

Generalizing the transfer in iterative error correction dissection decoding

Conference Paper**Author(s):**

Sorger, Ulrich; Heim, Axel

Publication date:

2010

Permanent link:

<https://doi.org/10.3929/ethz-a-006001389>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Generalizing the Transfer in Iterative Error Correction: Dissection Decoding

Ulrich Sorger

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

Axel Heim

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

Abstract—Iterative decoding with message-passing is considered. The message format is generalized from the classical, single probability value for each code symbol to a probability distribution by introducing an additional logarithmic probability measure. Thereby, the representation of the probability distributions underlying the constituent code constraints by the messages is improved in terms of the Kullback-Leibler divergence. Simulation shows that this improvement can transfer to the error correcting performance.

I. INTRODUCTION

PEARL's belief propagation algorithm (BPA) [1], [2] has attracted major attention in the communication community when it was applied to parallel concatenated convolutional codes (PCCCs) by BERROU et al. [3] in the early 90's. Using the BCJR algorithm [4] to efficiently compute symbol probabilities in the trellises of the constituent codes, the iterative exchange of so-called extrinsic information between the constituent decoders allows for error correcting performance close to the SHANNON limit [5] while maintaining low computational complexity. The field of application was quickly extended to other code constructions like serial concatenations [6] or low-density parity-check codes [7]. The basic principle of the decoding scheme, however, has remained the same ever since.

After recalling the abstract class of intersection codes in Section II, Section III-A emphasizes an observation made in [8]: The symbol probabilities computed in the constituent decoders minimize the KULLBACK-LEIBLER divergence between a) the probability distribution of the code words given the input beliefs and the code constraint, and b) the uncoded distribution given the objective variables. By replacing the latter distribution by a new one with a larger parameter space in Section III-B, this optimization is improved. Simulation in Section IV shows that this improvement can also transfer to the error correcting performance.

II. INTERSECTION CODES

The class of intersection (IS) codes [9] is equivalent to the class of embedding codes [10] or trellis-constrained codes. Every code can be expressed as the intersection of two (or more) super-codes, and hence as an IS code.

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

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

of the constituent codes (super codes) $\mathbb{C}^{(1)}$ and $\mathbb{C}^{(2)}$.

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

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

with

$$\mathbb{C}^{(\cap)} \subseteq \mathbb{C}^{(l)} \subseteq \mathbb{S}, \quad l = 1, 2,$$

where \mathbb{S} denotes the n -dimensional binary space.

Example 1 (Turbo Codes): Let

$$\mathbf{G}_{\text{CC}} = \begin{bmatrix} \mathbf{I} & \mathbf{G}^{(p)} \end{bmatrix}$$

denote the generator matrix of the two identical systematic convolutional encoders of a PCCC [3], where \mathbf{I} is the identity matrix and $\mathbf{G}^{(p)}$ generates the parity part of the convolutional code words, including termination bits from both the systematic and the parity output. Let Π denote the Turbo code permutation matrix. The generator matrix of the PCCC then is given by

$$\mathbf{G}^{(\cap)} = \begin{bmatrix} \mathbf{I} & \mathbf{G}^{(p)} & \Pi \mathbf{G}^{(p)} \end{bmatrix}.$$

For interpretation as constituent codes $\mathbb{C}^{(l)}$ of an IS code, the codes defined by \mathbf{G}_{CC} require *uncoded extension*, i.e.

$$\begin{aligned} \mathbb{C}^{(1)} &= \left\{ \begin{bmatrix} \mathbf{u} & \mathbf{u} \mathbf{G}^{(p)} & \mathbf{v} \end{bmatrix} : \mathbf{u} \in \mathbb{F}_2^k, \mathbf{v} \in \mathbb{F}_2^{k+2\kappa} \right\} \\ \mathbb{C}^{(2)} &= \left\{ \begin{bmatrix} \mathbf{u} \Pi & \mathbf{v} & \mathbf{u} \Pi \mathbf{G}^{(p)} \end{bmatrix} : \mathbf{u} \in \mathbb{F}_2^k, \mathbf{v} \in \mathbb{F}_2^{k+2\kappa} \right\}, \end{aligned}$$

where k is the dimension of the PCCC, κ is the encoder memory and \mathbb{F}_2^k denotes the binary space of dimension k .

