Low-Density Parity-Check codes
— An introduction —

June 9, 2016

Abstract

Low-density parity-check codes (LDPC codes) are efficient channel coding codes that allow transmission errors to be corrected. They were first described in 1960 by Gallager in his dissertation [Gal63]. These codes did not gain general acceptance, not least because of the computational effort required to calculate them, and they were forgotten again. Prompted by the groundbreaking development of Turbo codes by Berrou, Glavieux and Thitimajshima [Ber93], which allow coding close to the channel capacity predicted by Claude E. Shannon [Sha48], LDPC codes have been rediscovered [Mac96].

An important element for the success of Turbo codes was the use of iterative decoding where information at the output of two decoders is sent as a checkback signal to the input of the other decoder in each case. This type of decoding also provides powerful protection against errors for the LDPC codes.

The basic idea of LDPC codes is described in this article using a simple example. An important aspect here is the way in which iterative decoding works. It is assumed that the reader has a basic understanding of channel coding with linear block codes.

1 General description

We are considering the example of six data bits arranged in a 2×3 block. They are protected horizontally by two and vertically by three parity bits.

d_1	d_2	d_3	p_1
d_4	d_5	d_6	p_2
p_3	p_4	p_5	

The parity bits p_1, p_2, p_3, p_4, and p_5 are calculated according to the following equations

$$\begin{aligned} d_1 \oplus d_2 \oplus d_3 &= p_1 \\ d_4 \oplus d_5 \oplus d_6 &= p_2 \\ d_1 \oplus d_4 &= p_3 \\ d_2 \oplus d_5 &= p_4 \\ d_3 \oplus d_6 &= p_5 \,. \end{aligned} \tag{1}$$

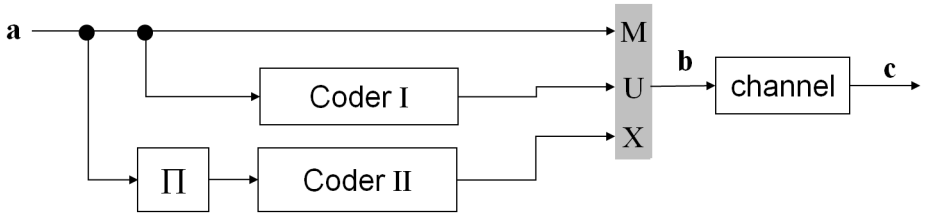

Figure 1: Systematic encoder with two (possibly identical) coders. Coder II processes the data bits in vector **a** in permuted sequence. The channel codeword **b** is produced as interleaving of the original bits and the output bits of the two coders.

The sign $\oplus$ symbolizes the addition modulo 2, i.e. an XOR operation.

Alternatively, you can also say that the six data bits

$$\mathbf{a} = (d_1\ d_2\ d_3\ d_4\ d_5\ d_6)$$

in coder I are protected by two parity bits (p_1, p_2). Parallel to this, the data bits are nested $(d_1, d_4, d_2, d_5, d_3, d_6)$ and protected with coder II by three parity bits (p_1, p_2, p_3), **Figure 1**. A channel codeword is produced as the result

$$\mathbf{b} = (d_1\ d_2\ d_3\ d_4\ d_5\ d_6\ p_1\ p_2\ p_3\ p_4\ p_5)$$

As the data bits can be read directly from the channel codeword, this code is called *systematic*.

With this coding, a constant number of l data bits is supplemented algebraically by a constant number k of parity bits. It is a linear block code and the coding can also be described with a parity-check matrix **H**

$$\mathbf{H} = \left(\mathbf{C}^{\mathrm{T}}|\mathbf{I}\right) = \begin{array}{c} \begin{array}{cccccc|ccccc} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & p_1 & p_2 & p_3 & p_4 & p_5 \end{array} \\ \left(\begin{array}{cccccc|ccccc} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \end{array} . \tag{2}$$

Each column before the vertical separation line corresponds to a data bit and each column after the separation line relates to a parity bit. In this example, the parity-check matrix consists of two connected matrices, $\mathbf{C}^{\mathrm{T}}$ and one identity matrix **I**.

The ones in each row from **H** specify which bits are involved in the corresponding parity equation from (1).

The characteristic of LDPC codes is now that the parity-check matrix **H** contains very few ones with reference to the total size. It is said that the matrix is sparse. This is where the name *low density* comes from. The parity equations (1) then generally contain few elements and it provides an advantage for iterative decoding (see below).

The description of the channel codes is also possible via a (*bipartite graph*), which combines the data bits with the parity bits (**Figure 2**). The upper nodes (*variable nodes*) correspond to the bits that have been transferred. The lower nodes (*check nodes*) combine

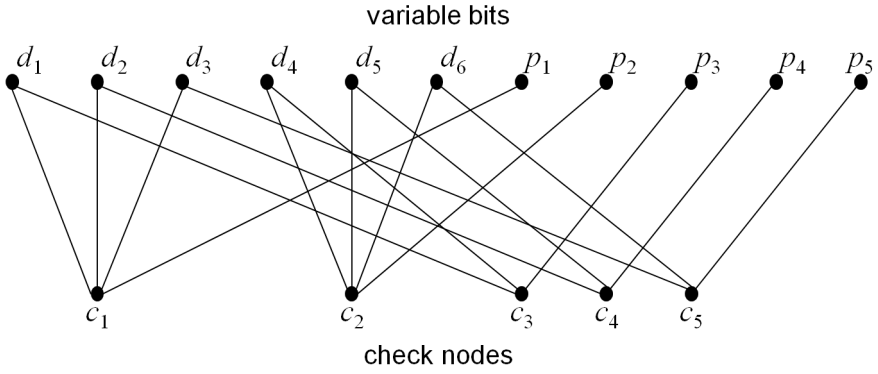

Figure 2: Bipartite graph of the parity-check matrix $\mathbf{H}$

the nodes of those bits whose sum (modulo 2) must be zero. This representation is generally used in conjunction with low-density parity-check codes (LDPC codes) and is called a Tanner graph [Tan81].

However, with LDPC codes it is usual that every node has at least two edges. In the example used above, the nodes of the parity bits only have one edge each, because the channel code is systematic.

As a rule, no distinction is made between data bits and parity bits in connection with LDPC codes. However, it must be ensured that at least one parity bit is involved in each parity equation. The parity bits must appear at least twice in the parity equations in order to generate a Tanner graph with at least two edges per node. However, in the equations (1) this only applies to the data bits. Nevertheless, one could linearly combine two or more parity equations with each other, e.g.

$$\begin{aligned} d_2 \oplus d_3 \oplus d_4 &= p_1 \oplus p_3 \\ d_2 \oplus d_4 \oplus d_6 &= p_2 \oplus p_4 \\ d_1 \oplus d_2 \oplus d_4 \oplus d_5 &= p_3 \oplus p_4 \\ d_2 \oplus d_3 \oplus d_5 \oplus d_6 &= p_4 \oplus p_5 \\ d_1 \oplus d_2 \oplus d_4 \oplus d_5 &= p_1 \oplus p_2 \oplus p_5 \ . \end{aligned} \tag{3}$$

The corresponding graph is somewhat more complex (**Figure 3**) and the parity-check matrix is more densely populated

$$\mathbf{H}_{\text{mod}} = \left(\begin{array}{cccccc|ccccc} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right) \ . \tag{4}$$

From the perspective of the channel coder, the parity-check matrices (2) and (4) are adequate, as both can be transferred into each other through row combination. The channel codewords can be generated with the same generator matrix. With the aid of the Gauß-Jordan algorithm, conversion of (4) into a matrix of a systematic channel code is possible.

