Complexity-Reduced Iterative Source-Channel Decoding by Conditional Quantization

Laurent Schmalen, Peter Vary
Institute of Communication Systems
and Data Processing (ind)
RWTH Aachen University, Germany
{schmalen,vary}@ind.rwth-aachen.de

Marc Adrat
FGAN e.V.
FKIE/KOM
Wachtberg, Germany
adrat@fgan.de

Thorsten Clevorn
Infineon Technologies AG
COM Development Center NRW
Duisburg, Germany
thorsten.clevorn@infineon.com

***Abstract*— Iterative source-channel decoding (ISCD) exploits the residual redundancy of source codec parameters by using the Turbo principle. However, ISCD might require more computational complexity than available. The main reason is that the utilized soft decision source decoder (SDSD) can be computationally quite expensive. In this paper we propose a joint source-channel coding approach which reduces the computational complexity of the SDSD by slightly modifying the quantizer such that the complexity of the source decoder is reduced. The complexity of the SDSD is further reduced by transforming the SDSD equations into the logarithmic domain. We give analytical expressions for the expected quality loss by the modified quantizer, simulation results showing the overall ISCD system performance, as well as complexity figures.**

I. Introduction

With the discovery of Turbo codes, channel coding close to the Shannon limit has become possible with moderate computational complexity. In the past years, the Turbo principle of exchanging extrinsic information between separate channel decoders has also been extended to other receiver components. In a Turbo-like process the residual redundancy of source codec parameters such as scale factors or predictor coefficients for speech, audio, and video signals can be exploited by *iterative source-channel decoding* (ISCD) [1], [2]. This residual redundancy occurs due to imperfect source encoding resulting for instance from delay and complexity constraints. It can be utilized by a *soft decision source decoder* (SDSD) [3], [4] which exchanges extrinsic reliabilities with a channel decoder.

The execution of the SDSD, however, can be computationally quite demanding, especially if large quantizer codebooks are employed. In non-iterative transmission systems, it is possible to execute the SDSD only for the most significant bits, as proposed in [5]. However, if such a source decoder is utilized in an ISCD transmission scheme, the source decoder can only generate extrinsic information for the most significant bits, leading to a sub-optimal performance.

Therefore, we propose a different approach for a complexity-reduced ISCD receiver. It has been observed that quite a high number of certain pairs of consecutive quantized values $(\bar{u}_\kappa, \bar{u}_{\kappa-1})$ occur with small probabilities if the sequence is correlated. If the transmitter is modified in a way that these transitions are not allowed, the SDSD does not need anymore a fully developed trellis, but a trellis with a reduced number of state transitions. Note that only the number of state transitions is reduced while the number of states remains the same. This differs from the M-algorithm [6], which works with a reduced number of states.

This work was financed by the European Union under Grant FP6-2002-IST-C 020023-2 FlexCode

II. System Model

In the following, we will briefly review the *iterative source-channel decoding* (ISCD) system. In Fig. 1 the baseband model of ISCD is depicted. At time instant t a source encoder generates a frame $\underline{u}_t = (u_1, \dots u_{K_S})$ of K_S unquantized source codec parameters u_κ, with $\kappa \in \{1, \dots, K_S\}$ denoting the position within the frame. Each value u_κ is individually mapped to a quantizer reproduction level $\bar{u}_\kappa$, with $\bar{u}_\kappa \in \mathbb{U} = \{\bar{u}^{(1)}, \dots, \bar{u}^{(Q)}\}$. The set $\mathbb{U}$ denotes the quantizer codebook with a total number of $|\mathbb{U}| = Q$ codebook entries. A unique (bipolar) bit pattern $\mathbf{x}_\kappa \in \mathbb{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(Q)}\}$ of M^* bits (i.e., $\mathbb{X} \subseteq \{\pm 1\}^{M^*}$), with $M^* \geq \lceil \log_2 Q \rceil \doteq M$, is assigned to each quantizer level $\bar{u}_\kappa$ according to the index assignment $\Gamma(\bar{u}^{(i)}) = \mathbf{x}^{(i)}$. The single bits of a bit pattern $\mathbf{x}_\kappa$ are indicated by $x_\kappa(m)$, $m \in \{1, \dots, M^*\}$. If $M^* > \log_2 Q$, the index assignment Γ is called *redundant index assignment* [7] and can be considered to be the composite function $\Gamma = \Gamma_{\mathrm{R}} \circ \Gamma_{\mathrm{NB}}$ (i.e., $\Gamma(\bar{u}) = (\Gamma_{\mathrm{R}} \circ \Gamma_{\mathrm{NB}})(\bar{u}) = \Gamma_{\mathrm{R}}(\Gamma_{\mathrm{NB}}(\bar{u}))$). The function Γ_{NB} performs a non-redundant *natural binary* index assignment, i.e., the binary representation of the codebook index of $\bar{u}$ is assigned to $\Gamma_{\mathrm{NB}}(\bar{u})$. The function Γ_{R}, which

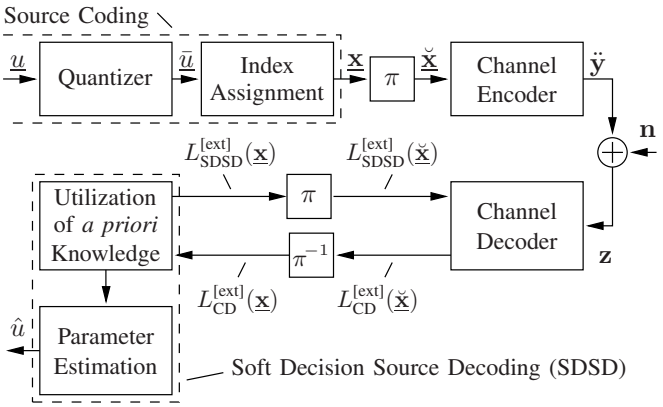

Fig. 1. Baseband model of the utilized ISCD system

is the redundancy introducing part, can be regarded as being a (linear or non-linear) block code of rate $r^{\mathsf{IA}} = M/M^*$. After index assignment, K_S bit patterns are grouped to a frame of bit patterns $\underline{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_{K_S})$ consisting of $K_S \cdot M^*$ bits. The frame $\underline{\mathbf{x}}$ of bits is then re-arranged by a bit interleaver π in a deterministic, pseudo-random like manner. The interleaved frame with $K_S \cdot M^*$ bits is denoted as $\breve{\underline{\mathbf{x}}}$.

