INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2010, VOL. 56, NO. 2, PP. 111-116
Manuscript received November 15, 2009; revised June, 2010. 10.2478/v10177-010-0014-3

Performance of Non-Binary LDPC Codes for Next Generation Mobile Systems

Henryk Gierszal, Witold Hołubowicz, Łukasz Kiedrowski, and Adam Flizikowski

***Abstract*—A new family of non-binary LDPC is presented that are based on a finite field GF(64). They may be successfully implemented in single-carrier and OFDM transmission system. Results prove that DAVINCI codes allow for improving the system performance and may be considered to be applied in the future mobile system.**

***Keywords*—LDPC, mobile systems, modulation schemes, non-binary.**

I. Introduction

LOW DENSITY PARITY CHECK codes (LDPC) are a class of linear error correcting block codes introduced by Gallager in 1962 [7] and rediscovered by MacKay and Neal [12]. Despite their simple construction they have excellent performance. Recent improvements of LDPC have allowed them to surpass the performance of turbo codes. LDPC codes and turbo codes are among the known near Shannon limit codes that can achieve very low bit error rates for low Signal-to-Noise Ratio (SNR) applications [2], [8].

LDPC codes are defined in terms of a sparse parity check matrix, in which most of the entries are zero and only a small fraction are nonzero values. Each codeword satisfies a number of linear constraints and each symbol of the codeword participates in a small number of constraints. For binary LDPC (B LDPC) in the receiver each bit of codeword is connected with a number of parity check nodes. In the case of errorless transmission at the output of each parity check node there are zeros for all codewords.

The constructions, description of an iterative probabilistic decoding algorithm and theory provided by Gallager [7] goes beyond what is known today for turbo codes. Arriving before the computing power that was to prove their effectiveness they were largely forgotten until the rediscovery by MacKay and Neal [19].

Now it is well known that binary low density parity check codes achieve rates close to the channel capacity for very long codeword lengths [2], and more and more LDPC solutions have been proposed in standards (Digital Video Broadcasting (DVB), Worldwide Interoperability for Microwave Access (Wi MAX), etc.). In terms of performance, binary LDPC codes start to show their weaknesses when the codeword length is small or moderate, or when higher order modulation is used for transmission.

For these cases advancements to LDPC codes include improvements in terms of non-binary versions of the codes and codes having variable number of non-zero values in the parity check matrix [4]. The non-binary of LDPC (NB LDPC) codes involved encoding the messages using symbols from a finite field with more than two elements, resulting in each parity check becoming complex but decoding remained tractable.

The main aim of the research done within DAVINCI project is to demonstrate the outstanding performance improvements and industrial feasibility of pioneering non-binary LDPC codes, as well as adapted link level technologies for next generation wireless communications IMT Advanced (International Mobile Telecom-munications), such as IEEE 802.16m or 3GPP LTE-Advanced (Long Term Evolution). Authors applied elaborated DAVINCI codes into a complete telecommunication chain in order to compare their performance in different transmission schemes.

II. DAVINCI Codes

In DAVINCI project three IMT Advanced candidates are examined, i.e., IEEE 802.16e/m, Wireless World Initiative New Radio (WINNER) and LTE. The performance obtained with the DAVINCI air interface, which starts from the common aspects of the air interfaces of the three IMT-candidate schemes, is compared with that obtainable by each candidate system equipped with its own link level technologies and coding schemes. In such way, it will be possible to evaluate the impact of new DAVINCI technologies within the three systems.

A. Encoding

An NB-LDPC code is defined by a very sparse random parity check matrix H, whose components belong to a finite field GF(q) (F_q). The matrix H consists of M rows and N columns; the code rate is defined by $r \leq (N - M) / N$. Decoding algorithms of LDPC codes are iterative message passing decoders based on a Tanner (or factor) graph representation of the matrix H [16]. In general, an LDPC code has a factor graph consisting of N variable nodes (i.e. for codewords) and M parity check nodes with various degrees (i.e. different number of edges in the graph connected to variable nodes or parity check nodes).

