Symbol-by-Symbol MAP Decoding of Variable Length Codes

Rainer Bauer and Joachim Hagenauer
Institute for Communications Engineering (LNT)
Munich University of Technology (TUM)
e-mail: Rainer.Bauer@ei.tum.de, Hagenauer@ei.tum.de
WWW: http://www.LNT.ei.tum.de

Abstract — **In this paper we introduce a new approach in the decoding of variable length codes. Based on the tree structure of these codes a trellis representation is derived which allows the application of the BCJR algorithm. This algorithm provides us with the a posteriori probabilities of the transmitted source symbols. Therefore we do not only use soft information from an outer decoding stage but also produce a per symbol soft output that can be used in a successive decoding stage. We also propose a scheme where the soft output can be used in an iterative decoding structure.**

I. INTRODUCTION

Variable length codes (VLC) are widely used in state of the art video and audio compression schemes. While they provide a reduction in data rate because redundancy is removed from the source symbols variable length coded data are very sensitive against channel noise. Especially when the compressed data has to be transmitted over mobile radio channels effective means of error correction have to be applied so that the VLC decoder gets an almost error free input. In a conventional VLC decoder the received bit sequence is decoded bit by bit using the prefix property of these codes. The input values to the decoder usually are hard decisions from a preceding decoding stage.

Recently in [2], [3], [4], [9], [10] some schemes have been proposed that look at the decoding of variable length codes as a sequence estimation problem, that can be solved by a modified Viterbi decoder or an alternative dynamic programming approach as described in [5]. Various trellis representations have been proposed for the decoding problem. Both schemes described in [3] and [2] use a graphical representation of the decoding problem that considers memory in the transmitted symbol sequence. While in [3] the memory stems from the given symbol source which was assumed to be first order Markovian in [2] also memory in terms of explicit channel coding by a nonbinary convolutional encoder was added before transmission. In the work of [8] and [10] i.i.d. sources are used which results in a significantly less complex trellis representation. Some of the above mentioned approaches like [3], [4], [5] utilize both the number of bits and the number of source symbols in the decoding procedure while the other schemes like [2], [9] and [10] only use either length information in terms of bits or symbols.

All of the above mentioned schemes are able to use soft input from a previous decoder stage but none except [3] proposes a symbol-level soft output. In [3] a soft-output algorithm for variable length codes is proposed but the soft values are not used for further processing.

The algorithm proposed in this paper is based on an intuitive trellis representation for variable length codes and applies the BCJR algorithm [1] to generate a per symbol soft output. This soft output is utilized in an iterative decoding structure.

After a brief introduction of the notation we use throughout this paper we describe a trellis representation of variable length coded sequences in section III. Our modified BCJR algorithm that takes care of the variable length nature of the codewords is based on this trellis and is described in section IV. In section V we propose a system, where the soft-output of the soft-in/soft-out VLC decoder is used for iterative decoding in a concatenated scheme consisting of an outer variable length code and an inner convolutional code. Some simulation results with this system are shown for the AWGN and the fully interleaved Rayleigh fading channel.

II. NOTATION

Let U be a discrete random variable with the alphabet $\mathcal{U} = \{0, ..., M-1\}$ and a probability mass function $p(u) = P(U = u)$. Consider an i.i.d. source that draws independently successive values of U. Using a variable length code $\mathcal{C}$ each symbol u is mapped to a codeword $\mathbf{c}(u)$ with length $l(\mathbf{c})$. We use the notations $l_{min} = \min_{\mathbf{c}\in\mathcal{C}} l(\mathbf{c})$ and $l_{max} = \max_{\mathbf{c}\in\mathcal{C}} l(\mathbf{c})$ for the minimal respectively maximal length of a codeword in $\mathcal{C}$.