For channel encoding of a frame $\breve{\underline{\mathbf{x}}}$, we use a convolutional code of constraint length $J+1$ and of rate r^{C}. In general, any channel code can be used as long as the respective decoder is able to provide the required *extrinsic reliabilities*. In this paper, we restrict ourselves to rate $r^{\mathsf{C}} = 1$, recursive, non-systematic convolutional codes as it has been shown [8] that the inner code of a serially concatenated system should be recursive in order to be capacity achieving. For the termination of the code, J tail bits are appended to $\breve{\underline{\mathbf{x}}}$. The encoded frame of length $K_S \cdot M^* + J$ is denoted by $\underline{\mathbf{y}}$. The bits y_k of $\underline{\mathbf{y}}$ are indexed by $k \in \{1, \ldots, K_S \cdot M^* + J\}$. Prior to transmission over the channel, the encoded bits y_k are mapped to bipolar values $\ddot{y}_k$ forming a sequence $\ddot{\mathbf{y}} \in \{\pm 1\}^{K_S \cdot M^* + J}$. We only consider BPSK modulation in this paper in order to demonstrate the concept, which can easily be extended to include higher order modulation schemes [9], [7].

On the channel, the modulation symbols $\ddot{y}_k$ (with symbol energy $E_{\mathrm{s}} = 1$) are subject to additive white Gaussian noise (AWGN) with known variance $\sigma_n^2 = N_0/2$.

The received symbols $z_k \in \{\pm 1\}$ are transformed to L-values [10] prior to being evaluated in a Turbo process which exchanges *extrinsic* reliabilities between the channel decoder (CD) and the soft decision source decoder (SDSD).

The channel decoder used in this paper is based on the LogMAP algorithm [11], [10]. For the derivations of the equations for computing the *extrinsic* probabilities of the SDSD, we refer the reader to the literature, e.g., [2], [3], [12], [13]. In Section IV, we will briefly revise the SDSD equations and give expressions in the logarithmic domain in order to evaluate the complexity and the complexity savings of the proposed algorithms.

III. Conditional Scalar Quantization

In this section we present the concept of conditional scalar quantization, which enables a very efficient realization (in terms of computational complexity) of the soft decision source decoder (SDSD). Although we present the concept for scalar quantization and a first order Markov model only, the extension to vector quantization as well as higher order Markov models is straightforward.

If $\mathbb{U}$ denotes the original quantizer codebook, let $\mathcal{C} = \{\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(Q)}\}$ denote the set of all quantization cells with

$$\mathcal{C}^{(q)} = \{u : |u - \bar{u}^{(q)}| < |u - \bar{u}^{(\ell)}|, \forall \bar{u}^{(\ell)} \in \mathbb{U}, \bar{u}^{(\ell)} \neq \bar{u}^{(q)}\}. \quad (1)$$

Conditional quantization exploits the correlation between successive samples in such a way that the quantization of the current value u_κ depends on the previously quantized value $\bar{u}_{\kappa-1}^{(i)}$. For quantizing the current sample u_κ, the conditional quantizer only considers codebook entries $\bar{u}^{(j)}$ where the conditional probability $P(\bar{u}_\kappa^{(j)} | \bar{u}_{\kappa-1}^{(i)})$ is above a certain threshold $\mathcal{T}$. We define a set of reduced codebooks $\mathbb{U}_{\text{red},i}$ with

$$\mathbb{U}_{\text{red},i} = \left\{ \bar{u}_\kappa^{(j)} : P\left(\bar{u}_\kappa^{(j)} | \bar{u}_{\kappa-1}^{(i)}\right) > \mathcal{T}, \forall\, \bar{u}_\kappa^{(j)} \in \mathbb{U} \right\}. \quad (2)$$

The conditional quantizer uses the reduced codebook $\mathbb{U}_{\text{red},i}$ to quantize the sample u_κ if the previous sample has been quantized to $\bar{u}_{\kappa-1}^{(i)}$. Let $|\mathbb{U}_{\text{red},i}|$ denote the number of entries in the reduced codebook $\mathbb{U}_{\text{red},i}$. The total number of allowed transitions $\bar{u}_{\kappa-1}^{(i)} \to \bar{u}_\kappa^{(j)}$ is thus reduced from $\mathcal{N}' = Q^2$ to

$$\mathcal{N} \doteq \sum_{i=1}^{Q} |\mathbb{U}_{\text{red},i}|. \quad (3)$$

This (reduced) number of transitions is directly linked to the complexity of the source decoder as shall be seen in Section IV. Let $\mathbb{X}_{\text{red},i}$ denote the set of all bit patterns assigned to the reduced codebook $\mathbb{U}_{\text{red},i}$. Furthermore, we define

$$\mathbb{U}'_{\text{red},j} \doteq \left\{ \bar{u}_{\kappa-1}^{(i)} : \bar{u}_\kappa^{(j)} \in \mathbb{U}_{\text{red},i}, \forall\, i \in \{1, \ldots, Q\} \right\} \quad (4)$$

to be the set of all codebook entries $\bar{u}_{\kappa-1}^{(i)}$ that allow a transition $\bar{u}_{\kappa-1}^{(i)} \to \bar{u}_\kappa^{(j)}$. Again, $\mathbb{X}'_{\text{red},j}$ denotes the set of assigned bit patterns to the entries of $\mathbb{U}'_{\text{red},j}$.