Davey and MacKay [5] showed that LDPC codes over GF(q) achieve superior performance to that of binary LDPC codes. In case of non-binary codes the H matrix can take values from the finite field GF(q). Again as for binary LDPC codes, H$\underline{x}$=0 (where $\underline{x}$ is a codeword), but the presence of elements from the GF(q) produces more stringent checks on the codewords of non-binary LDPC codes. Any non-zero value

This paper is prepared within a FP7 project DAVINCI (Design And Versatile Implementation of Nonbinary wireless Communications based on Innovative LDPC codes – INFSO-ICT-216203).

H. Gierszal and W. Hołubowicz are with Adam Mickiewicz University, Poznań, Poland (e-mail: {gierszal; holub}@amu.edu.pl).

Łukasz Kiedrowski is affiliated to ITTI, Poznań, Poland (e-mail: kiedrowski@itti.com.pl).

A. Flizikowski is with University of Technology and Life Sciences in Bydgoszcz, Poland (e-mail: adamfli@utp.edu.pl).

at H(i, j) indicates that there is an edge existing between i-th row (variable node) and j-th column (parity check node).

For NB-LDPC codes there is a weight on each edge of the Tanner graph. This weight is the matrix entry in the parity check matrix and is chosen from the finite field GF(q). So now for nodes defined over GF(q), a parity check m would require a following expression:

$$\sum_{j \in \mathrm{N}(m)} a_{mj} \otimes x_j = 0 \quad (1)$$

where N(m) is the set of variable nodes (representing possible codewords $x_j = [x_0\, x_1\, \dots\, x_{N-1}]_j$) connected to the parity check m and $a_{mj} \in$ GF(q), $a_{mj} \neq 0$. All operations are done in GF(q). MacKay showed that going from binary to non-binary field may reduce the number of cycles in the Tanner graph. This is one of the main reasons why it is expected that LDPC codes over non-binary fields should perform better than LDPC codes over the binary field.

MacKay and Davey [5] also show that the average entropy of messages passed in the graph for non-binary LDPC codes falls faster than the average entropy of messages passed in graph for binary LDPC codes only for certain mean column weight. Although the procedure provided in [5] finds the mean column weight where codes over GF(q) would outperform codes over GF(2) it does not give any insight into the convergence properties of the decoding algorithm.

However moving onto GF(q), $q > 2$, increases the state space of each node in the decoding graph by decoding over GF(q). In other words increasing the field order is comparable to increasing the memory of a convolutional code.

In the framework of the DAVINCI project, ultra-sparse non-binary LDPC codes are used that were designed in a Galois field GF(q) of order $q = 64$, which corresponds to the largest modulation order considered for wireless communications (i.e., 64 QAM — Quadrature Amplitude Modulation) in this project. The non-binary LDPC codes are described by a Tanner graph with regular and constant connection degree, with $d_v = 2$ edges at the variable node side, and varying parity-check connection d_c depending on the desired code rate. To each edge, a non-zero value belonging to the Galois field GF(64) is assigned, in order to define non-binary parity check equations. The choice of the nonzero values is especially important to obtain good performance and requires an optimization strategy. This type of non-binary LDPC code is also referred to as cycle codes and has two main advantages:

- Regular codes with $d_v = 2$ are very sparse and the corresponding Tanner graphs have very large girths compared to usual binary codes graphs. As a consequence, iterative decoders show very good performance, especially at small to moderate codeword lengths. For example, the girth of a binary irregular LDPC code with length $N = 848$ bits and rate $r = 1/2$ is at most $g_b = 6$, while the girth of a NB-LDPC code with same parameters is $g_{nb} = 14$ when a good graph construction is used [9], [17].
- As for the code design, it has been shown in the literature [15] that the finite length optimization of non-binary cycle codes can be decomposed into two steps: (i) first build a Tanner graph with the maximum possible girth and the minimum number of cycles with minimal length, then list all 'short' cycles and the combination of short cycles which define the smallest trapping sets, (ii) optimize iteratively the choice of the non-zero values on the edges of the cycles, such that the local binary minimum distance computed on the set of cycles and trapping sets is maximized. This optimization procedure allows gaining performance both in the waterfall and the error floor region, compared to a random choice of the Tanner graph structure and of the non-zero values assignment.