If we assume a packet of K source symbols $\mathbf{u} = (u_1, u_2, ..., u_K)$, the output of the VLC encoder is a sequence of variable length codewords $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_K)$ respectively a bit sequence $\mathbf{b} = (b_1, b_2, ..., b_N)$. A specific bit position in a particular codeword of the sequence $\mathbf{C}$ is addressed by $c_{k,j}$ which means the j-th bit in the k-th codeword of the sequence $\mathbf{C}$ and $j \in \{1, ..., l(\mathbf{c}_k)\}$. The bit sequence $\mathbf{b}$ is transmitted over a memoryless channel with transition probabilities $p(y|x)$ and we observe at the receiver a noisy sequence $\mathbf{y} = (y_1, y_2, ..., y_N)$. The input alphabet of the channel is assumed to be binary. If the variable length coded bit sequence is directly transmitted via the channel the channel input values x are identical to the components of the bit sequence $\mathbf{b}$.

In the following we denote a subsequence of $\mathbf{y}$ starting at position a and ending at position b by $\mathbf{y}_a^b = (y_a, y_{a+1}, ..., y_b)$.

III. Trellis Representation

Before we describe the decoding algorithm we introduce our trellis representation of a variable length coded sequence by a simple example:

Consider a VLC with alphabet size M=3 and codewords $\mathbf{c}(0)$=1, $\mathbf{c}(1)$=01, $\mathbf{c}(2)$=00, and a sequence of K = 4 source symbols $\mathbf{u} = (0, 2, 0, 1)$.
This sequence is mapped to a sequence of variable length codewords $\mathbf{c} = (1, 00, 1, 01)$. The length of this sequence in bits is N = 6. All sequences with K = 4 and N = 6 can be represented in a trellis diagram as shown in Fig. 1. In this diagram k denotes the symbol time and n identifies a particular state at symbol time k. The value of n is equivalent to the number of bits of a subsequence ending in this state. The set of possible states n at symbol time k is denoted by $\mathcal{N}_k$. The set $\mathcal{N}_2$ is depicted in Fig. 1. Each node

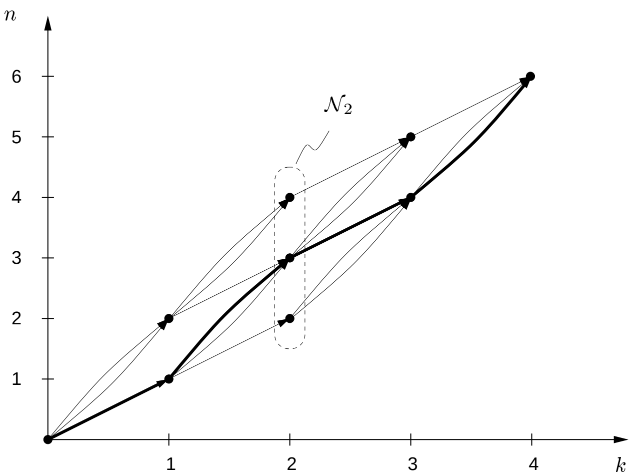

Fig. 1: Original trellis

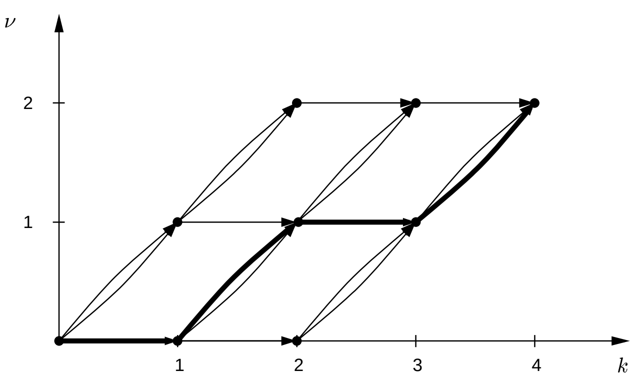

Fig. 2: Transformed trellis

$S_{n,k}$ in the trellis diagram represents a terminal node of all possible sequences consisting of k symbols and n bits. All possible sequences with K = 4 and N = 6 are paths through the trellis and terminate in $S_{6,4}$. At each node of the trellis synchronization between received bits and received symbols is guaranteed. With the transformation

$$\nu = n - kl_{min} \tag{1}$$

we obtain the trellis in Fig. 2. The maximal number of states at any symbol time k is

$$\nu_{max} = N - K \cdot l_{min} + 1 \tag{2}$$

The given sequence of messages is marked in the two trellises.

