A Generalized Framework for Iterative Source-Channel Decoding

Une Méthode Générale de Décodage Conjoint Itératif Source-Canal

Norbert Görtz

Institute for Communications Engineering

Technical University of Munich

80290 Munich / Germany

E-mail: norbert.goertz@ei.tum.de

Tel: +49 89 289 23494 Fax: +49 89 289 23490

Biography – *Norbert Görtz* received the Dr.-Ing. degree in electrical engineering from the University of Kiel, Germany, where he served as a member of the research staff until October 2000. In November 2000 he joined the Technical University of Munich as a lecturer and senior researcher. His interests include speech coding, joint source-channel (de)coding and the application of the turbo-principle in communications.

Abstract

Joint source-channel decoding is considered for a transmission system, in which the quantizer indices of several autocorrelated source signals are bit-interleaved, commonly channel encoded, and transmitted in parallel. Since the optimal decoding algorithm is not feasible in most practical situations, iterative source-channel decoding has been introduced. The latter is generalized in the present paper. Furthermore, it is shown in detail, that iterative source-channel decoding can be derived by insertion of appropriate approximations into the optimal joint decoding algorithm. The approximations allow the decomposition of the optimal decoder into two parts, which can be identified as the constituent decoders for the channel-code and the source-code redundancies. Similar as in other concatenated coding systems, the constituent decoders are applied in an iterative decoding scheme. Its performance is analyzed by simulation results.

Résumé

Dans cet article, nous considérons le décodage conjoint source-canal pour un système de transmission, dans lequel les indices des quantificateurs de plusieurs sources auto-corrélées sont entrelacés bit par bit, codés ensemble par un codage canal, et transmis en parallèle. Comme l'algorithme de décodage optimal est trop complexe, le décodage itératif source-canal est introduit et généralisé dans cet article. De plus, nous démontrons en détail que le décodage source-canal itératif peut se déduire après quelques approximations de l'algorithme de décodage conjoint optimal. Ces approximations permettent la décomposition du décodeur optimal en deux parties que l'on peut identifier comme les décodeurs constituants prenant en compte respectivement la redondance du codage canal et la redondance résiduelle du codage source. Comme dans d'autres systèmes concaténés de codage, les décodeurs constituants sont utilisés dans un processus de décodage itératif. Les performances sont analysées grâce aux résultats de simulation.

I. INTRODUCTION

In practical communication systems, especially in mobile telephony applications, the block-length for source and channel coding is limited due to complexity and delay constraints. Therefore, the data-bits that are issued by a source encoder usually contain residual redundancies, i.e., source coding can not be carried out perfectly in the sense of information theory. Furthermore, "perfect" channel codes are also only achievable with infinite block-length, i.e., the output bits of a practical channel decoder are not error-free, even if the code rate resides below the channel capacity. Thus, the application of the separation theorem of information theory is not really justified in practice.

In order to improve the performance of communication systems without sacrificing resources like bandwidth or transmission power, a lot of authors have considered joint source-channel coding and decoding.

One direction of joint source-channel coding is given by channel-optimized vector quantization [1], [2], and related approaches like optimization of the index-assignments, e.g., [3]. These techniques consider the bit-error probability that is to be expected in the data bits at the receiver side, possibly after a separate channel decoding step, in the design of the quantizer tables and the index-assignments for the reproduction levels, i.e., the transmitter is adapted to the channel.

However, much work in the area and also this paper concentrate on the receiver side, especially on joint source-channel *de*coding (JSCD). The key-idea of JSCD is to exploit the residual redundancies in the data bits to improve the overall quality of the transmission. In the past, two main approaches could be distinguished:

In the first type, the residual redundancies in the data bits are used as a-priori information in channel decoding in order to reduce the bit-error rate after decoding. One example is source-controlled channel decoding stated in [4] for the decoding of binary convolutional channel codes. The idea has been extended in [5] and [6] for the use of non-binary a-priori information. The problem of these approaches is, that the actual quality-criterion in the transmission of waveform signals (e.g., speech, audio) is not the bit-error probability, but the (possibly weighted) SNR between the input of the transmitter and the output of the receiver. However, powerful channel coding is required in such systems, especially for wireless transmission.

The algorithms of the second class process reliability informations of the received bits and statistical a-priori information on the quantizer indices to estimate the decoder output signal. Such algorithms have been stated, e.g., in [7], [8], [9], [10]. The techniques can be used after soft-output channel decoding [11], [12] or without application of any channel code. The a-posteriori probabilities (APPs) of the quantizer *indices* are computed for the estimation rather than the APPs of single index-*bits*, because the bit-APPs do not contain the correlation of the index-bits due to a non-uniform probability distribution of the indices, which usually results from imperfect source encoding. The drawback of this approach is, that a possibly required channel code is decoded independently of this estimation procedure.

Although both methods could be combined, e.g., by application of the APRI-SOVA algorithm [4] for channel decoding followed by the soft-bit source decoding [9], there is still a separation between source and channel decoding, since the channel decoder does not take advantage of the results of soft-output source-decoding.

In [13] an optimal algorithm for joint source-channel decoding called Channel-Coded Optimal Estimation (CCOE) was introduced, for a system with a single quantizer index that is channel-encoded by a systematic encoder. The algorithm was generalized in [14] for non-systematic channel encoders and the transmission of several quantizer indices in parallel. A generalization for the application of short non-linear block codes has been stated in [15].

Channel-Coded Optimal Estimation fully closes the gap between the two approaches mentioned above, since it exploits the reliabilities of the channel output-bits, the channel-code redundancies, and the residual source correlations in one step, and it allows to estimate the decoder-output signal directly. However, CCOE has the drawback of many optimal algorithms: It is not feasible for practical systems due to its complexity. Therefore, iterative source-channel decoding (ISCD) has been introduced in [16] and, more detailed, in [17], for systems with channel coding by a systematic encoder. For this special case, iterative source-channel decoding has been formulated as an approximation of the optimal algorithm in [18]. An iterative approach to source-channel decoding has also been stated in [19].

In this paper, iterative source-channel decoding (ISCD) is generalized for any binary channel code. It is practically feasible, if a channel code is selected, for which a symbol-by-symbol APP decoder can be implemented with realistic complexity. The ISCD-algorithm is no longer restricted to the use of systematic channel encoders.

This paper is organized as follows: First, a model of a transmission system is given, for which the optimal joint source-channel decoding algorithm is shortly reviewed. Then, iterative source-channel decoding is derived from the optimal algorithm by insertion of appropriate approximations. The latter allows the decomposition of the optimal decoder into two components, which can be identified as the constituent decoders for the channel-code and the source-code redundancies. Thus, new insight is gained into the relations of the optimal and the iterative decoding principles. Finally, after some remarks concerning the implementation, the performance of ISCD is analyzed by simulation results.

II. SYSTEM MODEL

Figure 1 shows the model of a transmission system that is used throughout the paper. The input vectors $X_k^1, ..., X_k^M$ have to be transmitted at each time k. The vectors can be thought of as M independent source-signals. In a practical case they might also be the parameters[1] of a speech, audio, or image codec.

The input vectors X_k^j, $j = 1, ..., M$, are quantized ("source-encoded") by the indices I_k^j by N^j bits, $j=1, ..., M$. Although for each input vector X_k^j, $j = 1, ..., M$, a different number N^j of bits can be used for the quantization, the number of output bits is fixed for each quantizer[2]. Additionally, it is assumed that it is *not*

[1]Usually, the codec-parameters are derived from a *block* of source-signal samples. For instance, in speech coding the duration of a signal block is typically around 20ms.

[2]The applications which are intended for the algorithms stated in this paper lie in the area of mobile communications, e.g., speech transmission. The possibility of variable-length encoding of the quantizer indices is not considered because of the well-known error-propagation and synchronization problems due to bit-errors. For instance, this is the reason why practical speech encoders for mobile radio systems usually do not use variable-length encoding of their quantizer indices. The combination of the algorithms described in this paper with more recent approaches of robust decoding of entropy-coded sources is an interesting topic for future work.

possible to jointly encode the input vectors of several time-steps (which might be desirable in order to achieve more efficient source-encoding) since this would cause additional delay at the receiver. This prerequisite is realistic in common two-way communication scenarios, e.g., in telephony applications.

