A-posteriori Probability Decoding of Variable-Length Codes Using a Three-Dimensional Trellis Representation

Jörg Kliewer and Ragnar Thobaben

University of Kiel, Institute for Circuits and Systems Theory, 24143 Kiel, Germany
{jkl, rat}@tf.uni-kiel.de

Abstract— **In this contribution we present an improved index-based a-posteriori probability (APP) decoding approach for variable-length encoded packetized data, where implicit residual source correlation is exploited for error protection. The proposed algorithm is based on a novel generalized two-dimensional state representation which leads to a three-dimensional trellis with unique state transitions. APP decoding on this trellis is realized by employing a two-dimensional version of the classical BCJR algorithm. This new method has the advantage that due to the unique state representation all available a-priori information can be fully exploited, which especially holds for the transition probabilities of the Markov model associated with the variable-length encoded source indices. Simulation results for an additional error protection by channel codes and iterative joint source-channel decoding show that the proposed approach leads to an increased error-correction performance compared to previously published results where a one-dimensional state representation is used.**

I. Introduction

Recently, robust transmission of variable-length encoded signals over wireless channels has become a very active research area. This is mainly due to the increased demand for mobile access of multimedia data where the source compression usually relies on standardized approaches employing variable-length codes (VLCs). Many authors have considered trellis-based joint source-channel decoding techniques for the reliable transmission of variable-length encoded data (see e.g. [1–7]). Especially, a bit-level VLC trellis representation is proposed in [1], [7], which allows to use the BCJR algorithm [8] as a-posteriori probability (APP) decoder. In [9] the bit-level trellises for a VLC and a convolutional channel code are merged for iterative joint source-channel decoding of turbo-encoded VLC source symbols. On the other hand, a symbol-level trellis representation for uncorrelated variable-length encoded source data is presented in [5]. This approach is extended in [10] to first-order Markov sources, where the residual source index correlation is additionally exploited in the decoding process.

In this paper we propose a new source decoding algorithm for packetized correlated variable-length encoded source sequences based on a symbol-based two-dimensional (2-D) state representation, which can be regarded as a generalization of the approach in [10]. APP decoding on the corresponding three-dimensional (3-D) VLC trellis can then be carried out by a 2-D modification of the classical BCJR algorithm. In combination with a subsequent mean-squares (MS) or maximum a-posteriori probability (MAP) estimation the proposed VLC decoder is optimal. We show that both the decoding approach from [10] and optimal decoding for the constant word length case can be regarded as special cases of the proposed technique.

II. VLC APP Source Decoding

A. Transmission model

The underlying transmission model is shown in Fig. 1, where $\mathbf{U} = [U_1, U_2, \ldots, U_K]$ denotes one packet consisting of K correlated source symbols. After subsequent (vector-) quantization the resulting indices $I_k \in \mathcal{I}$ from the finite alphabet $\mathcal{I} = \{0, 1, \ldots, 2^M - 1\}$ are represented with M bits. In the following we model the residual index correlation in the index vector $\mathbf{I} = [I_1, I_2, \ldots, I_K]$, which is due to delay and complexity constraints for the quantization stage, by a first-order (stationary) Markov process with index transition probabilities $P(I_k = \lambda \,|\, I_{k-1} = \mu)$ for $\lambda, \mu \in \mathcal{I}$. VLC encoding is then carried out by mapping a fixed-length index I_k to a variable-length bit vector $\mathbf{c}(\lambda) = \mathcal{C}(I_k = \lambda)$ with the prefix code $\mathcal{C}$. This leads to a binary sequence $\mathbf{v} = [v_1, v_2, \ldots, v_N]$ of length N where $v_n \in \{0, 1\}$ represents a single bit at bit index n. The transmission of one bit over the channel characterized by the p.d.f. $p(\hat{v}_n \,|\, v_n)$ results in a soft-bit $\hat{v}_n \in \mathbb{R}$ at the channel output.

