On the Convergence of Iterative Belief Propagation

Axel Heim
Institute of Telecommunications and
Applied Information Theory
Ulm University, Germany
Email: axel.heim@uni-ulm.de

Ulrich Sorger
Computer Science and Communications,
University of Luxembourg,
Luxembourg
Email: ulrich.sorger@uni.lu

***Abstract*—The convergence of iterative decoding schemes utilizing belief propagation is considered. A quantitative bound for the output L-values of a Turbo decoder is given that only depends on the received word and thus is independent from the decoder iterations. Further, it is shown that the exponential increase of the L-values in each iteration within an LDPC decoder is limited by the degree of the variable nodes.**

***Index Terms*—Turbo decoding, Belief-Propagation, Convergence**

I. Introduction

Turbo codes, or more precisely parallel concatenated convolutional codes (PCCCs) [1] have been introduced to the communication community in 1993. These codes can perform close to the SHANNON limit when being iteratively decoded using the so-called sum-product or belief-propagation algorithm (BPA). Extrinsic information in terms of logarithmic probability ratios (L-values, cf. Equation (3)) is passed between two constituent decoders until a maximum number of iterations is reached or a stopping criterion is fulfilled.

The BPA assumes that the information passed from the other constituent decoder(s) is independent of what is already known about a given code symbol, e.g., from the channel. This assumption does not generally hold true as is visualized in the bipartite (Tanner [2]) graph in Figure 1. Due to the code constraint and the resulting correlation between (neighboring) bits, information is implicitly transferred with each run of a constituent decoder (dashed arrows). As extrinsic information it is then transferred to the other decoder (solid arrows), and so on. The figure shows a cycle (loop) of length 4 (black arrows), i.e., already in the second decoding iteration the independence assumption would be wrong.

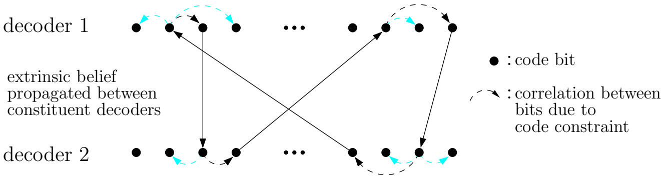

Figure 1. Belief propagation between two decoders with cycle of length 4

In [3] it was shown that if the output probabilities of a turbo decoder according to [1] converge, they converge to values between 0 and 1 – a property being frequently exploited for the design of stopping criteria [4], [5], yet on a heuristic basis. On the contrary, it was shown that the output probabilities of low-density parity-check (LDPC) [6], [7] decoders, which also apply the BPA, can converge to 0 or 1, that is the corresponding L-values can grow to ±∞.

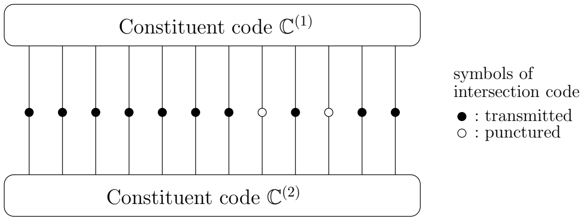
Constituent code $\mathbb{C}^{(1)}$
Constituent code $\mathbb{C}^{(2)}$
symbols of
intersection code
● : transmitted
○ : punctured

Figure 2. Tanner graph of punctured intersection code

In this paper we first recall the class of intersection (IS) codes which is a super class to both PCCCs and LDPC codes. Then we give quantitative bounds for the output L-values of a PCCC decoder and the exponential slope of LDPC decoder L-values.

II. Intersection Codes

The class of intersection (IS) codes [8] is equivalent to trellis-constrained codes [9]. We show that both PCCCs and LDPC codes belong to this code class which therefore provides a common basis for their comparison.

Definition 1 (Intersection Code): Let $\mathbb{C}^{(1)}$ and $\mathbb{C}^{(2)}$ be linear block codes of length n. An intersection code $\mathbb{C}^{(\cap)}$ is defined as the intersection

$$\mathbb{C}^{(\cap)} = \mathbb{C}^{(1)} \cap \mathbb{C}^{(2)}. \tag{1}$$

The parity check matrix of an intersection code is obtained by stacking the $h^{(l)} \times n$ parity check matrices $\boldsymbol{H}^{(l)}$, $l = 1, 2$ of its constituent codes $\mathbb{C}^{(l)}$. I.e., for $\boldsymbol{c} = [c_1\, c_2 \ldots c_n]$ being a binary vector, Equation (1) is equivalent to

$$\mathbb{C}^{(\cap)} = \left\{ \boldsymbol{c} : \boldsymbol{H}^{(\cap)} \cdot \boldsymbol{c}^T = \boldsymbol{0} \right\} \quad \text{with} \quad \boldsymbol{H}^{(\cap)} = \begin{bmatrix} \boldsymbol{H}^{(1)} \\ \boldsymbol{H}^{(2)} \end{bmatrix}.$$

Figure 2 shows the Tanner graph of an intersection code.[1] Each edge between the constituent codes indicates a code symbol of the intersection code. For illustration purposes, a

[1] In fact, this is not a graph according to Tanner's definition, as in his definition the sub-codes (here: constituent codes) are of shorter length than the overall code. However, this discrepancy is ignored.