In general the so constructed trellis can be split up in three sections along the k axis. We denote these sections as the diverging, the stationary and the converging section. In the diverging section the number of states per symbol period k increases when k increases. In the stationary section the number of states is constant in successive symbol intervals and takes on the maximal number of states from equation (2). In this section successive trellis segments are time invariant. In the converging section the number of states per symbol period decreases again until there remains only one possible state at $k = K$.

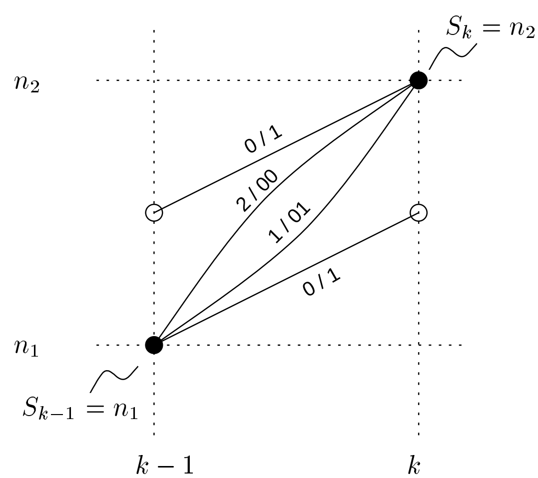

Fig. 3: Transitions from trellis in Fig. 1

Fig. 3 shows a more detailed representation of transitions in the trellis of Fig. 1 The branches are labeled with two values. The left one is the source symbol u_k that initiates the transition from state S_{k-1} to state S_k. The right value is the codeword of the variable length code that corresponds to the particular symbol u_k. We denote the transition probabilities by

$$t_k(n_2|n_1) = P(S_k = n_2|S_{k-1} = n_1). \tag{3}$$

Note that the parallel transitions occur when the variable length code contains codewords of the same codeword length. We obtain the state transition probabilities by taking the sum of the probabilities of all symbols that correspond to codewords of the same length. Further we denote the probability of a source symbol given a particular transition by

$$q_k(u_k|n_1,n_2) = P(U_k = u|S_{k-1} = n_1, S_k = n_2). \quad (4)$$

The values n_1 and n_2 in the above equations specify states in two successive symbol time instants. Using the original trellis in Fig. 1 these values also represent the bit length of a particular partial sequence up to symbol time $k-1$ and k respectively.

IV. Decoding Algorithm

Based on the above trellis representation of variable length coded sequences various decoding strategies can be applied. Using the Viterbi algorithm either maximum likelihood (ML) or maximum a posteriori (MAP) sequence estimation can be performed. But this trellis also allows the application of the BCJR algorithm to perform symbol by symbol MAP decoding. The objective of the decoder then is to determine the a posteriori probabilities (APP) of the transmitted symbols $u_k,\ (1 \le k \le K)$ from the observation sequence $\mathbf{y}$ and select the symbol with the largest APP. This can be written in the well known MAP decoding rule as

$$P(\hat{u}_k|\mathbf{y}) = \max_{u_k} P(u_k|\mathbf{y}). \quad (5)$$

If we apply this algorithm the decoder provides us not only with a sequence of symbol estimates but also delivers a per symbol reliability information. Although the transformed trellis of Fig. 2 is more convenient to represent the variable length coded sequence we describe the decoding algorithm using the original trellis in Fig. 1 with state index n instead of the transformed state index ν. With this notation it is easier to handle variable length sequences in a formal way. In the following we do not want to derive the BCJR algorithm completely but point out some aspects that are relevant for the application of the algorithm on the above trellis structure. The basic operations of the decoder to provide us with the a posteriori probabilities are a forward recursion to determine the values