After source encoding the index-bits are interleaved[3], commonly channel-encoded, and the codewords $V_k = \{v_{l,k} \in \{0,1\},\ l=1,...,N_V\}$, $N_V > N \doteq \sum_{j=1}^{M} N^j$, are transmitted over an AWGN-channel. Coherently detected binary phase-shift

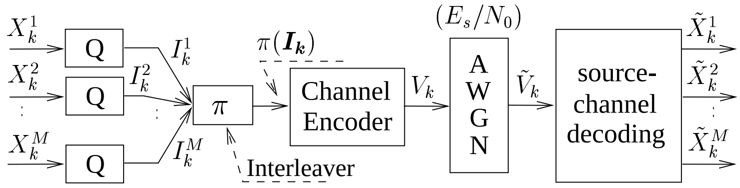

Fig. 1. Model of the transmission system — Modèle du système de transmission

keying is assumed for the modulation. Therefore, the conditional p.d.f. (**p**robability **d**ensity **f**unction) of the received value $\tilde{v}_{l,k} \in I\!R$ at the channel output, given that the code bit $v_{l,k} \in \{0,1\}$ has been transmitted, is given by

$$p_c\big(\tilde{v}_{l,k}|v_{l,k}\big) = \frac{e^{-\frac{1}{2\sigma_n^2}(\tilde{v}_{l,k}-v'_{l,k})^2}}{\sqrt{2\pi}\sigma_n}\Bigg|_{v'_{l,k}=1-2\cdot v_{l,k}}, \qquad (1)$$

with the variance $\sigma_n^2 = \frac{N_0}{2E_s}$. E_s is the energy that is used to transmit each channel-code bit and N_0 is the one-sided power spectral density of the channel noise. The joint conditional p.d.f. $p_c(\tilde{V}_k|V_k)$ that the vector $\tilde{V}_k \in I\!R^{N_V}$ has been received, given that the codeword $V_k \in \{0,1\}^{N_V}$ was transmitted, equals the product of (1) over all

[3]The interleaver, which is denoted by $\pi(.)$, is treated as a part of the channel encoder. Consequently, it is "hidden" in the algorithms for channel decoding in the sequel.

N_V code-bits, since the channel noise is "white" and normally distributed, i.e.,

$$p_c\big(\tilde{V}_k|V_k\big) = \prod_{l=1}^{N_V} p_c\big(\tilde{v}_{l,k} \mid v_{l,k}\big) \ . \qquad (2)$$

If the signals X_k^j are autocorrelated, adjacent indices I_{k-1}^j, I_k^j show dependencies. They are modeled by first-order stationary Markov-processes, which are described by the index-transition probabilities $P(I_k^j = \lambda^j | I_{k-1}^j = \mu^j)$, $\lambda^j, \mu^j = 0, ..., 2^{N^j} - 1$ and $j = 1, ..., M$. It is assumed that these probabilities and the probability-distributions of the indices are known. The indices are assumed to be *mutually* independent. This simplifies the analysis and the realization of source-channel decoding, and it is at least approximately fulfilled in many practically relevant situations. For instance, it was shown in [20] that the quantizer indices of CELP speech codecs [21] show relatively small mutual correlations[4] but some of them (e.g., the quantizer indices of the block-energy and the spectral shape) are strongly autocorrelated over a few adjacent time-steps (typically, a time-step corresponds to speech frame with a duration of 20ms). An appropriate model for this type of correlation is given by first-order Markov-processes.

In the terminology of turbo-codes the coding scheme is serially concatenated since the redundancies of the indices (i.e., the time-based dependencies and the non-uniform probability distributions) form the "outer" codes which are encoded once more by the "inner" channel code. The *bit*-interleaver was included, because it is a well-known fact that it improves the error-correction properties of concate-

[4]The cross-correlations of the *parameters* of a source codec can be exploited by vector quantization to reduce the bit rate without the error-propagation problems which occur in schemes that also exploit the autocorrelation of the parameters, e.g., by differential encoding.

nated channel codes since the *individual bits* of the indices, which are coupled by the residual redundancies (the "outer" codes) are spread by the bit-interleaver over the whole range of input bits of the "inner" channel code. Additionally, the interleaver makes adjacent bits in a block independent, which is an assumption of the frequently used APP decoding-algorithm in [11] (BCJR-algorithm).

On principle, it would be possible to use an interleaver-length that is larger than the number of bits that have to be transmitted at each time step. However, this case is not considered in this paper since such an interleaver would cause additional delay.

III. OPTIMAL JOINT SOURCE-CHANNEL DECODING

For the system model stated above, the goal is to minimize the distortion of the decoder output signals $\tilde{X}_k^j$ due to the channel noise, i.e., we would like to perform optimal joint source-channel *de*coding (JSCD) for a *fixed* transmitter. This allows to improve existing systems by the algorithms stated below with decoder-only modifications. The mathematical formulation of the optimization criterion is given by the conditional expectation of the mean-squared error (accumulated over all signal vectors):

$$E\Big\{ \sum_{j=1}^{M} \big\| \tilde{X}_k^j - X_q^j(I_k^j) \big\|_2^2 \,\Big|\, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}} \Big\} \underset{\tilde{X}_k^j}{\longrightarrow} \min \,. \tag{3}$$

This distortion measure has to be minimized over the output vectors $\tilde{X}_k^j$ of the joint source-channel decoder. In (3), $X_q^j(I_k^j)$ is the entry with the index I_k^j of the quantization table, which is used by the source encoder to quantize the vector X_k^j, and $\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}} \doteq \{\tilde{V}_k, \tilde{V}_{k-1}, ..., \tilde{V}_0\}$ is the set of channel words which were received up to the current time k.

Let the realizations $\boldsymbol{\mu} = \{\mu^1, ..., \mu^M\}$ of the set of indices

$$\boldsymbol{I_k} \doteq \{I_k^1, ..., I_k^M\}, \quad I_k^j = \{i_{l,k}^j \in \{0,1\},\ l = 1, ..., N^j\} \tag{4}$$

take on "values" from the set $\boldsymbol{S}$, i.e., $\boldsymbol{\mu} \in \boldsymbol{S}$, with

$$\boldsymbol{S} \doteq S^1 \times .... \times S^M\ , \quad S^j \doteq \{0, ..., 2^{N^j}-1\}\ . \tag{5}$$

Then, the minimization (3) results in mean-square estimators for the signals $\tilde{X}_k^j$, even if the indices are not mutually independent, i.e.,

$$\tilde{X}_k^j = \sum_{\forall \lambda^j \in S^j} X_q^j(\lambda^j) \cdot P(I_k^j = \lambda^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}})\ , \ \ j = 1, 2, ..., M\ , \tag{6}$$

with the *index a-posteriori probabilities* (APPs)

$$P(I_k^j = \lambda^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}})\ . \tag{7}$$

Hence, the actual problem of joint source-channel decoding is how to compute (7).