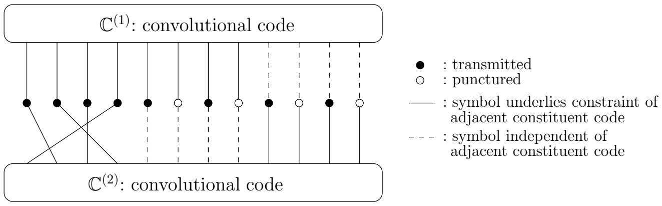
$\mathbb{C}^{(1)}$: convolutional code
$\mathbb{C}^{(2)}$: convolutional code
● : transmitted
○ : punctured
—— : symbol underlies constraint of adjacent constituent code
- - - : symbol independent of adjacent constituent code

Figure 3. Tanner graph of PCCC interpreted as IS code

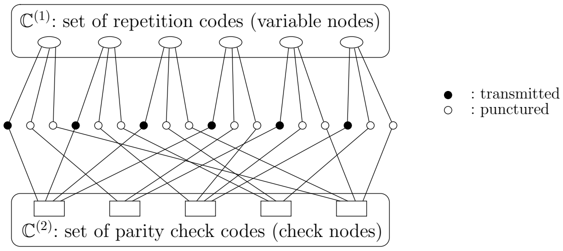
$\mathbb{C}^{(1)}$: set of repetition codes (variable nodes)
$\mathbb{C}^{(2)}$: set of parity check codes (check nodes)
● : transmitted
○ : punctured

Figure 4. Tanner graph of LDPC code interpreted as IS code

circle is added in the middle of each edge. An empty circle indicates that the particular bit is punctured, i.e., it is not transmitted.

The words of a PCCC[2] consist of three parts, namely information bits and parity bits of the two recursive systematic convolutional codes. The systematic information bits of the constituent encoders are the information bits of the PCCC encoder and an interleaved version of them, respectively. Figure 3 shows a PCCC interpreted as an IS code. The constituent codes $\mathbb{C}^{(l)}$ are a concatenation of each a terminated convolutional and a trivial code. All edges are connected to both constituent codes. However, the dashed lines - in contrast to the solid ones - indicate that a symbol of the IS code actually does not underlie a constraint of the adjacent constituent code, i.e., they connect to the trivial code part. E.g., the dashed lines at the right top represent the parity bits of $\mathbb{C}^{(2)}$ which are not considered by decoder 1. The crossing edges in the left part of the figure indicate the interleaving of the information bits.

An LDPC code can be interpreted as a highly punctured intersection code with constituent code $\mathbb{C}^{(1)}$ being a set of repetition codes and constituent code $\mathbb{C}^{(2)}$ being a set of parity check codes, cf. Figure 4. The ovals and squares within the constituent code boxes indicate variable nodes and check nodes, respectively. The IS code is punctured such that of each repetition code only one symbol is transmitted.

One major difference between the Tanner graphs for PCCC and LDPC codes is that for the LDPC case all symbols of the IS code underlie the constraints of both constituent codes.

III. Bounds for the Decoder Output

Let $C_i \in \{-1, +1\}$ denote a random variable for the value c_i of the i-th bit of a code word $\boldsymbol{c}$. Let

$$\boldsymbol{r} = \frac{1}{\sigma^2 \log(2)} \cdot (\boldsymbol{c} + \boldsymbol{\eta}) \tag{2}$$

be the scaled, noisy version of $\boldsymbol{c}$ after transmission over an additive white Gaussian noise (AWGN) channel, where $\boldsymbol{\eta}$ is the noise vector, σ^2 is the noise variance, and

$$P(\boldsymbol{r}|\boldsymbol{c}) \propto \exp\left(-\frac{\|\boldsymbol{r}-\boldsymbol{c}\|^2}{2\sigma^2}\right) \propto 2^{\boldsymbol{r}\boldsymbol{c}^T}$$