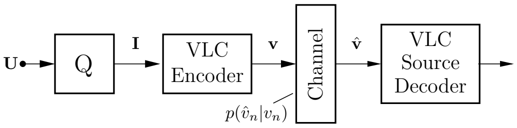

Fig. 1. Model of the transmission system

B. Trellis representation

A possible index-based VLC trellis representation is given in [5], where an example is depicted in Fig. 2(a) for $K = 5$, $N = 10$, and the Huffman code

$$\mathcal{C} = \{\mathbf{c}(0) = [1], \mathbf{c}(1) = [0, 1], \mathbf{c}(2) = [0, 0, 0], \mathbf{c}(3) = [0, 0, 1]\}. \quad (1)$$

The time-varying trellis with the one-dimensional (1-D) states $S_k = n$ represents all possible bit positions n for a given time instant k, where the transition from state $S_{k-1} = n_1$ to $S_k = n_2$ is caused by the source hypothesis $I_k = \lambda$, $\lambda \in \mathcal{I}$, and the corresponding Huffman codeword $\mathbf{c}(\lambda) = \mathcal{C}(I_k = \lambda)$ of length $l(\mathbf{c}(\lambda)) = n_2 - n_1$. In the following the set of all possible states $S_k = n$ at time instant k is denoted with $\mathcal{N}_k$. Note that parallel

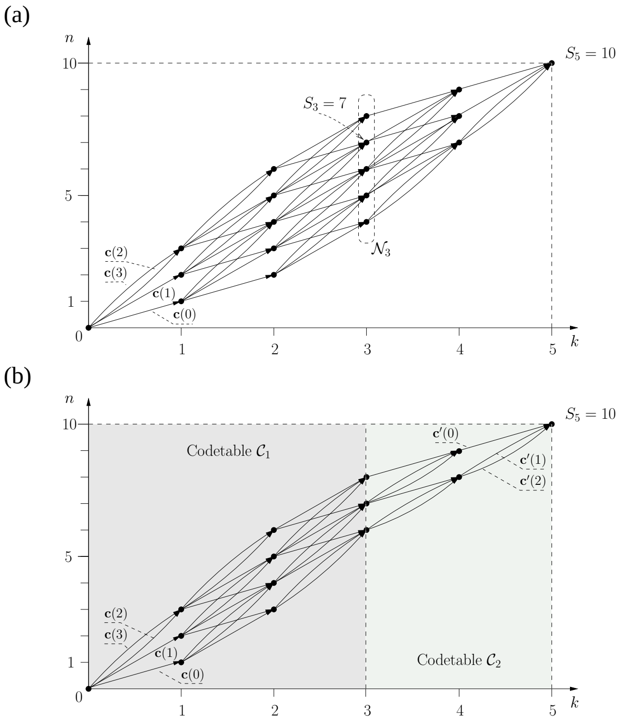

Fig. 2. VLC trellis representation for $K=5$, $N=10$: (a) Codetable $\mathcal{C} = \{\mathbf{c}(0)=[1], \mathbf{c}(1)=[0,1], \mathbf{c}(2)=[0,0,0], \mathbf{c}(3)=[0,0,1]\}$ [5], (b) codetables $\mathcal{C}_1=\mathcal{C}$ from (1) and $\mathcal{C}_2=\{\mathbf{c}'(0)=[1], \mathbf{c}'(1)=[0,1], \mathbf{c}'(2)=[0,0]\}$

state transitions may be possible when two or more entries of the codetable $\mathcal{C}$ have the same bit length.

As a new, to our knowledge previously unpublished feature regarding the VLC trellis in Fig. 2(a) we note that it is possible to change the codetable $\mathcal{C}$ and thus also the bit allocation within a source packet. This may be useful in practical source coding situations with short-time stationary source signals. An example is depicted in Fig. 2(b) where the codetable $\mathcal{C}_1=\mathcal{C}$ from (1) is used for $k \in \{0,\ldots,3\}$, whereas the code $\mathcal{C}_2 = \{\mathbf{c}'(0)=[1], \mathbf{c}'(1)=[0,1], \mathbf{c}'(2)=[0,0]\}$ is applied to the source symbols for the rest of the packet. We can see from this example that the trellis is now described by the quantities K and N, the different codetables, and their positions. Note that when an APP decoder as described in Section II-C is used the decoding operation for the new codetable already starts with properly initialized reliability values for the trellis states (i.e. in Fig. 2(b) for $k=3$).

