electronics

MDPI

Article

Role of the Product $\lambda'(0)\rho'(1)$ in Determining LDPC Code Performance

Francesca Vatta *,† , Fulvio Babich, Flavio Ellero, Matteo Noschese, Giulia Buttazzoni and Massimiliano Comisso

Department of Engineering and Architecture (DIA), University of Trieste, 34127 Trieste, Italy; babich@units.it (F.B.); flavio.ellero@studenti.units.it (F.E.); matteo.noschese@phd.units.it (M.N.); gbuttazzoni@units.it (G.B.); mcomisso@units.it (M.C.)

* Correspondence: vatta@units.it; Tel.: +39-040-558-3455

† Current address: Via A. Valerio, 10, I-34127 Trieste, Italy.

Received: 31 October 2019; Accepted: 6 December 2019; Published: 10 December 2019

Abstract: The objective of this work is to analyze the importance of the product $\lambda'(0)\rho'(1)$ in determining low density parity check (LDPC) code performance, as far as its influence on the weight distribution function and on the decoding thresholds. This analysis is based on the 2006 paper by Di et al., as far as the weight distribution function is concerned, and on the 2018 paper by Vatta et al., regarding the LDPC decoding thresholds. In particular, the first paper Di et al. analyzed the relation between the above mentioned product and the minimum weight of an ensemble of random LDPC codewords, whereas in the second some analytical upper bounds to the LDPC decoding thresholds were determined. In the present work, besides analyzing the performance of an ensemble of LDPC codes through the outcomes of Di et al.'s 2006 paper, we give the relation between one of the upper bounds found in Vatta et al.'s 2018 paper and the above mentioned product $\lambda'(0)\rho'(1)$, thus showing its role in also determining an upper bound to LDPC decoding thresholds.

Keywords: LDPC codes; threshold; gaussian approximation; density evolution; sum-product algorithm; capacity approximation; upper bound; minimum distance; growth rate

1. Introduction

Low Density Parity Check (LDPC) codes, belonging to the class of block channel codes, were introduced for the first time in the 1960s in Robert Gallager's doctoral thesis [1]. Now they are counted among the most promising and cutting-edge coding techniques in contemporary channel coding. Due to the technological impediments of the time in which they were introduced and of the decoder complexity, these codes were hardly contemplated for about 30 years, with the exception of Tanner's graphical description, introduced in [2], and successively designated as the Tanner graph. Appearing independently of Gallager's work, the authors of [3,4] were the re-inventors of LDPC codes in the mid 1990s. Afterwards, because of their astonishing performance approaching the Shannon limit along with Turbo codes [5], they were rapidly comprehended in current communication standards [6–8].

The asymptotic performance of a block channel code, such as an LDPC code, is determined by its minimum distance, i.e., by its minimum weight codewords and by their multiplicities. These determine its correction capability and its asymptotic performance expressed in terms of bit error rate (BER) and frame error rate (FER) vs. the signal-to-noise ratio (SNR) E_b/N_0 from moderate to high SNRs, i.e., away from capacity. Moreover, as far as LDPC codes are concerned, as first observed in [1], it presents the well-known "threshold phenomenon". In other words, the noise threshold defines a channel noise upper bound under which the lost information probability can be maintained as low as needed.

Since it is extremely difficult to study the performance of a LDPC codes ensemble analytically as the determination of its weight enumerating function is a complicated task, in this work we try to give an insight into this problem analyzing the importance of the product $\lambda'(0)\rho'(1)$ in influencing not only this function, that, as said above, determines the asymptotic BER and FER performance, away from capacity, but also the decoding thresholds, i.e., the performance next to the Shannon limit. Namely, on the basis of the results of [9,10], respectively, we are interested in determining, on one side, the relationship between the above mentioned product and the minimum distance of a LDPC codes ensemble, and, on the other side, the relationship between this product and the noise threshold of a LDPC codes ensemble. The first, namely the minimum distance, is related to LDPC's codewords weight distribution, i.e., to their distance spectrum. The second, namely the noise threshold, is related to the convergence behavior of their iterative decoding algorithm, which is particularly important because LDPC codes, as well as turbo-like codes, are "capacity-achieving" codes.

As far as the first of the above cited papers, namely [9], is concerned, its main result asserts that the minimum distance growth rate of irregular LDPC codes ensembles is only determined by the value of the product $\lambda'(0)\rho'(1)$, a number that is related to the degree distribution polynomial couple $\lambda(x)$ and $\rho(x)$. In particular, if $\lambda'(0)\rho'(1) > 1$, the minimum distance growth with the block length being sublinear, if not, the minimum distance growth with the block length is linear.

As far as the second of the above cited papers, namely Vatta et al.'s 2018 paper [10], is concerned, in this work we derived three computationally light upper bounds to LDPC belief-propagation decoding thresholds, usually determined "exactly" through density evolution, assuming, as in [11], the validity of the Gaussian Approximation (GA) (the employment of which is very convenient when an effective and computationally light method is required. See [10,12–15]), and by using the derivation of [16] (the clarifying vision and interpretation of [11] allows to determine the threshold as the ultimate value guaranteeing the convergence of the recurrent sequence, therein defined, but the authors of [11] did not provide any mathematical method to calculate it. Thus, in [16] we presented a mathematical method, based on the quadratic degeneracy theory, allowing the evaluation of noise thresholds by means of converting a convergence problem into a mathematical analysis problem. Using the algorithm shown in [16], a computationally light approximation of LDPC belief-propagation decoding thresholds, usually determined "exactly" through density evolution, can be obtained, assuming, as in [11], the validity of the GA) to determine the asymptotic analytical performance of $\Delta(s,t)$ defined, e.g., in [6].

In Section 2 we recall LDPC codes graphical representation and, in Section 3, the Gaussian Approximation approach. In Section 4 we review the mathematical method deduced in [16] to explain how the upper bounds of [10] were obtained. Furthermore, to analyze the relationship between the product $\lambda'(0)\rho'(1)$ and the minimum distance of a LDPC codes ensemble, in Section 6 the results of [9] are recalled, and, to analyze the relationship between this product and the noise threshold of a LDPC codes ensemble, in Section 7 we present a Lemma specifying the relationship between the third noise threshold upper bound found in [10] and the product $\lambda'(0)\rho'(1)$. In Section 8, the numeric results of these two analyses are reported. These results are discussed in Section 9 which also will report some simulation results to support the discussion. Finally, Section 10 summarizes the conclusions.