In the following we will implicitly use binary vectors and code words with bipolar values using the mapping

$$\{0, 1\} \mapsto \{+1, -1\}.$$

III. ITERATIVE DECODING

In general, an iterative decoder is a device consisting of two (or more) constituent decoders $\mathcal{D}^{(l)}$, $l = 1, 2$ corresponding to the constituent codes $\mathbb{C}^{(l)}$, which output a set of probabilities. For decoding, the noisy received word \mathbf{r} is input to the first constituent decoder $\mathcal{D}^{(1)}$ which computes conditional probabilities given \mathbf{r} and the constraint of $\mathbb{C}^{(1)}$. Together with \mathbf{r} these probabilities are input to decoder $\mathcal{D}^{(2)}$. $\mathcal{D}^{(2)}$ then computes probabilities under the constraint of $\mathbb{C}^{(2)}$ which are passed back to $\mathcal{D}^{(1)}$ and so forth until some stopping criterion is fulfilled.

We consider transmission over the additive white GAUSSIAN noise (AWGN) channel. Code words are transmitted with equal probability. Let

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

be the scaled, noisy version of a code word $\mathbf{c} \in \mathbb{C}^{(n)}$, where $\boldsymbol{\eta}$ is the noise vector, σ^2 is the noise variance,

$$p_{\mathbf{R}|\mathbf{S}}(\mathbf{r}|\mathbf{s}) = \frac{1}{(\sqrt{2\pi}\sigma)^n} \cdot \exp\left(-\frac{\|\mathbf{c} + \boldsymbol{\eta} - \mathbf{s}\|^2}{2\sigma^2}\right) \propto 2^{\mathbf{r}\mathbf{s}^T}$$

is the probability of \mathbf{r} given $\mathbf{s} \in \mathbb{S}$, and \mathbf{R} and \mathbf{S} denote the corresponding random variables, respectively.

Let $\mathbf{C}^{(l)}$, $l = 1, 2$ denote the random variable for the words of the codes $\mathbb{C}^{(l)}$, respectively, and S_i the random variable for the i -th bit of a binary vector. Denote by

$$P_{\mathbf{C}^{(l)}|\mathbf{R}}(\mathbf{s}|\mathbf{r}) \propto p_{\mathbf{R}|\mathbf{S}}(\mathbf{r}|\mathbf{s}) \cdot \langle \mathbf{s} \in \mathbb{C}^{(l)} \rangle, \quad l = 1, 2 \quad (3)$$

the probability of \mathbf{s} given \mathbf{r} and the constraint of code $\mathbb{C}^{(l)}$, where

$$\sum_{\mathbf{s} \in \mathbb{S}} P_{\mathbf{C}^{(l)}|\mathbf{R}}(\mathbf{s}|\mathbf{r}) = 1 \quad \text{and} \quad \langle b \rangle := \begin{cases} 1 & \text{if } b \text{ is true} \\ 0 & \text{else} \end{cases}$$

denotes the IVERSON bracket. Further, let

$$P_{S_i|\mathbf{R}}(x|\mathbf{r}, \mathbf{C}^{(l)}) = \sum_{\mathbf{s} \in \mathbb{S}: s_i = x} P_{\mathbf{C}^{(l)}|\mathbf{R}}(\mathbf{s}|\mathbf{r}), \quad x \in \{\pm 1\}$$

define the probability for $S_i = x$ given \mathbf{r} and the constituent code constraint $\mathbb{C}^{(l)}$, and

$$L_i^{(l)}(\mathbf{r}) := \frac{1}{2} \cdot \log_2 \frac{P_{S_i|\mathbf{R}}(+1|\mathbf{r}, \mathbf{C}^{(l)})}{P_{S_i|\mathbf{R}}(-1|\mathbf{r}, \mathbf{C}^{(l)})} \quad (4)$$

the corresponding logarithmic likelihood ratio (LLR). The probabilities $P_{\mathbf{S}|\mathbf{R}}(\mathbf{s}|\mathbf{r})$, $P_{S_i|\mathbf{R}}(x|\mathbf{r}, \mathbf{S})$ and $L_i(\mathbf{r})$ without a code constraint, i.e. $\mathbf{s} \in \mathbb{S}$, are defined accordingly.

In the following, subscripts may be neglected when clear from the context.