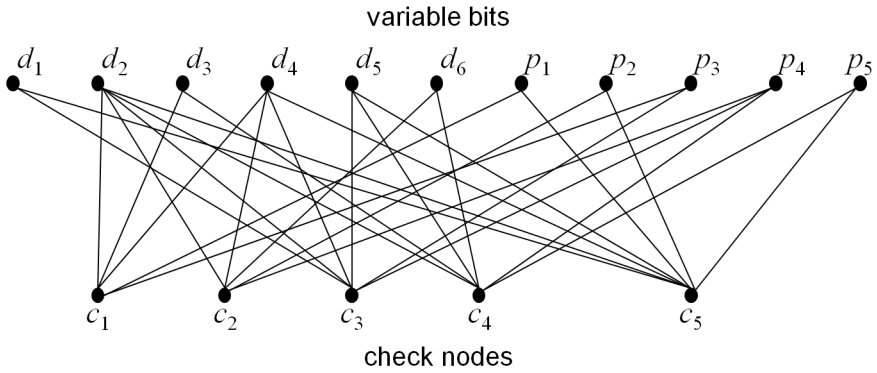

Figure 3: Bipartite graph of the parity-check matrix $\mathbf{H}_{\text{mod}}$

2 Encoding

The (canonical) generator matrix is derived from the parity-check matrix $\mathbf{H}$ in (2)

$$\mathbf{G} = (\mathbf{I}|\mathbf{C}) = \left(\begin{array}{cccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right) . \tag{5}$$

The vector of the channel codeword $\mathbf{b}$ is calculated from the vector of data bits $\mathbf{a}$ and the generator matrix $\mathbf{G}$ in accordance with

$$\mathbf{b} = \mathbf{a} \otimes \mathbf{G} . \tag{6}$$

The symbol $\otimes$ corresponds to a (matrix) multiplication in Modulo 2 arithmetic.

For example:

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Ideally, the generator matrix is also sparse and calculation of the parity bits may be achieved more quickly through special arithmetic than with direct multiplication as per (6).

3 Decoding

3.1 General decoding with hard decision

At the channel output, the (continuous) signal value is interpreted as either a zero bit or a one bit using a single threshold (*hard decision decoding*) and the received codeword $\mathbf{c}$ is compiled from this.

The parity-check matrix $\mathbf{H}$ is required on the receiver side to verify that this received codeword $\mathbf{c}$ is error free. Where there is no transmission error, $\mathbf{c} = \mathbf{b}$ applies and what is termed the *error syndrome*

$$\mathbf{s} = \mathbf{H} \otimes \mathbf{c}^{\mathrm{T}} \tag{7}$$

is equal to a zero vector (all elements of $\mathbf{s}$ are equal to zero.) One could also say that all parity equations (1) are satisfied. Where one or several elements not equal to zero in $\mathbf{s}$ indicate at least one transmission error.

Where hard decision decoding is used, syndrome $\mathbf{s}$ can provide an indication of the position of the bit error, providing that the number of bit errors does not exceed the correction capacity of the channel code used.

As $\mathbf{H}_{\mathrm{mod}}$ is produced from a linear combination of the rows from $\mathbf{H}$, the channel code generated with $\mathbf{G}$ can also be verified with $\mathbf{H}_{\mathrm{mod}}$.

3.2 Iterative decoding

Parallel decoding

In the same way as encoding was in parallel with two coders in the example in section 1, decoding can also be seen as parallel. The parity equations are accordingly divided into two groups. The horizontal line in the following sample parity-check matrix is intended to indicate the separation

$$\mathbf{H} = \left(\begin{array}{cccccc|ccccc} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \begin{array}{l} \text{decoder I} \\ \\ \\ \\ \text{decoder II} \end{array} . \tag{8}$$

The information about each data bit now originates from three sources, if it is a systematic channel code. The (possibly corrupted) data bits may be read from the received codeword. Also, both decoders provide information about the sent data bits. It is worth mentioning at this point that, depending on the number of ones in the columns of the parity-check matrix $\mathbf{H}$, more than two decoders can also be examined.

If no hard decision is made via zero and one at the channel output, but instead the signal level at the output is analysed, iterative decoding proves to be an advantage where information from these three sources is exchanged.

Let us take the above coding example with concrete bit values

$$\left|\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & \end{array}\right| .$$

With modulation of the type $x_i = 1 - 2 \cdot d_i$ is the channel input $\mathbf{x}$, so

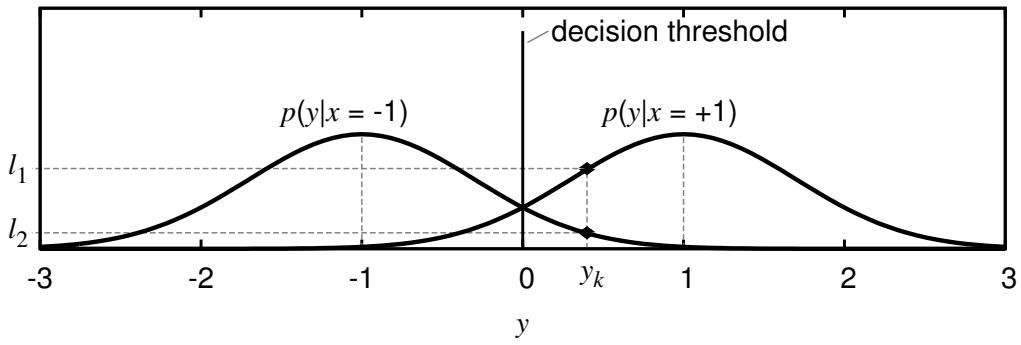

Figure 4: Likelihood of a value y_i at the output of a AWGN channel under the condition that signal value x_i at the channel input was either equal to -1 or $+1$

-1	-1	1	1
1	-1	-1	1
-1	1	-1	

.

Interference on the channel produces the signal values $y_i = x_i + \varepsilon$ at the channel output. Presuming an AWGN channel (*Additive White Gaussian Noise*), $\varepsilon \sim \mathcal{N}(0;\sigma^2)$ applies. The distribution of the values at the channel output are shown in **Figure 4** for $\sigma = 0.7$. For a measured signal value y_k at the channel output, there are two likelihoods l_1 and l_2 that either $x_k = +1$ or $x_k = -1$ were present at the channel input. If $l_1 = p(y|x = +1) > l_2 = p(y|x = -1)$ holds true, then $x_k = +1$ was probably sent and vice versa. The threshold is therefore $y = 0$ and only the sign of the measured value is decisive.

The received values y_i might take the following form

-0.8	-1.5	2	0.7
0.7	0.4	-1	2
0.2	0.4	-0.5	

.

A hard decision on the basis of the sign $x'_i = \mathrm{sgn}(y_i)$ and demodulation with $d'_i = (1 - x'_i)/2$ produces

1	1	0	0
0	0	1	0
0	0	1	

.

Three parities are violated; correction of the two bit errors is not possible.

Log Likelihood Algebra

As already alluded to in the previous section, iterative decoding involves an exchange of information or, to be more precise, a propagation of likelihoods. The multiplicative operations required here can be circumvented with the aid of log likelihood algebra [Hag96]. In Figure 4 a concrete signal value y_k that has been observed at instant k is drawn in. Values l_1 and l_2 of the ordinate are the likelihoods of this signal value under the condition that either $x = +1$

or $x = -1$ has been sent.