2. Graphical Representation of LDPC Codes

LDPC codes can be represented in a graphical way through a commonly named bipartite graph, commonly known as the Tanner graph [2]. As the trellis diagram gives an efficient graphical representation of a convolutional code behavior, so does the Tanner graph in providing an efficient graphical representation of how a LDPC encoder and decoder work.

The nodes in a Tanner graph are classified in variable nodes and check nodes. In the drawing of the Tanner graph, the following rule must be respected: An edge connects a variable node i to a check node j when the corresponding element h_{ij} in the parity-check matrix H is a 1. From this it

may be deduced that there is a check node for each of the $m = n - k$ check equations, and that there is a variable node for each of the n code bits c_i.

As far as irregular LDPC codes are concerned [4], the parameters d_l and d_r specify the maximum degrees of the nodes distributions in their Tanner graphs, defined, e.g., in [17]. In particular, the polynomial $\lambda(x)$ ($\rho(x)$) defines the variable (check) nodes edge-perspective degree distribution:

$$\lambda(x) = \sum_{i=2}^{d_l} \lambda_i x^{i-1} \tag{1}$$

$$\rho(x) = \sum_{j=2}^{d_r} \rho_j x^{j-1} \tag{2}$$

A LDPC code rate can be expressed as [18]:

$$r(\lambda, \rho) = 1 - \frac{\int_0^1 \rho(x)\mathrm{d}x}{\int_0^1 \lambda(x)\mathrm{d}x} = 1 - \frac{\sum_{j=2}^{d_r} \rho_j / j}{\sum_{j=2}^{d_l} \lambda_j / j} \tag{3}$$

The edge-perspective degree distributions $\lambda(x)$ and $\rho(x)$ are related to the node perspective degree distributions $L(x)$ and $R(x)$ as follows [9]:

$$L(x) = \sum_{i=2}^{d_l} L_i x^i \tag{4}$$

$$R(x) = \sum_{j=2}^{d_r} R_j x^j \tag{5}$$

being,

$$L_j = \frac{\lambda_j}{j \int_0^1 \lambda(x)} = \frac{\lambda_j / j}{\sum_{i=2}^{d_l} \lambda_i / i} \tag{6}$$

$$R_j = \frac{\rho_j}{j \int_0^1 \rho(x)} = \frac{\rho_j / j}{\sum_{i=2}^{d_r} \rho_i / i}. \tag{7}$$

3. Gaussian Approximation

Following the assumptions made in [11], the distributions of the messages involved in the LDPC iterative decoding process under appropriate hypotheses may be approximated as Gaussians. Since it may be observed that to totally specify a Gaussian only its mean and variance are needed, and during the iterative decoding process it is necessary to specify only the means and variances of a generic check node output message, u, and of a generic variable node output message, v. Furthermore, since in [11], assuming the validity of the symmetry condition, the variance σ^2 was shown to be connected to the mean m by the relation $\sigma^2 = 2m$, the means only can be kept in the computation process. The means of u and v have been denoted by $m_u^{(l)}$ and $m_v^{(l)}$ at the l-th iteration, respectively. Furthermore, the log-likelihood ratio (LLR) message u_0 from the channel can be taken as Gaussian with a mean of $m_{u_0} = 2/\sigma_n^2$ and a variance of $4/\sigma_n^2$, being $\sigma_n^2 = N_0/2$ the variance of the channel noise.

The mean of the output of a degree-i variable node at the l^{th} iteration is:

$$m_{v,i}^{(l)} = m_{u_0} + (i-1) m_u^{(l-1)} \tag{8}$$

where m_{u_0} is the mean of u_0 and $m_u^{(l-1)}$ is the mean of u at the $(l-1)$-th iteration.

Defining $\phi(x)$ as in Definition 1 in [11], the update rule for an irregular code becomes:

$$m_{u,j}^{(l)} = \phi^{-1}\left(1 - \left[1 - \sum_{i=2}^{d_l} \lambda_i \phi(m_{v,i}^{(l)})\right]^{j-1}\right). \tag{9}$$

The output of a variable node is specified by its mean $m_u^{(l)}$, that can be calculated as the linear combination of the means $m_{u,j}^{(l)}$:

$$m_u^{(l)} = \sum_{j=2}^{d_r} \rho_j \phi^{-1}\left(1 - \left[1 - \sum_{i=2}^{d_l} \lambda_i \phi\left(m_{u_0} + (i-1)m_u^{(l-1)}\right)\right]^{j-1}\right). \tag{10}$$

Defining $s = m_{u_0}$ and $t_l = m_u^{(l)}$, Equation (10) may be rewritten as:

$$t_l = f(s, t_{l-1}) \tag{11}$$

where the function $f(s,t)$ is expressed as:

$$f_j(s,t) := \phi^{-1}\left(1 - \left[1 - \sum_{i=2}^{d_l} \lambda_i \phi(s + (i-1)t)\right]^{j-1}\right) \tag{12}$$

$$f(s,t) := \sum_{j=2}^{d_r} \rho_j f_j(s,t). \tag{13}$$

4. Mathematical Method of [16]

As noted in [16], the problem of the determination of the convergence of Equation (10) may be solved by converting it to a problem of quadratic degeneracy, the solution of which can be given in charge to a commercial software. If the second partial derivative of $f(s,t)$ with respect to t, $f_{tt}(s,t)$, is $\neq 0$, the problem of the determination of the convergence of Equation (10) becomes the search for the solution of the following system:

$$\begin{cases} f(s,t) = t \\ f_t(s,t) = 1 \end{cases} \tag{14}$$

where $f_t(s,t)$ is the first partial derivative of $f(s,t)$ with respect to t. Its solution is the value $s^* = m_{u_0}^*$, which is the minimum $s = m_{u_0}$ guaranteeing the convergence of Equation (10).

Defining, as in [6],

$$\Delta(s,t) := f(s,t) - t \tag{15}$$

and

$$\Delta_t(s,t) = f_t(s,t) - 1 \tag{16}$$

Equation (14) is given by:

$$\begin{cases} \Delta(s,t) = 0 \\ \Delta_t(s,t) = 0 \end{cases} \tag{17}$$

Its solution (s^*, t^*) gives an estimate of the belief-propagation decoding threshold $\sigma^* := \sqrt{\frac{2}{s^*}}$ that may be determined exactly using density evolution. To solve Equation (17), an invertible approximation of the function $\phi(x)$ is needed (see, e.g., [11,19,20]).