A. Belief Propagation

In belief propagation (BP), the messages passed between the decoders are given by a vector of extrinsic LLRs denoted by $(\mathbf{d}^{(l)} - \mathbf{m}^{(l)})$. This vector is defined by the decoder input $\mathbf{m}^{(l)} = \mathbf{r} + (\mathbf{d}^{(h)} - \mathbf{m}^{(h)})$ and the decoder output LLRs

$$d_i^{(l)} = \frac{1}{2} \cdot \log_2 \frac{P_{S_i|\mathbf{R}}(+1|\mathbf{m}^{(l)}, \mathbf{C}^{(l)})}{P_{S_i|\mathbf{R}}(-1|\mathbf{m}^{(l)}, \mathbf{C}^{(l)})}, \quad i = 1, \dots, n \quad (5)$$

Algorithm 1 The Belief Propagation Algorithm

- 1) *initialize*
 - set $l = 1, h = 2$
 - set $\mathbf{m}^{(h)} = \mathbf{d}^{(h)} = \mathbf{0}$
 - 2) *iterate*
 - while** (stopping criterion not fulfilled)
 - $\mathcal{D}^{(l)} : (\mathbf{m}^{(l)} = \mathbf{r} + \mathbf{d}^{(h)} - \mathbf{m}^{(h)}) \mapsto \mathbf{d}^{(l)}$, cf. (5)
 - swap $l \leftrightarrow h$
 - end**
 - 3) *output* $\hat{\mathbf{c}} = \text{sgn}(\mathbf{d}^{(h)})$
-

given the constraint of code $\mathbb{C}^{(l)}$. This is summarized in Algorithm 1.

The computation (5) can be motivated as follows. For simplicity we consider one constituent decoder and disregard the indices l, h . Let \mathbf{d} be a vector of n independent LLRs

$$d_i = \frac{1}{2} \cdot \log_2 \frac{P_{S_i|\mathbf{R}}(+1|\mathbf{d}, \mathbf{S})}{P_{S_i|\mathbf{R}}(-1|\mathbf{d}, \mathbf{S})}, \quad i = 1, \dots, n,$$

where we deliberately choose \mathbf{R} as the corresponding random variable. Hence \mathbf{d} is considered as being obtained from the same channel as the received word, i.e. $p_{\mathbf{R}|\mathbf{S}}(\mathbf{d}|\mathbf{s}) \propto \exp_2(\mathbf{d}\mathbf{s}^T)$. The following lemma shows that the cross entropy between $P_{\mathbf{C}|\mathbf{R}}(\mathbf{s}|\mathbf{r})$ and $P_{\mathbf{S}|\mathbf{R}}(\mathbf{s}|\mathbf{d})$ is an objective function whose minimization with respect to \mathbf{d} yields Equation (5).

Lemma 1 (Cross Entropy [8]): Minimizing the cross entropy

$$H_{\mathbf{R}|\mathbf{R}}(\mathbf{C}|\mathbf{r}|\mathbf{S}|\mathbf{d}) := - \sum_{\mathbf{s} \in \mathbb{S}} P_{\mathbf{C}|\mathbf{R}}(\mathbf{s}|\mathbf{r}) \cdot \log_2 P_{\mathbf{S}|\mathbf{R}}(\mathbf{s}|\mathbf{d}) \quad (6)$$

between the distributions $P_{\mathbf{C}|\mathbf{R}}(\mathbf{s}|\mathbf{r})$ and $P_{\mathbf{S}|\mathbf{R}}(\mathbf{s}|\mathbf{d})$ with respect to the vector \mathbf{d} of LLRs yields the logarithmic symbol probability ratios

$$d_i = \arg \min_{v_i \in \mathbb{R}} H_{\mathbf{R}|\mathbf{R}}(\mathbf{C}|\mathbf{r}|\mathbf{S}|\mathbf{v}) = \frac{1}{2} \log_2 \frac{P_{S_i|\mathbf{R}}(+1|\mathbf{r}, \mathbf{C})}{P_{S_i|\mathbf{R}}(-1|\mathbf{r}, \mathbf{C})}$$

where \mathbb{R} denotes the set of real numbers.¹

The KULLBACK-LEIBLER divergence (KLD) is an information theoretic measure for the similarity between two distributions over the same probability space. It directly relates to the cross entropy by

$$D_{\text{KL}}(\mathbf{C}|\mathbf{r}|\mathbf{S}|\mathbf{d}) := H_{\mathbf{R}|\mathbf{R}}(\mathbf{C}|\mathbf{r}|\mathbf{S}|\mathbf{d}) + \sum_{\mathbf{s} \in \mathbb{S}} P_{\mathbf{C}|\mathbf{R}}(\mathbf{s}|\mathbf{r}) \cdot \log_2 P_{\mathbf{C}|\mathbf{R}}(\mathbf{s}|\mathbf{r})$$

and its minimum value is 0 for two identical distributions.