is the probability of $\boldsymbol{r}$ given $\boldsymbol{c}$. Further, let $P_{C_i}^{(l)}(x|\boldsymbol{r})$ denote the output probability for $C_i = x$ after decoding constituent code $\mathbb{C}^{(l)}$ given the input vector $\boldsymbol{r}$, and

$$L_i^{(l)}(\boldsymbol{r}) := \frac{1}{2} \cdot \log_2 \frac{P_{C_i}^{(l)}(+1|\boldsymbol{r})}{P_{C_i}^{(l)}(-1|\boldsymbol{r})} \tag{3}$$

the corresponding logarithmic probability ratio (L-value). The binary logarithm is motivated by the symbol set $\{\pm 1\}$. The factor $\frac{1}{2}$ in (3) then retains

$$L_i(\boldsymbol{r}) = \frac{1}{2} \cdot \log_2 \frac{P_{C_i}(+1|\boldsymbol{r})}{P_{C_i}(-1|\boldsymbol{r})} = r_i \tag{4}$$

for uncoded transmission as without a code constraint

$$P_{C_i}(x|\boldsymbol{r}) = P_{C_i}(x|r_i) \propto 2^{r_i \cdot x}. \tag{5}$$

Hence, r_i is an L-value. The extrinsic L-value is defined by

$$\breve{L}_i^{(l)}(\boldsymbol{r}) := L_i^{(l)}(\boldsymbol{r}) - r_i. \tag{6}$$

For iterative decoding, the input vector $\boldsymbol{r}$ of decoders 1 and 2 in Equations (3) and (6) is replaced by

$$\begin{aligned} \boldsymbol{m}^{(1,\nu)} &= \boldsymbol{r} + \boldsymbol{w}^{(2,\nu-1)} \quad \text{and} \\ \boldsymbol{m}^{(2,\nu)} &= \boldsymbol{r} + \boldsymbol{w}^{(1,\nu)}, \end{aligned} \tag{7}$$

respectively, where $\boldsymbol{w}^{(l,\nu)}$ is a transfer vector whose elements $w_i^{(l,\nu)}$ contain the extrinsic beliefs of decoder l after iteration ν if position i is in the transfer set $\mathbb{T}$ (see below), i.e.,

$$w_i^{(l,\nu)} = \begin{cases} \breve{L}_i^{(l)}\left(\boldsymbol{m}^{(l,\nu)}\right) & i \in \mathbb{T} \\ 0 & \text{else} \end{cases}. \tag{8}$$

In the following superscripts for code and iteration are omitted if clear from the context.

We now present some definitions and qualitative results of [3] (without proof), leading to the derivation of quantitative bounds for the output of Turbo and LDPC decoders in Sections III-A and III-B.

Definition 2 (Transfer Set [3]): We define the transfer set $\mathbb{T}$ as the set of positions in a word of an intersection code (edges in the Tanner graph) on which extrinsic information is exchanged between the constituent decoders during the decoding process.

Definition 3 (Information Set [3]): An information set $\mathbb{I}^{(l)}$ of a code $\mathbb{C}^{(l)}$ is defined as a set of positions i for which the code symbols c_i completely define the entire code word.

Definition 4 (Bit-Information Set [3]): We define a bit-information set $\mathbb{I}_i^{(l)}$ for bit c_i as a set of positions $j \neq i$ of a code $\mathbb{C}^{(l)}$ for which the code symbols c_j determine the value of c_i.

[2] We restrict ourselves to rate 1/2 convolutional constituent codes.

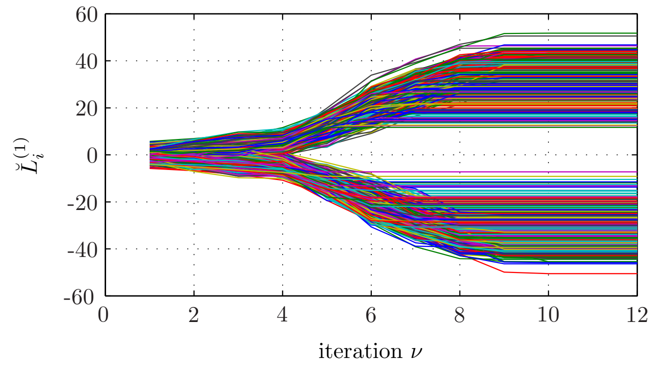

Figure 5. Turbo code: extrinsic L-values of constituent decoder 1

Lemma 1 ([3]): A necessary condition for the extrinsic L-values $\breve{L}_i^{(l)}\left(\boldsymbol{m}^{(l,\nu)}\right)$ of an intersection code to grow to plus or minus infinity for $\nu \to \infty$ is that

$$\exists\, \mathbb{I}_i^{(l)} : \mathbb{I}_i^{(l)} \subseteq \mathbb{T}, \tag{9}$$

i.e., there has to exist a bit-information set $\mathbb{I}_i^{(l)}$ for bit i of code $\mathbb{C}^{(l)}$ which is contained in the transfer set $\mathbb{T}$.