C. A-posteriori probabilities

The VLC trellis representation may now be used in order to calculate the a-posteriori probabilities (APPs) $P(I_k=\lambda\,|\,\hat{\mathbf{v}})$ via a modification of the classical BCJR algorithm [10], where in the following this method is denoted as 1-D APP decoding. For the sake of clarity we restrict ourselves to just a single codetable for the whole packet. The first step of the derivation is given as

$$P(I_k=\lambda\,|\,\hat{\mathbf{v}}) = \frac{1}{p(\hat{\mathbf{v}})} \sum_{n_2\in\mathcal{N}_k} \sum_{n_1\in\mathcal{N}_{k-1}} \underbrace{p(\hat{\mathbf{v}}_{n_2+1}^{N}\,|\,S_k=n_2)}_{=:\,\beta_k(n_2)} \cdot$$

$$\cdot \underbrace{p(\hat{\mathbf{v}}_{n_1+1}^{n_2}, I_k=\lambda, S_k=n_2\,|\,S_{k-1}=n_1, \hat{\mathbf{v}}_1^{n_1})}_{=:\,\gamma_k(\lambda,n_2,n_1)} \cdot$$

$$\cdot \underbrace{p(S_{k-1}=n_1, \hat{\mathbf{v}}_1^{n_1})}_{=:\,\alpha_{k-1}(n_1)}, \qquad (2)$$

where $\hat{\mathbf{v}}_a^b = [\hat{v}_a, \hat{v}_{a+1}, \ldots, \hat{v}_b]$ denotes a subsequence from bit position a to b. It is shown in [10] that by including the first-order Markov model for the source indices the following expression for the term $\gamma_k(\lambda, n_2, n_1)$ is obtained:

$$\gamma_k(\lambda,n_2,n_1) = p(\hat{\mathbf{v}}_{n_1+1}^{n_2}\,|\,I_k=\lambda) \cdot$$

$$\cdot \sum_{\mu=0}^{2^M-1} \underbrace{P(I_k=\lambda, S_k=n_2\,|\,I_{k-1}=\mu, S_{k-1}=n_1)}_{=:\,P_c(\lambda,n_2,\mu,n_1)} \cdot$$

$$\cdot \frac{1}{C_1(n_1)}\,\gamma_{k-1}(\mu,n_1,n_0)\,\alpha_{k-2}(n_0) \qquad (3)$$

with $n_0 = n_1 - l(\mathbf{c}(\mu))$ and $C_1(n_1)$ denoting a constant depending on n_1. The conditional probability $P_c(\lambda,n_2,\mu,n_1)$ in (3) represents the transition probability of the Markov source adapted to the time-varying VLC trellis according to

$$P_c(\lambda,n_2,\mu,n_1) =$$

$$\frac{1}{C_2(n_1,\mu)} \begin{cases} P(I_k=\lambda\,|\,I_{k-1}=\mu) & n_2-n_1=l(\mathbf{c}(\lambda)), \\ 0 & \text{otherwise.} \end{cases} \qquad (4)$$

The normalization factor $C_2(n_1,\mu)$ considers that due to the non-stationary VLC trellis only certain transitions from $S_{k-1}=n_1$ to $S_k = n_1 + l(\mathbf{c}(\lambda))$ are possible. Finally, the α-term can now be calculated using the forward recursion

$$\alpha_k(n_2) = \sum_{n_1\in\mathcal{N}_{k-1}} \sum_{\lambda=0}^{2^M-1} \gamma_k(\lambda,n_2,n_1)\cdot\alpha_{k-1}(n_1),\ \alpha_0(0)=1, \qquad (5)$$

which exploits the residual source index correlation as additional a-priori information by using the modified γ-term in (3). Similarly, the $\beta_k(n_1)$ are calculated with the backward recursion [10]

$$\beta_k(n_1) = \sum_{n_2\in\mathcal{N}_{k+1}} \sum_{\lambda=0}^{2^M-1} \beta_{k+1}(n_2)\cdot\gamma'_{k+1}(\lambda,n_2,n_1) \qquad (6)$$

with $\beta_K(N)=1$. Note that due to the causal definition of the source transition probabilities from time instant $(k-1)$ to k it is not possible to incorporate this additional a-priori information in the backward recursion. Therefore, only the source probability distribution $P(I_k=\lambda), \lambda \in \mathcal{I}$, adapted to the time-varying VLC trellis can be considered in the corresponding γ'-term in (6).

The APPs $P(I_k=\lambda\,|\,\hat{\mathbf{v}})$ can now be utilized for reconstructing the source symbol packet $\hat{\mathbf{U}}$ via mean-squares (MS) [11] or MAP estimation.