Note that the utilization of conditional quantization also modifies the *a priori* knowledge of first order which is exploited in the source decoder. We denote this modified conditional probability $P_{\text{red}}(\bar{u}^{(j)} | \bar{u}^{(i)})$ with $\bar{u}^{(j)} \in \mathbb{U}_{\text{red},i}$ and $\bar{u}^{(i)} \in \mathbb{U}$. Again, for the conditional quantizer, quantization cells $\mathcal{C}_{\text{red},i}^{(q)}$ with

$$\mathcal{C}_{\text{red},i}^{(q)} = \{u : |u - \bar{u}^{(q)}| < |u - \bar{u}^{(\ell)}|, \forall \bar{u}^{(\ell)} \in \mathbb{U}_{\text{red},i}, \bar{u}^{(\ell)} \neq \bar{u}^{(q)}\}$$

can be defined for $q = 1, \ldots, |\mathbb{U}_{\text{red},i}|$. For a given (stationary) source with (two-dimensional) joint probability function $p_U(u_\kappa, u_{\kappa-1}) = p_U(u_\kappa | u_{\kappa-1}) \cdot p_U(u_{\kappa-1})$ the quantization noise amounts to (e.g. [4], [14])

$$N = \sum_{\bar{u}_{\kappa-1}^{(i)} \in \mathbb{U}} \sum_{\bar{u}_\kappa^{(j)} \in \mathbb{U}_{\text{red},i}} \int_{\mathcal{C}^{(i)}} \int_{\mathcal{C}_{\text{red},i}^{(j)}} \left(\zeta - \bar{u}_\kappa^{(j)}\right)^2 p_U(\zeta, \nu)\, \mathrm{d}\zeta\, \mathrm{d}\nu. \quad (5)$$

The quantization noise is determined by considering all possible previous samples $\bar{u}_{\kappa-1}^{(i)} \in \mathbb{U}$ and then calculating the quantization noise amount of the pair $(\bar{u}_{\kappa-1}^{(i)}, \bar{u}_\kappa^{(j)})$ with $\bar{u}_\kappa^{(j)} \in \mathbb{U}_{\text{red},i}$ by solving the double integral in (5). The total quantization noise is then obtained by summing over all combinations of $(\bar{u}_{\kappa-1}^{(i)}, \bar{u}_\kappa^{(j)})$

As an example, we assume that the source realizes a Gauss-Markov process of first order with correlation ρ, zero mean and variance $\sigma_u^2 = 1$. The two-dimensional joint distribution of the source amounts to (e.g., [4])

$$p_U(u_\kappa, u_{\kappa-1}) = \frac{1}{2\pi\sqrt{1-\rho^2}} \cdot \mathrm{e}^{-\frac{u_\kappa^2 + u_{\kappa-1}^2 - 2\rho u_\kappa u_{\kappa-1}}{2(1-\rho^2)}}. \quad (6)$$

Figure 2 depicts the number of transitions $\mathcal{N}$ as a function of the threshold $\mathcal{T}$ for $\rho \in \{0.5; 0.7; 0.9\}$ and $Q = 16$ quantizer

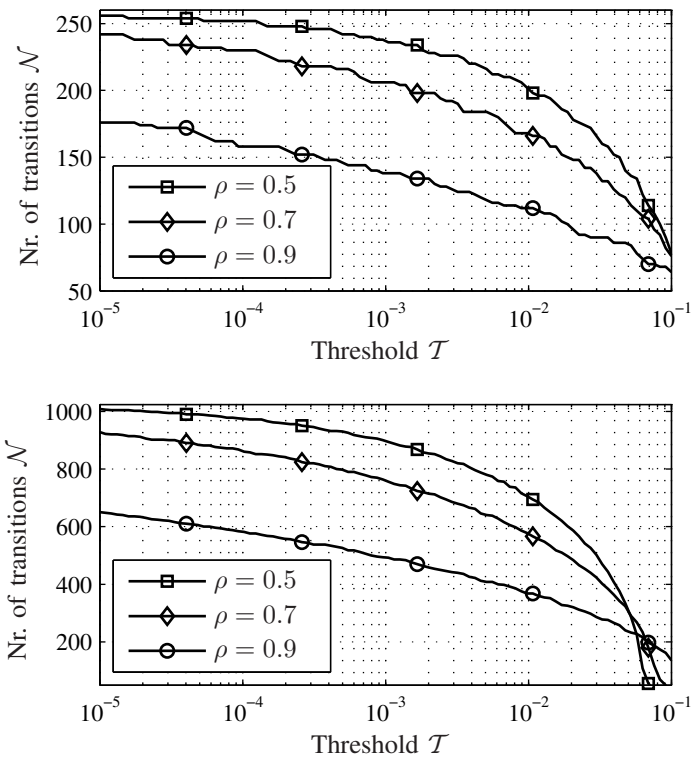

Fig. 2. Number of transitions $\mathcal{N}$ as a function of the threshold $\mathcal{T}$ for $Q = 16$ (top) and $Q = 32$ (bottom).

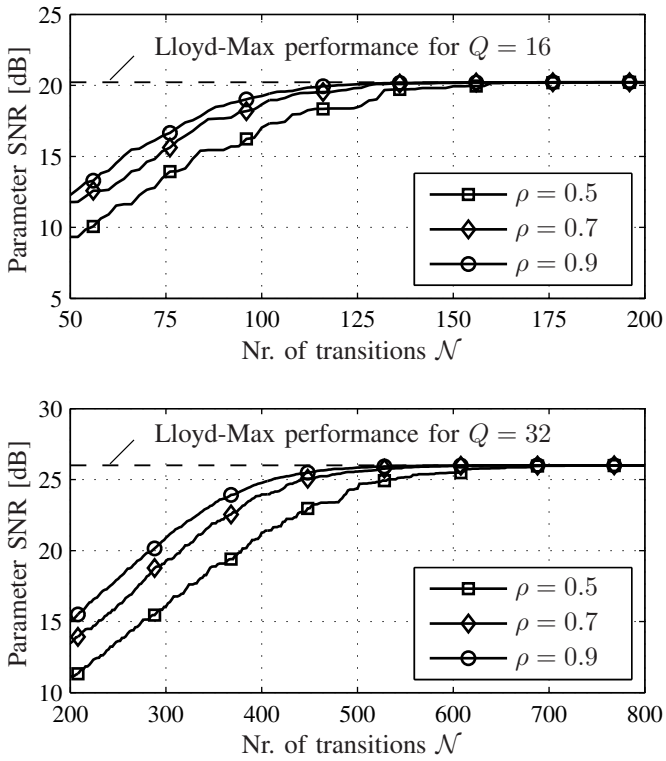

Fig. 3. Parameter SNR as a function of the number of transitions for different values of auto-correlation ρ for $Q = 16$ (top) and $Q = 32$ (bottom).

levels (top subplot) as well as for $Q = 32$ (bottom subplot). As it is expected intuitively, a higher correlation ρ leads to a lower number of transitions $\mathcal{N}$ as transitions $u_{\kappa-1} \rightarrow u_\kappa$ with $|u_\kappa - u_{\kappa-1}| \gg 0$ occur less frequently.