The observation that for belief propagation the computation within the constituent decoders corresponds to the optimization of (6) – or, equivalently, the minimization of the KLD – is essential for the concept of Dissection Decoding below.

¹When extrinsic information from another decoder is available, \mathbf{r} is replaced by the appropriate input \mathbf{m} .

B. Dissection Decoding

In belief propagation the transfer message can be written as a vector of LLRs. We now increase the transfer complexity by introducing a new dimension to these messages. This new dimension is spanned by the discrete random variable U whose realizations

$$u(s) := H_{S|R}(s|\mathbf{d}) \quad (7)$$

are given by the *conditional word uncertainties*

$$H_{S|R}(s|\mathbf{d}) := -\log_2(P(s|\mathbf{d})) = -\log_2 \prod_{i=1}^n \frac{2^{s_i d_i}}{2^{d_i} + 2^{-d_i}}$$

of s , given a *dissector* \mathbf{d} . For now, \mathbf{d} is assumed to be constant and is disregarded in the notation for better readability. The finite probability space of U is denoted by \mathbb{U} . Its size is determined by \mathbf{d} . Rather than a transfer vector of length n , we employ a matrix $\mathbf{m} = [m_1[u], \dots, m_n[u]]$ of size $|\mathbb{U}| \times n$. We also introduce a new transfer vector \mathbf{q} of length $|\mathbb{U}|$. Let

$$\begin{aligned} P_{S|M,Q}(s|\mathbf{m}, \mathbf{q}) &\propto q[u(s)] \cdot P_{S|R}(s|\mathbf{m}[u(s)]) \\ &= q[u(s)] \cdot \prod_{i=1}^n \frac{2^{s_i \cdot m_i[u(s)]}}{2^{m_i[u(s)]} + 2^{-m_i[u(s)]}} \end{aligned} \quad (8)$$

with

$$\sum_{s \in \mathbb{S}} P_{S|M,Q}(s|\mathbf{m}, \mathbf{q}) = 1$$

denote the symbol-based probability of s given \mathbf{m} and \mathbf{q} . Further define

$$P_{S_i,C,U|R}(x, s, u|\mathbf{r}) := P_{C|R}(s|\mathbf{r}) \cdot \langle S_i = x \rangle \cdot \langle H(s|\mathbf{d}) = u \rangle$$

from which we obtain probabilities such as

$$P_{S_i,U|R}(x, u|\mathbf{r}, \mathbf{C}) = \sum_{s \in \mathbb{S}} P_{S_i,C,U|R}(x, s, u|\mathbf{r}, \mathbf{C})$$

by marginalization.

Akin to Lemma 1, the following theorem defines the optimum pair (\mathbf{m}, \mathbf{q}) for representing the distribution $P_{C|R}(s|\mathbf{r})$ in terms of the (uncoded) distribution $P_{S|M,Q}(s|\mathbf{m}, \mathbf{q})$.

Theorem 1: Minimizing the cross entropy

$$\begin{aligned} H_{R|M,Q}(C|\mathbf{r}|\mathbf{S}|\mathbf{m}, \mathbf{q}) &:= \sum_{s \in \mathbb{S}} P_{C|R}(s|\mathbf{r}) \cdot H_{S|M,Q}(s|\mathbf{m}, \mathbf{q}) \\ &= - \sum_{s \in \mathbb{S}} P_{C|R}(s|\mathbf{r}) \cdot \log_2(P_{S|M,Q}(s|\mathbf{m}, \mathbf{q})) \end{aligned} \quad (9)$$

with respect to \mathbf{m} and \mathbf{q} yields

$$q[u] \propto \frac{P_{U|R}(u|\mathbf{r}, \mathbf{C})}{P_{U|R}(u|\mathbf{m}[u], \mathbf{S})}, \quad (10)$$

and \mathbf{m} is given by the implicit solution

$$P_{S_i|R,U}(x|\mathbf{m}[u], u, \mathbf{S}) = P_{S_i|R,U}(x|\mathbf{r}, u, \mathbf{C}), \quad i = 1, \dots, n. \quad (11)$$