D. Special case: Constant codeword length

In the special case of a constant codeword length of M bits for the whole packet (which is equivalent to not using a VLC at all) the trellises in Fig. 2 degenerate to a straight line with 2^M parallel branches between the uniquely determined trellis states

$$S_{k-1} = n_1 = (k-1)\cdot M \quad \text{and} \quad S_k = n_2 = k\cdot M. \tag{7}$$

Therefore, the α- and β-terms in (2) do not depend on the state hypotheses anymore, and thus can be written as simple p.d.f.s

$$\alpha_{k-1}(n_1) = p(\hat{\mathbf{v}}_1^{n_1}) \quad \text{and} \quad \beta_k(n_2) = p(\hat{\mathbf{v}}_{n_2+1}^N). \tag{8}$$

By inserting these p.d.f.s into (2) and applying (7) to the γ-term we obtain

$$\begin{aligned} P(I_k = \lambda \,|\, \hat{\mathbf{v}}) &= \sum_{n_2\in\mathcal{N}_k} \sum_{n_1\in\mathcal{N}_{k-1}} \frac{p(\hat{\mathbf{v}}_{n_1+1}^{n_2}, I_k = \lambda \,|\, \hat{\mathbf{v}}_1^{n_1})}{p(\hat{\mathbf{v}}_{n_1+1}^{n_2})} \\ &= P(I_k = \lambda \,|\, \hat{\mathbf{v}}_1^{n_2}). \end{aligned} \tag{9}$$

We can see that for constant-length codewords the APPs are no longer conditioned on all received soft-bits for the whole data packet, instead only the previously received part of the channel observation $\hat{\mathbf{v}}_1^{n_2} = \hat{\mathbf{v}}_1^{kM}$ up to the time instant k is utilized. The APPs $P(I_k = \lambda \,|\, \hat{\mathbf{v}}) = P(I_k = \lambda \,|\, \hat{\mathbf{v}}_1^{kM})$ may be then derived by simply applying the normalized BCJR forward recursion according to

$$\begin{aligned} P(I_k = \lambda \,|\, \hat{\mathbf{v}}) = P(I_k = \lambda \,|\, \hat{\mathbf{v}}_1^{n_2}) = \frac{p(\hat{\mathbf{v}}_{n_1+1}^{n_2} \,|\, I_k = \lambda)}{p(\hat{\mathbf{v}}_{n_1+1}^{n_2})} \cdot \\ \cdot \sum_{\mu=0}^{2^M-1} P(I_k = \lambda \,|\, I_{k-1} = \mu)\, P(I_{k-1} = \mu \,|\, \hat{\mathbf{v}}_1^{n_1}) \end{aligned}$$

with $n_1 = (k–1)\cdot M$, $n_2 = k\cdot M$, and $P(I_k = \lambda \,|\, I_{k-1} = \mu)$ denoting the transition probabilities between adjacent source indices.

III. Improved APP decoding by using a 3-D trellis

In this section we present a novel generalized decoding algorithm based on a 3-D VLC trellis representation, where both the bit position n and the source hypotheses $I_k = \lambda$, $\lambda \in \mathcal{I}$, are treated as separate variables. We show that in contrast to the 1-D APP decoding method from above the proposed approach now allows to exploit all available a-priori information for error protection, which also holds for constant codeword lengths.

A. Trellis representation and decoding algorithm

In a first step we define 2-D states $S'_k = (\lambda, n) \in (\mathcal{I} \times \mathcal{N}_k)$ leading to a general 3-D trellis representation, where for the codetables used in Fig. 2(b) the resulting 3-D trellis is shown in Fig. 3. Note that now all parallel branches have been removed from the trellis, and the state transitions are now unique due to the separation of the state variables n and λ. As it can be observed from Fig. 3 multiple codetables within the same source packet may also be used with this type of trellis. However, for simplicity reasons in the following we again consider only the single codetable case, but an extension to multiple tables can be obtained in a straightforward way.