Theorem 1 (Convergence of Turbo Decoder Output [3]): The symbol output probabilities of a Turbo decoder according to [1] can never converge to 0 or 1.

Example 1: Figure 5 shows how the extrinsic L-values of a Turbo decoder converge to finite values.

In the following we assume that $\hat{\boldsymbol{c}}$ is the 'correct' code word to which the decoder output converges, i.e., for $i \in \mathbb{T}$ it holds

$$\operatorname{sgn}(w_i) = \hat{c}_i.$$

As the magnitude of the received word elements $|r_i| \ll |w_i|$ is negligibly small compared to the extrinsic beliefs for $i \in \mathbb{T}$ we can approximate

$$\begin{aligned} m_i^{(1,\nu)} &\approx w_i^{(2,\nu-1)} \quad \text{and} \\ m_i^{(2,\nu)} &\approx w_i^{(1,\nu)}. \end{aligned} \tag{10}$$

Without loss of generality let $\hat{c}_i = +1$ for some $i \in \{1, \ldots, n\}$.[3] The likelihood ratio for bit c_i is then

$$\begin{aligned} \frac{P_{C_i}^{(l)}(+1|\boldsymbol{m})}{P_{C_i}^{(l)}(-1|\boldsymbol{m})} &= \frac{\displaystyle\sum_{\boldsymbol{c}\in\mathbb{C}^{(l)}:c_i=+1} P^{(l)}(\boldsymbol{c}|\boldsymbol{m})}{\displaystyle\sum_{\boldsymbol{c}\in\mathbb{C}^{(l)}:c_i=-1} P^{(l)}(\boldsymbol{c}|\boldsymbol{m})} \\ &= \frac{P^{(l)}(\hat{\boldsymbol{c}}|\boldsymbol{m})(1+\varepsilon)}{\displaystyle\sum_{\boldsymbol{c}\in\mathbb{C}^{(l)}:c_i=-1} P^{(l)}(\boldsymbol{c}|\boldsymbol{m})} \\ &= \frac{2^{\boldsymbol{m}\hat{\boldsymbol{c}}^T}}{\displaystyle\sum_{\boldsymbol{c}\in\mathbb{C}^{(l)}:c_i=-1} 2^{\boldsymbol{m}\boldsymbol{c}^T}} + \varepsilon' \end{aligned} \tag{11}$$

where

$$\varepsilon := \frac{P_{C_i}^{(l)}(+1|\boldsymbol{m})}{P^{(l)}(\hat{\boldsymbol{c}}|\boldsymbol{m})} - 1 \gtrapprox 0$$

is greater than and approximately Zero as $\boldsymbol{m}$ is assumed to converge towards $\hat{\boldsymbol{c}}$.

A. Turbo Codes

In this section we first consider unpunctured turbo codes for simplicity. In the end some aspects of puncturing are addressed.

Denote by $\hat{\boldsymbol{u}}$ the information sequence corresponding to the code word $\hat{\boldsymbol{c}}$. Let $\tilde{\boldsymbol{u}}$ be the information sequence which differs from $\hat{\boldsymbol{u}}$ only in position i,

$$\tilde{u}_j := \begin{cases} -\hat{u}_j & j = i \\ \hat{u}_j & \text{else} \end{cases}, \tag{12}$$

and let $\tilde{\boldsymbol{c}}$ be the corresponding code sequence. Theorem 2 gives an asymptotic[4] bound for the extrinsic L-values of a turbo decoder which depends only on the received word and thus is unaffected by the decoding iterations.

Theorem 2 (Turbo Decoder Output): The magnitude $\left|\breve{L}_i^{(l)}\right|$ of the i-th extrinsic L-value of constituent code $\mathbb{C}^{(l)}$ is upper bounded by

$$\left|\breve{L}_i^{(l)}(\boldsymbol{m})\right| \le \left|\sum_{\substack{j=1\\ j\neq i}:\tilde{c}_j\neq\hat{c}_j}^{n} \hat{c}_j \cdot r_j + \varepsilon''\right| =: \left|\breve{L}_{i,\max}^{(l)} + \varepsilon''\right| \tag{13}$$

where $\boldsymbol{r}$ is the received word, and $\varepsilon'' \gtrapprox 0$ as the magnitudes $|m_j|$, $j \in \mathbb{T}$, grow large relative to $|r_j|$.