As for binary decoders, there are two possible representations for messages: probability weights vectors or LDR (Log-Density-Ratios) vectors. The use of the LDR form for messages has been advised by many authors who proposed practical LDPC decoders. The LDR values, which represent real reliability measures on the bits or the symbols are less sensitive to quantization errors due to the finite precision coding of the messages [14]. Also, LDR measures operate in the logarithm domain, which avoids complicated operations like multiplications or divisions.

B. Decoding

Improvement of the coding performance using NB-LDPC is related to a greater computing power needed for the decoding due to increased decoding complexity. As in all practical coding schemes, an important feature is the complexity/performance trade-off, it is very important to try to reduce the decoding complexity of non-binary LDPC codes, especially for high order fields GF(64). The base iterative decoder of non-binary LDPC codes is the Belief Propagation (BP) decoder over the Tanner graph representation of the code. The main difference with the binary BP decoder is that for GF(q) LDPC codes, the messages from variable nodes to check nodes and from check nodes to variable nodes are defined by q probability weights, or q–1 log-density-ratios. As a result, the decoder complexity scales as O(q^2) per check node [21], which is too complex for practical applications.

Computing the check node in the Fourier-domain reduces the complexity of the BP decoder to O(q log q) per check node [1], [9], but adapting the Fourier-domain decoder to practical implementation is tedious due to complicated operators like exponentials or real multiplications.

Recently, sub-optimum decoders based on the generalization of the Min-Sum (MS) decoder have been developed [6], [18]. One of them is Extended-Min-Sum (EMS) algorithm [18] proposed for NB-LDPC codes [18], [19], [20]. A particularity of this algorithm is that it takes into accounts the memory problem of the non-binary LDPC decoders, together with a significant complexity reduction per decoding iteration.

The core idea of the Extended Min-Sum (EMS) decoder is to only use a limited number of LLR (Log-Likelihood Ratio) values $n_m << q$ both for the storage of messages, and for the computation of symbol and check nodes. In the decoder the vector messages of the decoder (represented by LDR vector) is truncated to a limited number n_m of values in order to reduce the memory requirements. This algorithm promises the best complexity/performance trade-off for LDPC codes in high

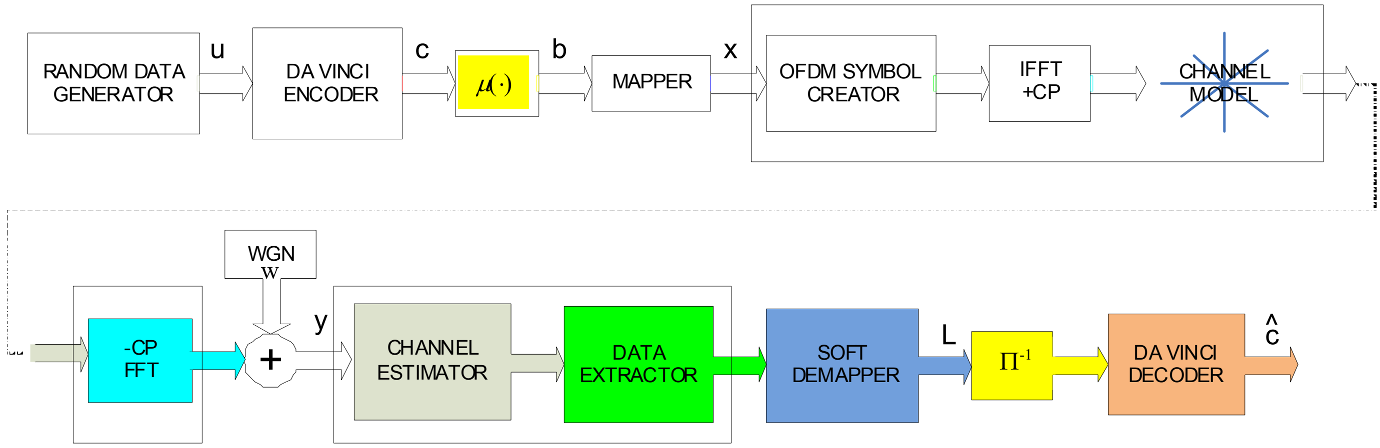

Fig. 1. Transmitter and receiver structure for BICM; CP – Circular Prefix adder/remover; WGN - White Gaussion Noise adder.

order fields, and the complexity scales as $O(n_m \log(n_m))$ with $n_m << q$. The performance loss compared to BP decoding is small (around 0.1 dB) to negligible, depending on the decoder complexity which is tuned by the value of n_m.