However, what is more interesting is a statement on the likelihood of signal x at the input, if value y at the output is observed. That is the a posteriori likelihood $p(x|y)$. The following applies in accordance with Bayes' theorem

$$p(x|y) = \frac{p(y|x) \cdot p(x)}{p(y)} \propto p(y|x) \cdot p(x) \; . \tag{9}$$

As $p(y)$ only has a normalising character, this variable can henceforth be omitted.

The hypothesis $x = +1$ is correct, if $p(x = +1|y) > p(x = -1|y)$ and vice versa. We can rewrite the test $p(x = +1|y) \lessgtr p(x = -1|y)$ with the aid of equation (9) as

$$\begin{aligned} p(x = +1|y) &\lessgtr p(x = -1|y) \\ p(y|x = +1) \cdot p(x = +1) &\lessgtr p(y|x = -1) \cdot p(x = -1) \\ \frac{p(y|x = +1) \cdot p(x = +1)}{p(y|x = -1) \cdot p(x = -1)} &\lessgtr 1 \; . \end{aligned}$$

If logarithms are used, the relationship is simplified as follows

$$\begin{aligned} \log \left[\frac{p(y|x = +1) \cdot p(x = +1)}{p(y|x = -1) \cdot p(x = -1)} \right] &\lessgtr \log(1) \\ \log \frac{p(y|x = +1)}{p(y|x = -1)} + \log \frac{p(x = +1)}{p(x = -1)} &\lessgtr 0 \; . \end{aligned}$$

or in general

$$L(x|y) = L_\mathrm{c}(y) + L(x) \lessgtr 0 \; .$$

The L values are described as LLR values (*Log-Likelihood Ratio*). The term $L_\mathrm{c}(y)$ is determined by the characteristics of the channel; the term $L(x)$ describes the a priori information and is initially equal to zero at the start of decoding, assuming that the bits $p(x = +1) = p(x = -1)$are uniformly distributed. If $L(x|y)$ is greater than zero, this points to a modulated value of $x_i = +1$ and consequently to a data bit $d_i = 0$; if $L(x|y) < 0$, then $d_i = 1$ is decoded. Each decoder provides what is termed *extrinsic* information, which is obtained on the basis of parity equations. As already mentioned above, the decoders exchange this information. As a result, $L(x)$ becomes more accurate and the decision in respect of zero or one is improved.

The L value $L_\mathrm{c}(y)$ for an AWGN channel applies in accordance with Figure 4

$$\begin{aligned} L_\mathrm{c}(y) &= \log\frac{p(y|x=+1)}{p(y|x=-1)} = \log\frac{\exp\left[-\frac{1}{2\sigma^2}\cdot(y-1)^2\right]}{\exp\left[-\frac{1}{2\sigma^2}\cdot(y+1)^2\right]} \\ &= \log\left(\exp\left[-\frac{1}{2\sigma^2}\cdot(y-1)^2+\frac{1}{2\sigma^2}\cdot(y+1)^2\right]\right) \\ &= \frac{1}{2\sigma^2}\cdot\left((y+1)^2-(y-1)^2\right) = \frac{1}{2\sigma^2}\cdot 4y \\ L_\mathrm{c}(y) &= \frac{2}{\sigma^2}\cdot y\;. \end{aligned} \tag{10}$$

This calculation must be adapted accordingly for other channel models. For the exchange of extrinsic information it is important to know how the L value for the sum of two or more bits is calculated. The a priori information

$$L(x) = \log\left(\frac{p(x=+1)}{p(x=-1)}\right) = \log\left(\frac{p(x=+1)}{1-p(x=+1)}\right)$$

can be rearranged according to $p(x=+1)$

$$\begin{aligned} \mathrm{e}^{L(x)} &= \frac{p(x=+1)}{1-p(x=+1)} \\ \mathrm{e}^{L(x)}-\mathrm{e}^{L(x)}\cdot p(x=+1) &= p(x=+1) \\ \mathrm{e}^{L(x)} &= p(x=+1)\cdot(1+\mathrm{e}^{L(x)}) \\ p(x=+1) &= \frac{\mathrm{e}^{L(x)}}{1+\mathrm{e}^{L(x)}} \end{aligned}$$

It applies analogously

$$p(x=-1) = 1-p(x=+1) = 1-\frac{\mathrm{e}^{L(x)}}{1+\mathrm{e}^{L(x)}} = \frac{1}{1+\mathrm{e}^{L(x)}}$$

Also, it is likely that the BPSK value from the XOR sum of two bits is equal to one, depending on the likelihood that both bits have the same value ($0\oplus 0 = 1\oplus 1 = 0$).

$$p(x_1\oplus x_2=+1) = p(x_1=+1)\cdot p(x_2=+1) + p(x_1=-1)\cdot p(x_2=-1) \tag{11}$$

For the sum of two bits, assuming their statistical independence, the following now ap-

plies:

$$
\begin{aligned}
L(x_1 \oplus x_2) &= \log \frac{p(x_1=+1)\cdot p(x_2=+1) + p(x_1=-1)\cdot p(x_2=-1)}{1-[p(x_1=+1)\cdot p(x_2=+1) + p(x_1=-1)\cdot p(x_2=-1)]} \\
&= \log\left(\frac{\frac{\mathrm{e}^{L(x_1)}}{1+\mathrm{e}^{L(x_1)}}\cdot\frac{\mathrm{e}^{L(x_2)}}{1+\mathrm{e}^{L(x_2)}} + \frac{1}{1+\mathrm{e}^{L(x_1)}}\cdot\frac{1}{1+\mathrm{e}^{L(x_2)}}}{1-\left[\frac{\mathrm{e}^{L(x_1)}}{1+\mathrm{e}^{L(x_1)}}\cdot\frac{\mathrm{e}^{L(x_2)}}{1+\mathrm{e}^{L(x_2)}} + \frac{1}{1+\mathrm{e}^{L(x_1)}}\cdot\frac{1}{1+\mathrm{e}^{L(x_2)}}\right]}\right) \\
&= \log\left(\frac{\frac{\mathrm{e}^{L(x_1)}\cdot\mathrm{e}^{L(x_2)}+1}{[1+\mathrm{e}^{L(x_1)}]\cdot[1+\mathrm{e}^{L(x_2)}]}}{1-\left[\frac{\mathrm{e}^{L(x_1)}\cdot\mathrm{e}^{L(x_2)}+1}{[1+\mathrm{e}^{L(x_1)}]\cdot[1+\mathrm{e}^{L(x_2)}]}\right]}\right) \\
&= \log\left(\frac{\frac{\mathrm{e}^{L(x_1)}\cdot\mathrm{e}^{L(x_2)}+1}{[1+\mathrm{e}^{L(x_1)}]\cdot[1+\mathrm{e}^{L(x_2)}]}}{\frac{[1+\mathrm{e}^{L(x_1)}]\cdot[1+\mathrm{e}^{L(x_2)}]}{[1+\mathrm{e}^{L(x_1)}]\cdot[1+\mathrm{e}^{L(x_2)}]} - \left[\frac{\mathrm{e}^{L(x_1)}\cdot\mathrm{e}^{L(x_2)}+1}{[1+\mathrm{e}^{L(x_1)}]\cdot[1+\mathrm{e}^{L(x_2)}]}\right]}\right) \\
&= \log\left(\frac{\frac{\mathrm{e}^{L(x_1)}\cdot\mathrm{e}^{L(x_2)}+1}{[1+\mathrm{e}^{L(x_1)}]\cdot[1+\mathrm{e}^{L(x_2)}]}}{\left[\frac{\mathrm{e}^{L(x_1)}+\mathrm{e}^{L(x_2)}}{[1+\mathrm{e}^{L(x_1)}]\cdot[1+\mathrm{e}^{L(x_2)}]}\right]}\right) \\
&= \log\left(\frac{\mathrm{e}^{L(x_1)}\cdot\mathrm{e}^{L(x_2)}+1}{\mathrm{e}^{L(x_1)}+\mathrm{e}^{L(x_2)}}\right) \qquad (12)
\end{aligned}
$$