The most interesting question however is how much the signal quality is affected by conditional quantization. Therefore, (5) is evaluated for the same source and the signal-to-noise ratio after quantization is determined as a function of the reduced number of transitions $\mathcal{N}$. The original codebook $\mathbb{U}$ is assumed to be the optimum Lloyd-Max codebook [14]. The results are depicted in Fig. 3, for $Q = 16$ (top subplot) and $Q = 32$ (bottom subplot). It can be seen that for $\rho \geq 0.7$ the number of transitions can be halved (e.g., from $Q^2 = 1024$ to $\mathcal{N} \approx 500$ for $Q = 32$) without affecting the SNR considerably.

Note that although we introduced the concept of conditional quantization for intra-frame correlation only, it is also easily applicable to inter-frame correlation.

IV. Complexity Considerations

In Section IV-A we first revise the SDSD equations [1], [2], [13] and then modify the expressions such that the SDSD operates in the logarithmic domain in Section IV-B. The given expressions are already modified in such a way that conditional quantization is incorporated. The full expressions are needed in order to determine the complexity figures in Section IV-C.

A. SDSD Revisited

The SDSD may be interpreted as a modification of the well-known BCJR algorithm [11], operating on a fully developed trellis diagram. In this section, we assume that only intra-frame correlation is exploited by the SDSD, the extension towards inter-frame correlation is straightforward by exchanging the position indices κ with time indices t.

The input to the soft decision source decoder (SDSD) are the extrinsic L-values generated by the channel decoder

$$\begin{aligned} L_{\text{SDSD}}^{[\text{input}]}(x_\kappa(m)) &= L_{\text{CD}}^{[\text{ext}]}(x_\kappa(m)) \\ &= \ln\left(\frac{P_{\text{CD}}^{[\text{ext}]}(x_\kappa(m)=+1)}{P_{\text{CD}}^{[\text{ext}]}(x_\kappa(m)=-1)}\right). \end{aligned} \tag{7}$$

The first step of the SDSD consists in determining the factors $\theta(\mathbf{x}_\kappa^{(j)})$ for each distinct bit pattern $\mathbf{x}_\kappa^{(j)}$ with

$$\theta(\mathbf{x}_\kappa^{(j)}) = \exp\left(\sum_{m=1}^{M^*} \frac{x_\kappa^{(j)}(m)}{2} L_{\text{SDSD}}^{[\text{input}]}(x_\kappa(m))\right). \tag{8}$$

Note that $x_\kappa^{(j)}(m) \in \{\pm 1\}$. The forward and backward recursion of the SDSD are given by

$$\alpha(\mathbf{x}_\kappa^{(j)}) = \theta(\mathbf{x}_\kappa^{(j)}) \sum_{\mathbf{x}_{\kappa-1}^{(i)} \in \mathbb{X}'_{\text{red},\,j}} \alpha(\mathbf{x}_{\kappa-1}^{(i)}) P_{\text{red}}(\mathbf{x}_\kappa^{(j)} | \mathbf{x}_{\kappa-1}^{(i)}) \tag{9}$$

$$= \theta(\mathbf{x}_\kappa^{(j)}) \cdot A(\mathbf{x}_\kappa^{(j)}) \tag{10}$$

$$\beta(\mathbf{x}_{\kappa-1}^{(i)}) = \sum_{\mathbf{x}_\kappa^{(j)} \in \mathbb{X}_{\text{red},\,i}} \beta(\mathbf{x}_\kappa^{(j)}) \theta(\mathbf{x}_\kappa^{(j)}) P_{\text{red}}(\mathbf{x}_\kappa^{(j)} | \mathbf{x}_{\kappa-1}^{(i)}) \tag{11}$$

with the initialization $\alpha(\mathbf{x}_0^{(\ell)}) = P(\mathbf{x}^{(\ell)})$, $\beta(\mathbf{x}_{K_S}^{(\ell)}) = 1$, $\forall \ell \in \{1, \ldots, Q\}$, (see, e.g., [13]). We furthermore define

$A(\mathbf{x}_\kappa^{(j)}) \doteq \sum_{\mathbf{x}_{\kappa-1}^{(i)} \in \mathbb{X}'_{\text{red},j}} \alpha(\mathbf{x}_{\kappa-1}^{(i)}) P_{\text{red}}(\mathbf{x}_\kappa^{(j)} | \mathbf{x}_{\kappa-1}^{(i)})$. Of course, the summations in (9) and (11) only have to be evaluated for those pairs of $(\mathbf{x}_{\kappa-1}^{(i)}, \mathbf{x}_\kappa^{(j)})$ with an existing transition between the corresponding $(\bar{u}_{\kappa-1}^{(i)}, \bar{u}_\kappa^{(j)})$ (i.e., $P(\bar{u}_\kappa^{(j)} | \bar{u}_{\kappa-1}^{(i)}) > \mathcal{T}$). Further note that due to conditional quantization the exploited *a priori* information changes. Therefore $P_{\text{red}}(\mathbf{x}_\kappa^{(j)} | \mathbf{x}_{\kappa-1}^{(i)})$ has to be utilized instead of $P(\mathbf{x}_\kappa^{(j)} | \mathbf{x}_{\kappa-1}^{(i)})$ (see also Section III).