Using the Bayes-rule, the *index* APPs (7) can be expressed as

$$P\big(I_k^j = \lambda^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big) = \frac{p\big(I_k^j = \lambda^j, \tilde{V}_k, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)}{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)} = \frac{p\big(\tilde{V}_k \mid I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) \cdot p\big(I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)}{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)}. \tag{8}$$

This results in

$$P\big(I_k^j = \lambda^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big) = \frac{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)}{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)} \cdot p\big(\tilde{V}_k \mid I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) \cdot P\big(I_k^j = \lambda^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big), \tag{9}$$

which can be written more compact according to

$$P(I_k^j = \lambda^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}) = B_k \ \cdot \ p\big(\tilde{V}_k \mid I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) \ \cdot \ P_a\big(I_k^j = \lambda^j\big)\ . \tag{10}$$

The term

$$P_a\big(I_k^j = \lambda^j\big) \doteq P\big(I_k^j = \lambda^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) \tag{11}$$

can be interpreted as *index* a-priori probability, since it is computed only from channel words that have been received in the past, i.e., the information carried by the currently received channel-word $\tilde{V}_k$ is not used. The factor

$$B_k \doteq \frac{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)}{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)} \tag{12}$$

does not depend on the random variable I_k^j, i.e., it is a normalizing constant, that makes the left-hand side of (10) a true probability that sums up to one over all possible values of I_k^j. Thus, the p.d.f.s $p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)$ and $p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)$ do not have to be explicitly computed, since the normalization of (10) can be easily achieved by computing B_k according to

$$B_k = 1 \Big/ \sum_{\forall \lambda^j \in S^j} p\big(\tilde{V}_k \mid I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) \cdot P_a\big(I_k^j = \lambda^j\big) \ . \tag{13}$$

Considering the mutual independence of the indices and the Markov-models for their time-based dependencies, it can be shown by using the results[5] in [14], that the two terms on the right-hand side of (10) are given by

$$p\big(\tilde{V}_k \mid I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) \doteq \sum_{\forall \boldsymbol{\mu} \in \boldsymbol{S}:\, \mu^j = \lambda^j} \!\!\cdots \sum p_c\big(\tilde{V}_k \mid V_k^{(\boldsymbol{\mu})}\big) \ \cdot \prod_{\substack{m=1,\\ m \neq j}}^{M} P_a\big(I_k^m = \mu^m\big) \ , \tag{14}$$

for $\lambda^j \in S^j$, $j = 1, 2, ..., M$, and the *index a-priori probabilities*

$$P_a\big(I_k^j = \lambda^j\big) \doteq \sum_{\forall \mu^j \in S^j} \underbrace{P(I_k^j = \lambda^j | I_{k-1}^j = \mu^j)}_{\text{Markov-model}} \cdot P(I_{k-1}^j = \mu^j | \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}). \tag{15}$$

[5] The optimal algorithm for joint source-channel decoding has been stated in [14] and called Channel-Coded Optimal Estimation. In this paper, the general formula from [14] is used in a specialized form for mutually independent quantizer indices.

The notation $V_k^{(\boldsymbol{\mu})}$ corresponds to the channel codeword that results from some input index-combination $\boldsymbol{\mu} \in \boldsymbol{S}$. No restrictions apply for the channel code and the channel encoder, i.e., the result is quite general. Additionally, it should be noticed that the *index* a-priori probabilities (15) are computed from the "old" *index* APPs (10) at time $k-1$, i.e., the equations (10), (14), and (15) form a recursion that allows the computation of the *index* APPs using only the known quantities, which were defined in section II. At $k=0$, the "old" APPs are initialized by the "unconditioned" probability distributions of the indices in order to compute (15) for the first time.

IV. ITERATIVE SOURCE-CHANNEL DECODING

The optimal algorithm for JSCD stated in section III is practically not feasible, if realistic numbers of index-bits are first spread by an interleaver and then are commonly channel encoded. This is due to the tremendous complexity of the summation in (14), which is taken over all possible combinations of the indices, excluding only the index I_k^j under consideration.

On principle, the a-posteriori probabilities (10) can be computed by the symbol-by-symbol APP-algorithm in [11] (BCJR-algorithm), if a non-binary channel code is used, with code-symbols that are matched to the numbers of bits in the quantizer indices. In this approach, the bits cannot be interleaved prior to channel encoding, because the goal is to exploit the a-priori probabilities of the indices in one step of the trellis of the non-binary code. Thus, the overall concatenated code has worse distance properties than a scheme with an interleaver. Additionally, the implementation of an APP-decoder for a non-binary channel code is practically feasible only for a small number of bits (e.g., two or three) in the code-symbols. Nonetheless, practical source encoders often issue quantizer indices with more than five bits.

An efficient implementation of the APP-algorithm is possible, if a *binary* channel code with a regular trellis with a small number of states is used[6]. This will be assumed in the sequel. Still, the quantizer-indices are non-binary: In general they consist of arbitrary numbers of bits. In order to be able to efficiently compute a-posteriori probabilities for the output data bits of the channel code, but to exploit the correlations of the bits of the *non-binary* indices, the *iterative source-channel decoding* (ISCD) can be used. It has been introduced in [16], [17] for channel codes with systematic encoders. However, this restriction is not necessary as we will see in the more general framework in the sequel.

The basic idea of iterative source-channel decoding is to use a binary channel code and to simplify the computation of the index APPs (10) in such a way, that the APP-algorithm [11] (BCJR-algorithm) can be efficiently applied to exploit the channel-code redundancies, while the bit-correlations, which are contained in the index a-priori probabilities (15), are still utilized for the computation of the index APPs (7).

As a consequence of this idea, the term $p\big(\tilde{V}_k \,|\, I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)$ in (10), that contains the channel-code redundancies and which is given by the highly complex formula (14), must be approximated by the *bit* APPs $P^{(C)}\big(i_{l.k}^j = \lambda_l^j | \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)$ (or some other

[6] This is the case for the frequently used binary convolutional codes with short constraint-length.

quantity, that can be efficiently computed by the APP-algorithm). The superscript "(C)" is introduced, to identify quantities that belong to the APP-algorithm for channel decoding. In the next section, this algorithm is analyzed in some detail in order to break up the recursion (10), (14), and (15) into two constituent algorithms that operate iteratively with drastically lower complexity than the optimal scheme.

A. General Description of the Symbol-by-Symbol APP-algorithm

The symbol-by-symbol APP-algorithm for decoding of a binary channel code (i.e., the code symbols are bits) computes the *bit* APPs

$$P^{(C)}(i_{l,k}^{j}=\lambda_l^j \mid \tilde{V}_k)\ ,\ j=1,2,...,M,\ l=1,2,...,N^j\ . \qquad (16)$$

The probability of each data bit $i_{l,k}^j$ is "conditioned" on the whole received channel-word[7] $\tilde{V}_k$ (and on the underlying code constraints). In (16) the same (a little complex) bit-notation as above was used, to make the results of this section directly applicable in the rest of the paper.

The *bit* APPs (16) can be rewritten as the marginal probability-distribution

$$P^{(C)}(i_{l,k}^{j}=\lambda_l^j|\tilde{V}_k) = \sum_{\forall \boldsymbol{\mu}\in \boldsymbol{S}:\ \mu_l^j=\lambda_l^j} \cdots \sum P(V_k^{(\boldsymbol{\mu})}|\tilde{V}_k) = \sum_{\forall \boldsymbol{\mu}\in \boldsymbol{S}:\ \mu_l^j=\lambda_l^j} \cdots \sum \frac{P(V_k^{(\boldsymbol{\mu})})}{p(\tilde{V}_k)}\ p(\tilde{V}_k|V_k^{(\boldsymbol{\mu})})\ . \qquad (17)$$

The a-priori probability $P(V_k^{(\boldsymbol{\mu})})$ of the code word can be replaced by the product of the a-priori probabilities $P_a^{(C)}(i_{l,k}^m=\mu_l^m)$ of the data-bits, if the latter are assumed

[7]From the channel-coding point of view, there is no need for the time index k that identifies adjacent codewords, because the channel decoder only exploits the redundancies within one code word to compute the output *bit*-APPs (16). The latter only have a time index, because the a-priori information, that is passed to the channel decoder, depends on the previously transmitted codewords due to the correlation of the source-encoder indices.

to be mutually independent (which is not true in our setup). Therefore, we obtain:

$$P^{(C)}(i^j_{l,k}=\lambda^j_l \mid \tilde{V}_k) = \frac{1}{p(\tilde{V}_k)} \sum \cdots \sum_{\forall \boldsymbol{\mu}\in \boldsymbol{S}:\ \mu^j_l=\lambda^j_l} p(\tilde{V}_k \mid V^{(\boldsymbol{\mu})}_k) \prod_{m=1}^{M} \prod_{\nu=1}^{N^m} P^{(C)}_a(i^m_{\nu,k}=\mu^m_\nu)\ . \quad (18)$$