Proof: We give the proof for $\hat{c}_i = 1$, the proof for $\hat{c}_i = -1$ is analogous. During the first decoding iterations, the magnitudes $|m_j|$ of input values for systematic bits, parity bits and termination bits are of the same order, as the extrinsic belief – which is available only for systematic bits – is still small. Hence, the denominator sum in (11) is governed by code words $\boldsymbol{c}$ at small HAMMING distance $\mathrm{d}_H(\boldsymbol{c}, \hat{\boldsymbol{c}})$ to $\hat{\boldsymbol{c}}$.

The larger the magnitudes of the extrinsic L-values grow, the more increases the contribution of $\tilde{\boldsymbol{c}}$ to the denominator sum in (11) yielding

$$\frac{P_{C_i}^{(l)}(+1|\boldsymbol{m})}{P_{C_i}^{(l)}(-1|\boldsymbol{m})} = \frac{2^{\boldsymbol{m}\hat{\boldsymbol{c}}^T}}{\sum_{\boldsymbol{c}\in\mathbb{C}^{(l)}:c_i=-1} 2^{\boldsymbol{m}\boldsymbol{c}^T}} + \varepsilon' \le \frac{2^{\boldsymbol{m}\hat{\boldsymbol{c}}^T}}{2^{\boldsymbol{m}\tilde{\boldsymbol{c}}^T}} + \varepsilon'. \tag{14}$$

Thus the extrinsic L-value is bounded by

$$\begin{aligned} \breve{L}_i^{(l)}(\boldsymbol{m}) &\le \frac{1}{2}\left(\boldsymbol{m}\hat{\boldsymbol{c}}^T - \boldsymbol{m}\tilde{\boldsymbol{c}}^T\right) + \varepsilon'' - m_i \\ &= \sum_{\substack{j=1\\ j\neq i}:\tilde{c}_j\neq\hat{c}_j}^{n} \hat{c}_j \cdot m_j + \varepsilon'' \end{aligned} \tag{15}$$

As $\tilde{\boldsymbol{c}}$ differs from $\hat{\boldsymbol{c}}$ except for position i only in parity and termination bit positions j for which there is no extrinsic

[3] For $\hat{c}_i = -1$, the derivation is analog but with nominator/denominator exchanged.

[4] Actually the bound is not asymptotic as it does not hold $\varepsilon'' \to 0$ for Turbo codes according to Lemma 1. Still, as ε'' becomes very small, we use the notation $\varepsilon'' \gtrapprox 0$.

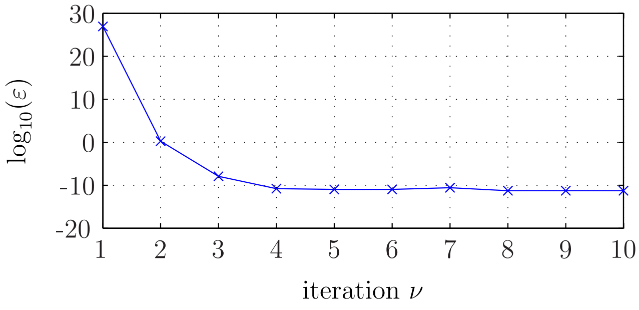

Figure 6. Logarithmic magnitude of ε

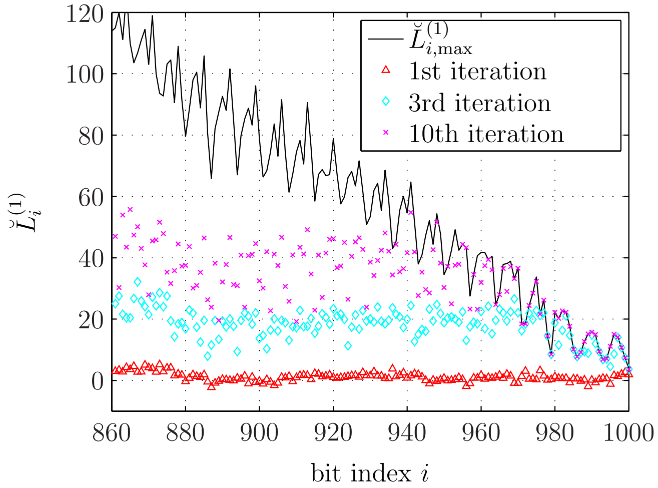

Figure 7. Turbo code: extrinsic L-values of constituent decoder 1

information, cf. Equation (12), the right hand side of (15) does not change over the iterations. ■