The final step of the SDSD consists in determining the extrinsic information for each bit $x_\kappa(m)$ which is given by [1] (in terms of L-values)

$$L_{\text{SDSD}}^{[\text{ext}]}(x_\kappa(m)) = \ln \frac{\sum_{\substack{j=1 \\ x_\kappa^{(j)}(m)=+1}}^{Q} \beta(\mathbf{x}_\kappa^{(j)}) \theta_m^{[\text{ext}]}(\mathbf{x}_\kappa^{(j)}) A(\mathbf{x}_\kappa^{(j)})}{\sum_{\substack{j=1 \\ x_\kappa^{(j)}(m)=-1}}^{Q} \beta(\mathbf{x}_\kappa^{(j)}) \theta_m^{[\text{ext}]}(\mathbf{x}_\kappa^{(j)}) A(\mathbf{x}_\kappa^{(j)})} \quad (12)$$

with

$$\theta_m^{[\text{ext}]}(\mathbf{x}_\kappa^{(j)}) = \exp\left(\sum_{\substack{\ell=1 \\ \ell \neq m}}^{M^*} \frac{x_\kappa^{(j)}(\ell)}{2} L_{\text{SDSD}}^{[\text{input}]}(x_\kappa(\ell)) \right). \quad (13)$$

B. SDSD in the Logarithmic Domain

In a practical implementation the translation of the BCJR algorithm to the logarithmic domain offers several advantages such as, e.g., better numerical stability [15]. In the following, we derive the equations for the SDSD in the logarithmic domain. Therefore, we define $\tilde{\theta}(\mathbf{x}_\kappa^{(j)}) \doteq \ln \theta(\mathbf{x}_\kappa^{(j)})$, $\tilde{\alpha}(\mathbf{x}_\kappa^{(j)}) \doteq \ln \alpha(\mathbf{x}_\kappa^{(j)})$, as well as $\tilde{\beta}(\mathbf{x}_\kappa^{(j)}) \doteq \ln \beta(\mathbf{x}_\kappa^{(j)})$. With (8), the expression of $\tilde{\theta}(\mathbf{x}_\kappa^{(j)})$ becomes

$$\tilde{\theta}(\mathbf{x}_\kappa^{(j)}) = \sum_{m=1}^{M^*} \frac{1}{2} x_\kappa^{(j)}(m) L_{\text{SDSD}}^{[\text{input}]}(x_\kappa(m)). \quad (14)$$

Taking the natural logarithm of (9) and using $\tilde{\alpha}(\mathbf{x}_\kappa^{(j)})$ as well as $\tilde{\theta}(\mathbf{x}_\kappa^{(j)})$ leads to (with $\tilde{P}_{\text{red}}(\mathbf{x}_\kappa^{(j)} | \mathbf{x}_{\kappa-1}^{(i)}) \doteq \ln P_{\text{red}}(\mathbf{x}_\kappa^{(j)} | \mathbf{x}_{\kappa-1}^{(i)})$) [15]

$$\begin{aligned} \tilde{\alpha}(\mathbf{x}_\kappa^{(j)}) &= \tilde{\theta}(\mathbf{x}_\kappa^{(j)}) + \ln \left(\sum_{\mathbf{x}_{\kappa-1}^{(i)} \in \mathbb{X}'_{\text{red},j}} \mathrm{e}^{\ln\left(\alpha(\mathbf{x}_{\kappa-1}^{(i)}) \cdot P_{\text{red}}(\mathbf{x}_\kappa^{(j)} | \mathbf{x}_{\kappa-1}^{(i)})\right)} \right) \\ &= \tilde{\theta}(\mathbf{x}_\kappa^{(j)}) + \ln \left(\sum_{\mathbf{x}_{\kappa-1}^{(i)} \in \mathbb{X}'_{\text{red},j}} \mathrm{e}^{\tilde{\alpha}(\mathbf{x}_{\kappa-1}^{(i)}) + \tilde{P}_{\text{red}}(\mathbf{x}_\kappa^{(j)} | \mathbf{x}_{\kappa-1}^{(i)})} \right) \quad (15) \end{aligned}$$

The expression $\ln\left(\mathrm{e}^{\delta_1} + \mathrm{e}^{\delta_2}\right)$ which is part of (15) can be computed using the Jacobian logarithm [15]:

$$\begin{aligned} \ln\left(\mathrm{e}^{\delta_1} + \mathrm{e}^{\delta_2}\right) &= \max(\delta_1, \delta_2) + f_c(|\delta_1 - \delta_2|) \\ &\doteq \max^*(\delta_1, \delta_2) \quad (16) \end{aligned}$$

with $f_c(\zeta) = \ln\left(1 + \mathrm{e}^{-\zeta}\right)$ and $\ln\left(\mathrm{e}^{\delta_1} + \mathrm{e}^{\delta_2} + \mathrm{e}^{\delta_3} + \ldots\right) = \max^*(\delta_1, \delta_2, \delta_2, \ldots) = \max^*(\delta_1, \max^*(\delta_2, \max^*(\delta_3, \ldots)))$. Furthermore, $\max^*(\delta_1, -\infty) = \max^*(-\infty, \delta_1) = \delta_1$. The $\max^*$ function can be efficiently implemented using, e.g., a lookup table.

Using the $\max^*$ function, (15) can then be rewritten as

$$\tilde{\alpha}(\mathbf{x}_\kappa^{(j)}) = \tilde{\theta}(\mathbf{x}_\kappa^{(j)}) + \max_{\mathbf{x}_{\kappa-1}^{(i)} \in \mathbb{X}'_{\text{red},j}}{}^* \left(\tilde{\alpha}(\mathbf{x}_{\kappa-1}^{(i)}) + \tilde{P}_{\text{red}}(\mathbf{x}_\kappa^{(j)} | \mathbf{x}_{\kappa-1}^{(i)}) \right). \quad (17)$$

Similarly, the backward recursion (11) can be rewritten as

$$\tilde{\beta}(\mathbf{x}_{\kappa-1}^{(i)}) = \max_{\mathbf{x}_\kappa^{(j)} \in \mathbb{X}_{\text{red},i}}{}^* \left(\tilde{\beta}(\mathbf{x}_\kappa^{(j)}) + \tilde{\theta}(\mathbf{x}_\kappa^{(j)}) + \tilde{P}_{\text{red}}(\mathbf{x}_\kappa^{(j)} | \mathbf{x}_{\kappa-1}^{(i)}) \right). \quad (18)$$