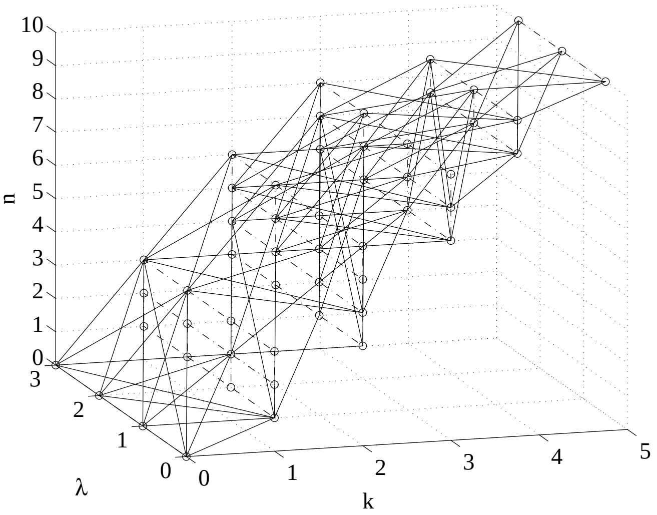

Fig. 3. 3-D VLC trellis for $K = 5$, $N = 10$, $\mathcal{C}_1 = \mathcal{C}$ from (1), and $\mathcal{C}_2 = \{\mathbf{c}'(0) = [1], \mathbf{c}'(1) = [0,1], \mathbf{c}'(2) = [0,0]\}$

We now derive a 2-D extension of the BCJR algorithm denoted as 2-D APP decoding in the following, which is based on the proposed 3-D trellis representation. Here, the APPs $P(S'_k = (\lambda, n_2) \,|\, \hat{\mathbf{v}})$ are decomposed analog to [8] as

$$\begin{aligned} P(S'_k = (\lambda, n_2) \,|\, \hat{\mathbf{v}}) = \frac{1}{p(\hat{\mathbf{v}})} \cdot \\ \cdot \underbrace{p(\hat{\mathbf{v}}_{n_2+1}^N \,|\, S'_k = (\lambda, n_2))}_{=:\, \beta_k^{(2D)}(\lambda, n_2)} \cdot \underbrace{p(S'_k = (\lambda, n_2), \hat{\mathbf{v}}_1^{n_2})}_{=:\, \alpha_k^{(2D)}(\lambda, n_2)}. \end{aligned} \tag{10}$$

In this case the 2-D forward recursion for the calculation of $\alpha_k^{(2D)}(\lambda, n_2)$ can be stated as

$$\alpha_k^{(2D)}(\lambda, n_2) = \sum_{\substack{(\mu, n_1) \\ \in (\mathcal{I}\times\mathcal{N}_{k-1})}} \gamma_k^{(2D)}(\lambda, n_2, \mu, n_1)\, \alpha_{k-1}^{(2D)}(\mu, n_1), \tag{11}$$

and for obtaining $\beta_k^{(2D)}(\lambda, n_1)$ we have the backward recursion

$$\beta_k^{(2D)}(\lambda, n_1) = \sum_{\substack{(\mu, n_2) \\ \in (\mathcal{I}\times\mathcal{N}_{k+1})}} \beta_{k+1}^{(2D)}(\mu, n_2)\, \gamma_{k+1}^{(2D)}(\mu, n_2, \lambda, n_1). \tag{12}$$

Both forward and backward recursion may be initialized with the source index probability distribution, where

$$\alpha_0^{(2D)}(\mu, 0) = P(I_0 = \mu),\ \beta_K^{(2D)}(\mu, N) = P(I_K = \mu),\ \mu \in \mathcal{I}.$$

The term $\gamma_k^{(2D)}(\cdot)$ in (11) and (12), resp., corresponds to the reliability information of a single trellis branch which is given with $P_c(\lambda, n_2, \mu, n_1)$ from (4) as

$$\begin{aligned}\gamma_k^{(2D)}&(\lambda, n_2, \mu, n_1)\\ &= p(\hat{\mathbf{v}}_{n_1+1}^{n_2}, S'_k = (\lambda, n_2) \mid S'_{k-1} = (\mu, n_1))\\ &= p(\hat{\mathbf{v}}_{n_1+1}^{n_2} \mid S'_k = (\lambda, n_2)) \cdot P_c(\lambda, n_2, \mu, n_1). \qquad (13)\end{aligned}$$

Analog to the γ-term for the 1-D APP decoding approach in (3) here $\gamma_k^{(2D)}(\cdot)$ also includes bit reliability information from the soft-output channel specified by the p.d.f. $p(\hat{\mathbf{v}}_{n_1+1}^{n_2} \mid S'_k = (\lambda, n_2))$. Due to the 2-D state representation the normalized transition probabilities of the source symbols $P_c(\lambda, n_2, \mu, n_1)$ are included directly and not via a marginal probability as in (3). Note that in contrast to 1-D APP decoding where the residual source index correlation is only used in the forward recursion (5), here the first-order Markov property of the source is exploited as a-priori information in *both* forward and backward recursion.