Example 2: Consider a PCCC of dimension $k = 1000$ and rate $1/3$ whose constituent encoders have the generator polynomial $G(D) = \left[1 \; \frac{1+D+D^3}{1+D^2+D^3}\right]$. The all-plus word $\hat{\boldsymbol{c}} = \mathbf{1}$ is transmitted over an AWGN channel. Figure 6 shows the value of ε before decoding constituent code $\mathbb{C}^{(1)}$. Already for iteration 3, ε is negligibly small. Figure 7 shows the extrinsic L-values of the first constituent decoder for the systematic bits after 1, 3 and 10 decoding iterations, and the bound $\breve{L}_{i,\max}$ is given as reference.
As expected, the extrinsic L-values grow with the iterations. However, for $i > 950$ they are limited by $\breve{L}_{i,\max}$ which is a (in the mean) decreasing curve overlaid by a noisy periodic pattern. The values of $\breve{L}_{i,\max}$ are explained as follows. For $i = k$, $\tilde{\boldsymbol{c}}$ differs[5] from $\hat{\boldsymbol{c}}$ only in termination bits and the i-th parity bit. The smaller $i < k$, the more parity bits can differ, hence more received parity symbols can contribute to $\breve{L}_{i,\max}$ in (13) thus increasing the bound. However, with decreasing i also the number of words $\boldsymbol{c}$ with $c_i = -1$ at small distance to $\hat{\boldsymbol{c}}$ rises, governing the true value of $\breve{L}_i$ which explains the looseness of the bound.

The average slope ϕ of the bound values is the negative ratio of weight and length of the parity sequence after exciting the recursive systematic encoder with a single 1 followed by zeros, multiplied with the mean of the received symbols, and the period ξ of the overlying noisy pattern is equal to the number of states the encoder goes through. For a primitive feedback polynomial, $\xi = 2^\kappa - 1$ where κ is the encoder memory. For $G(D) = \left[1 \; \frac{1+D+D^3}{1+D^2+D^3}\right]$ as in Figure 7, $\xi = 2^3 - 1 = 7$.

Remark 1: The result that extrinsic L-values of bits close to the termination are limited in magnitude may be counter-intuitive as trellis termination does make beliefs on these bits more reliable.

Figure 7 visualizes how the magnitudes of extrinsic L-values are limited by $\breve{L}_{i,\max}$. Puncturing parity bits causes the (mean) slope of $\breve{L}_{i,\max}$ to decrease as less received symbols contribute to the sum in (13), thus inhibiting the propagation of extrinsic beliefs. This is particularly relevant when both all termination bits and some parity bits close to the termination are punctured. Then for information bits close to the termination $\breve{L}_i^{(l)}(\boldsymbol{m}) = \breve{L}_{i,\max} = 0$ and hence no extrinsic belief can be transferred. E.g., for even information lengths, the puncturing pattern $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ is more suggestive than $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$. Also, high rate PCCCs should have systematic bits punctured while leaving some parity. Common methods for finding puncturing tables like the free-distance criterion (e.g., [10]) implicitly reduce this degradation by maximizing the free-distance of a punctured code.

B. LDPC Codes

In [3] it was shown that the output L-values of a belief propagation LDPC decoder can grow to $\pm\infty$. We now give a quantitative bound for the exponential slope, i.e. the factor by which L-values increase in each decoding iteration. As an LDPC code is a concatenation of a set of repetition codes (variable nodes) and a set of single parity check (SPC) codes (check nodes), cf. Section II, we first consider these two code types individually.

1) Repetition Code: A repetition code consists of only two words, the all-plus word $\mathbf{1}$ and the all-minus word $-\mathbf{1}$. The L-value and the extrinsic L-value are thus given by

$$L_i^{(\text{rep})}(\boldsymbol{m}) = \frac{1}{2}\log_2 \frac{P^{(\text{rep})}(\mathbf{1}|\boldsymbol{m})}{P^{(\text{rep})}(-\mathbf{1}|\boldsymbol{m})} = \sum_{j=1}^{n^{(\text{rep})}} m_j$$

and

$$\breve{L}_i^{(\text{rep})}(\boldsymbol{m}) = L_i^{(\text{rep})}(\boldsymbol{m}) - m_i = \sum_{j=1, j\neq i}^{n^{(\text{rep})}} m_j \quad \forall i = 1,\ldots,n^{(\text{rep})}, \tag{16}$$

respectively, where $n^{(\text{rep})}$ denotes the length of the code.