5. Upper Bounds on LDPC Codes Decoding Thresholds

The upper bounds on thresholds have been determined in [10] from the asymptotic performance of Equation (17). The first bound, called s^*_{bound} in [10], was obtained determining an approximation of the function $\phi(x)$ valid for when $x \geq 10$. To obtain the second upper bound, called s^*_{approx} in [10], we have used the approximation Equation (16) of [10] (that was implicitly used in [11]). Finally,

involving the Jensen's inequality to manipulate the second upper bound, the third bound, called s^*_{Jensen} in [10], was determined.

5.1. Upper Bound on LDPC Codes Decoding Thresholds Holding for $i_1 \geq 2$

As far as the above mentioned first upper bound on LDPC codes thresholds (called s^*_{bound} in [10]) is concerned, in [20] the following lemma was proved.

Lemma: Given $\lambda(x)$ of minimum degree $i_1 \geq 2$, and defining $z(s,t)$ as $z(s,t) := \frac{s+(i_1-1)t}{2}$ and A_j as $A_j := \frac{1}{(j-1)^2\lambda_{i_1}^2}$, being $\mathrm{W}(\cdot)$ the Lambert-W function, the following asymptotic approximation holds:

$$f(s,t) = 2\sum_{j=2}^{d_r} \rho_j \mathrm{W}(A_j z(s,t)\, \mathrm{e}^{z(s,t)}) + O(t^{-1}). \tag{18}$$

Recalling that $\frac{\mathrm{dW}(x)}{\mathrm{d}x} = \frac{1}{x+\mathrm{e}^{\mathrm{W}(x)}}$:

$$f_t(s,t) = 2\sum_{j=2}^{d_r} \rho_j \frac{z_t(s,t)\mathrm{e}^{z(s,t)}(1+z(s,t))}{z(s,t)\mathrm{e}^{z(s,t)} + \mathrm{e}^{\mathrm{W}(A_j z(s,t)\mathrm{e}^{z(s,t)}) - \log A_j}}. \tag{19}$$

Applying Equation (14) to Equations (18) and (19) we get:

$$\begin{cases} 2\sum_{j=2}^{d_r} \rho_j \mathrm{W}(A_j z(s,t)\mathrm{e}^{z(s,t)}) = t \\ 2\sum_{j=2}^{d_r} \rho_j \frac{z_t(s,t)\mathrm{e}^{z(s,t)}(1+z(s,t))}{z(s,t)\mathrm{e}^{z(s,t)} + \mathrm{e}^{\mathrm{W}(A_j z(s,t)\mathrm{e}^{z(s,t)}) - \log A_j}} = 1 \end{cases}. \tag{20}$$

and Equation (17) can be rewritten as:

$$\begin{cases} 2\sum_{j=2}^{d_r} \rho_j \mathrm{W}(A_j z(s,t)\mathrm{e}^{z(s,t)}) - t = 0 \\ 2\sum_{j=2}^{d_r} \rho_j \frac{z_t(s,t)\mathrm{e}^{z(s,t)}(1+z(s,t))}{z(s,t)\mathrm{e}^{z(s,t)} + \mathrm{e}^{\mathrm{W}(A_j z(s,t)\mathrm{e}^{z(s,t)}) - \log A_j}} - 1 = 0 \end{cases}. \tag{21}$$

Its solution $(s^*_{\text{bound}}, t^*_{\text{bound}})$ determines the bound $\sigma^*_{\text{bound}} = \sqrt{\frac{2}{s^*_{\text{bound}}}}$, which is valid $\forall i_1$, unlike the other two (s^*_{approx} and s^*_{Jensen}) reported in [10], which hold both for $i_1 = 2$ only.

5.2. Further Upper Bounds on LDPC Codes Decoding Thresholds Holding for $i_1 = 2$

With $a_j := \log A_j = -2\log((j-1)\lambda_{i_1})$, Equation (18) can be rewritten as:

$$f(s,t) = 2\sum_{j=2}^{d_r} \rho_j \mathrm{W}\left(z(s,t)\mathrm{e}^{z(s,t)+a_j}\right) + O(t^{-1}). \tag{22}$$

Assuming that the following simplified approximation holds:

$$\mathrm{W}\left(z(s,t)\mathrm{e}^{z(s,t)+a_j}\right) \simeq \mathrm{W}\left((z(s,t)+a_j)\mathrm{e}^{z(s,t)+a_j}\right), \tag{23}$$

calling $x := z_s(t) + a_j$, and being $W(x\mathrm{e}^x) \equiv x$ for $x > 0$, we find a much simpler asymptotic expression for $f(s,t)$ (simpler than the one of Equation (18)):

$$f(s,t) \simeq 2\sum_{j=2}^{d_r} \rho_j \mathrm{W}\left((z(s,t)+a_j)e^{z(s,t)+a_j}\right) = 2\sum_{j=2}^{d_r} \rho_j (z(s,t)+a_j) = \sum_{j=2}^{d_r} \rho_j (s+(i_1-1)t - 4\log((j-1)\lambda_{i_1})). \tag{24}$$

Thus, the asymptotic performance of $f(s,t)$ is given by:

$$f(s,t) = s + (i_1 - 1)t - 4\log\lambda_{i_1} - 4\sum_{j=2}^{d_r}\rho_j\log(j-1) + O(t^{-1}). \tag{25}$$

This is the same asymptotic expression obtained, following a different derivation, in [11], where the approximation in Equation (23) was implicitly used in the proof of Lemma 3.

Thus, ignoring the $O(t^{-1})$, for large t Equation (17) can be rewritten as:

$$\begin{cases} s + (i_1 - 2)t - 4\log\lambda_{i_1} - 4\sum_{j=2}^{d_r}\rho_j\log(j-1) = 0 \\ i_1 - 2 = 0 \end{cases}. \tag{26}$$

The solution of Equation (26) is:

$$s^*_{\text{approx}} = 4\log\lambda_{i_1} + 4\sum_{j=2}^{d_r}\rho_j\log(j-1) \tag{27}$$

and $\sigma^*_{\text{approx}} = \sqrt{\frac{2}{s^*_{\text{approx}}}}$ gives the second upper bound of [10].