Finally, (18) can be rewritten as

$$P^{(C)}(i^j_{l,k}=\lambda^j_l \mid \tilde{V}_k) = P^{(C)}_a(i^j_{l,k}=\lambda^j_l) \cdot \frac{1}{p(\tilde{V}_k)} \sum \cdots \sum_{\forall \boldsymbol{\mu}\in \boldsymbol{S}:\ \mu^j_l=\lambda^j_l} p(\tilde{V}_k \mid V^{(\boldsymbol{\mu})}_k)$$
$$\prod_{\substack{m=1,\\ m\neq j}}^{M} \prod_{\nu=1}^{N^m} P^{(C)}_a(i^m_{l,k}=\mu^m_l) \cdot \prod_{\substack{\eta=1,\\ \eta\neq l}}^{N^j} P^{(C)}_a(i^j_{\eta,k}=\mu^j_\eta)\ . \quad (19)$$

The *bit* APPs given by this formula can be efficiently computed by the APP-algorithm in [11], if the channel code is appropriately structured.

For the application in an iterative decoding scheme it is convenient to define

$$P^{(C)}_{er}(i^j_{l,k}=\lambda^j_l) \doteq \frac{P^{(C)}(i^j_{l,k}=\lambda^j_l \mid \tilde{V}_k)}{P^{(C)}_a(i^j_{l,k}=\lambda^j_l)} = \frac{1}{p(\tilde{V}_k)} \sum \cdots \sum_{\forall \boldsymbol{\mu}\in \boldsymbol{S}:\ \mu^j_l=\lambda^j_l} p(\tilde{V}_k \mid V^{(\boldsymbol{\mu})}_k)$$
$$\prod_{\substack{m=1,\\ m\neq j}}^{M} \prod_{\nu=1}^{N^m} P^{(C)}_a(i^m_{l,k}=\mu^m_l) \cdot \prod_{\substack{\eta=1,\\ \eta\neq l}}^{N^j} P^{(C)}_a(i^j_{\eta,k}=\mu^j_\eta)\ , \quad (20)$$

using (19) for the second equality. This quantity contains the information on the data bits, that has been derived by the APP-decoder by exploiting the received channel word $\tilde{V}_k$ and the channel-code redundancies, i.e., it contains the extrinsic [22] and the received channel information. The a-priori information $P^{(C)}_a(i^j_{l,k}=\lambda^j_l)$ is removed (by the division in (20)) in order to extract the new information that can be used by further constituent decoders without feeding back information to them that they produced themselves or that they have already used.

Since, due to the correlations of adjacent indices in the system model, the *index* a-priori probabilities depend on the previously received channel words, the notation is formally adapted to that by setting

$$P_a^{(C)}(i_{l,k}^j = \lambda_l^j) \doteq P\big(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) \tag{21}$$

and

$$P_{er}^{(C)}(i_{l,k}^j = \lambda_l^j) \doteq \frac{P^{(C)}(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}})}{P_a^{(C)}\big(i_{l,k}^j = \lambda_l^j\big)} \; . \tag{22}$$

B. Derivation of Iterative Source-Channel Decoding (ISCD)

Since we would like to make the APP-algorithm for decoding of a binary channel codes, as summarized in section IV-A, applicable in a source-channel decoding scheme that involves non-binary quantizer indices, we have to

- find appropriate *bit* a-priori information for the APP-algorithm.
- find a way to use the *bit*-based output-information of the APP-decoder in source-channel decoding.

As a first step towards the solution of these two problems, the left-hand side of (14) is reformulated:

$$p\big(\tilde{V}_k \mid I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) = \frac{p\big(\tilde{V}_k, I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)}{p\big(I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)} \; . \tag{23}$$

Now, in the numerator and in the denominator, the *index* probabilities are approximated by the product over the corresponding *bit* probabilities:

$$p\big(\tilde{V}_k \mid I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) \approx p_\xi^{(C)}\big(\tilde{V}_k \mid I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) \doteq \frac{\prod_{l=1}^{N^j} p_\xi^{(C)}\big(\tilde{V}_k, i_{l,k}^j = \lambda_l^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)}{\prod_{l=1}^{N^j} p_\xi^{(C)}\big(i_{l,k}^j = \lambda_l^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)} \; . \tag{24}$$

Thus, the mutual dependencies of the bits, which result from the residual redundancies in the quantizer indices, are neglected. Since, later on, the approximately computed quantities will vary within iterations, the iteration-counter $\xi = 1, 2, ...$ has been introduced in (24). The latter equation can be rewritten as

$$p_\xi^{(C)}\big(\tilde{V}_k \mid I_k^j = \lambda^j, \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) = \frac{\prod_{l=1}^{N^j} P_\xi^{(C)}\big(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big) \cdot p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)}{\prod_{l=1}^{N^j} P_\xi^{(C)}\big(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big) \cdot p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)} \tag{25}$$

$$= \frac{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)}{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)} \cdot \prod_{l=1}^{N^j} \frac{P_\xi^{(C)}\big(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)}{P_\xi^{(C)}\big(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)} \,. \tag{26}$$

If we insert (26) into (10) we obtain

$$P_\xi(I_k^j = \lambda^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}) = \frac{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)}{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)} \cdot \frac{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)}{p\big(\tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)} \cdot \prod_{l=1}^{N^j} \frac{P_\xi^{(C)}\big(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}\big)}{P_\xi^{(C)}\big(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k-1}}\big)} \cdot P_a\big(I_k^j = \lambda^j\big) \tag{27}$$

as the approximation of $P(I_k^j = \lambda^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}})$. The use of the *bit* a-priori probabilities (21) and the definition (22) leads to

$$P_\xi(I_k^j = \lambda^j \mid \tilde{\boldsymbol{V}}_{\boldsymbol{0}}^{\boldsymbol{k}}) = \left[\prod_{l=1}^{N^j} P_{er,\xi}^{(C)}(i_{l,k}^j = \lambda_l^j)\right] \cdot P_a\big(I_k^j = \lambda^j\big) \,. \tag{28}$$

It should be noticed that the probabilities $P_{er,\xi}^{(C)}(i_{l,k}^j = \lambda_l^j)$ are given by (20), which is computed by the APP-algorithm for channel decoding. The formula (28) is strongly similar to the one that is used in the Optimal-Estimation (OE) algorithm [14], that has been introduced in [9] for soft-bit source decoding: Instead of a pure channel-term that inserts the reliabilities of the received index-bits into the estimation, we have a modified channel-term in (28) (between the brackets) that includes the reliabilities of the received bits *and* the additional information derived by the APP-algorithm from the channel-code redundancies. Thus, APP channel decoding produces a more reliable "virtual" channel for the OE-algorithm. For the concatenated code, that consists of the explicit channel code and the implicit index-dependencies, the APP-algorithm and the OE-algorithm are the constituent decoders in the iterative decoding scheme described in the sequel.

In the first "iteration", i.e., $\xi = 1$, the *bit* a-priori probabilities for the APP-algorithm can be computed from the *index* a-priori probabilities according to

$$P_{a,1}^{(C)}\left(i_{l,k}^{j}=\lambda_l^j\right) = \sum_{\forall \mu^j \in S^j : \mu_l^j = \lambda_l^j} P_a\left(I_k^j = \mu^j\right) . \tag{29}$$

This is all available a-priori information, before the first run of the APP-algorithm for channel decoding. Similar as in (15), the *index* a-priori probabilities are given by

$$P_a\left(I_k^j = \lambda^j\right) \doteq \sum_{\forall \mu^j \in S^j} \underbrace{P(I_k^j = \lambda^j | I_{k-1}^j = \mu^j)}_{\text{Markov-model}} \cdot P'(I_{k-1}^j = \mu^j | \tilde{\mathbf{V}}_{\mathbf{0}}^{\boldsymbol{k-1}}) , \tag{30}$$

with the difference, that the term $P'(I_{k-1}^j = \mu^j | \tilde{\mathbf{V}}_{\mathbf{0}}^{\boldsymbol{k-1}})$ denotes the *index* APPs from the previous time $k-1$ that resulted from (28) after the last iteration. It should be noticed that the *index* a-priori probabilities (30) are fixed quantities within the iterations, i.e., they have to be computed only once at each time k.

