Information Technology and Electrical Engineering - Devices and Systems, Materials and Technologies for the Future

Faculty of Electrical Engineering and Information Technology

Startseite / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=14089
COMPARATIVE STUDY OF A CDMA2000 TURBO CODE
AND A LINEAR TIME ENCODABLE PEG LDPC CODE OVER GF(q)

Wolfgang Proß, Franz Quint
University of Applied Sciences Karlsruhe
Department of Electrical Engineering and Information Technology

ABSTRACT
In this paper we compare a CDMA2000 Turbo code and two linear-time encodable irregular PEG LDPC codes, a binary one and one defined over the Galois field GF(64). An overview over the construction methods of the Shannon limit approaching codes is given. For each code the simulation results are depicted in terms of Bit Error Rate (BER) where several decoding iterations have been applied. Thereby the simulated codes share the relatively short binary code word length of \( n = 1008 \) and a code rate of \( r = 1/2 \). For a channel model the additive white Gaussian noise channel (AWGNC) has been used and the decoding was done with the belief propagation (BP)-decoding algorithm.

Index Terms – Turbo codes, LDPC codes, progressive edge-growth (PEG), GF(q), nonbinary, irregular, linear-time encodable codes, girth

1. INTRODUCTION
In 1948 Claude E. Shannon published *A mathematical theory of communication* [1], where the definition of the Shannon limit stems from. This limit provides a lower bound on the signal to noise ratio. With the introduction of Turbo codes in 1993 by Berrou, Galvieux and Thitimajshima [2] a concatenated coding scheme was proposed that yielded a near Shannon limit decoding. Low-Density Parity-Check (LDPC) codes had already been published in 1962 by Robert Gallager [3] but then have been forgotten for three decades. When they were rediscovered by MacKay and Neal in 1995 [4], a real competitor to Turbo codes arose. This competition has become fierce even for short code word length \( n \), due to various improvements for the LDPC code’s underlying parity-check matrix that have been published since then. Most notably progressive edge growth (PEG) LDPC codes presented by Hu, Eleftheriou and Arnold in [5] appear to be outstanding.

2. SIMULATIONS
The simulations in this paper are executed using the all-zero codeword which is always a valid codeword of any linear code. The codeword is BPSK modulated and then applied to the additive white Gaussian noise channel (AWGNC). Thereby white Gaussian noise gets added depending on the signal to noise ratio (SNR) which in channel coding simulations is usually defined as \( E_b/N_0 \) (energy per bit) divided by \( N_0 \) (spectral noise density). The variance of the Gaussian amplitude distribution is

\[
\sigma^2 = \frac{x_i^2}{10(\text{SNR})^{10}/10} \cdot 2r
\]

and thus is not only dependent on \( E_b/N_0 \) but also on the code rate \( r \). This way it is possible to compare codes that exhibit different code rates. With \( x_i \in \{1,-1\} \) being a sent bit, the conditional probability of a bit \( y_i \) being received by the decoder is then distributed as follows:

\[
p(y_i|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i-x_i)^2}{2\sigma^2}}, \quad 1 \leq i \leq n
\]

For a decoder that processes soft-decision values in the log-domain, a convenient format is the log likelihood ratio (LLR) which for the AWGNC is expressed as

\[
L(y_i|x_i) = \ln \frac{p(y_i|x_i=1)}{p(y_i|x_i=-1)} = \frac{2}{\sigma^2} \cdot y_i
\]

To depict the error correcting capabilities of a code, the bit error ratio (BER) is plotted on the y-axis of a graph while the according \( E_b/N_0 \)-values are plotted on the x-axis. The BER is obtained by dividing all errors occurring in a decoded codeword by the length of the code word \( n \) where \( n = 1008 \) for all simulations throughout this paper.
3. TURBO CODES

Turbo codes in general are defined by a serial or parallel concatenation of several channel-codes and an appropriate decoder that processes soft decisions in an iterative way. Thereby the Turbo decoder constantly refines the decoding result by an exchange of data (called extrinsic data) among the component decoders. The name Turbo code stems from the similarity of the decoder’s functional principle to a turbocharger.

3.1. Turbo Encoder

Here a Turbo encoder of rate $r = 1/2$ (Figure 1) is established as recommended in the CDMA2000 standard [6]. It consists of a parallel concatenation of two 8-state convolutional encoders, where one of these encodes the information sequence $x_{sk}$ (also called the systematic part) and the other one a random-interleaved version $\pi(x_{sk})$ of $x_{sk}$ with $1 \leq k \leq n/2$. Each convolutional encoder is described by the generator polynomial in octal notation:

$$G = \frac{P}{q} = \frac{13}{15} \quad (4)$$

The output of the first component encoder is denoted as $x_{pk}$ (parity part one) and the coded bit of the second one as $x_{p2k}$. By use of an appropriate puncturing pattern $x_{p1k}$ and $x_{p2k}$ get compacted into $x_{pk}$. A Turbo code word then comprises of the systematic part and the packed parity part so that $x_{sp} = \{x_{sk} \ x_{pk}\}$.