Finally, we obtain the APPs $P(I_k = \lambda \mid \hat{\mathbf{v}})$ for the source hypotheses $I_k = \lambda$ via calculating the marginal distribution of the APPs $P(S'_k = (\lambda, n_2) \mid \hat{\mathbf{v}})$ in (10) according to

$$P(I_k = \lambda \mid \hat{\mathbf{v}}) = \frac{1}{p(\hat{\mathbf{v}})} \sum_{n_2 \in \mathcal{N}_k} \beta_k^{(2D)}(\lambda, n_2)\, \alpha_k^{(2D)}(\lambda, n_2). \qquad (14)$$

Again, the APPs $P(I_k = \lambda \mid \hat{\mathbf{v}})$ can be utilized for a MS or MAP estimation to reconstruct the source symbol packet $\hat{\mathbf{U}}$, which leads to an optimal decoding process in the MS or MAP sense.

B. Special cases

1) Relation to 1-D APP decoding: We now show that the 1-D APP decoding technique summarized in Section II-C can be regarded as special case of the 2-D approach from above. Since (2) can also be written as

$$P(I_k = \lambda \mid \hat{\mathbf{v}}) = \frac{1}{p(\hat{\mathbf{v}})} \sum_{n_2 \in \mathcal{N}_k} \beta_k(n_2) \underbrace{\sum_{n_1 \in \mathcal{N}_{k-1}} \gamma_k(\lambda, n_2, n_1)\, \alpha_{k-1}(n_1)}_{=:\, p(\lambda, n_2)} \qquad (15)$$

where

$$\begin{aligned} p(\lambda, n_2) &= \sum_{n_1 \in \mathcal{N}_{k-1}} p(I_k = \lambda, S_k = n_2, S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_2})\\ &= p(I_k = \lambda, S_k = n_2, \hat{\mathbf{v}}_1^{n_2}) = \alpha_k^{(2D)}(\lambda, n_2), \qquad (16)\end{aligned}$$

we now obtain the relation

$$P(I_k = \lambda \mid \hat{\mathbf{v}}) = \frac{1}{p(\hat{\mathbf{v}})} \sum_{n_2 \in \mathcal{N}_k} \beta_k(n_2)\, \alpha_k^{(2D)}(\lambda, n_2). \qquad (17)$$

By comparing (14) and (17) it can be observed that both approaches use the same forward recursion, but differ in the type of the backward recursion. The $\beta_k(n_2)$ in (17) are derived by only using the 1-D states $S_k = n_2$, $n_2 \in \mathcal{N}_2$, which confirms the fact that 1-D APP decoding cannot exploit residual source correlation in the backward recursion.

2) APP decoding for constant codeword length: In the following we verify that 2-D APP decoding allows a full backward recursion also for constant-length codewords. Since in the constant-length case the bit position n_2 is uniquely given by $n_2 = k \cdot M$ (see eq. (7)) the states $S'_k = (\lambda, n_2)$ do not depend on n_2 anymore and can thus be directly expressed as source hypotheses $S'_k = I_k = \lambda$. Inserting this into (10) leads to

$$P(I_k = \lambda \mid \hat{\mathbf{v}}) = \frac{1}{p(\hat{\mathbf{v}})}\, \beta_k(\lambda)\, \alpha_k(\lambda),$$

where the α- and β-recursions are identical to those for the classical BCJR algorithm formulated with the 2^M-ary source indices I_k. This property is especially useful in the case of multiple codetables where now constant-length codes (i.e. codetables with an identity mapping) can be optimally decoded within a single trellis as well.