Using the *index* APPs that result from (28), the mean square estimations $\tilde{X}_k^j$ for transmitted signals could now be determined by (6). However, the *index* APPs are only approximations of the optimal values, since the *bit* a-priori informations, that were used for channel decoding, did not contain the mutual dependencies of the bits: They were "summed away" in (29).

The idea how to improve the accuracy of the *index* APPs is taken over from iterative decoding of turbo codes [23]: From the intermediate results for the *index*

APPs derived from (28), new *bit* APPs are computed by

$$P_\xi^{(S)}(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\mathbf{0}}^{\boldsymbol{k}}) = \sum_{\forall \mu^j \in S^j : \mu_l^j = \lambda_l^j} P_\xi(I_k^j = \mu^j \mid \tilde{\boldsymbol{V}}_{\mathbf{0}}^{\boldsymbol{k}}) \; . \tag{31}$$

The superscript "(S)" was introduced, since (31) is computed after source decoding (which included the results of the previous channel-decoding step). Inserting (28) into (31) leads to

$$P_\xi^{(S)}(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\mathbf{0}}^{\boldsymbol{k}}) = \sum_{\forall \mu^j \in S^j : \mu_l^j = \lambda_l^j} P_a\big(I_k^j = \mu^j\big) \cdot \prod_{\nu=1}^{N^j} P_{er,\xi}^{(C)}\big(i_{\nu,k}^j = \mu_\nu^j\big) \; , \tag{32}$$

which equals

$$P_\xi^{(S)}(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\mathbf{0}}^{\boldsymbol{k}}) = P_{er,\xi}^{(C)}\big(i_{l,k}^j = \lambda_l^j\big) \cdot \sum_{\forall \mu^j \in S^j : \mu_l^j = \lambda_l^j} P_a\big(I_k^j = \mu^j\big) \cdot \prod_{\substack{\nu=1, \\ \nu \neq l}}^{N^j} P_{er,\xi}^{(C)}\big(i_{\nu,k}^j = \mu_\nu^j\big) \; . \tag{33}$$

By definition of the extrinsic *bit* probabilities

$$P_{e,\xi}^{(S)}(i_{l,k}^j = \lambda_l^j) \doteq \frac{P_\xi^{(S)}(i_{l,k}^j = \lambda_l^j \mid \tilde{\boldsymbol{V}}_{\mathbf{0}}^{\boldsymbol{k}})}{P_{er,\xi}^{(C)}\big(i_{l,k}^j = \lambda_l^j\big)} \; , \tag{34}$$

(33) can be written as

$$P_{e,\xi}^{(S)}(i_{l,k}^j = \lambda_l^j) = \sum_{\forall \mu^j \in S^j : \mu_l^j = \lambda_l^j} P_a\big(I_k^j = \mu^j\big) \cdot \prod_{\substack{\nu=1, \\ \nu \neq l}}^{N^j} P_{er,\xi}^{(C)}\big(i_{\nu,k}^j = \mu_\nu^j\big) \; . \tag{35}$$

A comparison of (35) and (29) reveals, that both equation are structurally similar. In both equations a summation is carried out over the a-priori probabilities of the index with the fixed realization $\lambda_l^j \in \{0, 1\}$ for the index-bit $i_{l,k}^j$. While in (29) no further information on the bit $i_{l,k}^j$ was available, now, after the run of the APP channel decoder, additional information $P_{er,\xi}^{(C)}(i_{l,k}^j = \lambda_l^j)$ on the bits has been computed. Our intention is to provide the APP channel decoder with improved a-priori information for the next iteration. Therefore, the information, which is given by $P^{(C)}_{er,\xi}(i^j_{l,k})$, should not be included in the new a-priori information for this bit, since it has been generated by the APP channel decoder itself. This is the reason, why it has been excluded in the definition (35) of the extrinsic information. On the other hand, the channel-decoder output-information $P^{(C)}_{er,\xi}(i^j_{\nu,k})$, $\nu \neq l$, on the *other* bits *is* used in (35) to weight the *index* a-priori probabilities before summing them up. This way, new (extrinsic) information $P^{(S)}_{e,\xi}(i^j_{l,k})$ is computed compared to (29), because additional knowledge on the data bit $i^j_{l,k}$ can be derived from the probabilities $P^{(C)}_{er,\xi}(i^j_{\nu,k})$, $\nu \neq l$, and the coupling of the index-bits by the *index* a-priori probabilities. For the next iteration, which starts with a new run of the APP-algorithm, the required new a-priori information is therefore given by

$$P^{(C)}_{a,\xi+1}\big(i^j_{l,k}=\lambda^j_l\big) \doteq P^{(S)}_{e,\xi}\big(i^j_{l,k}=\lambda^j_l\big)\ . \tag{36}$$

Finally, the iteration counter is incremented, i.e., $\xi \rightarrow \xi + 1$, and the next iteration is started.

<u>Iterative source-channel decoding (ISCD) can be summarized as follows:</u>

1. At each time k, compute the initial *index* a-priori probabilities by (30).
2. Insert the results from step 1 in (29) to determine the *bit* a-priori information for the APP channel decoder for the first iteration.
3. Set the iteration counter to one, i.e., $\xi = 1$.
4. Perform APP channel decoding by an efficient realization of (20).
5. Source decoding through Optimal Estimation by inserting the results of APP channel decoding into (28) to compute new (temporary) *index* APPs.

6. If this was the last iteration, proceed with step 10, otherwise continue with 7
7. Use (31) and (34) to compute extrinsic *bit* information from the source-code redundancies.
8. Set the extrinsic information from step 7 equal to the new a-priori information for the APP channel decoder in the next iteration, i.e., perform (36).
9. Increment the iteration counter, i.e., $\xi \to \xi + 1$, and proceed with step 4
10. Estimate the receiver output signals by (6) using the *index* APPs from step 5

C. Implementation of Iterative Source-Channel Decoding

An iterative source-channel decoder that works as summarized at the end of the previous section is depicted in Fig. 2. It directly fits into the transmission system in Fig. 1.

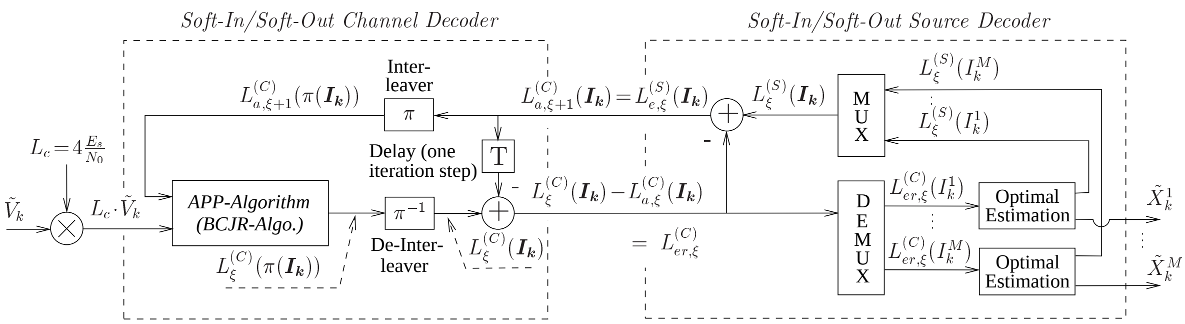

Fig. 2. Iterative source-channel decoding — Décodage itératif source-canal

The iterative source-channel decoder consists of two constituent decoders: The APP-algorithm for channel decoding and the Optimal-Estimation algorithm for source-decoding. Both were described in the previous section. As usual in iterative decoding schemes, they are denoted as Soft-In/Soft-Out decoders (SISO decoders), since both process and issue bit-reliability informations ("soft-values").