Applying the Jensen's inequality:

$$\prod_{j=2}^{d_r}(j-1)^{\rho_j} \leq \sum_{j=2}^{d_r}\rho_j(j-1) \tag{28}$$

and ignoring the $O(t^{-1})$, for large t Equation (17) can be rewritten as:

$$\begin{cases} s + (i_1 - 2)t - 4\log\lambda_{i_1} - 4\log\left(\sum_{j=2}^{d_r}(j-1)\rho_j\right) = 0 \\ i_1 - 2 = 0 \end{cases} \tag{29}$$

The solution of Equation (29) is:

$$s^*_{\text{Jensen}} = 4\log\lambda_{i_1} + 4\log\left(\sum_{j=2}^{d_r}(j-1)\rho_j\right) \tag{30}$$

Taking $\sigma^*_{\text{Jensen}} = \sqrt{\frac{2}{s^*_{\text{Jensen}}}}$ we obtain the third upper bound of [10]. Since, for the Jensen's inequality for Equation (28):

$$\sum_{j=2}^{d_r}\rho_j\log(j-1) \leq \log\left(\sum_{j=2}^{d_r}(j-1)\rho_j\right) \tag{31}$$

it results $s^*_{\text{approx}} \leq s^*_{\text{Jensen}}$ and thus $\sigma^*_{\text{Jensen}} \leq \sigma^*_{\text{approx}}$, i.e., σ^*_{Jensen} gives a tighter upper bound.

6. Role of the Product $\lambda'(0)\rho'(1)$ in Determining the Weight Distribution of LDPC Codes

In Reference [9] it was demonstrated that $\lambda'(0) = \lambda_2$ occupies a key position in determining LDPC codes performance, both theoretically and in practice. In particular, the minimum distance growth rate, namely, whether it is linear or not, was shown to depend only on the product $\lambda'(0)\rho'(1)$. Namely, if $\lambda'(0)\rho'(1) > 1$, the minimum distance growth rate is sublinear with the block length, otherwise, i.e., if $\lambda'(0)\rho'(1) < 1$, it is linear with the block length.

Moreover, in Reference [9] it was demonstrated that, when $\lambda'(0)\rho'(1) > 1$ (condition examined in this paper), for any integer l such that:

$$l \leq \min\{L_2 n, (1-r)n\} \tag{32}$$

being n the codeword length, the probability that a randomly chosen code $\mathcal{G} \in \mathcal{C}(n, \lambda, \rho)$ has any weight-l codeword is:

$$1 - \Pr(X_l = 0) = 1 - e^{-\frac{(\lambda'(0)\rho'(1))^l}{2l}} + O(n^{-1/3}) \tag{33}$$

From Equations (32) and (33), it may be concluded that both the parameter L_2 and the product $\lambda'(0)\rho'(1)$ have an effect on the weight distribution of a LDPC code.

7. Role of the Product $\lambda'(0)\rho'(1)$ in Determining the Decoding Threshold of LDPC Codes

The bound of Equation (30) is strictly related to the product $\lambda'(0)\rho'(1)$ investigated in this paper, but this relationship was not explicitly put in evidence in [10]. In particular, extending the result of [10], the following Lemma can be easily proven.

Lemma 1. *The above mentioned third bound on threshold in Equation (30) may be rewritten as:*

$$s^*_{\text{Jensen}} = 4\log(\lambda'(0)\rho'(1)) \tag{34}$$

Proof of Lemma 1. Remembering the expression of $\lambda(x)$ and $\rho(x)$ in (1) and (2), respectively, $\lambda'(0)$ may be alternatively expressed as

$$\lambda'(0) = \sum_{i=2}^{d_l} (i-1)\lambda_i x_{x=0}^{i-2} = \lambda_2 \tag{35}$$

and $\rho'(1)$ as,

$$\rho'(1) = \sum_{j=2}^{d_r} (j-1)\rho_j x_{x=1}^{j-2} = \sum_{j=2}^{d_r} (j-1)\rho_j \tag{36}$$

Thus, remembering the expression of s^*_{Jensen} found in Equation (30), which holds for $i_1 = 2$ only, this can be rewritten as:

$$s^*_{\text{Jensen}} = 4\log\lambda_2 + 4\log\Big(\sum_{j=2}^{d_r} (j-1)\rho_j\Big) \tag{37}$$

and thus, given the above mentioned expressions found for $\lambda'(0)$ and $\rho'(1)$, also as:

$$s^*_{\text{Jensen}} = 4\log\lambda'(0) + 4\log\rho'(1) \tag{38}$$

from which we get the result. □

Taking $\sigma^*_{\text{Jensen}} = \sqrt{\frac{2}{s^*_{\text{Jensen}}}}$, the third upper bound of [10] may be rewritten as:

$$\sigma^*_{\text{Jensen}} = \frac{1}{\sqrt{2\log(\lambda'(0)\rho'(1))}} \tag{39}$$

8. Numeric Results

Being $\lambda'(0)$ and $\rho'(1)$ given in Equations (35) and (36), respectively, we obtain:

$$\lambda'(0)\rho'(1) = \lambda_2 \sum_{j=2}^{d_r} (j-1)\rho_j \tag{40}$$

In Tables 1–3 we report $\lambda(x)$ and $\rho(x)$ for the irregular LDPC codes of Table I and II in [18]. For each pair of degree distributions, we also report the values of the product $\lambda'(0)\rho'(1)$ and of the

parameter L_2. Moreover, we give the σ^* values obtained in [18] by applying the density evolution analysis, jointly with the upper bound $\sigma^*_{\text{Jensen}} = \sqrt{2/s^*_{\text{Jensen}}}$ defined in Equation (30).

Table 1. Computed parameters and decoding threshold bound regarding rate-1/2 codes deduced from Table I of [18], with $d_l = 4$, 6, 7, and 8.

d_l	4	6	7	8
λ_2	0.38354	0.33241	0.31570	0.30013
λ_3	0.04237	0.24632	0.41672	0.28395
λ_4	0.57409	0.11014		
λ_5				
λ_6		0.31112		
λ_7			0.43810	
λ_8				0.41592
ρ_5	0.24123			
ρ_6	0.75877	0.76611	0.43810	0.22919
ρ_7		0.23389	0.56190	0.77081
$\lambda'(0)$	0.38354	0.33241	0.31570	0.30013
$\rho'(1)$	4.75877	5.23389	5.56190	5.77081
$\lambda'(0)\rho'(1)$	1.82518	1.73980	1.75589	1.73199
$\int_0^1 \lambda(x)$	0.34942	0.32770	0.35934	0.29671
L_2	0.54883	0.50719	0.43927	0.50577
σ^*	0.9114	0.9304	0.9424	0.9497
σ^*_{Jensen}	0.91160	0.95021	0.94241	0.95409
SNR_{gap}	0.60903	0.24872	0.32032	0.21333
$\text{SNR}_{\text{gap}-\text{spline}}$	0.61687	0.25656	0.32815	0.22117

Table 2. Computed parameters and decoding threshold bound regarding rate-1/2 codes deduced from Table I of [18], with $d_l = 9$, 10, 11, and 12.