We observe that for $\mathbf{d} = \mathbf{0}$, i.e. $|\mathbb{U}| = 1$ it follows from Theorem 1 that

$$m_i[u] = \frac{1}{2} \cdot \log_2 \frac{P_{S_i|R,U}(+1|\mathbf{r}, u, \mathbf{C})}{P_{S_i|R,U}(-1|\mathbf{r}, u, \mathbf{C})} \quad (12)$$

are the symbol beliefs given \mathbf{r} and the code \mathbb{C} , and $q[u]$ is a constant, i.e. the computation is as for the BPA. Moreover, due to the larger parameter space of the objective function (9) for $|\mathbb{U}| > 1$ the cross entropy can only decrease. Closer investigation shows that in this case (12) is a near optimum approximation of (11).

We have thus found a (near) optimum pair (\mathbf{m}, \mathbf{q}) with respect to the objective function (9) and a given dissector \mathbf{d} .

From Theorem 1 it does not directly follow how to apply the transfer message (\mathbf{m}, \mathbf{q}) in iterative decoding. We now reintroduce superscripts to indicate constituent codes or the decoder where variables originate from. Given a message pair $(\mathbf{m}^{(h)}, \mathbf{q}^{(h)})$ from decoder $\mathcal{D}^{(h)}$ we first need a new dissector $\mathbf{d}^{(l)}$ from which then a new message $(\mathbf{m}^{(l)}, \mathbf{q}^{(l)})$ can be computed in $\mathcal{D}^{(l)}$, where $l, h = 1, 2, l \neq h$. Define by

$$\begin{aligned} H_{M,Q|R}(C|\mathbf{m}, \mathbf{q}|\mathbf{S}|\mathbf{d}) &:= \\ &- \sum_{s \in \mathbb{S}} P_{C|M,Q}(s|\mathbf{m}, \mathbf{q}) \cdot \log_2(P_{S|R}(s|\mathbf{d})) \end{aligned}$$

the cross entropy between the uncoded distribution of s given \mathbf{d} and the distribution of $s \in \mathbb{C}$ given the message pair (\mathbf{m}, \mathbf{q}) . A possible optimization rule for the dissectors $\mathbf{d}^{(l)}$ is given in the following.

$$\begin{aligned} \text{Proposal 1: Find the dissectors } \mathbf{d}^{(1)}, \mathbf{d}^{(2)} \text{ minimizing} \\ (\mathbf{d}^{(1)}, \mathbf{d}^{(2)}) = \arg \min_{(\mathbf{v}^{(1)}, \mathbf{v}^{(2)})} H_{M,Q|R}(C^{(1)}|\mathbf{m}^{(2)}, \mathbf{q}^{(2)}|\mathbf{S}|\mathbf{v}^{(1)}) \\ + H_{M,Q|R}(C^{(2)}|\mathbf{m}^{(1)}, \mathbf{q}^{(1)}|\mathbf{S}|\mathbf{v}^{(2)}) \end{aligned} \quad (13)$$

with $(\mathbf{m}^{(l)}, \mathbf{q}^{(l)})$, $l = 1, 2$ chosen to minimize

$$H_{R|M,Q}(C^{(l)}|\mathbf{r}|\mathbf{S}|\mathbf{m}^{(l)}, \mathbf{q}^{(l)})$$

given $\mathbf{d}^{(l)}$ according to Theorem 1.