The soft-values that are passed between the SISO decoders in Fig. 2 are not the probabilities from the previous section, but log-likelihood-ratios (L-values, [22]), which are directly related to them. For instance, the extrinsic L-value from the SISO source decoder for the bit $i^j_{l,k}$ in the iteration with the number ξ is defined by

$$L^{(S)}_{e,\xi}(i^j_{l,k}) \doteq \log \frac{P^{(S)}_{e,\xi}(i^j_{l,k}=0)}{P^{(S)}_{e,\xi}(i^j_{l,k}=1)} \; , \tag{37}$$

using the natural logarithm. The other L-values are related to the corresponding probabilities in a similar fashion. In order to simplify the drawing, *vectors* $L(\boldsymbol{I_k})$ of L-values that correspond to the index set defined in (4) are noted in Fig. 2, i.e.:

$$L(\boldsymbol{I_k}) \doteq \left\{ L(i^j_{l,k}),\; j=1,...,M,\; l=1,...,N^j \right\} \, . \tag{38}$$

The advantage of L-values is, that they cause less numerical problems than probability-values in the implementation of algorithms with finite word-length on a digital computer. The BCJR-algorithm, an efficient implementation of the APP-algorithm, e.g., for decoding of binary convolutional channel codes, can be completely carried out in the L-value domain (Log-MAP-algorithm, [24]). The use of such an algorithm is assumed in Fig. 2. The received channel values $\tilde{v}_{l,k}$ are converted to L-values at the input of the SISO channel decoder by multiplication with the factor $L_c = 4\frac{E_s}{N_0}$. This follows from the definition of the L-values, the p.d.f. (1), and the usual assumption that the code-bits are equally likely:

$$L(\tilde{v}_{l,k}) = \log \frac{P(v_{l,k}=0|\tilde{v}_{l,k})}{P(v_{l,k}=1|\tilde{v}_{l,k})} = \log \frac{p_c(\tilde{v}_{l,k}|v_{l,k}=0)}{p_c(\tilde{v}_{l,k}|v_{l,k}=1)} = 4\frac{E_s}{N_0}\tilde{v}_{l,k} \; . \tag{39}$$

Since the SISO source decoder, as it has been stated above, processes probabilities, the interface to the APP-algorithm operating in the L-value domain is shortly discussed in the sequel:

The computation of the index APPs by the Optimal-Estimation algorithm (28) requires the bit-probabilities $P^{(C)}_{er,\xi}(i^j_{l,k} = \lambda^j_l)$, $l = 1, ..., N^j$. The latter can be computed from the output L-values $L^{(C)}_{er,\xi}(i^j_{l,k})$ of the SISO channel decoder by

$$P^{(C)}_{er,\xi}(i^j_{l,k} = \lambda^j_l) = \frac{\exp\big(L^{(C)}_{er,\xi}(i^j_{l,k})\big)}{1 + \exp\big(L^{(C)}_{er,\xi}(i^j_{l,k})\big)} \cdot \exp\big(-L^{(C)}_{er,\xi}(i^j_{l,k}) \cdot \lambda^j_l\big) \ , \quad \lambda^j_l \in \{0,1\} \tag{40}$$

by inversion of the L-value definition (37). Since the Optimal-Estimation algorithm computes the product over all these probabilities for one index, this operation can be simplified by inserting (40) into (28), i.e.,

$$P_\xi(I^j_k = \lambda^j \mid \tilde{\boldsymbol{V}}^{\boldsymbol{k}}_{\boldsymbol{0}}) = D^j_k \cdot \left[\prod_{l=1}^{N^j} \exp\Big(-L^{(C)}_{er,\xi}(i^j_{l,k}) \cdot \lambda^j_l\Big)\right] \cdot P_a\big(I^j_k = \lambda^j\big) \ , \tag{41}$$

with the normalizing constant

$$D^j_k = \prod_{l=1}^{N^j} \frac{\exp\big(L^{(C)}_{er,\xi}(i^j_{l,k})\big)}{1 + \exp\big(L^{(C)}_{er,\xi}(i^j_{l,k})\big)} \ , \tag{42}$$

that does not depend on the variable λ^j. Now, the product in (41) can be turned into a summation in the L-value domain, i.e.:

$$P_\xi(I^j_k = \lambda^j \mid \tilde{\boldsymbol{V}}^{\boldsymbol{k}}_{\boldsymbol{0}}) = D^j_k \cdot \exp\left(-\sum_{l=1}^{N^j} L^{(C)}_{er,\xi}(i^j_{l,k}) \cdot \lambda^j_l\right) \cdot P_a\big(I^j_k = \lambda^j\big) \ . \tag{43}$$

Thus, the L-values from the SISO channel decoder can be integrated into the Optimal-Estimation algorithm for SISO source decoding without converting the single L-values back to probabilities if (43) is used instead of (28). This is a simplification that also has numerical advantages. Additionally, the left-hand side of (43) is a probability that must sum up to one over all index-realizations $\lambda^j \in S^j$. Hence, the constant D^j_k can be computed from this condition instead of using (42).

The computation of new bit APPs within the iterations is still carried out by (31), but the derivation of the extrinsic L-values $L^{(S)}_{e,\xi}(i^j_{l,k})$, that are issued by SISO source decoder, can be simplified, since (34) requires a division which is turned into a simple subtraction in the L-value-domain:

$$L^{(S)}_{e,\xi}(i^j_{l,k}) \doteq \log \frac{P^{(S)}_{\xi}(i^j_{l,k}=0 \mid \tilde{\boldsymbol{V}}^{\boldsymbol{k}}_{\boldsymbol{0}})/P^{(C)}_{er,\xi}(i^j_{l,k}=0)}{P^{(S)}_{\xi}(i^j_{l,k}=1 \mid \tilde{\boldsymbol{V}}^{\boldsymbol{k}}_{\boldsymbol{0}})/P^{(C)}_{er,\xi}(i^j_{l,k}=1)} = L^{(S)}_{\xi}(i^j_{l,k}) - L^{(C)}_{er,\xi}(i^j_{l,k}) \ . \quad (44)$$

Thus, in the whole ISCD algorithm the L-values $L^{(C)}_{er,\xi}(i^j_{l,k})$ from the SISO channel decoder are used and the probabilities $P^{(C)}_{er,\xi}(i^j_{l,k}=\lambda^j_l)$ are no longer required.

V. SIMULATION RESULTS

Independent Gaussian random signals were correlated by a first-order recursive low-pass filter (coefficient $a=0.9$) in order to generate the signals $X^j_k, j=1,...,M$. In the first simulations 5-bit optimal scalar quantizers (i.e., $N^j=5\ \forall j$) were used as source encoders. Their output-bits were spread by a random-interleaver at each time k and afterwards they were commonly channel-encoded by a terminated rate-1/2 recursive systematic convolutional code (RSC-code[8], [23]). The codewords were transmitted over the AWGN-channel.

In the first simulation $M=2$ signals and a terminated memory-2 rate-1/2 RSC-channel-code[9] were considered (code-rate $R=10/24$). The results are depicted in Fig. 3. The mean SNR (averaged over all signals) is plotted over E_b/N_0, the ratio of the energy E_b per transmitted *data*-bit and the one-sided power spectral density N_0 of the channel-noise.

[8]Recursive Systematic Convolutional Codes, [23]

[9]Generator polynomials: $g_0(D)=1+D+D^2$ and $g_1(D)=1+D^2$; $g_0(D)$ was used for the feedback part.