The determination of the extrinsic information (12) can also be expressed using the $\max^*$ operator

$$\begin{aligned} L_{\text{SDSD}}^{[\text{ext}]}(x_\kappa(m)) = &\max_{\substack{j=1 \\ x_\kappa^{(j)}(m)=+1}}^{Q}{}^* \left(\tilde{\alpha}(\mathbf{x}_\kappa^{(j)}) + \tilde{\beta}(\mathbf{x}_\kappa^{(j)}) - \frac{1}{2} L_{\text{SDSD}}^{[\text{input}]}(x_\kappa(m)) \right) \\ &- \max_{\substack{j=1 \\ x_\kappa^{(j)}(m)=-1}}^{Q}{}^* \left(\tilde{\alpha}(\mathbf{x}_\kappa^{(j)}) + \tilde{\beta}(\mathbf{x}_\kappa^{(j)}) + \frac{1}{2} L_{\text{SDSD}}^{[\text{input}]}(x_\kappa(m)) \right) \quad (19) \end{aligned}$$

by consecutively exploiting the facts that $\ln \theta_m^{[\text{ext}]}(\mathbf{x}_\kappa^{(j)}) = \tilde{\theta}(\mathbf{x}_\kappa^{(j)}) - \frac{1}{2} x_\kappa^{(j)}(m) L_{\text{SDSD}}^{[\text{input}]}(x_\kappa(m))$ (compare (8) and (13)) and $\tilde{\theta}(\mathbf{x}_\kappa^{(j)}) + \ln A(\mathbf{x}_\kappa^{(j)}) = \tilde{\alpha}(\mathbf{x}_\kappa^{(j)})$ (see (10)).

The last step of the SDSD consists in estimating the parameters $\hat{u}$ which is done here using an MMSE estimation

$$\hat{u} = \sum_{i=1}^{Q} \bar{u}^{(i)} \exp\left(\tilde{\alpha}(\mathbf{x}_\kappa^{(i)}) + \tilde{\beta}(\mathbf{x}_\kappa^{(i)}) + \tilde{C}_1 \right) \quad (20)$$

with the constant $\tilde{C}_1 \in \mathbb{R}$ which is chosen such that $\sum_{i=1}^{Q} \exp\left(\tilde{\alpha}(\mathbf{x}_\kappa^{(i)}) + \tilde{\beta}(\mathbf{x}_\kappa^{(i)}) + \tilde{C}_1 \right) \stackrel{!}{=} 1$. Note that the parameter estimation only has to be performed once per frame and not for each iteration.

C. Complexity of the SDSD

The evaluation of (14) requires $Q \cdot M^*$ additions per parameter as the $\tilde{\theta}(\mathbf{x}_\kappa^{(j)})$ have to be determined for each possible bit pattern $\mathbf{x}_\kappa^{(j)} \in \mathbb{X}$. The factors $\frac{1}{2} L_{\text{SDSD}}^{[\text{input}]}(x_\kappa(m))$ can be calculated and stored (as they are needed a second time in the run-time of the algorithm) using M^* multiplications per parameter. The multiplication by $x_\kappa(m)$ corresponds to a sign change only as $x_\kappa(m) \in \{\pm 1\}$. The forward and backward recursions also have to be calculated for each $\mathbf{x}_\kappa^{(j)} \in \mathbb{X}$. In the case of conventional SDSD (i.e., $|\mathbb{U}_{\text{red},i}| = |\mathbb{X}_{\text{red},i}| = |\mathbb{U}'_{\text{red},i}| = |\mathbb{X}'_{\text{red},i}| = Q$), the evaluation of (17) requires Q^2 $\max^*$ operations as well as $Q + Q^2$ additions per parameter while the evaluation of (18) requires Q^2 $\max^*$ operations and $2Q^2$ additions per parameter. If conditional quantization is utilized, the number of $\max^*$ operations each reduces to $\mathcal{N}$ and the number of additions to $Q + \mathcal{N}$ (forward recursion), respectively $2\mathcal{N}$ (backward recursion). Finally, the evaluation of (19) requires $M^* Q$ $\max^*$ operations as well as $M^*(2Q+1)$ additions for the conventional as well as for the reduced-complexity SDSD.

In some cases (e.g. if inter-frame correlation is exploited) only the forward recursion can be carried out due to delay

TABLE I

OPERATIONS PER PARAMETERS NEEDED BY THE SDSD

• Full algorithm:

	Standard	Conditional Quantization
$\max^*$	$2Q^2 + M^*Q$	$2\mathcal{N} + M^*Q$
ADD	$3Q^2 + (3M^* + 1)Q + M^*$	$3\mathcal{N} + (3M^* + 1)Q + M^*$
MUL	M^*	M^*

• Forward only algorithm:

	Standard	Conditional Quantization
$\max^*$	$Q^2 + M^*Q$	$\mathcal{N} + M^*Q$
ADD	$Q^2 + (2M^* + 1)Q + M^*$	$\mathcal{N} + (2M^* + 1)Q + M^*$
MUL	M^*	M^*

constraints. In this case $\tilde{\beta}(\mathbf{x}_\kappa^{(j)})$ is set to zero in (19) as the backward recursion (18) does not need to be carried out. Table I summarizes the number of operations required for both the standard SDSD and the complexity reduced SDSD by conditional quantization. Both cases are considered: the forward/backward algorithm and the forward-only variant. Note that we do not consider the complexity of the parameter estimation as this step, which is only performed once per frame, is required in any case. Thus the complexity figures given here only include the operations performed in the block denoted "Utilization of *a priori* knowledge" in Fig. 1.

