
24. X-Y Hu, MPC Fossorier, E Eleftheriou, in *Proc. 2004 IEEE International Conference on Communications (ICC 2004)*. On the computation of the minimum distance of low-density parity-check codes, vol. 2, (Paris, 2004), pp. 767–771
25. A Kabat, F Guilloud, R Pyndiah, in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '07)*. New approach to order statistics decoding of long linear block codes, (Washington, 2007), pp. 1467–1471
26. M Fossorier, Iterative reliability-based decoding of low-density parity check codes. IEEE J. Select. Areas Commun. **19**(5), 908–917 (2001)
27. M Baldi, F Chiaraluce, G Cancellieri, Finite-precision analysis of demappers and decoders for LDPC-coded M-QAM systems. IEEE Trans. Broadcast. **55**(2), 239–250 (2009)