IV. Results

Due to the high sensitivity of the variable-length encoded bit stream exploiting the residual source redundancy alone may not provide a reasonable error protection in many transmission situations. In the following simulations therefore an additional protection of the interleaved output of the VLC encoder by a channel code is considered, as depicted in Fig. 4. For this purpose we use a terminated memory-4 rate $R = 3/4$ recursive systematic convolutional code derived from a rate-1/2 mother code by puncturing. As joint source-channel decoder the iterative decoding scheme from [10] is applied where the outer constituent decoder corresponds to the generalized APP VLC decoder from Section III-A. As inner decoder a BCJR APP channel decoder is used, where both decoders exchange extrinsic information in form of conditional log-likelihood values (L-values).

In the simulations we furthermore utilize a standard Huffman code as VLC, and the source redundancy is modeled by an AR(1) process with correlation coefficient $a = 0.9$ quantized with a scalar uniform quantizer. Furthermore, it is assumed that all side information (including N and K) is transmitted without errors. The block-length is chosen as $K = 150$ where the ability of changing the codetable within one source packet is demonstrated by using different Huffman tables for the first, the middle, and the last 50 symbols ($K_1 = K_2 = K_3 = 50$) in the source packet corresponding to a bit length of $M_1 = 4$, $M_2 = 3$, and $M_3 = 4$, respectively.

The simulation results for a fully-interleaved BPSK-modulated Rayleigh channel averaged over 50 source realizations and 100 simulated transmissions are depicted in Fig. 5. The results for a MAP estimation are shown in Fig. 5(a) where the symbol error rate (SER) is plotted over the channel parameter E_b/N_0. Both the JSCD(0) and JSCD(1) techniques utilize 1-D APP decoding from Section II. The JSCD(0) approach only exploits the probability distribution of the source indices, whereas the JSCD(1) method also uses the mutual residual index correlation between the indices I_k due to the first-order Markov source model. The latter also applies to the JSCD-2D method, however, here the 2-D APP decoding technique

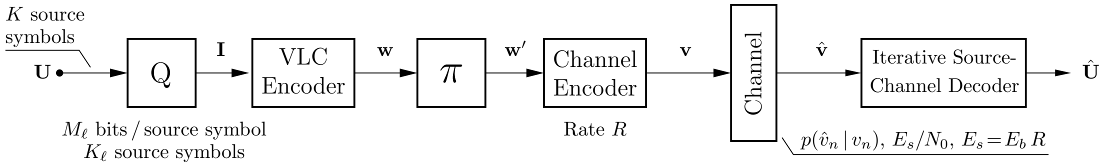

Fig. 4. Transmission system with additional error protection by a channel code

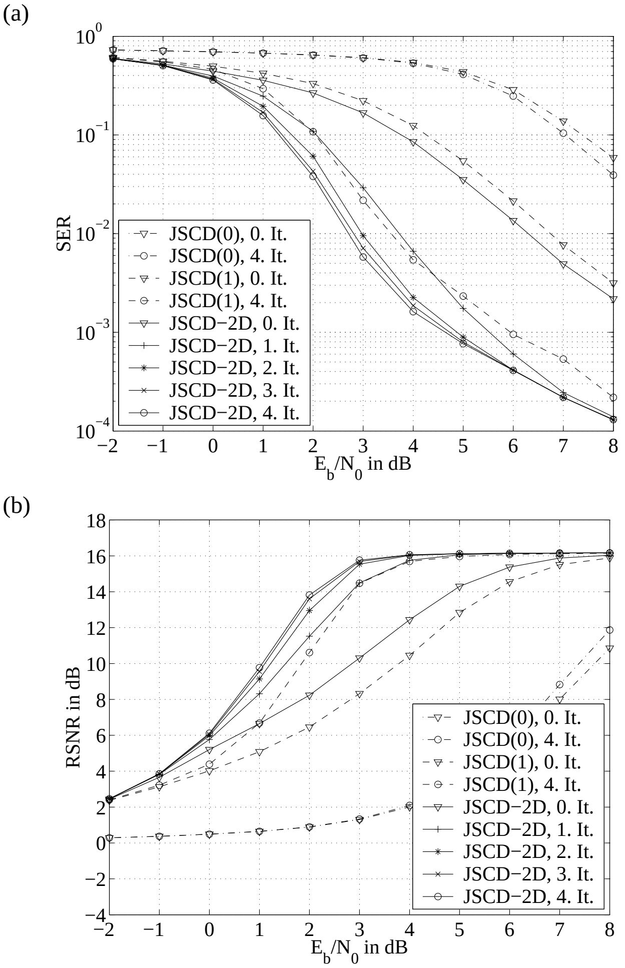

Fig. 5. Simulation results for a fully interleaved flat Rayleigh channel ($K = 150$, three different Huffman tables with $M_1 = 4$, $M_2 = 3$, $M_3 = 4$, each generated for a block of 50 symbols from an AR(1) process with $a = 0.9$): (a) MAP estimation and symbol error rate (SER), (b) mean-squares estimation and reconstruction SNR (RSNR).