In [Hag96], the following approximation has been stated

$$L(x_1 \oplus x_2) \approx \mathrm{sgn}[L(x_1)]\cdot\mathrm{sgn}[L(x_2)]\cdot\min\{|L(x_1)|,|L(x_1)|\} \ .$$

Consequently, only the smallest L value in terms of the amount and the product of the sign are significant. This can be expressed in general terms as

$$L\left(\sum_i \oplus\, x_i\right) \approx \left\{\prod_i \mathrm{sgn}[L(x_i)]\right\}\cdot\min_i\{|L(x_i)|\} \ . \qquad (13)$$

Thus one does not only have an L value for an individual bit but also for the additive operation of bits in Modulo 2 arithmetic. This relationship is required for the iterative decoding to analyse the parity equations and for generating the extrinsic information.

Decoding example

Let us return to the above example with the received values y_i

-0.8	-1.5	2	0.7
0.7	0.4	-1	2
0.2	0.4	-0.5	

For simplification purposes, we can assume that $\sigma^2 = 2$ and the numerical values are, according to (10), simultaneously the L values of the channe $L_c(y)$. The information for the channel is therefore $L_c(y) = (-0.8\ -1.5\ 2\ 0.7\ 0.4\ -1\ 0.7\ 2\ 0.2\ 0.4\ -0.5)$. In the structure of the parity-check matrix $\mathbf{H}$ from equation (8) we enter these L values where the corresponding bit is included in a parity equation. The positions of the zero bits in the matrix are left blank to provide a better overview. The values are initially unchanged, because the extrinsic information $L_{e_1}(x)$ and $L_{e_2}(x)$ of the two decoders is not yet known.

$L_c(y)$	-0.8	-1.5	2.0	0.7	$\underline{0.4}$	-1.0	0.7	2.0	$\underline{0.2}$	0.4	-0.5
$L_c(y) + L_{e_2}(x)$	-0.8	-1.5	2.0				0.7				
				0.7	$\underline{0.4}$	-1.0		2.0			
$L_c(y) + L_{e_1}(x)$	-0.8			0.7					$\underline{0.2}$		
		-1.5			$\underline{0.4}$					0.4	
			2.0			-1.0					-0.5

(14)

The underscored L values lead to bit errors when hard decisions are made. The number of negative values must be an even number in each row so that the parity equations are satisfied.

The extrinsic information of the two decoders will now be determined with the assistance of the relationship from the equation (13) The question is: How large is the L value for one bit, if the information of all other bits that are connected to this bit via a parity equation is analysed?

Let us take as an example bit d_1. It is connected via the equation $d_1 + d_2 + d_3 + p_1 = 0$ to three other bits (decoder I). If one knows the L values of d_2, d_3, and p_1, one can obtain the L value of d_1. The product of the signs of these three values is equal to minus one and the smallest value for the amount is equal to 0.7. The extrinsic information from decoder I for bit d_1 is therefore -0.7. The same procedure applies to all other bits and it produces the following extrinsic information

1. $L_{e_1}(x)$	-0.7	-0.7	0.7				0.8				
				-0.4	-0.7	0.4		-0.4			
$L_{e_2}(x)$	0.2			-0.2					-0.7		
		0.4			-0.4					-0.4	
			0.5			-0.5					-1.0

. (15)

The extrinsic information is now exchanged between the decoders and one obtains

$L_c(y)$	-0.8	-1.5	2.0	0.7	$\underline{0.4}$	-1.0	0.7	2.0	$\underline{0.2}$	0.4	-0.5
$L_c(y) + L_{e_2}(x)$	-0.6	-1.1	2.5				0.7				
				0.5	$\underline{0.0}$	-1.5		2.0			
$L_c(y) + L_{e_1}(x)$	-1.5			0.3					$\underline{0.2}$		
		-2.2			-0.3					0.4	
			2.7			-0.6					-0.5
$L_c + L_{e_1} + L_{e_2}$	-1.3	-1.8	3.2	0.1	-0.7	-1.1	1.5	1.6	-0.5	$\underline{0.0}$	-1.5

. (16)

As there is only one entry per column in the last five columns of the matrix, every decoder calculates extrinsic information for its bits, but an exchange is not possible. The overall

result of decoding after the first pass is produced from the sum of the L values from the three sources of information (channel, decoder I and II) for

$$L(\hat{y}) = L_{\mathrm{c}}(y) + L_{e_1}(x) + L_{e_2}(x)\ . \tag{17}$$

The sign of $L(\hat{y})$ provides the transmitted signal value x, the absolute value of $L(\hat{y})$ provides information about the reliability of the decoded bit. It can be seen that, by adding the extrinsic information, the two values for d_5 and p_3 that were originally incorrect have been corrected. In contrast to this, the L value for p_4 no longer allows a decision. A further pass is required. The extrinsic information is now determined on the basis of the matrix (16)

2.													
$L_{e_1}(x)$	-0.7	-0.6	0.6					0.6					
				0.0	-0.5	0.0			0.0				
	0.2			-0.2						-0.3			
$L_{e_2}(x)$		-0.3			-0.4						0.3		
			0.5			-0.5						-0.6	

. (18)

The input for the next iteration and the overall result are

$L_{\mathrm{c}}(y)$	-0.8	-1.5	2.0	0.7	$\underline{0.4}$	-1.0	0.7	2.0	$\underline{0.2}$	0.4	-0.5
$L_{\mathrm{c}}(y) + L_{e_2}(x)$	-0.6	-1.8	2.5				0.7				
				0.5	$\underline{0.0}$	-1.5		2.0			
$L_{\mathrm{c}}(y) + L_{e_1}(x)$	-1.5			0.7					$\underline{0.2}$		
		-2.1			-0.1					0.4	
			2.6			-1.0					-0.5
$L_{\mathrm{c}} + L_{e_1} + L_{e_2}$	-1.3	-2.4	3.1	0.5	-0.5	-1.5	1.3	2.0	-0.1	0.7	-1.1

. (19)

All signs, i.e. all parity equations, are thus correct, the bits are decoded without errors and decoding is ended. A further pass would only change the absolute values of $L(\hat{y})$ slightly but would no longer change the sign and thus the decoding result.

Serial decoding with decoding example

The typical characteristic of parallel processing is that all decoders work simultaneously and then exchange the extrinsic information. With serial processing, the decoders work one after another. When it is finished, its output is used as extrinsic information for the other[1] decoder. This therefore creates a better initial position and produces more accurate results. Its output is then in turn included by the first decoder. The decoders therefore work alternately, which in general leads to faster convergence.