To derive an algorithm from this proposal, compute the partial derivatives of the entropy terms in (13). We obtain

$$\begin{aligned} \frac{\partial}{\partial d_i^{(l)}} H_{M,Q|R}(C^{(l)}|\mathbf{m}^{(h)}, \mathbf{q}^{(h)}|\mathbf{S}|\mathbf{d}^{(l)}) \\ = \sum_{s \in \mathbb{S}} P_{C^{(l)}|M,Q}(s|\mathbf{m}^{(h)}, \mathbf{q}^{(h)}) \cdot \left(\tanh_2(d_i^{(l)}) - s_i \right), \end{aligned} \quad (14)$$

and the derivative of the second term is approximately zero. Hence we set (14) equal to zero and obtain

$$d_i^{(l)} = \frac{1}{2} \cdot \log_2 \frac{P_{S_i|M,Q}(+1|\mathbf{m}^{(h)}, \mathbf{q}^{(h)}, C^{(l)})}{P_{S_i|M,Q}(-1|\mathbf{m}^{(h)}, \mathbf{q}^{(h)}, C^{(l)})} \quad (15)$$

which is a calculation rule. Note that, though not explicitly stated in the formula, the computation (15) requires knowledge of $\mathbf{d}^{(h)}$ as $\mathbf{m}^{(h)}$ and $\mathbf{q}^{(h)}$ are functions of u .

The results in (12) and (15) motivate the Dissection Decoding Algorithm 2 for the decoding of a noisy IS code word. In the beginning, nothing is known about either constituent code and thus $\mathbf{d}^{(2)} = \mathbf{0}$, $\mathbf{m}^{(2)} = \mathbf{0}$ and $\mathbf{q}^{(2)} = \mathbf{1}$ are initialized as all-zero and all-one, respectively, which directly results in $\mathbf{d}^{(1)} = \mathbf{0}$ when assuming equiprobable code symbols. ‘Normal’ symbol beliefs $\mathbf{m}^{(1)}[u]$ are computed in $\mathcal{D}^{(1)}$ according to (12) and passed to $\mathcal{D}^{(2)}$. There the dissector $\mathbf{d}^{(2)}$

Algorithm 2 Dissection Decoding

-
- 1) *initialize*
 - set $l = 1$, $h = 2$, $\mathbf{d}^{(2)} = \mathbf{0}$, $\mathbf{m}^{(2)} = \mathbf{0}$, $\mathbf{q}^{(2)} = \mathbf{1}$
 - 2) *iterate*
 - **while** (stopping criterion not fulfilled)
 - $\mathcal{D}^{(l)} : (\mathbf{m}^{(h)}, \mathbf{q}^{(h)}, \mathbf{d}^{(h)}) \mapsto \mathbf{d}^{(l)}$ cf. (15)
 - $\mathbf{d}^{(l)} \mapsto (\mathbf{m}^{(l)}, \mathbf{q}^{(l)})$ cf. (10), (12)
 - swap $l \leftrightarrow h$
 - **end**
 - 3) *output* $\hat{\mathbf{c}} = \text{sgn}(\mathbf{d}^{(h)})$
-

is computed according to (15). Up to this point the algorithm is identical to the BPA and all computations can be accomplished with the BCJR algorithm. But rather than computing extrinsic symbol beliefs, $\mathbf{d}^{(2)}$ is taken to *dissect* (hence the name) the code space $\mathbb{C}^{(2)}$ and to compute the message pair $(\mathbf{m}^{(2)}, \mathbf{q}^{(2)})$ according to (10) and (12) with which the iterative procedure continues in $\mathcal{D}^{(1)}$.

IV. IMPLEMENTATION AND SIMULATION

For a dissector \mathbf{d} with non-zero real-valued elements d_i , the set size or *resolution* $|\mathbb{U}|$ is very large. The result would be a maximum likelihood (ML) decoder with huge matrices \mathbf{m} and thus impracticable decoding complexity. Therefore we uniformly quantize the elements of \mathbf{d} with a granularity Δ , and limit their magnitude to $|d_i| \leq d_{\max}$. Thus the set of possible word uncertainties is reduced to the values

$$H_{S|R}(\mathbf{s}|\mathbf{d}) \in \{u_{\min}, u_{\min} + 2 \cdot \Delta, u_{\min} + 4 \cdot \Delta, \dots\}$$

where

$$u_{\min} = \sum_{i=1}^n (\log_2(2^{d_i} + 2^{-d_i}) - |d_i|)$$

is the minimum possible word uncertainty given \mathbf{d} . We further limit the resolution $|\mathbb{U}|$ by setting

$$u(\mathbf{s}) = \begin{cases} H_{S|R}(\mathbf{s}|\mathbf{d}) & : H_{S|R}(\mathbf{s}|\mathbf{d}) \leq u_{\max} \\ u_{\max} & : \text{else} \end{cases}$$

with

$$u_{\max} = u_{\min} + (|\mathbb{U}| - 1) \cdot 2 \cdot \Delta.$$

The computations of the distributions in the matrices \mathbf{m} are accomplished in the constituent code trellises, cf. [11].