3.2. Turbo decoder

The Turbo decoder shown in Figure 2 consists of two BCJR component decoders that exchange extrinsic data in an iterative process. The BCJR decoder was presented in 1974 by Bahl, Cocke, Jelinek, and Raviv [7]. It describes a soft input/soft output decoder that maximizes the a posteriori probability $p(x_t = b|y_t)$ (MAP) with the BPSK modulated bit $b \in \{-1, 1\}$. Using Bayes’ theorem [8] the LLR is derived which in case of the AWGNC is $L(y_t|x_t) = \frac{2}{\sigma^2} \cdot y_t$. Thus the decoder is called a log MAP decoder.

3.3. Turbo code simulations

As mentioned before, the simulation was executing using a code word of length $n = 1008$. The result is seen in Figure 3.

![Figure 3: BER of Turbo code; n=1008; r=1/2](image)

4. LDPC CODES

LDPC codes are asymptotically optimal regarding the Shannon-limit. To achieve this optimum, the code-length has to be very large and thus the decoding complexity gets extremely high.

4.1. Parity-check matrix & Tanner graph

The name of LDPC codes stems from their underlying sparse Parity-Check Matrix (PCM) which exhibits a low density of nonzero elements. In the following example (Figure 4) a PCM is shown together with their corresponding Parity-check equations.
Figure 4: PCM example

With the help of the Gaussian elimination any PCM can be transformed to

\[ H_{m \times n} = \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mn} \end{bmatrix} = \]

\[
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{bmatrix} \rightarrow \begin{bmatrix}
c_1 \rightarrow s_1 + s_2 + s_3 = 0 \\
c_2 \rightarrow s_4 + s_5 + s_6 = 0 \\
c_3 \rightarrow s_7 + s_8 + s_9 = 0 \\
c_4 \rightarrow s_2 + s_3 + s_7 = 0 \\
c_5 \rightarrow s_1 + s_4 + s_6 = 0 \\
c_6 \rightarrow s_2 + s_5 + s_9 = 0 \\
\end{bmatrix}
\]

\[ s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9 \]

Figure 5: Tanner graph example

4.2. Cycles & girth

Any set of consecutive edges that connect node \( x \) with node \( y \) is called a path. If such a path connects a node \( x \) with itself, it is called a cycle of \( x \). In Figure 5 a cycle of length 4 is shown. Local girth \( g_{s_i} \) refers to the length of the shortest cycle a symbol-node \( s_i \) participates in. Global girth \( g \) is defined as \( g = \min \{g_{s_i}\} \) and so equals the length of the shortest cycle in the graph [5]. A low global girth has a harmful impact on the decoding performance which is thus mainly dependent on the construction of the PCM or the Tanner graph. This is the reason for optimizing the construction method in reference to the resulting decoding performance.

4.3. Regular & Irregular LDPC codes

The PCM of regular LDPC codes possesses exactly \( \gamma \) nonzero elements in each column and \( \rho \) in each row and thus all check-nodes and symbol-nodes share the same number of adjacent edges respectively. When Gallager introduced LDPC codes in [3] he also proposed a pseudorandom construction of a regular PCM. Though he suggested avoiding 4-cycles, he did not provide any advice on how to do so. Figure 4 shows the PCM of a \((n, \rho, \gamma) = (9,3,2)\) Gallager code and Figure 5 the appropriate Tanner graph.

In contrast to regular LDPC codes, irregular codes exhibit several row and column weights. They are described through the use of the symbol-node degree distribution

\[ \Lambda(x) = \sum_{i=2}^{\max} \Lambda_i \cdot x^i, \]

where \( d_{\max}^s \) is the maximum number of edges connected to a symbol-node in the graph and \( \Lambda_i \) is the fraction of symbol-nodes connected to \( i \) check-nodes. Since it is a distribution it follows:

\[ \sum_{i=2}^{\max} \Lambda_i = 1. \]

4.4. PEG LDPC codes

In [5] a construction method is described which is based on Tanner graphs and maximizes the global girth. This is done by progressively establishing edges between the symbol- and check-nodes. For each placement of an edge \((s_i, c_j)\), the check-node \( c_j \) to get connected to the current symbol-node \( s_i \) is chosen in a way that the local girth \( g_{s_i} \) is maximized. This way the global girth \( g \) is maximized as well because \( g = \min \{g_{s_i}\} \). There are three different situations when choosing a check-node \( c_j \) in order to establish an edge \((s_i, c_j)\):

1. If it is the first edge to get connected to a symbol-node \( s_i \) choose the check-node having the lowest check-node degree (fewest connected edges) under the current graph settings.
2. If there are still check-nodes that are not already connected to the current graph choose one of them.
3. If neither of the two former cases are true establish a PEG-tree with \( s_i \) as a root of that tree. Then choose a check-node of the bottom-layer.