Taking as a basis the received signal values from the example above

$L_{\mathrm{c}}(y)$	-0.8	-1.5	2.0	0.7	$\underline{0.4}$	-1.0	0.7	2.0	$\underline{0.2}$	0.4	-0.5
$L_{\mathrm{c}}(y) + L_{e_2}(x)$	-0.8	-1.5	2.0				0.7				
				0.7	$\underline{0.4}$	-1.0		2.0			
$L_{\mathrm{c}}(y) + L_{e_1}(x)$	-0.8			0.7					$\underline{0.2}$		
		-1.5			$\underline{0.4}$					0.4	
			2.0			-1.0					-0.5

(20)

[1] or the next one, where there are more than two decoders

serial decoding is explained. The extrinsic information from decoder I is initially

$1.I$	$L_{e_1}(x)$	-0.7	-0.7	0.7						0.8		
						-0.4	-0.7	0.4			-0.4	

. (21)

This information is now processed by decoder II

$L_\mathrm{c}(y)$	-0.8	-1.5	2.0	0.7	$\underline{0.4}$	-1.0	0.7	2.0	$\underline{0.2}$	0.4	-0.5
$L_\mathrm{c}(y)+L_{e_1}(x)$	-1.5			0.3					0.2		
		-2.2			-0.3					0.4	
			2.7			-0.6					-0.5

(22)

and its extrinsic information $L_{e_2}(x)$ is calculated

$1.II$	$L_{e_2}(x)$	0.2			-0.2					-0.3		
			-0.3				-0.4				0.3	
				0.5				-0.5				-0.6

. (23)

The connection of the received signal values $L_\mathrm{c}(y)$ to the two items of extrinsic information $L_{e_1}(x)$ and $L_{e_2}(x)$ produces

$L_\mathrm{c}+L_{e_1}+L_{e_2}$	-1.3	-2.5	3.2	0.1	-0.7	-1.1	1.5	1.6	-0.1	0.7	-1.1

. (24)

All parity equations are already satisfied after this first iteration and decoding is ended. If this is not the case, the information $L_{e_2}(x)$ is supplied to decoder I again.

> Comment:
> For LDPC codes each row of parity-check matrix **H** can be interpreted as an encoder. The extrinsic information is generated for each row and in each case made available for all other rows.

Parallel decoding with modified parity-check matrix

In contrast to the above examples, verification is now carried out with the parity-check matrix from equation (4). The number of the bits involved in the parity equations is therefore greater. The bits are also involved in more equations.

Taking as a basis the received signal values from the above example, the extrinsic information is determined for each row:

L_c	-0.8	-1.5	2.0	0.7	$\underline{0.4}$	-1.0	0.7	2.0	$\underline{0.2}$	0.4	-0.5
L_e		0.20	-0.20	-0.20			-0.20		-0.70		
L_e		-0.40		0.40		-0.40		0.40		0.70	
L_e	-0.20	-0.20		0.20	0.20				0.40	0.20	
L_e		0.40	-0.40		-0.40	0.40				-0.40	0.40
L_e	0.40	0.40		-0.40	-0.50		-0.40	-0.40			0.40
$\sum$	-0.60	-1.10	1.40	0.70	-0.30	-1.00	0.10	2.00	-0.10	0.90	0.30

(25)

All positions not involved in a parity equation are shown as blanks. For the totals row

$$\sum = L_c + \sum_i L_e(i) \ .$$

all parity equations are checked. As not all are satisfied (the last two rows of $H_{\text{mod}} \Rightarrow$ value in the last column of $\sum$ should be negative), the L_e values are now exchanged. For row r the following applies

$$L_u(r) = L_c + \sum_{i \neq r} L_e(i)$$

or, using the sum already calculated,

$$L_u(r) = \sum -L_e(r) \ . \tag{26}$$

$L_c(y)$	−0.8	−1.5	2.0	0.7	0.4	−1.0	0.7	2.0	0.2	0.4	−0.5
$L_u(1)$		−1.30	1.60	0.90			0.30		0.60		
$L_u(2)$		−0.70		0.30	−	−0.60		1.60		0.20	
$L_u(3)$	−0.40	−0.90		0.50	−0.50				−0.50	0.70	
$L_u(4)$		−1.50	1.80		0.10	−1.40				1.30	−0.10
$L_u(5)$	−1.00	−1.50		1.10	0.20		0.50	2.40			−0.10

(27)

These total values are used as a basis for the next iteration, i.e., for the next calculation of extrinsic information.

L_c	−0.8	−1.5	2.0	0.7	0.4	−1.0	0.7	2.0	0.2	0.4	−0.5
L_e		0.30	−0.30	−0.30			−0.60		−0.30		
L_e		−0.20		0.20		−0.20		0.20		0.30	
L_e	−0.50	−0.40		0.40	−0.40				−0.40	0.40	
L_e		0.10	−0.10		−0.10	0.10				−0.10	0.10
L_e	0.10	0.10		−0.10	−0.10		−0.10	−0.10			0.20
$\sum$	−1.20	−1.60	1.60	0.90	−0.20	−1.10	0.00	2.10	−0.50	1.00	−0.20

(28)

One position (0.00) is still not correct, even after this calculation. The sums in accordance with equation (26) produce

$L_c(y)$	−0.8	−1.5	2.0	0.7	0.4	−1.0	0.7	2.0	0.2	0.4	−0.5
$L_u(1)$		−1.90	1.90	1.20			0.60		−0.20		
$L_u(2)$		−1.40		0.70		−0.90		1.90		0.70	
$L_u(3)$	−0.70	−1.20		0.50	0.20				−0.10	0.60	
$L_u(4)$		−1.70	1.70		−0.10	−1.20				1.10	−0.30
$L_u(5)$	−1.30	−1.70		1.00	−0.10		0.10	2.20			−0.40

(29)

L_e		−0.20	0.20	0.20			0.20		−0.60		
L_e		−0.70		0.70		−0.70		0.70		0.70	
L_e	0.10	0.10		−0.10	−0.10				0.20	−0.10	
L_e		−0.10	0.10		−0.30	−0.10				0.10	−0.10
L_e	−0.10	−0.10		0.10	−0.10		0.10	0.10			−0.10
$\sum$	−0.80	−2.50	2.30	1.60	−0.10	−1.80	1.00	2.80	−0.20	1.10	−0.70

(30)

Now all parities for $\sum$ are correct and decoding can be ended for this codeword.

4 Characteristics of LDPC codes

4.1 Parameters

In order to develop their capability, LDPC codes typically have a block length n of several hundred or even several thousand bits. The parity-check matrix $\mathbf{H}$ of order $k \times n$ is correspondingly huge. The above example is therefore not an LDPC code in the strict sense of the word. The block length n is the sum of l data bits and k parity bits, i.e., $n = l + k$ applies.

The number w_r of the 1-bits per row and the number w_c of the 1-bits per column in $\mathbf{H}$ are important indicators, for which the following should apply

$$w_r \ll n \qquad \text{and} \qquad w_c \ll k \; . \tag{31}$$

The number of ones in $\mathbf{H}$ is therefore very low overall, the matrix is sparse. The requirement for a sparse matrix is not only due to the capability of iterative decoding but it is also intended to limit the effort required for calculation. If w_r is constant for all rows and if w_c is constant for all columns of $\mathbf{H}$, the code is called *regular*. If l is the number of data bits and k the number of parity bits in a codeword, then $w_\mathrm{r} = w_\mathrm{c} \cdot (l+k)/k$ for regular codes.

If the number of ones varies, it is an *irregular* code. The LDPC codes originally proposed by Gallager were regular [Gal63].

With reference to the visualization with a bipartite graph, w_c corresponds to the number of edges which leave a node of the variable bits and w_r is the number of the edges which leave a check node. Consequently, $w_\mathrm{r} \in \{3,4\}$ and $w_\mathrm{c} \in \{1,2\}$ apply to the block code in Figure 2. It is an irregular code. Typically, irregular codes perform better than regular codes.