d_l	9	10	11	12
λ_2	0.27684	0.25105	0.23882	0.24426
λ_3	0.28342	0.30938	0.29515	0.25907
λ_4		0.00104	0.03261	0.01054
λ_5				0.05510
λ_6				
λ_7				
λ_8				0.01455
λ_9	0.43974			
λ_{10}		0.43853		0.01275
λ_{11}			0.43342	
λ_{12}				0.40373
ρ_6	0.01568			
ρ_7	0.85244	0.63676	0.43011	0.25475
ρ_8	0.13188	0.36324	0.56989	0.73438
ρ_9				0.01087
$\lambda'(0)$	0.27684	0.25105	0.23882	0.24426
$\rho'(1)$	6.11620	6.36324	6.56989	6.75612
$\lambda'(0)\rho'(1)$	1.69321	1.59749	1.56902	1.65025
$\int_0^1 \lambda(x)$	0.28175	0.27276	0.26535	0.25888
L_2	0.49128	0.46020	0.45001	0.47176
σ^*	0.9540	0.9558	0.9572	0.9580
σ^*_{Jensen}	0.97439	1.03314	1.05356	0.99908
SNR_{gap}	0.03046	0.47807	0.64807	0.18689
$\text{SNR}_{\text{gap}-\text{spline}}$	0.03830	0.47023	0.64023	0.17905

Table 3. Computed parameters and decoding threshold bound regarding rate-1/2 codes deduced from Table II of [18] with $d_l = 15$, 20, 30, and 50.

d_l	**15**	**20**	**30**	**50**
λ_2	0.23802	0.21991	0.19606	0.17120
λ_3	0.20997	0.23328	0.24039	0.21053
λ_4	0.03492	0.02058		0.00273
λ_5	0.12015			
λ_6		0.08543	0.00228	
λ_7	0.01587	0.06540	0.05516	0.00009
λ_8		0.04767	0.16602	0.15269
λ_9		0.01912	0.04088	0.09227
λ_{10}			0.01064	0.02802
λ_{14}	0.00480			
λ_{15}	0.37627			0.01206
λ_{19}		0.08064		
λ_{20}		0.22798		
λ_{28}			0.00221	
λ_{30}			0.28636	0.07212
λ_{50}				0.25830
ρ_8	0.98013	0.64854	0.00749	
ρ_9	0.01987	0.34747	0.99101	0.33620
ρ_{10}		0.00399	0.00150	0.08883
ρ_{11}				0.57497
$\lambda'(0)$	0.23802	0.21991	0.19606	0.17120
$\rho'(1)$	7.01987	7.35545	7.99401	9.23877
$\lambda'(0)\rho'(1)$	1.67087	1.61754	1.56731	1.58168
$\int_0^1 \lambda(x)$	0.24945	0.24017	0.22240	0.19699
L_2	0.47708	0.45783	0.44078	0.43454
σ^*	0.9622	0.9649	0.9690	0.9718
σ^*_{Jensen}	0.98692	1.01966	1.05485	1.04429
SNR_{gap}	0.08052	0.36399	0.65870	0.57131
$\text{SNR}_{\text{gap}-\text{spline}}$	0.07268	0.35615	0.65086	0.56347

Besides these results, we also reported the SNR_{gap} and $\text{SNR}_{\text{gap}-\text{spline}}$ values in dB, given by:

$$\text{SNR}_{\text{gap}}(d_l) := \frac{\frac{1}{2(\sigma^*_{\text{Jensen}}(d_l))^2}}{C_1^{-1}(1/2)} \tag{41}$$

$$\text{SNR}_{\text{gap}-\text{spline}}(d_l) := \frac{\frac{1}{2(\sigma^*_{\text{Jensen}}(d_l))^2}}{C^{-1}_{\text{spline}}(1/2)} \tag{42}$$

defining the gap from the Shannon limit of the bounds σ_{Jensen}. In Equations (41) and (42), $C^{-1}(1/2)$ was evaluated using an approximation or a spline interpolation of $C^{-1}(r)$, respectively, both recalled in the Appendix A. Applying the approximation $C_1^{-1}(1/2)$, we get the Shannon limit as $\sigma_{\text{Shannon1}} = 0.977813$, which in good agreement with [21]. Applying the spline interpolation we find $\sigma_{\text{Shannon}-\text{spline}} = 0.978696$ is in good agreement with [18]. In regards to the noise power, the Shannon limit is given by $C_1^{-1}(1/2) = (2\sigma^2_{\text{Shannon1}})^{-1} = 0.522948$ and $C^{-1}_{\text{spline}}(1/2) = (2\sigma^2_{\text{Shannon}-\text{spline}})^{-1} = 0.522005$.

9. Discussion and Simulation Results

Given the pairs $\lambda(x)$ and $\rho(x)$ of Tables 1–3, for each of them, an ensemble with a random rate-1/2 LDPC codes has been generated, and their performance simulated using a customized software built on the basis of [22], assuming a memoryless binary input with an additive white Gaussian noise (BI-AWGN) channel.

Since l in Equation (33) is upper bounded by Equation (32), the discussion of the results may be conducted separately for the following two conditions:

1. $L_2 n > (1-r)n$ and
2. $L_2 n < (1-r)n$.

9.1. Case 1: $L_2 n > (1-r)n$

The rate r of the codes whose pairs $\lambda(x)$ and $\rho(x)$ were listed in Tables 1–3, is 1/2. It follows that, for the examples considered in the paper, $1-r = 1/2$. Thus, the degree distributions fulfilling the condition $L_2 n > (1-r)n$ are those with $L_2 n > n/2$, i.e., $L_2 > 1/2$. This condition is fulfilled by the distributions having $d_l = 4$, 6, and 8 of Table 1. This implicates that $l \leq n/2$ for all these codes, i.e., that l is fixed. Given this fixed l value, the performance of the random selected LDPC codes having degree distributions given in the 2-nd, 3-rd, and last column of Table 1, are characterized by an asymptotic BER performance that depends on the probability of Equation (33) only, i.e., only on the value of the product $\lambda'(0)\rho'(1)$. Taken these three degree distributions, the random selected LDPC code having a pair $\lambda(x)$ and $\rho(x)$ that minimizes Equation (33), and thus is characterized by the best asymptotic BER performance, is the one for which the product $\lambda'(0)\rho'(1)$ is minimum. This is the case of the code ensemble with $d_l = 8$ in Table 1. On the other hand, the random selected LDPC code having a pair $\lambda(x)$ and $\rho(x)$ that maximize Equation (33) is the one for which the product $\lambda'(0)\rho'(1)$ is the maximum. This is the case of the code ensemble with $d_l = 4$ in Table 1. Moreover, the minimization of the product $\lambda'(0)\rho'(1)$ implies, as expected from Equation (39), the maximization of σ^*_{Jensen} and, thus, a minimization of SNR_{gap} and of $\text{SNR}_{\text{gap}-\text{spline}}$.