The NB-LDPC iterative decoding algorithms are characterized by three main steps corresponding to the different nodes: (i) the variable node update, (ii) the permutation of the messages due to non-zeros values in the matrix H and (iii) the check node update which is the bottleneck of the decoder complexity, since the BP operation at the check node is a convolution of the input messages, which makes the computational complexity grow in $O(q^2)$ with a straightforward implementation.

Although interesting in terms of memory and computation reduction, the truncation of messages from q–1 to n_m values obviously loses potentially valuable information that leads to performance degradation on the error rate curves. This loss of performance could be mitigated by using a proper compensation of the information that has been truncated. Before truncation to n_m entries, which are assumed to be the largest reliability values, the values in a message are sorted in decreasing order. Because the concern is the development of low complexity decoders, a single scalar value γ was chosen to compensate for the q–n_m truncated values. It is the simplest model that one can use.

C. *Modulation for q-ary channel codes*

There are basically two possibilities to combine modulation and coding: coded modulation (e.g. Trellis Coded Modulation (TCM)) and Bit-Interleaved Coded Modulation (BICM). While the former considers both operations jointly and is therefore theoretically superior for many cases, the latter is preferred in nearly all wireless communications systems due to its lower complexity. Additionally, it has been shown that even theoretically, the performance loss of BICM is insignificant [3].

For these reasons and since BICM is also used in all three reference systems, this scheme which is depicted in Fig. 1, is also applied to the q-ary channel codes.

The channel encoder and decoder are described in [11]. The message $\mathbf{u} \in \mathrm{F}_q^K$ is encoded into a codeword $\mathbf{c} = (c_0, c_1, \ldots, c_{N-1}) \in \mathrm{F}_q^K$, which is interleaved and mapped to QAM symbols.

Moreover there are mapping and demapping transformations [10]. The mapping function $\boldsymbol{\mu}(\cdot)$ is responsible for assigning symbols out of a M-QAM constellation A_x to the interleaved code symbols which are taken out of a Galois field of order q. Since the cardinality of both sets is generally not identical, one has to gather m_1 code symbols and map them onto m_2 QAM symbols:

$$\boldsymbol{\mu} : \mathrm{F}_q^{m_1} \rightarrow \mathrm{A}_x^{m_2} \tag{2}$$

In order to have a bijective mapping, the number of elements on both sides must be equal, i.e. m_1 code symbols out of F_q are mapped onto m_2 M-QAM symbols, such that:

$$q^{m_1} = M^{m_2}, \quad m_1, m_2 \in \mathrm{N}$$
$$m_1 = \frac{\operatorname{lcm}(\operatorname{ld} M, \operatorname{ld} q)}{\operatorname{ld} q}, \quad m_2 = \frac{\operatorname{lcm}(\operatorname{ld} M, \operatorname{ld} q)}{\operatorname{ld} M} \tag{3}$$

where $\operatorname{ld}(\cdot)$ is $\log_2(\cdot)$, lcm stands for *least common multiple* and $M = |\mathrm{A}_x|$ is the number of constellation points.

For common values of q and M, this gives the values for (m_1, m_2) as denoted in the following Table I [11].