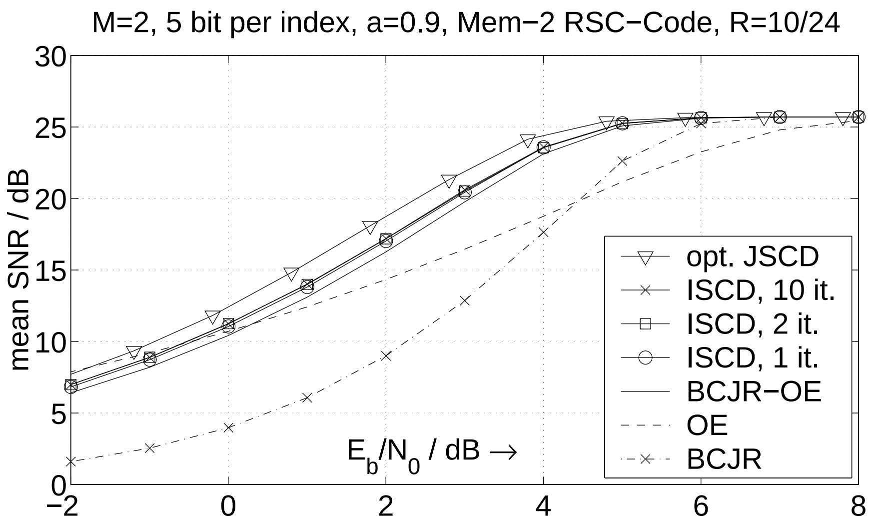

Fig. 3. Performance of iterative and optimal joint source-channel decoding. Transmission of $M=2$ autocorrelated signals ($a=0.9$) in parallel. — Performances des décodages conjoints source-canal itératif et optimal. Transmission de $M=2$ signaux auto-corrélées ($a=0.9$) en parallèle.

Clearly, the "optimal JSCD" decoder works best, followed by the iterative decoding (curves "ISCD"). It should be noticed that the difference between the "opt. JSCD" and the "ISCD"-curves is only due to the approximations in the iterative decoding scheme, i.e., the neglect of the bit-correlations of the quantizer indices in (24) and (31).

An additional decoder was realized (curve "BCJR-OE") which used the BCJR-algorithm for channel decoding with "zero" a-priori information followed by Optimal Estimation (OE) for source decoding, i.e., the scheme had the same constituent algorithms as ISCD. Due to its better initialization ISCD outperforms the "BCJR-OE"-decoder, even if only one iteration is carried out. In contrast to "BCJR-OE", ISCD utilizes the time-based correlations of the index-bits for channel decoding.

Obviously, ISCD can not take advantage of the mutual dependencies of the index-bits, since the performance hardly increases with the number of iterations. This can be explained by the short length of the channel code which does not allow sufficient interleaving[10] to make the extrinsic bit-information from the source decoder "virtually" independent for the channel decoder.

It is known that the performance of concatenated codes and their iterative decoding schemes is better, if a code with "long" codewords is used. Therefore, a similar simulation as above was carried out, but with $M=50$ commonly channel-encoded indices and a terminated memory-4 RSC-code[11] (code-rate $R = 250/508$). The results are depicted in Fig. 4: Now, the second iteration leads to a strong improvement compared to "ISCD, 1 it." and "BCJR-OE", i.e., the mutual correlations of the bits can be exploited in part by the iterative decoding scheme. Unfortunately, the simulation of the optimal decoder (including the interleaver) is too complex in this case, so a comparison with the best possible performance of the concatenated code (without the approximations inherent in ISCD) can not be carried out.

It is interesting to notice that more than two iterations do not significantly increase the SNR in Fig. 4. This is due to the "short" source codewords, which consist of just five bits, so that they influence only a small number of RSC-code-bits.

In Fig. 5 the results of a simulation similar to the previous one are shown. The difference is that the source signal is only weakly autocorrelated by setting

[10] One should recall that the length of the interleaver is limited to the number of information bits at each time k in order to avoid additional delay.

[11] Generator polynomials: $g_0(D) = 1+D^3+D^4$ and $g_1(D) = 1+D+D^3+D^4$; $g_0(D)$ was used for the feedback part.

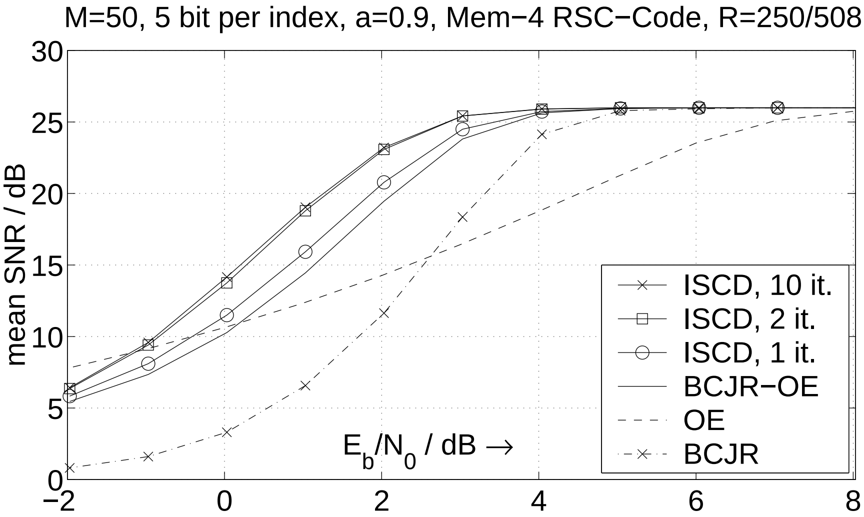

Fig. 4. Performance of iterative source-channel decoding and its constituent algorithms. Transmission of $M=50$ autocorrelated signals ($a=0.9$) in parallel. — Performances du décodage conjoint source-canal itératif, et de ses décodeurs constitutifs. Transmission de $M=50$ signaux auto-corrélées ($a=0.9$) en parallèle.

the low-pass-filter coefficient to $a=0.5$. This time, there is almost no difference between "BCJR-OE" and "ISCD, 1 it.". This means, that the bits have almost no autocorrelation. Qualitatively, the behavior of ISCD is the same as above. The second iteration produces a gain (a small one in this case) but more than two iterations again do not further increase the performance. Here, the gain by the second iteration is due to the well known [25] non-uniform probability distribution of the indices at the output of an optimal quantizer, which leads to bit-correlations in the quantizer index that can only be exploited by the iterations.

In Fig. 6 the influence of the interleaver on the performance of ISCD is investigated. For this purpose a terminated memory-6 rate-1/2 RSC-code[12] was used

[12]Generator polynomials: $g_0(D)=1+D^2+D^3+D^5+D^6$ and $g_1(D)=1+D+D^2+D^3+D^6$; $g_0(D)$ was used for the feedback part.

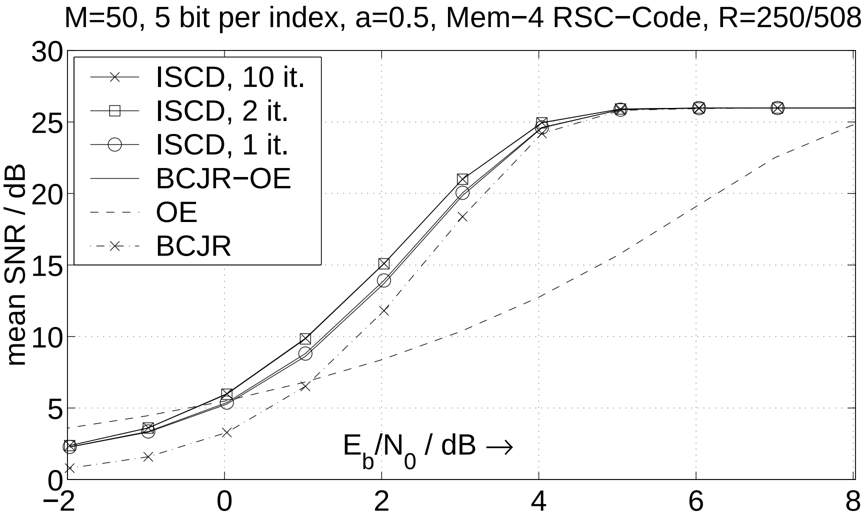

Fig. 5. Performance of iterative source-channel decoding and its constituent algorithms. Transmission of $M = 50$ autocorrelated signals ($a = 0.5$) in parallel. — Performances du décodage conjoint source-canal itératif, et de ses décodeurs constitutifs. Transmission de $M = 50$ signaux auto-corrélées ($a = 0.5$) en parallèle.

and the source signals were quantized by 4 bits. The filter-coefficient was set to $a = 0.9$, i.e., the signals were strongly autocorrelated. The performance of ISCD with 2 iterations was compared for $M = 15$ indices (resulting in 60 data bits) and $M = 100$ indices (resulting in 400 data bits), in both cases with a random interleaver and without an interleaver. Fig. 6 shows that the application of an interleaver significantly improves the performance of ISCD. The gain[13] gets larger with increasing block length.