2) Single Parity Check Code: First, assume that $\hat{c}_i = +1$. An SPC code of length $n^{(\text{SPC})}$ contains $(n-1)$ words $\boldsymbol{c}$ with $c_i = -1$ and distance $\mathrm{d}_H(\boldsymbol{c}, \hat{\boldsymbol{c}}) = 2$ to $\hat{\boldsymbol{c}}$. The probability of words $\boldsymbol{c}$ with $c_i = -1$ but larger distance to $\hat{\boldsymbol{c}}$ is assumed to be negligibly small, which is the case when $|m_j| \gg 0$ for all

[5] Within the part of the IS code being relevant for $\mathbb{C}^{(1)}$.

$j \in \{1,\ldots,n^{(\mathrm{SPC})}\}$. Then the denominator in (11) and the L-value become

$$\begin{aligned}\sum_{\boldsymbol{c}\in\mathbb{C}^{(\mathrm{SPC})}:c_i=-1} 2^{\boldsymbol{m}\boldsymbol{c}^T} &\gtrapprox \sum_{\boldsymbol{c}\in\mathbb{C}^{(\mathrm{SPC})}:c_i=-1,\mathrm{d}_H(\boldsymbol{c},\hat{\boldsymbol{c}})=2} 2^{\boldsymbol{m}\boldsymbol{c}^T}\\ &= 2^{\boldsymbol{m}\hat{\boldsymbol{c}}^T}\cdot\sum_{j=1,j\neq i}^{n^{(\mathrm{SPC})}} 2^{-2(m_i\hat{c}_i+m_j\hat{c}_j)}\end{aligned}$$

and (recall that $\mathrm{sgn}(m_j)=\hat{c}_j$ and $\hat{c}_i=+1$)

$$\begin{aligned}L_i^{(\mathrm{SPC})}(\boldsymbol{m}) &= \frac{1}{2}\log_2\frac{P_{C_i}^{(\mathrm{SPC})}(+1|\boldsymbol{m})}{P_{C_i}^{(\mathrm{SPC})}(-1|\boldsymbol{m})}\\ &\lessapprox \frac{1}{2}\log_2\frac{(1+\varepsilon)}{\sum_{j=1,j\neq i}^{n^{(\mathrm{SPC})}} 2^{-2(m_i+|m_j|)}}\\ &= m_i-\frac{1}{2}\log_2\sum_{j=1,j\neq i}^{n^{(\mathrm{SPC})}} 2^{-2|m_j|}+\varepsilon'\\ &\leq m_i+|m_q|+\varepsilon', \qquad (17)\end{aligned}$$

respectively,[6] where

$$q=\arg\min_{j\in\{1,\ldots,n^{(\mathrm{SPC})}\}\setminus i}|m_j|,$$

and with equality in the last line of (17) only for $n^{(\mathrm{SPC})}=2$. Thus the extrinsic L-value is

$$\breve{L}_i^{(\mathrm{SPC})}(\boldsymbol{m})=L_i^{(\mathrm{SPC})}(\boldsymbol{m})-m_i\leq|m_q|+\varepsilon'.$$

Accordingly, for $\hat{c}_i=\pm1$ we obtain

$$\left|\breve{L}_i^{(\mathrm{SPC})}(\boldsymbol{m})\right|\leq|m_q|+\varepsilon'. \qquad (18)$$

Theorem 3 (Mean Exponential Slope): Consider the decoding of a regular LDPC code with variable and check node degrees d_v and d_c, respectively. For the ratio between the extrinsic norms in iterations $(\nu+1)$ and ν holds

$$\frac{\left\|\boldsymbol{w}^{(1,\nu+1)}\right\|}{\left\|\boldsymbol{w}^{(1,\nu)}\right\|}\leq(\mathrm{d}_v-1)\cdot\left(1+\frac{2\|\boldsymbol{r}\|+\varepsilon'''}{\left\|\boldsymbol{w}^{(1,\nu)}\right\|}\right)\asymp\mathrm{d}_v-1,$$

where $\left\|\boldsymbol{w}^{(l,\nu)}\right\|:=\sum_{i=1}^n\left|w_i^{(l,\nu)}\right|$, $\|\boldsymbol{r}\|=\sum_{i=1}^n|r_i|$, and $\asymp$ denotes asymptotic equality for $|w_i|\to\infty$.