TABLE I
VALUES FOR M_1 AND M_2 FOR DIFFERENT MODULATIONS AND GALOIS FIELD ORDERS

$M \setminus q$	64	256
2	(1,6)	(1,8)
4	(1,3)	(1,4)
16	(2,3)	(1,2)
64	(1,1)	(3,4)
256	(4,3)	(1,1)

The mapping function hence gathers m_1 code symbols to $\mathbf{b} = (b_0, ..., b_{m_1-1})$ and maps them onto m_2 QAM symbols:

$$\mathbf{x} = (x_0, ..., x_{m_2-1}) = \boldsymbol{\mu}(\mathbf{b}) = (\mu_0(\mathbf{b}), ..., \mu_{m_2-1}(\mathbf{b})) \tag{4}$$

For the following, a mapping from binary vectors to QAM symbols is considered. This is possible because the code symbols $c_i \in F_q$ can be represented by their binary images $(\bar{c}_{i,0}, \bar{c}_{i,1}, ..., \bar{c}_{i,p-1})$ where $p = 1\mathrm{d}q$ and $\bar{c}_{i,j} \in F_2$.

Let $\chi : F_2^{\mathrm{ld}M} \to A_x$ be the mapping function, which assigns to each bit vector of length ld$M$ a QAM symbol out of the constellation A_x.

For each code symbol $b_n \in F_q$, a vector of q APP (A Posteriori Probability) L-values $\mathbf{L}_n = (L_{n,0}, L_{n,1}, \ldots, L_{n,q-1})$ has to be calculated, with:

$$L_{n,k} = \ln \frac{P[b_n = \alpha_k \mid \mathbf{y}]}{P[b_n = \alpha_0 \mid \mathbf{y}]}, \quad \begin{matrix} n \in \{0,1,\ldots,m_1-1\} \\ k \in Z_q \end{matrix} \tag{5}$$

where α_k are the Galois field elements, i.e. $F_q = \{\alpha_0, \alpha_1, \ldots, \alpha_{q-1}\}$, and $Z_q = \{0, 1, \ldots, q-1\}$ is a set of integers from 0 to $q-1$.

Since generally more than one code symbol is involved in the mapping, in order to calculate this vector we require a marginalization:

$$L_{n,k} = \ln \frac{\sum_{\mathbf{b}\in B_n^k} P[\mathbf{b} \mid \mathbf{y}]}{\sum_{\mathbf{b}\in B_n^0} P[\mathbf{b} \mid \mathbf{y}]} = \ln \frac{\sum_{\mathbf{b}\in B_n^k} p(\mathbf{y} \mid \mathbf{b}) P[\mathbf{b}]}{\sum_{\mathbf{b}\in B_n^0} p(\mathbf{y} \mid \mathbf{b}) P[\mathbf{b}]} \tag{6}$$

where $B_n^k = \{\mathbf{b} : b_n = \alpha_k\}$ is the set of all code symbol vectors $\mathbf{b} = (b_0, ..., b_{m_1-1})$ with the n-th component fixed to α_k.

It can assumed that all code vectors are equiprobable, i.e. $P[\mathbf{b}] = q^{-m_1}$, and the channel is memoryless, i.e., $p(\mathbf{y} \mid \mathbf{b}) = \prod_{i=0}^{m_2-1} p(y_i \mid \mathbf{b})$. Then:

$$L_{n,k} = \ln \frac{\sum_{\mathbf{b}\in B_n^k} \prod_{i=0}^{m_2-1} p(y_i \mid \mathbf{b})}{\sum_{\mathbf{b}\in B_n^0} \prod_{i=0}^{m_2-1} p(y_i \mid \mathbf{b})} \tag{7}$$

For a flat fading channel, given by:

$$\begin{aligned} & y_i = h_i \cdot x_i + w_i, i = 0,1,\ldots,m_2-1 \\ & x_i = \mu(\mathbf{b}), w_i \sim \mathrm{CN}(0, N_0) \end{aligned} \tag{8}$$

where h_i is a channel coefficient, and CN(0, N_0) is a noise sample of complex Gaussian distribution with mean equal 0 and variance N_0, there is the conditional probability density function (pdf) of the received signal:

$$p(y_i \mid \mathbf{b}) = p(y_i \mid x_i) = \frac{1}{\pi N_0} \exp\left(-\frac{|y_i - h_i x_i|^2}{N_0} \right) \tag{9}$$

which leads to the LLR vector:

$$\begin{aligned} L_{n,k} &= \ln \frac{\sum_{\mathbf{b}\in B_n^k} \prod_{i=0}^{m_2-1} \exp\left(-\frac{|y_i - h_i \mu(\mathbf{b})|^2}{N_0} \right)}{\sum_{\mathbf{b}\in B_n^0} \prod_{i=0}^{m_2-1} \exp\left(-\frac{|y_i - h_i \mu(\mathbf{b})|^2}{N_0} \right)} = \\ &= \ln \frac{\sum_{\mathbf{b}\in B_n^k} \exp\left(-\sum_{i=0}^{m_2-1} \frac{|y_i - h_i \mu(\mathbf{b})|^2}{N_0} \right)}{\sum_{\mathbf{b}\in B_n^0} \exp\left(-\sum_{i=0}^{m_2-1} \frac{|y_i - h_i \mu(\mathbf{b})|^2}{N_0} \right)} \end{aligned} \tag{10}$$

$$n = 0,1,\ldots,m_1-1; \; k = 0,1,\ldots,q-1$$

The cardinality of the set for the first summation is $|B_n^k| = q^{m_1-1}$. Since the denominator does not depend on k, one can compute only the first term and then normalize such that $L_{n,0} = 0$:

$$\begin{aligned} & L_{n,k} = a_{n,k} - b_n, \\ & a_{n,k} = \ln\left(\sum_{\mathbf{b}\in B_n^k} \exp\left(-\sum_{i=0}^{m_2-1} \frac{|y_i - h_i \cdot \mu_i(\mathbf{b})|^2}{N_0} \right) \right) \end{aligned} \tag{11}$$

In the following, it is assumed that $q = 64$ and $p = \mathrm{ld}\, q = 6$, and let $(\alpha_{k,0}, \alpha_{k,1}, \ldots, \alpha_{k,p-1})$ be the binary image of the GF element α_k.

For QPSK, one GF(64) symbols is mapped onto 3 QPSK symbols, i.e. $m_1 = 1$ and $m_2 = (\mathrm{ld}\, q) / 2 = 3$ and hence the expression (1) simplifies to:

$$a_k = -\frac{1}{N_0} \sum_{i=0}^{m_2-1} |y_i - h_i \mu_i(\alpha_k)|^2 \tag{12}$$

with

$\mu_0(\alpha_k) = \chi(\bar{\alpha}_{k0}\bar{\alpha}_{k1}),\ \mu_1(\alpha_k) = \chi(\bar{\alpha}_{k2}\bar{\alpha}_{k3}),\ \mu_2(\alpha_k) = \chi(\bar{\alpha}_{k4}\bar{\alpha}_{k5}),$

where $(\bar{\alpha}_{k,0}, \ldots, \bar{\alpha}_{k,5})$ is the binary image of the GF(64) symbol α_k.

For 64-QAM modulation, the mapping and demapping functions are especially simple since one code symbol corresponds to one QAM symbol, i.e., $m_1 = m_2 = 1$ and hence the formula (1) simplifies to:

$$a_k = -\frac{1}{N_0} |\, y - h \cdot \mu(\alpha_k) \,|^2 \tag{13}$$

For higher-order modulations, very similar expressions can be derived from the general formula (1). It should be noted here, that so far no efforts have been undertaken to optimize these mappings.

The decoder expects as input parameter a truncated and sorted version of the LLR vector defined above. Given the LLR vector as $\mathbf{a} = (a_0, a_1, \ldots, a_{q-1})$, first these values are sorted in descending order such that:

$$b_k = a_{\pi(k)} \text{ with } b_0 \ge b_1 \ge \cdots \ge b_{q-1} \tag{14}$$

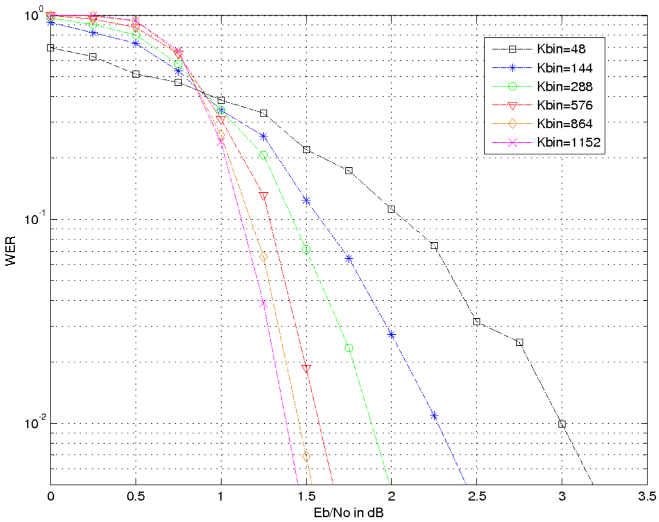

Fig. 2. WER in a function of E_b/N0 for DAVINCI codes of different codeword lengths for BPSK.