LDPC codes continue to be characterized by a *girth*. This states how many edges are required as a minimum in order to leave any node in the bipartite graph and to return to it again. Because of the structure of the graph, such a loop length is always an even number and the smallest possible loop length is equal to four. In Figure 2, a loop from node d_1 and back has a length of eight ($d_1 \to c_1 \to d_2 \to c_4 \to d_5 \to c_2 \to d_4 \to c_3 \to d_1$). Transposition into a non-systematic channel code (Figure 3) reduces the girth considerably to a loop length of four ($d_1 \to c_3 \to d_4 \to c_5 \to d_1$).

As for other block codes, the minimum Hamming distance d_min, which is calculated by comparing two channel codewords in pairs, is also significant. If

$$d_\mathrm{H}(\mathbf{b}_i, \mathbf{b}_j) = \sum_{g=1}^{n} b_i[g] \oplus b_j[g] \qquad \forall\, \mathbf{b}_i, \mathbf{b}_j \in \text{set of channel codewords} \tag{32}$$

is the Hamming distance between the two channel codewords $\mathbf{b}_i$ and $\mathbf{b}_j$ with the elements $b_i[g]$ and $b_j[g]$, respectively, then the minimum Hamming distance of this channel code is

$$d_\mathrm{min} = \min[d_\mathrm{H}(\mathbf{b}_i, \mathbf{b}_j)] \qquad \forall (i,j); i \neq j \; . \tag{33}$$

The minimum Hamming distance determines how many symbol errors within the codeword can be reliably identified ($d_\mathrm{min} - 1$) or corrected in the case of hard decisions ($\lfloor (d_\mathrm{min} - 1)/2 \rfloor$).

4.2 Design

Shannon's theory states that long and random codes permit transmission to the limits of channel capacity. LDPC codes provide the solution for this. An LDPC code with a code rate of $R = l/n = 1/2$ was presented by Richardson and Urbanke, which has a bit error rate (BER) of less than 10^6 with a distance of just 0.13 dB to the theoretical threshold predicted by Shannon [Ric01]. However, the block length n for this is 10^6 Bits.

Good results are generally achieved, if the ones are entered in the parity-check matrix randomly. However, for this, several ones should appear in every row and every column so that the extrinsic information described above can be exchanged. Unfortunately, this characteristic does not exist with systematic codes, as in equation (2). Furthermore, all parity equations produced from this should be linearly independent of each other.

It seemed originally that codes with a girth of only four are generally bad. Such short loops have therefore been removed again from a (randomly) developed graph. However in [Che04] it was demonstrated that there are also powerful codes with a girth of just four.

The minimum Hamming distance $d_{\min}$ of a channel code is maximised, if one minimises the similarity of columns in $\mathbf{H}$.

The following parity-check matrix ($\overline{w}_\mathrm{r} = 6.875$, $\overline{w}_\mathrm{c} = 3.4375$) has been generated randomly

$$\mathbf{H} = \begin{pmatrix} |\overline{1} & \overline{0} & \overline{1}| & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ |\underline{1} & \underline{0} & \underline{1}| & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} . \tag{34}$$

When two columns have two ones at the same position (see marking in the matrix) then this generates a loop with a length of four in the bipartite graph.

The major problem with random matrices is, however, that both the encoding and the decoding are relatively complex in computational terms. Simple circuits based on digital registers, as are used for many other channel coding processes, are not suitable for this. For this reason, there is considerable interest in finding parity-check matrices, which have a random character yet still have a certain inner structure, which reduces the effort required for calculation. These include quasi-cyclical codes [Fos04] or what are termed protograph constructions [Tho04], which involve the bipartite graph being compiled of prototypes of smaller graphs.

A more detailed description can also be found in[Joh00] or [Joh02].

4.3 Decoding of LDPC codes

LDPC codes are decoded in principle according to the iterative procedure described above. It is also called the "turbo principle", named to reflect the way a turbo-charged internal combustion engine works in as far as the exhaust gases are fed back into the engine system.

As scientists from different technical backgrounds were and are involved in the development of this channel coding technique, there are variations of the iterative principle that are given other names such as *sum-product algorithm*, *message-passing algorithm* or *belief propagation*. The last algorithm has its origin in the graph theory and is based on the description of LDPC codes using bipartite graphs (see section 1).

In the example of the parity-check matrix $\mathbf{H}$ in equation (2) there are exactly two ones per column for each data bit and a single "one" for each parity bit. Typical LDPC matrices do not differentiate between data and parity bits (together: variable bits) and have on average a large number of ones per column (see equation 34). They should at least be equal to three ($\overline{w}_\mathrm{c} \geq 3$). Each parity equation (each row in $\mathbf{H}$) can be interpreted as a separate decoder, which then only processes a subset of the variable bits. The variable bit in column j with the weight $w_{\mathrm{c},j}$ is therefore involved in $w_{\mathrm{c},j}$ parity equations and can be combined with $w_{\mathrm{c},j} - 1$ extrinsic information.

The iteration of decoding is ended when all parity equations are satisfied, i.e., a valid channel codeword has been found. Further passes do not change the result in general.

The number of passes required falls with the rising signal-to-noise ratio (SNR) of the transmission. At the same SNR, codes with large block lengths require fewer passes than short LDPC codes. As it may not be possible to reproduce the message if the distortion of the signal is too great, the maximum number of iterations must be limited. This also limits the calculation effort and the time delay. However, this method of working leads to a variable decoding time. Nevertheless, the average working time is still less than if the number of iterations were to be constant and sufficiently high. If this maximum number of passes is reached, the decoder signals to the remaining processing chain that a transmission error cannot be corrected.

The rate of undetected errors is very low and similar as for Reed-Solomon codes. Only if the minimal Hamming distance of the channel code is too small in comparison to the interference that occurs, it is possible that the decoder generates a false valid channel codeword and the transmission error therefore remains undetected. Turbo codes, in contrast, which exploit the iterative decoding in conjunction with recursive, systematic convolution coding, cannot generate any information on transmission errors that cannot be corrected; decoding errors remain undetected, if no additional measures are taken, such as a cyclic-redundancy check (CRC) as an outer code.

4.4 Performance

Studies

Various programs are available on the Internet that can be used for LDPC encoding and decoding. The results presented here are based on simulations with the software package of Radford Neal (coauthor of [Mac96]). The source code (ANSI-C, [wwwSC]) compiles under Unix/Linux without problems. Under Windows slight adjustments are required. The software can read and use any parity-check matrices so that studies with self-developed matrices are possible.

The software package contains among others the following components:

- Generation of parity-check matrices and generator matrices,

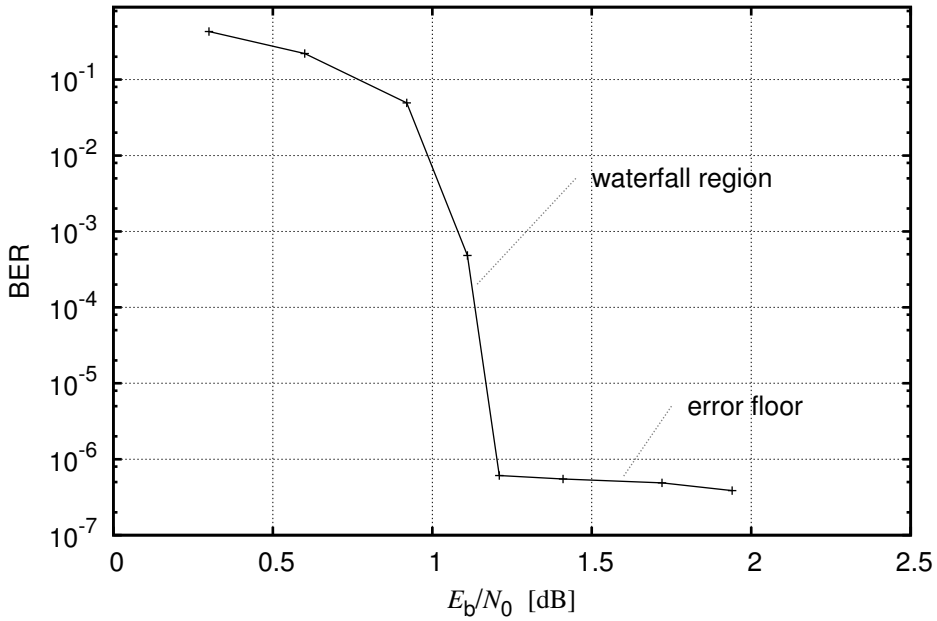

Figure 5: Characteristic curve for channel coding processes with iterative soft-input/soft-output decoding

- Generation of a random bit stream,
- Encoding the bit stream,
- Simulation of a channel,
- Decoding, and
- Checking the decoded message.