For easy comparison with the BPA we consider a Turbo code according to [3] with dimension $k = 20$ and terminated rate $R = \frac{1}{2}$ constituent codes with the generator polynomial $G(D) = [1 \frac{1+D+D^2}{1+D^2}]$. The choice of the rather small code dimension is on purpose as the BPA is known to not perform well for short codes, thus leaving room for improvement, and to keep the requirements for the resolution $|\mathbb{U}|$ small which grow with the code length. On the latter account, the dissector \mathbf{d} is allowed to take non-zero values only for the k systematic positions of the code. Figure 1 shows the simulation results for $\Delta = 0.1$, $d_{\max} = 4$, 8 decoding iterations and $|\mathbb{U}| = 50, 75, 100$. We observe that the error correcting performance is superior to Turbo decoding with the

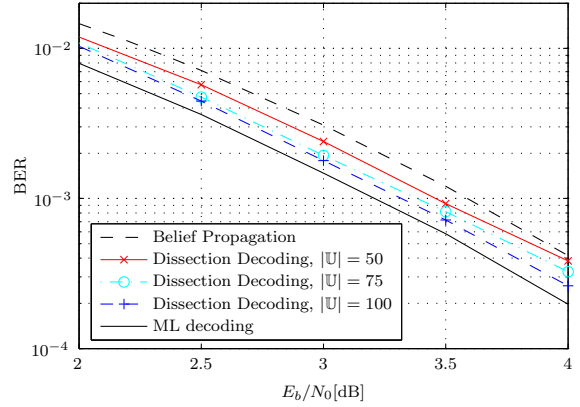


Figure 1. Error Correcting Performance for Turbo Code

BPA, and that it increases with $|\mathbb{U}|$ towards the maximum-likelihood (ML) bound. The gain compared to the BPA is up to 0.3 dB for $|\mathbb{U}| = 100$.

V. DISCUSSION & CONCLUSIONS

The proposed algorithm shows a distinct error correcting performance gain compared to belief propagation. However, the requirements for the set size $|\mathbb{U}|$ grow approximately proportional to the dissector length – and thus the code length – n . Taking into account the computation of two-dimensional functions over $u \in \mathbb{U}$ in the trellis of length n , the overall decoding complexity is $\mathcal{O}(n^3)$. Ongoing work focuses on GAUSSIAN approximation of the distributions over u [11], leading to $\mathcal{O}(n)$ as for the belief propagation algorithm.

REFERENCES

- [1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [2] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, “Turbo decoding as an instance of Pearl’s belief propagation algorithm,” *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 140–152, Feb. 1998.
- [3] 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.
- [4] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *Information Theory, IEEE Transactions on*, vol. 20, pp. 284 – 287, March 1974.
- [5] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, July, October 1948.
- [6] S. Benedetto and G. Montorsi, “Serial concatenation of block and convolutional codes,” *Electronics Letters*, vol. 32, pp. 887–888, May 1996.
- [7] D. J. MacKay and R. M. Neal, “Good codes based on very sparse matrices,” in *5th IMA Conference on Cryptography and Coding* (C. Boyd, ed.), no. 1025 in *Lecture Notes in Computer Science*, (Berlin, Germany), pp. 100–111, Springer, 1995.
- [8] U. Sorger, “Discriminated Belief Propagation,” Tech. Rep. TR-CSC-07-01, Computer Science and Communications Research Unit, Université du Luxembourg, 2007. <http://arxiv.org/abs/0710.5501>.
- [9] U. Sorger, *Communication Theory*. Books On Demand, 2009.
- [10] I. Land, *Reliability Information in Channel Decoding*. PhD thesis, Christian-Albrechts-University of Kiel, Faculty of Engineering, Information and Coding Theory Lab, 2005.
- [11] A. Heim, V. Sidorenko, and U. Sorger, “Computation of distributions and their moments in the trellis,” *Advances in Mathematics of Communications (AMC)*, vol. 2, pp. 373–391, November 2008.