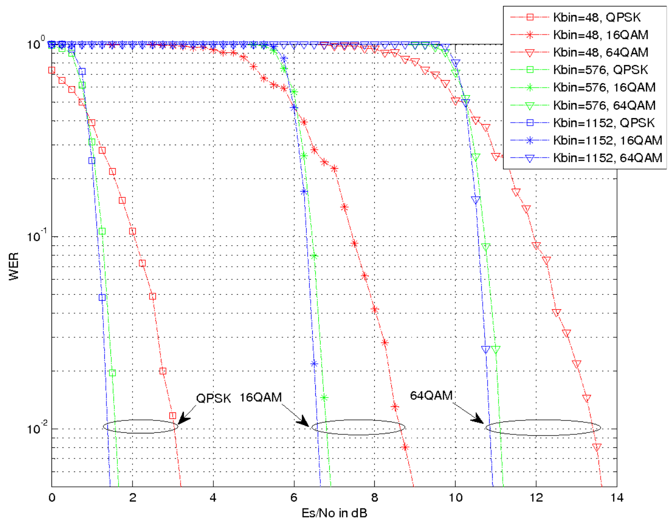

Fig. 3. WER in a function of E_b/N0 for DAVINCI codes of different codeword lengths and modulation schemes in OFDM transmission.

Next, only the greatest $n_m = nbMax$ values are kept and one subtracts a constant such that the last value is zero:

$$\mathbf{b} = \begin{pmatrix} b_0 - b_{n_m-1} \\ b_1 - b_{n_m-1} \\ \vdots \\ 0 \end{pmatrix} \quad (15)$$

There is one such vector for each code symbol, i.e. for each codeword, there are N such vectors. Together with the truncated LLR vector, the permutation ($\pi(0)$, $\pi(1)$, …, $\pi(n_m-1)$) is fed into the decoder.

III. Simulations Results

In this section results of simulation research are presented that allow comparing the performance of DAVINCI codes in a function of their parameters. In the physical layer there are BPSK, QPSK, 16-, or 64-QAM modulation schemes and Additive White Gaussian Noise (AWGN) channel. The maximum number of iterations in the algorithm has been fixed to 30, and a stopping criterion based on the syndrome check is used. A LLR vector is truncated to 16 values. Simulations are done for single-carrier and OFDM transmission mode. The value γ that replaces the truncated ones in the LLR vector was chosen equal to 1.

In Fig. 2 WER (Word Error Rate) is presented for DAVINCI codes of different codeword lengths $K_{\mathrm{bin}} = r{\cdot}N{\cdot}\mathrm{ld}\ q$ and rate $r = 1/2$. The longer codewords, the better performance can be obviously obtained. The gain between the longest and shortest ones is about 1,6 dB at WER = 10^{-2}.

In Fig. 3 the WER of DAVINCI codes for OFDM transmission and different modulation schemes (QPSK, 16-, and 64-QAM) is shown. The relation between codeword lengths remains the same as described in the case of a single-carrier transmission. Gains between K_{bin} = 48 and 1152 for each modulation schemes are listed in Table II. The longer codeword the more steep waterfall region of WER is obtained. In OFDM transmission mode DAVINCI codes are also efficient and may be successfully deployed in future mobile systems.

IV. Conclusions

Channel coding based on NB LDPC becomes an attractive solution for mobile radio systems. They can be seen as a patent-free competitor for turbo codes.

Using the truncated messages, an efficient implementation of the Extended-Min-Sum decoder can be proposed, which starts to be reasonable enough to compete with binary decoders. The performance of the low complexity algorithm with proper compensation is a quite good approach for complexity reduction.