The sample studies were only carried out with a code rate of $R = 1/2$ and an AWGN channel. The bit error rate (BER), plotted against the signal-to-noise ratio E_b/N_0, is the quality criterion of a channel code. The bit error rate is the likelihood that a bit cannot be corrected. E_b is the energy per information bit and N_0 the one-sided noise-power density. The ratio is calculated as a function of the noise variance σ_z^2 and the code rate R in accordance with the following equation [Mac96]

$$\frac{E_\mathrm{b}}{N_0}\ [\mathrm{dB}] = 10 \cdot \log_{10}\left(\frac{a^2}{2 \cdot R \cdot \sigma_\mathrm{z}^2}\right)\ , \tag{35}$$

if the transmitted signal values are $\pm a$.

The performance of a channel code becomes apparent with the extent to which the bit error likelihood at a certain signal-to-noise ratio falls. **Figure 5** shows a typical curve. The bit error rate falls clearly after a certain signal-to-noise ratio. This region is termed *waterfall region*. What may possibly happen is that the drastic decline in the curve may be substantially reduced again and a significantly higher signal-to-noise ratio may be required

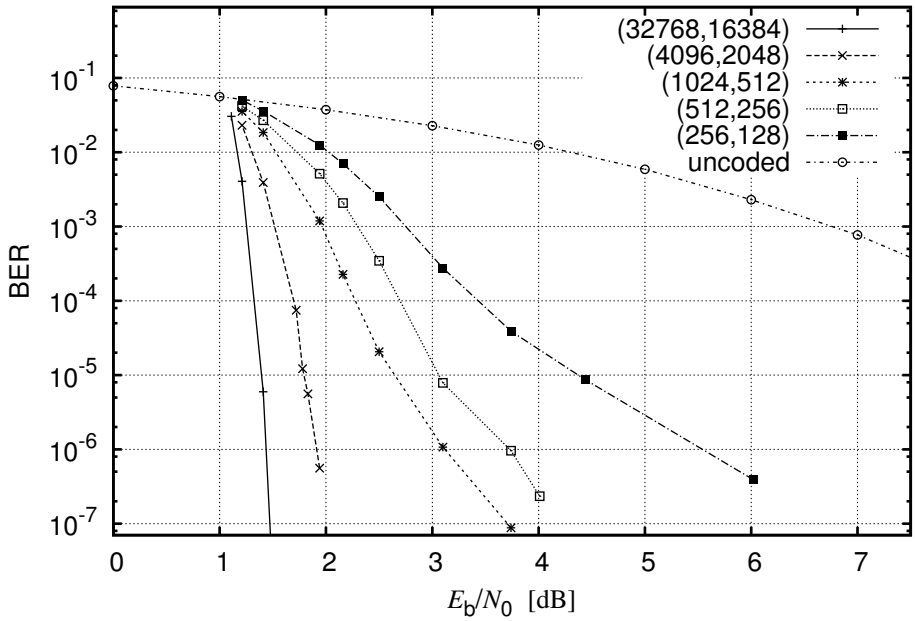

Figure 6: Comparison of bit error rates (BER) for regular LDPC codes with $w_c = 3$, $w_r = 6$ as a function of the signal-to-noise ratio

to reduce the bit error rate. This region is called the *error floor*. The cause generally lies with bits that are only incorporated in a single parity equation because these then have an increased probability of not being corrected.

Figure 6 compares regular LDPC codes with different block lengths. All were developed using Neal's software. The ones are distributed randomly in the parity-check matrix **H** and $w_r = 6$ and $w_c = 3$ apply. What can clearly be seen is that the necessary energy per bit falls, the greater the length of the codes is. When the number of ones per column or row is not constant, the LDPC code is called irregular, as has already been mentioned above.

Interestingly, the performance of irregular codes is strongly dependent on the distribution of ones.

Figure 7 compares irregular LDPC codes with different distributions for w_c. By way of comparison, the curve of the regular code of the same length is also drawn in. The distributions have been selected arbitrarily as follows:

	w_c						
Code	2	3	4	5	6	7	$\overline{w}_c$
regular	0	1	0	0	0	0	3.00
1	1/8	5/8	1/8	0	0	1/8	3.11
2	2/9	5/9	0	0	0	2/9	3.67
3	3/13	7/13	2/13	1/13	0	0	3.07
4	3/11	7/11	0	0	1/11	0	3.00

.

There is obviously a compromise between the position of the waterfall region and the level of the error floor.

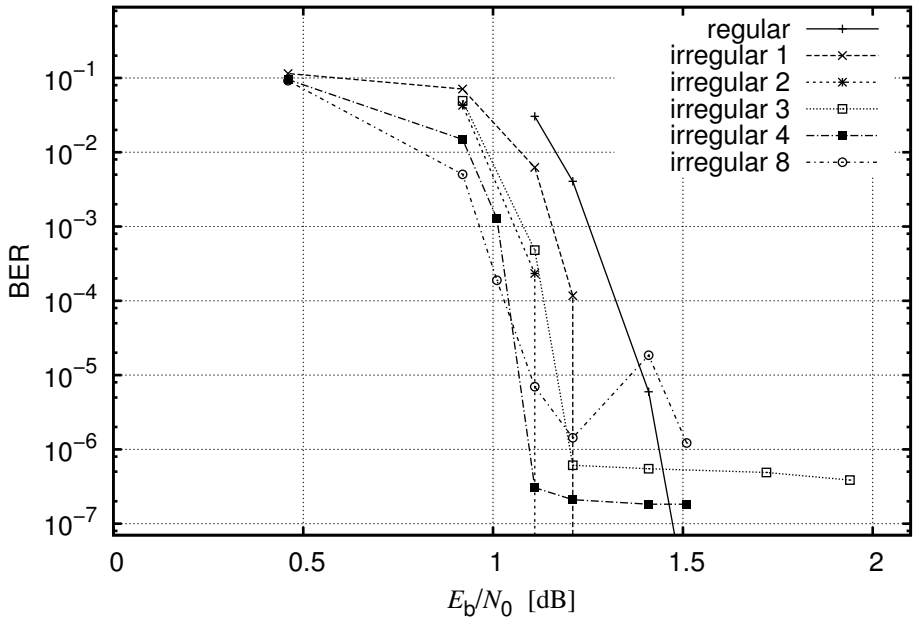

Figure 7: Bit error rate (BER) for irregular LDPC codes with $l = 16384$ and $R = 1/2$, see text for details

Shannon limit

Where does the so-called Shannon limit now lie? Essentially, this limit depends on the type of modulation and the code rate. For an AWGN channel with a binary input and continuous-value output, this limit in **Figure 8** is represented as a function of the code rate R [Lin04]. Here BPSK modulation[2] is assumed. The curve shows which signal-to-noise ratio is required in theory in order to allow error-free transmission at a certain code rate. If the code rate is close zero, i.e., the effort for channel coding will be infinitely great, then a minimum of -1.59 dB is required. For a code rate of $R = 1/2$, as has been used for the above studies, $E_\mathrm{b}/N_0 \geq 0.188$ dB applies.