V. SIMULATION RESULTS

The capabilities of the proposed ISCD system are demonstrated by two simulation examples. The *parameter signal-to-noise ratio* (SNR) between the originally generated parameters u and the reconstructed estimated parameters $\hat{u}$ is used for quality evaluation. The parameter SNR is plotted for different values of E_u/N_0, with E_u denoting the energy per source parameter u ($E_u = M^* \cdot \frac{1}{r^{\mathsf{C}}} \cdot E_\mathrm{s}$). The source is realized by a Gauss-Markov (autoregressive) process with correlation coefficient ρ fixed to $\rho = 0.9$. This autocorrelation value can be observed in typical speech and audio codecs, e.g., for the scale factors in CELP codecs or MP3. The utilized channel code is a rate $r^{\mathsf{C}} = 1$ recursive non-systematic convolutional code of constraint length $J = 4$ with generator polynomial $G^{\mathsf{C}}(D) = \left(\frac{1}{1+D+D^2+D^3}\right)$. The non-iterative reference scheme uses optimized components for non-iterative systems, i.e., a natural binary index assignment with $M^* = M = \log_2\lceil Q \rceil$ and a rate $r^{\mathsf{C}} = \frac{1}{2}$ recursive, systematic convolutional code of constraint length $J = 4$ with $G^{\mathsf{C}}(D) = \left(\frac{1+D^2+D^3}{1+D+D^3}\right)$.

In a first experiment (denoted "experiment A"), we assume that the source exhibits intra-frame correlation, i.e., the single elements u_κ of $\underline{u}_t$ are modelled by a 1st order Gauss-Markov process and a frame consists of $K_S = 50000$ parameters. The quantization is performed using a $Q = 16$ level Lloyd-Max codebook $\mathbb{U}$ and the redundant index assignment Γ_R consists of the $(8, 4)$ block code with generator matrix

$$\mathbf{G}^{\Gamma}_{\mathrm{BC}(8,4)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{21}$$

proposed in [16]. This index assignment results in $M^* = 8$ bit. The overall coding rate of the system amounts to $r^{\mathsf{IA}} \cdot r^{\mathsf{C}} = \frac{1}{2}$.

TABLE II

OPERATIONS PER PARAMETER AND ITERATION NEEDED BY THE SDSD FOR EXPERIMENT A (TOP) AND EXPERIMENT B (BOTTOM)

$Q = 16$ $M^* = 8$ Forw./Backw.	Standard	Conditional $\mathcal{T} = 0.01$ $\mathcal{N} = 112$	 $\mathcal{T} = 0.03$ $\mathcal{N} = 90$	 $\mathcal{T} = 0.05$ $\mathcal{N} = 84$
$\max^*$	640	352	308	296
ADD	1176	744	678	660

$Q = 32$ $M^* = 10$ Forw. only	Standard	Conditional $\mathcal{T} = 0.005$ $\mathcal{N} = 408$	 $\mathcal{T} = 0.01$ $\mathcal{N} = 368$	 $\mathcal{T} = 0.03$ $\mathcal{N} = 284$
$\max^*$	1344	728	688	604
ADD	1706	1090	1050	966

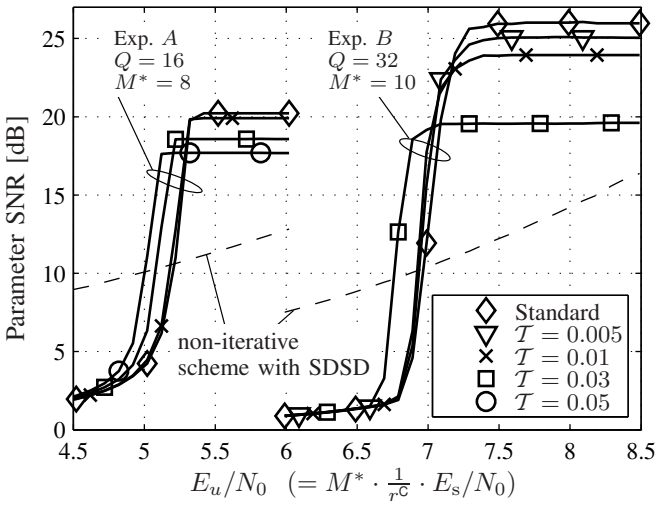

Fig. 4. Simulation results for experiment A and experiment B

At the receiver, 15 iterations are carried out. The simulation results are depicted in Fig. 4 for the standard SDSD with "regular" quantization as well as for the complexity-reduced system with conditional quantization for $\mathcal{T} \in \{0.01; 0.03; 0.05\}$. Note that the depicted non-iterative scheme uses SDSD while today's systems with hard-decision source decoding perform even worse. The resulting number of transitions $\mathcal{N}$ as well as the number of required $\max^*$ operations and additions are summarized in Table II. As predicted by (5), the maximum attainable parameter SNR is reduced due to the influence of the conditional quantizer. Interestingly, the lower the number of transitions (i.e., for higher values of $\mathcal{T}$), the more the waterfall region is moved towards lower channel qualities.

This behavior can be explained by an EXIT chart analysis [17]: The EXIT characteristics for the conventional SDSD as well as for the complexity-reduced SDSD with conditional quantization are depicted in Fig. 5-a). The area $\mathcal{A}_{\mathrm{SDSD}}$ under the EXIT curve of the complexity-reduced SDSD is larger than the area under the EXIT curve of the conventional SDSD. This leads to an earlier convergence (for lower channel qualities) as the area underneath the EXIT characteristic is linked to the channel quality where the waterfall region occurs [18]. The larger area is obvious as $1 - \mathcal{A}_{\mathrm{SDSD}}$ is a function of the

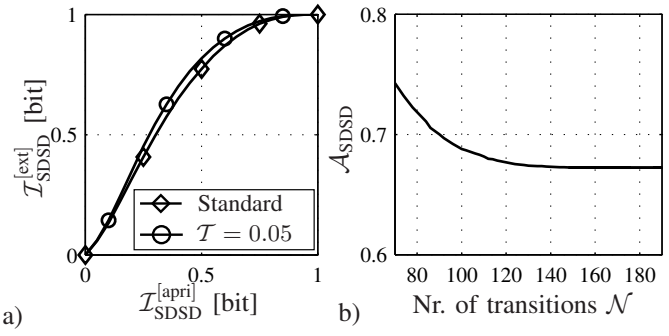

Fig. 5. EXIT chart analysis for $Q = 16$ and $M^* = 8$ (index assignment generator matrix from (21)) for conventional SDSD ("Standard") and for conditional quantization SDSD with $\mathcal{T} = 0.05$, i.e., $\mathcal{N} = 84$. Area under EXIT characteristic as a function of $\mathcal{N}$.

conditional entropy $H(\bar{u}_\kappa|\bar{u}_{\kappa-1})$ [19], which is reduced by the conditional quantizer. The reduction is caused by the fact that now $P_{\text{red}}(\mathbf{x}_\kappa^{(j)}|\mathbf{x}_{\kappa-1}^{(i)})$ is exploited by the SDSD (instead of $P(\mathbf{x}_\kappa^{(j)}|\mathbf{x}_{\kappa-1}^{(i)})$). Figure 5-b) depicts $\mathcal{A}_{\text{SDSD}}$ as a function of the number of transitions $\mathcal{N}$ for the given setup.