Figure 1 illustrates the BER performances of two randomly generated codes with pairs $\lambda(x)$ and $\rho(x)$ having $d_l = 4$ and 8, respectively, minimizing and, respectively, maximizing the product $\lambda'(0)\rho'(1)$.

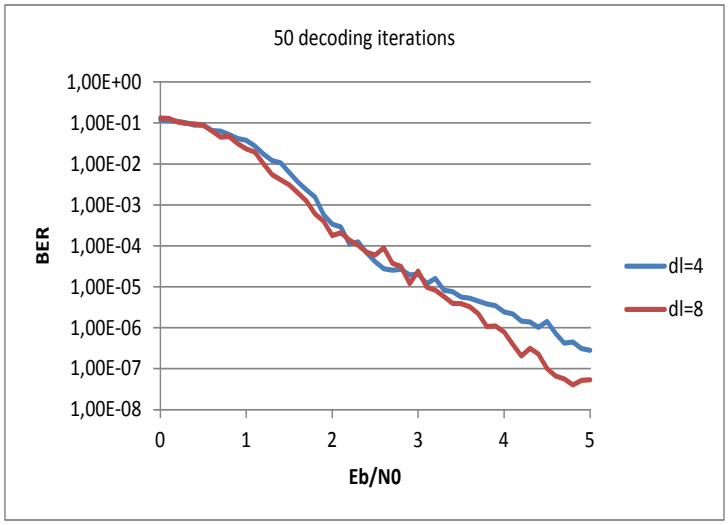

Figure 1. BER with respect to E_b/N_0 in dB with $d_l = 4$ and $d_l = 8$, $n = 1000$, and iterations number $I = 50$.

As shown in the figure above, the simulated performance achieved by the codes having $d_l = 8$ is better than that achieved by the codes having $d_l = 4$. For instance, at $E_b/N_0 = 5$ dB, the BER obtained with $d_l = 4$ is $\sim 3 \times 10^{-7}$, whereas the BER obtained with $d_l = 8$ is $\sim 5 \times 10^{-8}$.

9.2. Case 2: $L_2 n < (1-r)n$

Since, as said above, for the examples considered in the paper, $1-r = 1/2$, it follows that the degree distributions fulfilling the condition $L_2 n < (1-r)n$ are those with $L_2 n < n/2$, i.e., $L_2 < 1/2$.

This condition is fulfilled by the codes with $d_l = 7$ of Table 1, and by all the other codes of Table 2 and 3. This implies that $l \leq L_2 n$ for all these codes, i.e., that, given the block length n, l varies with L_2. In this case, we expect that a higher upper bound of Equation (32) on the codeword weight l will provide in general a better BER performance. Thus, the dependence of this performance from the product $\lambda'(0)\rho'(1)$ must be evaluated case by case. For instance, considering Figures 2–4, where the performances of three randomly generated codes with pairs $\lambda(x)$ and $\rho(x)$ having $d_l = 7, 9$ and 10 are shown respectively, the BER curve obtained with $d_l = 7$ (Figure 2) is the worse of the three (at $E_b/N_0 = 5$ dB the BER is $\sim 2 \times 10^{-7}$) and the one obtained with $d_l = 9$ (Figure 3) is the best of the three (at $E_b/N_0 = 5$ dB the BER is $\sim 2 \times 10^{-8}$), whereas the one obtained with $d_l = 10$ (Figure 4) is intermediate between the above mentioned two (at $E_b/N_0 = 5$ dB the BER is $\sim 4 \times 10^{-8}$) since the code with $d_l = 7$ presents the lowest L_2 value ($L_2 = 0.43927$), that with $d_l = 9$ presents the highest L_2 value ($L_2 = 0.49128$), whereas the one with $d_l = 10$ presents an intermediate L_2 value ($L_2 = 0.46020$) in respect to the other two. As far as the role of the product $\lambda'(0)\rho'(1)$ is concerned, in this case it may be seen that, even if the code with $d_l = 9$ presents a higher value of this product with respect to the code with $d_l = 10$ (1.69321 vs. 1.59749), its performance is better (at $E_b/N_0 = 5$ dB the BER is $\sim 2 \times 10^{-8}$) than that of the code with $d_l = 10$ (presenting at $E_b/N_0 = 5$ dB a BER of $\sim 4 \times 10^{-8}$) because it presents a higher value of L_2.

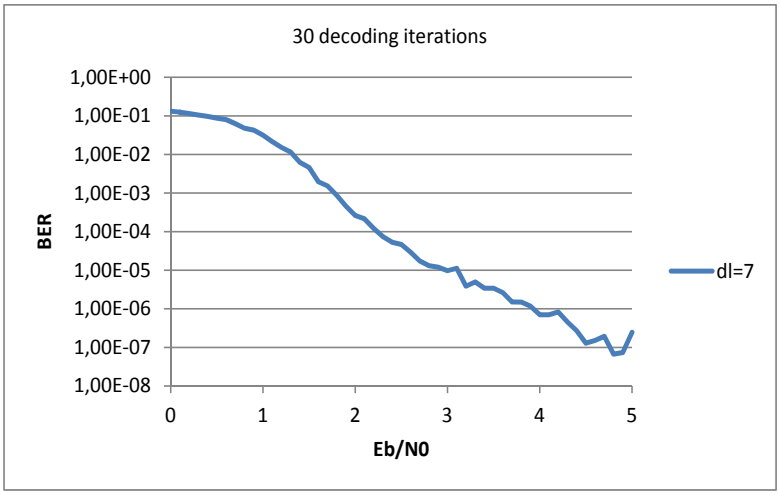

Figure 2. BER with respect to E_b/N_0 in dB with $d_l = 7$, $n = 1000$, and iterations number $I = 30$.