Proof: All bit positions $i\in\{1,\ldots,n\}$ are within the transfer set $\mathbb{T}$. With (18) holds for the SPC equations

$$\left\|\boldsymbol{w}^{(2,\nu)}\right\|\leq\sum_{i=1}^n\left|m_i^{(2,\nu)}\right|+\varepsilon'''\leq\left\|\boldsymbol{w}^{(1,\nu)}\right\|+\|\boldsymbol{r}\|+\varepsilon'''$$

using the triangle inequality, and as each input of a repetition code appears in (d_v-1) extrinsic outputs, cf. (16), follows

$$\begin{aligned}\left\|\boldsymbol{w}^{(1,\nu+1)}\right\| &= (\mathrm{d}_v-1)\cdot\sum_{i=1}^n\left|m_i^{(1,\nu+1)}\right|\\ &\leq (\mathrm{d}_v-1)\cdot\left(\left\|\boldsymbol{w}^{(1,\nu)}\right\|+2\|\boldsymbol{r}\|+\varepsilon'''\right).\end{aligned}$$

$\blacksquare$

Theorem 4: Consider the decoding of a regular LDPC code. If for the extrinsic L-values $w_i^{(2,\nu)}$ holds that

$$w_i^{(2,\nu+1)}=\theta\cdot w_i^{(2,\nu)}\qquad\forall i\text{ and }\nu\to\infty \qquad (19)$$

with $1<\theta\leq(d_v-1)$, then for any integer $\lambda\geq0$

$$w_i^{(2,\nu+\lambda)}=\theta^\lambda\cdot w_i^{(2,\nu)}.$$

Proof: As (16) is a linear equation, it directly follows

$$w_i^{(1,\nu+1)}=\theta\cdot w_i^{(1,\nu)}\quad\text{if}\quad m_i^{(1,\nu+1)}=\theta\cdot m_i^{(1,\nu)}$$

$\forall i\in\{1,\ldots,n\}$. Hence it remains to prove that

$$w_i^{(2,\nu+\lambda+1)}=\theta\cdot w_i^{(2,\nu+\lambda)}\quad\text{if}\quad m_i^{(2,\nu+\lambda+1)}=\theta\cdot m_i^{(2,\nu+\lambda)}. \qquad (20)$$

Equation (19) requires that a single term $2^{-2|m_q|}$ dominates the sum in each parity-check equation (17) which implies $||m_q|-|m_j||\to\infty$ for all $j=1,\ldots,n^{(\mathrm{SPC})},j\neq q$ of each check node. Let $m_j^{(2,\nu)}=\delta+m_q^{(2,\nu)}$ with some $\delta\neq0$. As with (19) and (20)

$$m_j^{(2,\nu+1)}=\theta\cdot m_j^{(2,\nu)}=\theta\cdot\delta+\theta\cdot m_q^{(2,\nu)}=\theta\cdot\delta+m_q^{(2,\nu+1)},$$

$||m_q|-|m_j||$ is increased in each iteration and thus (20) holds for $\lambda\geq0$. The bound for θ directly follows from Theorem 3. $\blacksquare$

Acknowledgment

The authors would like to thank Stephan Stiglmayr and Steffen Schober for fruitful discussions, and Martin Bossert for giving the opportunity to do this work.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes (1),” in *Proc. IEEE International Conference on Communications*, vol. 2, (Geneva, Switzerland), pp. 1064–1070, May 1993.

[2] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Transactions on Information Theory*, vol. 47, pp. 533–547, September 1981.

[3] A. Heim and U. Sorger, “Turbo decoding: Why stopping criteria do work,” in *Proc. 5th International Symposium on Turbo Codes & Related Topics*, Sep. 2008.

[4] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” *IEEE Transactions on Information Theory*, vol. 42, pp. 429–445, March 1996.

[5] R. Y. Shao, S. Lin, and M. P. Fossorier, “Two simple stopping criteria for turbo decoding,” *IEEE Transactions on Communications*, vol. 47, pp. 1117–1120, August 1999.

[6] R. G. Gallager, *Information Theory and Reliable Communications*. John Wiley, New York, 1968.

[7] D. MacKay and R. Neal, “Near Shannon limit performance of low density parity check codes,” *IEE Electronics Letters*, vol. 32, pp. 1645–1646, August 1996.

[8] U. Sorger, *Communication Theory*. Books On Demand, 2009.

[9] B. J. Frey and D. J. MacKay, “Trellis-constrained codes,” in *Proc. of the 35th Allerton Conference on Communication, Control and Computing*, 1997.

[10] F. Babich, G. Montorsi, and F. Vatta, “Some notes on rate-compatible punctured turbo codes (RCPTC) design,” *IEEE Transaction on Communications*, vol. 52, no. 5, pp. 681–684, May 2004.

[11] N. Wiberg, *Codes and Decoding on General Graphs*. PhD thesis, Linkoping University, 1996.

[6] The last inequality actually is the Min-Sum approximation [11].