The gain achieved by DAVINCI codes was found to vary with respect to the codeword length, coding rate and modulation order. The gain increases with an increase of the coding rate, or modulation order. On another hand, the gain increases when the codeword length decreases. This leads to the conclusion that DAVINCI codes are very promising solutions to achieve high spectral efficiency even in the challenging scenarios of short codeword lengths.

TABLE II
A Gain for Three Modulation Schemes Due to Increase of Codeword Length by 1104 Bits

Modulation	Gain at WER = 10^{-2}
QPSK	1,5
16-QAM	2,3
64-QAM	2,5

REFERENCES

[1] L. Barnault and D. Declercq, "Fast decoding algorithm for LDPC over GF(2q)," in *Proc. IEEE Information Theory Workshop*, Paris, France, 2003.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE International Conference on Communications*, Geneve, Switzerland, May 1993, pp. 1064–1070.

[3] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation", *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[4] M. C. Davey, "Error-correction using low density parity check codes", Ph.D. thesis, Cambridge.

[5] M. Davey and D. MacKay, "Low density parity check codes over GF (q)," in *Proc. Information Theory Workshop*, Killarney, Ireland, 1998, pp. 70–71.

[6] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Transactions on Communications*, vol. 55, April 2007, pp. 633–643.

[7] R. G. Gallager, "Low density parity check codes," Ph.D. thesis, MIT, Cambridge, Mass., September 1960.

[8] R. G. Gallager, "Low-Density Parity-Check Codes," *IEEE Transactions on Information Theory*, pp. 21–28, Jan. 1962.

[9] X. Y. Hu and E. Eleftheriou, "Binary representation of cycle Tanner-graph GF(2q) codes," in *Proc. IEEE International Conference on Communications*, Paris, France, June 2004, pp. 528–532.

[10] INFSO-ICT-216203 DAVINCI, "D2.2.1 Link Level Evaluation," November 2008.

[11] INFSO-ICT-216203 DA VINCI, "D4.1 Benchmark Non Binary LDPC codes: Report and Software (C-ANSI)", July 2008.

[12] D. J. C. MacKay and R. M. Neal, "Good codes based on very sparse matrices," in *Lecture Notes in Computer Science, Cryptography and Coding*, vol. 1025/1995, Berlin/Heidelberg: Springer, 1995, pp. 100–111.

[13] Madhurima Potluri, "Distributed Source Coding using Non Binary LDPC," M.Sc. thesis, Visweswaraiach Technological University, India, 2005.

[14] L. Ping and W. K. Leung, "Decoding low density parity check codes with finite quantization bits," *IEEE Communications Letters*, vol. 4, no. 2, Feb. 2000, pp. 62–64.

[15] M. F. C. Poulliat and D. Declercq, "Design of regular (2,dc)-LDPC codes over GF(q) using their binary images," *IEEE Transactions on Communications*, vol. 56, October 2008, pp. 1626–1635.

[16] R. M. Tanner, "A Recursive Approach to Low Complexity Codes," *IEEE Transactions on Information Theory*, vol. 27, 1981, pp. 533–547.

[17] D. D. A. Venkiah and C. Poulliat, "Design of cages with a randomized progressive edge growth algorithm," *IEEE Communications Letters*, vol. 12, April 2008, pp. 301–303.

[18] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity decoding algorithm for non-binary LDPC codes in high order fields," *IEEE Transactions on Communications*, 2008.

[19] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low complexity, low memory EMS algorithm for non-binary LDPC codes," in *Proc. International Conference on Communications*, Glasgow, UK, June 2007, pp. 671–676.

[20] A. Voicila, F. Verdier, D. Declercq, M. Fossorier, and P. Urard, "Architecture of a low-complexity non-binary LDPC decoder for high order fields", in *Proc. 7th International Symposium on Communications and Information Technologies*, Sydney, Australia, October 2007.

[21] H. S. H. Wymeersch and M. Moeneclaey, "Log-domain decoding of LDPC codes over GF(q)," in *Proc. IEEE International Conference on Communications*, Paris, France, June 2004, pp. 772–776.