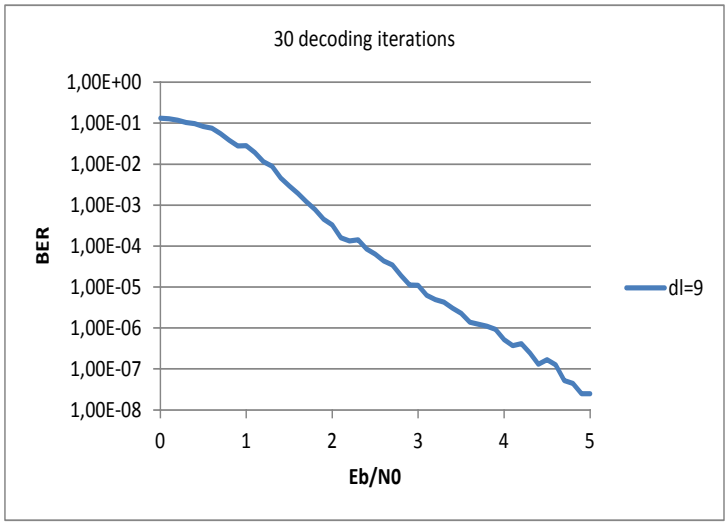

Figure 3. BER with respect to E_b/N_0 in dB with $d_l = 9$, $n = 1000$, and iterations number $I = 30$.

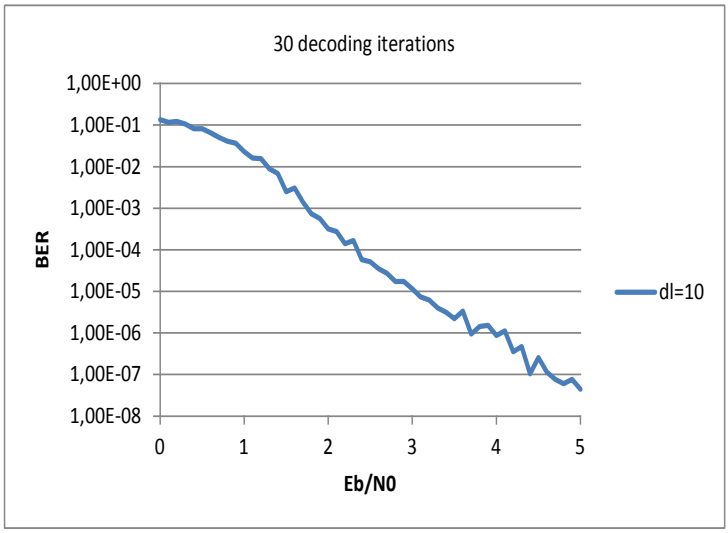

Figure 4. BER with respect to E_b/N_0 in dB with $d_l = 10$, $n = 1000$, and iterations number $I = 30$.

Thus, in conclusion, the randomly generated code having a degree distribution with $d_l = 9$ (Figure 3) is the best since as it has the highest value of L_2 (0.49128) even if it presents an intermediate value of the product $\lambda'(0)\rho'(1)$ (1.69321). Moreover, the code with the lowest value of the product $\lambda'(0)\rho'(1)$ (1.59749), i.e., the one with $d_l = 10$ also presents, as expected from (39), the highest value of σ^*_{Jensen} (1.03314).

10. Conclusions

The objective of this work was the analysis of the importance of the product $\lambda'(0)\rho'(1)$ in determining LDPC codes performance, as far as both the weight distribution function and the decoding thresholds are concerned. This analysis was based on [9], as far as the weight distribution function was concerned, and on [10], regarding the LDPC decoding thresholds. The analysis was conducted on two main conditions, i.e., the one for which $L_2 > (1 - r)$ (being r the code rate) and the one for which $L_2 < (1 - r)$. In the first case, the role of the product alone was fundamental in determining the performance. In the second, parameter L_2 was also important with the product. The best case was that presenting the highest value of L_2 together with the lowest value of the product $\lambda'(0)\rho'(1) > 1$. Moreover, a lower value of the product $\lambda'(0)\rho'(1)$ implied, as expected from (39), a higher value of the upper bound on the decoding threshold σ^*_{Jensen} and thus, in terms of noise power, a smaller gap from the Shannon limit.

Author Contributions: Conceptualization, F.V. and F.B.; methodology, F.V.; software, F.E. and M.N.; validation, F.V., G.B., and M.C.; formal analysis, F.V.; investigation, F.V. and F.B.; resources, F.B. and M.C.; data curation, F.E. and M.N.; writing—original draft preparation, F.V.; writing—review and editing, F.V. and F.B.; visualization, F.V.; supervision, M.C.; project administration, M.C.; funding acquisition, F.V., G.B., and M.C.

Funding: This research was partly funded by the Italian Ministry of University and Research (MIUR) within the project FRA 2018 (University of Trieste, Italy), entitled "UBER-5G: Cubesat 5G networks—Access layer analysis and antenna system development."

Acknowledgments: The authors wish to thank Alessandro Soranzo for his kind support in the formal analysis.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Approximation of the Functions $C(\gamma)$ and $C^{-1}(r)$

We approximated the function $C(\gamma)$, reported in Equation (8) of [16], with Mathematica®, using a numeric integration therein specified.

Here we report 66 (approximate) values ($\times 10^4$) of $C(\gamma)$, corresponding to $\gamma = 0.1, ..., 3.35$ with steps of 0.05. (The first value is 0.1314, the second is 0.1889, and so on.)

1314	5738	7804	8841	9379	9664
1889	5993	7929	8905	9413	9682
2417	6231	8047	8966	9445	9699
2905	6454	8158	9023	9475	9715
3356	6663	8263	9077	9504	9730
3774	6859	8361	9128	9530	9745
4162	7042	8453	9176	9556	9759
4523	7215	8540	9222	9580	9771
4859	7376	8622	9264	9603	9784
5173	7528	8699	9305	9624	9795
5465	7670	8772	9343	9645	9806

Figure A1 reports a graph of $C(\gamma)$ obtained with the same software.

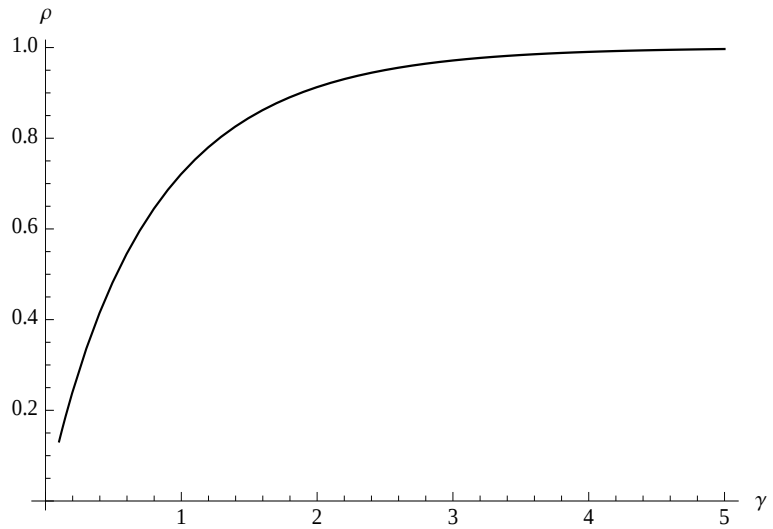

Figure A1. Graph of the function $C(\gamma)$.