As an example for a PEG-tree the creation process with symbol-node \( s_1 \) as a root is depicted in Figure 6.
The encoding time of LDPC codes usually increases with \( n^2 \). Hu, Eleftheriou and Arnold also propose an improved version of irregular PEG LDPC codes in [5]. It is based on a partitioning of the PCM so that

\[
H_{m \times n} = [H_{m \times m}^p, H_{m \times (n-m)}^d], \quad \text{with}
\]

\[
H^p = \begin{bmatrix}
1 & h^p_{12} & \cdots & h^p_{1m} \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & h^p_{(m-1)m}
\end{bmatrix}_{m \times m}
\]  

(9)

The parity bits can then be calculated according to

\[
p_i = \sum_{j=1}^{m} h^p_{ij} p_j + \sum_{j=1}^{n-m} h^d_{ij} d_j \pmod{2}
\]

(10)

where \( i = m, m-1, \ldots, 2, 1 \).

4.5. Nonbinary LDPC codes

By an increase of a binary PCM’s column weight, the Hamming weight spectrum and hence the decoding performance gets improved. The drawback is that if the PCM possesses more nonzero entries, the number of cycles increases which results in a degradation of the codes error correction capabilities. When moving to GF(q) the mean column weight increases while the number of cycles in the nonbinary Tanner graph remains the same [10]. The construction methods to attain a nonbinary PCM do not differ from those of binary LDPC codes. But in contrast to the latter, the PCM of nonbinary LDPC codes possesses elements defined over the Galois field \( GF(q) = GF(2^p) \). Thereby the nonzero entries in \( H \) are generated through the use of a primitive polynomial \( p(z) \) where \( p(0) = 0 \). It is also essential to realize calculations required during the decoding process in the Galois field \( GF(q) \). They are based on a polynomial representation of the elements. A \( GF(q) \) symbol is represented by \( p \) bit, whereas the exponents of the corresponding polynomial stand for the indices of the several bits and the coefficients for their value.

4.6. LDPC code simulations

Here a binary irregular PEG LDPC code that is linear-time encodable is simulated. The length of the codeword is \( n = 1008 \) and the symbol-node degree distribution is \( \Lambda(x) = 0.47532x^2 + 0.279537x^3 + 0.0348672x^4 + 0.108891x^5 + 0.101385x^{15} \), which is the same as the one in [5]. The belief propagation (BP) decoder applied for this simulation works in the log domain as is described in [11]. The simulation results can be seen in Figure 7.

Figure 6: PEG-tree for \( s_1 \)

Figure 7: BER of linear-time encodable PEG LDPC code; \( n=1008; r=1/2 \)
For constructing the nonbinary LDPC code, GF(2^6) = GF(64) has been chosen with \( \Lambda(x) = 0.94x^2 + 0.05x^3 + 0.01x^4 \) which is taken from [7]. The code word length is \( n = 168 \) in symbols and \( n \cdot b = 1008 \) in bits. The decoding was done with a FFT-based BP-decoder as described in [12].

**5. COMPARISON**

In Figure 9 the BER of the following codes is showed:
1. CDMA2000 Turbo code, \( n = 1008, r = 1/2, 100 \) decoding iterations
2. Irregular linear-time encodable PEG LDPC code; \( n = 1008, r = 1/2, 100 \) decoding iterations
3. Irregular linear-time encodable PEG LDPC code over GF(64), \( n = 168, n \cdot b = 1008, r = 1/2, 100 \) decoding iterations

As seen in Figure 9 the LDPC code defined over GF(64) clearly beats the Turbo code in terms of BER. Furthermore it is important to point out that LDPC-decoders are fully parallelizable and should therefore offer extremely high-speed applications. Moreover the parity-check matrix (PCM) of LDPC codes can be used to check after each iteration if a valid codeword has been found and thus the number of computations can be strongly reduced. In contrast to Turbo codes the minimum distance is higher and thus the LDPC decoder can detect its own errors.

![Figure 8: BER of linear-time encodable PEG LDPC code defined over GF(64); n=168; r=1/2](image)

**6. CONCLUSION**

The PEG algorithm offers an effective construction method for high girth LDPC codes that are competitive to Turbo codes. Especially when moving to higher order Galois fields GF(q) irregular PEG LDPC codes beat the applied Turbo code even for a short code word length. As a result of this comparison we construct a near Shannon limit coding scheme for 2D-Data Matrix code applications using the explored nonbinary linear-time encoding PEG LDPC code. This leads to better results in terms of BER and computational burden.

**7. ACKNOWLEDGEMENT**

This work is part of the project MERSES and has been supported by the European Union through its European regional development fund (ERDF) and by the German state Baden-Württemberg.


[12] D. Declercq, and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF (q)", *IEEE Transactions on Communications*, vol. 55, no. 4, p. 633.