from Section III is used. We can see from Fig. 5(a) that the JSCD-2D approach leads to an increased SER reduction by almost half an order of magnitude in the waterfall region compared to the JSCD(1) technique. This is due to the fact that the JSCD-2D approach is capable of exploiting *all* available a-priori information also in the backward recursion which is the case for the JSCD(1) decoder only in the forward recursion of the BCJR algorithm. Similar results hold for the MS estimation displayed in Fig. 5(b), where the reconstruction SNR (RSNR) versus the channel E_b/N_0 is shown. Here, for $E_b/N_0 > 3\,\text{dB}$ we almost obtain clear channel quality for the JSCD-2D decoding approach.

V. Conclusion

As a new result we have presented an improved VLC APP decoding algorithm based on a general 2-D state representation which exploits the residual source correlation in the variable-length encoded data packet for error protection. This leads to an index-based 3-D VLC trellis with unique transitions between adjacent trellis states, where the corresponding APP decoder employs a 2-D version of the BCJR algorithm. Besides, the proposed 3-D trellis allows to use multiple codetables within one source packet, which is advantageous for short-time stationary sources. We have then shown that both APP decoding using the VLC trellis from [5, 10] and BCJR APP decoding for the fixed-length case are included as special cases in our approach. When employing an additional error protection by channel codes and an iterative source-channel decoding scheme, where the proposed VLC APP source decoder is used as outer constituent decoder, a subsequent MAP estimation leads to a reduction in symbol error rate up to half an order of magnitude compared to the APP decoding approach previously published in [10].

References

[1] V. B. Balakirsky, "Joint source-channel coding with variable length codes," in *Proc. IEEE Int. Sympos. Information Theory*, Ulm, Germany, June 1997, p. 419.

[2] J. Wen and J. D. Villasenor, "Utilizing soft information in decoding of variable length codes," in *Proc. IEEE Data Compression Conference*, Snowbird, UT, USA, Mar. 1999, pp. 131–139.

[3] K. Sayood, H. H. Otu, and N. Demir, "Joint source/channel coding for variable length codes," *IEEE Trans. on Comm.*, vol. 48, no. 5, pp. 787–794, May 2000.

[4] M. Park and D. J. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," *IEEE Trans. on Comm.*, vol. 48, no. 1, pp. 1–6, Jan. 2000.

[5] R. Bauer and J. Hagenauer, "Symbol-by-symbol MAP decoding of variable length codes," in *Proc. 3. ITG Conf. on Source and Channel Coding*, Munich, Germany, Jan. 2000, pp. 111–116.

[6] E. Fabre, A. Guyader, C. Guillemot, and M. Robert, "Joint source-channel turbo decoding of entropy-coded sources," *IEEE Journal on Sel. Areas in Comm.*, vol. 19, no. 9, pp. 1680–1696, Sept. 2001.

[7] R. Bauer and J. Hagenauer, "On variable length codes for iterative source/channel decoding," in *Proc. IEEE Data Compression Conference*, Snowbird, UT, USA, Mar. 2001, pp. 273–282.

[8] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. on Inf. Theory*, pp. 294–287, Mar. 1974.

[9] K. Laković and J. Villasenor, "Combining variable length codes and turbo codes," in *Proc. IEEE 55th Vehicular Technology Conference*, Birmingham, AL, USA, May 2002, pp. 1719–1723.

[10] J. Kliewer and R. Thobaben, "Combining FEC and optimal soft-input source decoding for the reliable transmission of correlated variable-length encoded signals," in *Proc. IEEE Data Compression Conference*, Snowbird, UT, USA, Apr. 2002, pp. 83–91.

[11] T. Fingscheid and P. Vary, "Robust speech decoding: A universal approach to bit error concealment," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Munich, Germany, Apr. 1997, pp. 1667–1670.