To implement an approximation of the inverse $C^{-1}(r)$ we used a spline interpolation of the above approximatively reported values $(\gamma_i, C(\gamma_i)) = (C^{-1}(r_i), r_i)$, with $0.1314 < r_i < 0.9806$:

```
s[y_] = Interpolation[v, y];
```

where `v` is the array of the 66 pairs of values $(C^{-1}(r_i), r_i)$.

Moreover, in [16] we also reported a couple of functions, one inverse of the other, approximating $C(\gamma)$ and its inverse $C^{-1}(r)$, with forms, respectively,

$$C_1(\gamma) = 1 - e^{u\,\gamma^w + v} \qquad C_1^{-1}(r) = \left(\frac{\log(1-r) + v}{u}\right)^{\frac{1}{w}}$$

with

$$u = -1.286 \qquad v = 0.01022 \qquad w = 0.9308.$$

References

1. Gallager, R.G. *Low-Density Parity-Check Codes*; MIT Press: Cambridge, UK, 1963.
2. Tanner, R.M. A recursive approach to low complexity codes. *IEEE Trans. Inf. Theory* **1981**, *5*, 533–547. [CrossRef]

3. Mackay, D.J.C. Good error correcting codes based on very sparse matrices. *IEEE Trans. Inf. Theory* **1999**, *2*, 399–431. [CrossRef]
4. Luby, M.G.; Mitzenmacher, M.; Shokrollahi, M.A.; Spielman, D.A. Improved low-density parity-check codes using irregular graphs. *IEEE Trans. Inf. Theory* **2001**, *2*, 585–598. [CrossRef]
5. Babich, F.; Vatta, F. Turbo Codes Construction for Robust Hybrid Multitransmission Schemes. *J. Commun. Softw. Syst. (JCOMSS)* **2011**, *4*, 128–135. [CrossRef]
6. Vatta, F.; Soranzo, A.; Comisso, M.; Buttazzoni, G.; Babich, F. Performance Study of a Class of Irregular LDPC Codes through Low Complexity Bounds on Their Belief-Propagation Decoding Thresholds. In Proceedings of the 2019 AEIT International Annual Conference, AEIT'19, Florence, Italy, 18–20 September 2019.
7. Babich, F.; Comisso, M. Multi-Packet Communication in Heterogeneous Wireless Networks Adopting Spatial Reuse: Capture Analysis. *IEEE Trans. Wirel. Commun.* **2013**, *10*, 5346–5359. [CrossRef]
8. Babich, F.; D'Orlando, M.; Vatta, F. Distortion Estimation Algorithms for Real-Time Video Streaming: An Application Scenario. In Proceedings of the 2011 International Conference on Software, Telecommunications and Computer Networks, SoftCOM'11, Split, Croatia, 15–17 September 2011.
9. Di, C.; Richardson, T.J.; Urbanke, R. Weight distribution of low-density parity-check codes. *IEEE Trans. Inf. Theory* **2006**, *11*, 4839–4855. [CrossRef]
10. Vatta, F.; Soranzo, A.; Babich, F. Low-Complexity bound on irregular LDPC belief-propagation decoding thresholds using a Gaussian approximation. *Electron. Lett.* **2018**, *17*, 1038–1040. [CrossRef]
11. Chung, S.-Y.; Richardson, T.J.; Urbanke, R. Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation. *IEEE Trans. Inf. Theory* **2001**, *2*, 657–670. [CrossRef]
12. Babich, F.; Noschese, M.; Soranzo, A.; Vatta, F. Low complexity rate compatible puncturing patterns design for LDPC codes. In Proceedings of the 2017 International Conference on Software, Telecommunications and Computer Networks, SoftCOM'17, Split, Croatia, 21–23 September 2017.
13. Babich, F.; Noschese, M.; Vatta, F. Analysis and design of rate compatible LDPC codes. In Proceedings of the 27th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC '16, Valencia, Spain, 4–8 September 2016.
14. Tan, B.S.; Li, K.H.; Teh, K.C. Bit-error rate analysis of low-density parity-check codes with generalised selection combining over a Rayleigh-fading channel using Gaussian approximation. *IET Commun.* **2012**, *1*, 90–96. [CrossRef]
15. Chen, X.; Lau, F.C.M. Optimization of LDPC codes with deterministic UEP properties. *IET Commun.* **2011**, *11*, 1560–1565. [CrossRef]
16. Babich, F.; Soranzo, A.; Vatta, F. Useful mathematical tools for capacity approaching codes design. *IEEE Commun. Lett.* **2017**, *9*, 1949–1952. [CrossRef]
17. Vatta, F.; Babich, F.; Ellero, F.; Noschese, M.; Buttazzoni, G.; Comisso, M. Performance study of a class of irregular LDPC codes based on their weight distribution analysis. In Proceedings of the 2019 International Conference on Software, Telecommunications and Computer Networks, SoftCOM'19, Split, Croatia, 19–21 September 2019.
18. Richardson, T.J.; Shokrollahi, A.; Urbanke, R. Design of capacity- approaching irregular low-density parity-check codes. *IEEE Trans. Inf. Theory* **2001**, *2*, 619–637. [CrossRef]
19. Vatta, F.; Soranzo, A.; Comisso, M.; Buttazzoni, G.; Babich, F. New explicitly invertible approximation of the function involved in LDPC codes density evolution analysis using a Gaussian approximation. *Electron. Lett.* **2019**, *22*, 1183–1186. [CrossRef]
20. Vatta, F.; Soranzo, A.; Babich, F. More accurate analysis of sum-product decoding of LDPC codes using a Gaussian approximation. *IEEE Commun. Lett.* **2019**, *2*, 230–233. [CrossRef]
21. Chung, S.-Y. On the Construction of Some Capacity-Approaching Coding Schemes. Ph.D. Thesis, MIT, Cambridge, MA, USA, 2000. Available online: http://dspace.mit.edu/handle/1721.1/8981 (accessed on 9 December 2019).
22. Boscolo, A.; Vatta, F.; Armani, F.; Viviani, E.; Salvalaggio, D. Physical AWGN channel emulator for Bit Error Rate test of digital baseband communication. *Appl. Mech. Mater.* **2013**, *241–244*, 2491–2495. [CrossRef]