In a second experiment (denoted "experiment B") we assume that the source exhibits inter-frame correlation, i.e., all the single elements u_κ of $\underline{u}_t$ are assumed to be statistically independent from each other. The different samples u_κ are correlated with their counterpart from previous frames. In this experiment, a frame consists of $K_S = 250$ parameters. In order not to introduce any additional delay, the forward-only SDSD has to be employed. The block coded index assignment Γ_{R} utilized for experiment B corresponds to a $(31, 26)$ BCH code shortened to $(10, 5)$ [20]. The generator matrix of the shortened systematic code is given by

$$\mathbf{G}_{\text{BCH}(10,5)}^{\Gamma} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}. \qquad (22)$$

Note that we do not use a specially optimized index assignment but a standard small block code for demonstrating the concept. One possibility to optimize the index assignments are irregular index assignments introduced in [21].

At the receiver 20 iterations are carried out. The simulation results are depicted in Fig. 4 and the number of utilized operations are summarized in Table II. The results behave as expected: again a shift of the waterfall region towards lower channel qualities is observed.

VI. Conclusion

In this paper we presented a quantizer modification called conditional quantization which leads to an efficient, complexity-reduced soft-decision source decoder (SDSD) employed in an iterative source-channel decoding scheme. We have derived an analytical expression for the quality loss resulting from the non-optimum quantizer. Furthermore we have modified the expressions for determining the extrinsic information of the SDSD leading to a numerically stable, low-complexity implementation in the logarithmic domain and further reduced the complexity by adapting the SDSD to the conditional quantizer. Simulation results confirm the expected behavior. A similar reconstruction quality is observed with almost half the number of operations. It has been observed that the utilization of conditional quantization shifts the waterfall region towards lower channel qualities. This behavior can be explained by the EXIT chart, using the fact that the modified quantizer reduces the conditional entropy between consecutive source samples and thus modifies the area under the EXIT characteristic.

References

[1] M. Adrat, P. Vary, and J. Spittka, "Iterative Source-Channel Decoder Using Extrinsic Information from Softbit-Source Decoding," in *IEEE ICASSP*, Salt Lake City, May 2001.

[2] N. Görtz, "On the Iterative Approximation of Optimal Joint Source-Channel Decoding," *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1662–1670, Sept. 2001.

[3] T. Fingscheidt and P. Vary, "Softbit Speech Decoding: A New Approach to Error Concealment," *IEEE Trans. Speech Audio Processing*, vol. 9, no. 3, pp. 240–251, Mar. 2001.

[4] P. Vary and R. Martin, *Digital Speech Transmission - Enhancement, Coding and Error Concealment*. John Wiley & Sons, Ltd., 2006.

[5] X. Liu and S. N. Koh, "Simplification of Soft-Bit Speech Decoding and Application to MELP Encoded Speech," *Electronics Letters*, vol. 39, no. 3, pp. 332–333, Feb. 2003.

[6] V. Franz and J. B. Anderson, "Concatenated Decoding with a Reduced-Search BCJR Algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 186–195, Feb. 1998.

[7] T. Clevorn, M. Adrat, and P. Vary, "Turbo DeCodulation using Highly Redundant Index Assignments and Multi-Dimensional Mappings," in *Intl. Symposium on Turbo Codes & Related Topics*, Munich, Apr. 2006.

[8] J. Kliewer, A. Huebner, and D. J. Costello, Jr., "On the Achievable Extrinsic Information of Inner Decoders in Serial Concatenation," in *IEEE ISIT*, Seattle, July 2006.

[9] T. Clevorn, J. Brauers, M. Adrat, and P. Vary, "Turbo DeCodulation: Iterative Combined Demodulation and Source-Channel Decoding," *IEEE Commun. Lett.*, vol. 9, no. 9, pp. 820–822, Sept. 2005.

[10] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[11] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symol Error Rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[12] M. Adrat and P. Vary, "Iterative Source-Channel Decoding: Improved System Design Using EXIT Charts," *EURASIP Jour. Appl. Sig. Proc.*, vol. 205, no. 6, pp. 928–941, May 2005.

[13] R. Martin, U. Heute, and C. Antweiler, Eds., *Advances in Digital Speech Transmission*. John Wiley & Sons, Ltd., Jan. 2008, ch. 13, pp. 365–398.

[14] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Audio*. Prentice-Hall, 1984.

[15] P. Robertson, E. Villebrun, and P. Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," in *IEEE ICC*, Seattle, 1995, pp. 1009–1013.

[16] T. Clevorn, P. Vary, and M. Adrat, "Iterative Source-Channel Decoding using Short Block Codes," in *IEEE ICASSP*, Toulouse, May 2006.

[17] S. ten Brink, "Convergence Behaviour of Iteratively Decoded Parallel Concatenated Codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[18] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic Information Transfer Functions: Model and Erasure Channel Properties," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[19] R. Thobaben, "A New Transmitter Concept for Iteratively-Decoded Source-Channel Coding Schemes," in *IEEE SPAWC*, Helsinki, June 2007.

[20] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Pearson Higher Education, 2003.

[21] L. Schmalen, P. Vary, T. Clevorn, and B. Schotsch, "Efficient Iterative Source-Channel Decoding Using Irregular Index Assignments," in *Int. ITG Conf. on Source and Channel Coding (SCC)*, Ulm, Jan. 2008.