5 Applications and outlook

Low-density parity-check codes are now used in many applications. In space travel, in particular, energy efficiency plays an exceptionally important role. Channel codes that can work close to the channel capacity limit are given precedence [And07]. Turbo codes [Ber93] have a certain advantage over LDPC codes at very low code rates ($R < 1/2$). LDPC codes were selected for NASA's Constellation Program (manned flights, e.g. to the moon or Mars) and the Mars Science Lab (MSL). However, at the start of 2010 the Constellation Program was postponed for financial reasons.

The LDPC codes are also used in many updates of the standards. Digital television via satellite (DVB-S2 ... Digital Video Broadcast via Satellite 2nd generation, 2003) uses

[2]Binary Phase Shift Keying

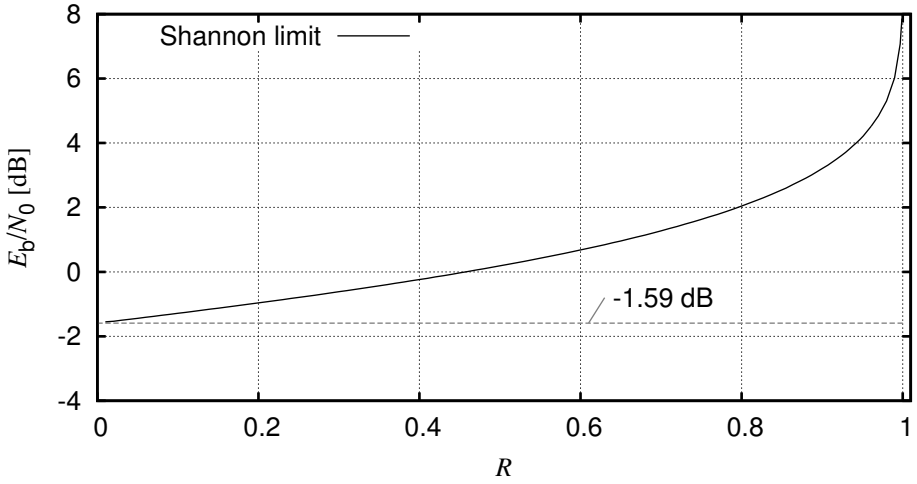

Figure 8: Minimum signal-to-noise ratio that theoretically allows fault-free transmission at a given code rate R (AWGN channel with BPSK signalling and continuous output)

LDPC codes in conjunction with an outer Bose-Chaudhuri-Hocquenghem (BCH) code at a high rate. The latter is intended to lower the error floor and allow very low bit error rates [Mor05]. This combination prevailed over other proposals for standardization purposes because of its performance in the presence of maximum hardware complexity. The block length is 64800 or 16200 bits, depending on the application. There are a variety of code rates (1/4 ... 8/9). The selection is made according to the type of modulation and the application.

The IEEE 802.16e (Wi-MAX) Standard specifies quasi-cyclic LDPC codes with block sizes of $n = 576$ to $n = 2304$ at code rates of $R = 1/2$, $R = 2/3$, $R = 3/4$, or $R = 5/6$. An outer channel code to lower the bit error rate is not required here, because care was taken when constructing the parity-check matrix to ensure that bits at critical positions of the quasi-cyclical structure are involved in at least two parity equations. The IEEE 802.11n (Wi-Fi) Standard also uses quasi-cyclic codes with block sizes of $n = 648$, $n = 1296$, and $n = 1944$ and four different code rates at Wi-MAX.

Further improvements in the performance of this type of channel coding are very unlikely, because it is currently already possible to achieve a distance to the Shannon limit of less than one decibel.

However, in general the computational outlay is still considerable. There is scope for further developments here. The number of good LDPC codes with different code rates is already high and is increasing even further. With growing popularity, they will displace conventional channel coding methods from many applications and standards. Applications with different system designs, such as multiple-input and multiple-output (MIMO) systems, also offer significant potential.

References

[And07] Andrews, K.S.; Divsalar, D.; Dolinar, S.; Hamkins, J.; Jones, C.R.; Pollara, F.: The Development of Turbo and LDPC Codes for Deep-Space Applications. *Proc. of IEEE*, Vol.95, No.11, Nov. 2007, 2142–2156

[Ber93] Berrou, C.; Glavieux, A.; Thitimajshima, P.: Near Shannon limit error-corecting coding and decoding. *Proc. IEEE Intern. Conf. on Communications*, Geneva, Switzerland, May 1993, 1064–1070

[Che04] Chen, L.; Xu, J.; Djurdjevic, I.; Lin, S.: Near-Shannon-Limit Quasi-Cyclic Low-Density Parity-Check Codes. *IEEE Trans. on Communications*, Vol.52, No.7, 2004, 1038–1042

[Fos04] Fossier, M.P.C.: Quasi-Cyclic Low-Density Parity-Check Codes From Circulant Permutation Matrices. *IEEE Trans. on Information Theory*, Vol.50, No.8, 2004, 1788–1793

[Gal63] Gallager, R.G.: Low-Density Parity-Check Codes. Cambridge, MA: MIT Press, 1963

[Hag96] Hagenauer, J.; Offer, E.; Papje, L.: Iterative Decoding of Binary Block and Convolutional Codes. *IEEE Trans. on Information Theory*, Vol.42, No.2, 1996, 429–445

[Joh00] Johnson, S.J.: *Introducing Low-density Parity-check codes*. Published Internal Technical Report, Department of Electrical and Computer Engineering, University of Newcastle, Australia 2000

[Joh02] Johnson, S.J.; Weller, S.R.: Low-density parity-check codes: Design and decoding. Chapter in Wiley *Encyclopedia of Telecommunications*, 2003

[Lin04] Lin, S.; Costello, D.J.: *Error Control Coding.*, Prentice Hall, 2nd edition, 2004

[Mac96] MacKay, D.J.C.; Neal, R.M.: Near Shannon limit performance of low density parity check codes. *Electron. Letters*, Vol.32, August 1996, 1645-1646 (reprinted with printing errors corrected in Vol.33, 457–458)

[Mor05] Morello, A.; Mignone, V.: DVB-S2: The Second Generation Standard for Satellite Broad-band Services. *Proceedings of IEEE*, Vol.94, No.1, January 2005, 210–227

[Ric01] Richardson, T.J.; Shokrollahi, A.; Urbanke, R.L.: Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes. *IEEE Trans. on Information Theory*, Vol.47, No.2, 2001, 619–637

[Sha48] Shannon, C.E.: A Mathematical Theory of Communication. *Bell System Technical Journal*, Vol.27, 1948

[Tan81] Tanner, R.M.: A recursive approach to low complexity codes. *IEEE Trans. on Information Theory*, Vol.27, No.5, 1981, 533–547

[Tho04] Thorpe, J.; Andrews, K.; Dolinar, S.: Methodologies for Designing LDPC Codes Using Protographs and Circulants. *Proc. of IEEE Int. Symp. Inf. Theory (ISIT)*, Chicago, USA, 27 June - 2 July, 2004

[wwwSC] http://www.cs.toronto.edu/∼radford/ldpc.software.html, besichtigt am 22.Juni 2010