[13]Even if no interleaver is used, a gain caused by the increasing block-length can be observed. The reason is that the rate-loss due to the tail-bits of the terminated convolutional code is smaller if the block-length is enlarged (E_b/N_0 is noted on the x-axis, where E_b is the energy per transmitted *data*-bit.).

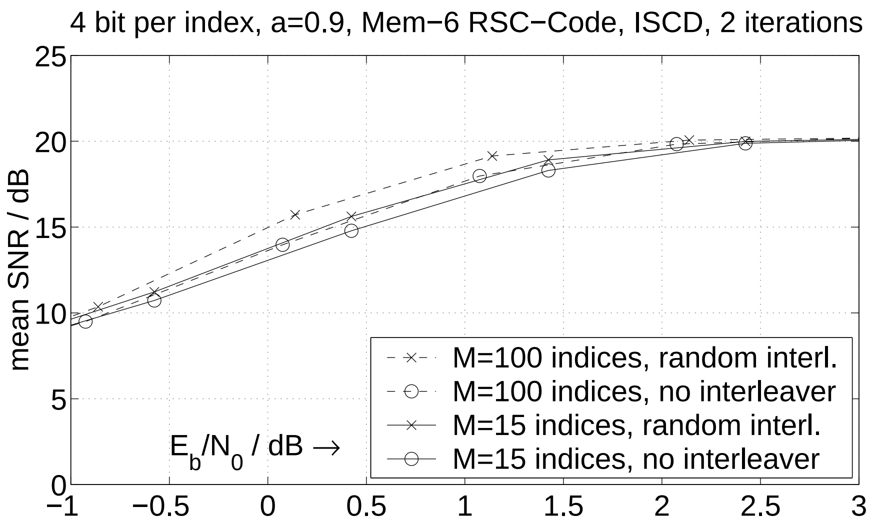

Fig. 6. Performance of iterative source-channel decoding for several block lengths and interleavers. — Performances du décodage source-canal itératif pour plusieurs longueurs de blocs et entrelaceurs.

VI. CONCLUSIONS

Iterative source-channel decoding was stated in a generalized framework for communication systems that use binary channel codes. It was derived from the optimal joint decoding algorithm by insertion of an approximation, that allows to decompose the optimal scheme into two constituent algorithms, namely the well-known symbol-by-symbol APP-algorithm for the channel-decoding part and the Optimal-Estimation algorithm for the source-decoding part. Both algorithms are applied alternately in several iterations to compensate for the loss of performance due to the approximation and they exploit only "their" type of redundancy, i.e., either the channel-code redundancies (APP-algorithm), or the residual redundancies in the source-encoder indices (OE-algorithm).

In the simulations, in which convolutional channel codes were used, it was shown, that ISCD achieves a strongly better quality of transmission than its constituent algorithms applied only once. Furthermore, the application of a random interleaver leads to a significantly better performance compared to a system without an interleaver. Both the absolute performance of ISCD and the gain due to the interleaver grow if the block length is increased. In all simulations only two iterations were required to nearly achieve the maximum performance.

Overall, ISCD works best in applications where long blocks of correlated data bits are transmitted. It is well suited for a combination with practical source-codecs for speech, audio, and image signals, because it exploits the remaining redundancies in a systematic way and the quantizer-indices are allowed to have different, arbitrary numbers of bits. Additionally, it offers scalable complexity at the decoder, since it is up to the system designer if he wants to perform iterations or not.

References

[1] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Transactions on Information Theory*, vol. 36, no. 4, pp. 799–809, July 1990.

[2] D. Miller and K. Rose, "Combined source-channel vector quantization using deterministic annealing," *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, pp. 347–356, Feb./Mar./Apr. 1994.

[3] K. Zeger and A. Gersho, "Pseudo-Gray coding," *IEEE Transactions on Communications*, vol. 38, no. 12, pp. 2147–2158, Dec. 1990.

[4] J. Hagenauer, "Source-controlled channel decoding," *IEEE Transactions on Communications*, vol. 43, no. 9, pp. 2449–2457, Sept. 1995.

[5] T. Hindelang and A. Ruscitto, "Kanaldecodierung mit Apriori-Wissen bei nicht binären Quellensymbolen," in *Proceedings of the ITG-Fachtagung "Codierung für Quelle, Kanal und Übertragung"*, Mar. 1998, pp. 163–167.

[6] S. Heinen, A. Geiler, and P. Vary, "MAP channel decoding by exploiting multilevel source a priori knowledge," in *Proceedings of the ITG-Fachtagung "Codierung für Quelle, Kanal und Übertragung"*, Mar. 1998, pp. 89–94.

[7] K. Sayood and J.C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Transactions on Communications*, vol. 39, no. 6, pp. 838–846, June 1991.

[8] N. Phamdo and N. Farvardin, "Optimal detection of discrete Markov sources over discrete memoryless channels – applications to combined source-channel coding," *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 186–193, Jan. 1994.

[9] T. Fingscheidt and P. Vary, "Robust speech decoding: A universal approach to bit error concealment," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 1997, vol. 3, pp. 1667–1670.

[10] M. Skoglund, "Hadamard-based soft decoding for vector quantization over noisy channels," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 515–532, Mar. 1999.

[11] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[12] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, Nov. 1989, vol. 3, pp. 1680–1686.

[13] N. Görtz, "Joint source channel decoding using bit-reliability information and source statistics," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, Aug. 1998, p. 9.

[14] N. Görtz, "Joint source-channel decoding by channel-coded optimal estimation," in *Proceedings of the 3rd ITG Conference Source and Channel Coding*, Jan. 2000, pp. 267–272.

[15] S. Heinen, S. Bleck, and P. Vary, "Robust speech transmission over noisy channels employing non-linear block codes," in *Proceedings of the IEEE Speech Coding Workshop*, June 1999, pp. 72–74.

[16] N. Görtz, "Iterative source-channel decoding using soft-in/soft-out decoders," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, June 2000, p. 173.

[17] N. Görtz, "Analysis and performance of iterative source-channel decoding," in *Proceedings of the International Symposium on Turbo Codes & Related Topics*, Sept. 2000, pp. 251–254.

[18] N. Görtz, "On the iterative approximation of optimal joint source-channel decoding," *IEEE Journal on Selected Areas in Communications*, accepted for publication.

[19] T. Hindelang, T. Fingscheidt, N. Seshadri, and R.V. Cox, "Combined source/channel (de-)coding: Can a priori information be used twice?," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, June 2000, p. 266.

[20] N. Görtz, "Zero-redundancy error protection for CELP speech codecs," in *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH)*, Sept. 1997, vol. 3, pp. 1283–1286.

[21] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP): high-quality speech at very low bit rates," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Mar. 1985, vol. 1, pp. 25.1.1–25.1.4.

[22] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[23] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[24] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Transactions on Telecommunications (ETT)*, vol. 8, no. 2, pp. 119–125, Mar. 1997.

[25] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1984.

List of Figures