$$\alpha_k(n) = P(S_k = n, \mathbf{y}_1^n) \quad (6)$$

and the backward recursion to obtain the values

$$\beta_k(n) = P(\mathbf{y}_{n+1}^N|S_k = n). \quad (7)$$

Both quantities have to be calculated for all symbol times k and all possible states $n \in \mathcal{N}_k$ at symbol time k. For the recursions we also need the probability function

$$\gamma_i(\mathbf{y}_{n'+1}^n, n', n) = \\ q_k(u_k|n', n) \cdot p(\mathbf{y}_{n'+1}^n|u_k = i) \cdot t_k(n|n') \quad (8)$$

which includes the symbol and transition probabilities from the equations (3), (4) and the channel characteristics in terms of the channel transition probability $p(\mathbf{y}_{n'+1}^n|u_k = i)$. For a memoryless channel we can express the transition probability in equation (8) by the product of the bitwise transition probabilities:

$$p(\mathbf{y}_{n'+1}^n|u_k = i) = \prod_{j=1}^{l(\mathbf{c}_i)} p(y_{n'+j}|c_{k,j}). \quad (9)$$

Now we have introduced all quantities we need to compute the a posteriori probabilities of the information symbols u_k. We performed the computations analogous to [12] to obtain

$$P(u_k = m|\mathbf{y}) = \quad (10)$$
$$\frac{\sum_{n\in\mathcal{N}_k}\sum_{n'\in\mathcal{N}_{k-1}} \gamma_m(\mathbf{y}_{n'+1}^n, n', n)\cdot\tilde{\alpha}_{k-1}(n')\cdot\tilde{\beta}_k(n)}{\sum_{n\in\mathcal{N}_k}\sum_{n'\in\mathcal{N}_{k-1}}\sum_{i=0}^{M-1} \gamma_i(\mathbf{y}_{n'+1}^n, n', n)\cdot\tilde{\alpha}_{k-1}(n')\cdot\tilde{\beta}_k(n)}.$$

The forward recursion can be written as:

$$\tilde{\alpha}_k(n) = \quad (11)$$
$$\frac{\sum_{n'\in\mathcal{N}_{k-1}}\sum_{i=0}^{M-1} \gamma_i(\mathbf{y}_{n'+1}^n, n', n)\cdot\tilde{\alpha}_{k-1}(n')}{\sum_{n\in\mathcal{N}_k}\sum_{n'\in\mathcal{N}_{k-1}}\sum_{i=0}^{M-1} \gamma_i(\mathbf{y}_{n'+1}^n, n', n)\cdot\tilde{\alpha}_{k-1}(n')},$$

with $\alpha_0(0)$= 1 and $k = 1, ..., K$, and the backward recursion as:

$$\tilde{\beta}_k(n) = \quad (12)$$
$$\frac{\sum_{n'\in\mathcal{N}_{k+1}}\sum_{i=0}^{M-1} \gamma_i(\mathbf{y}_{n+1}^{n'}, n, n')\cdot\tilde{\beta}_{k+1}(n')}{\sum_{n\in\mathcal{N}_k}\sum_{n'\in\mathcal{N}_{k+1}}\sum_{i=0}^{M-1} \gamma_i(\mathbf{y}_{n+1}^{n'}, n, n')\cdot\tilde{\alpha}_k(n)},$$

with $\tilde{\beta}_K(N)$ = 1 and $k = K-1, ..., 1$.

Because of the normalization in equations (11) and (12) we use $\tilde{\alpha_k}(n)$ $\tilde{\beta}_k(n)$ instead of $\alpha_k(n)$ and $\beta_k(n)$. This does not affect the result of the APP computation because it cancels out in equation (10). A further simplification can be obtained by using the logarithms of the above quantities instead of the quantities themselves.

In the following sections we denote a decoder that is based on the above algorithm a VLC-MAP decoder.

V. Proposed System

In the preceding section we described the symbol by symbol MAP decoding algorithm without channel coding. In a practical transmission system with non negligible bit error probability one would always use a concatenated scheme with an inner error correcting code that provides the outer VLC with a sufficiently small residual bit error rate.

In the following we describe a transmission system consisting of an outer VLC and an inner convolutional code.

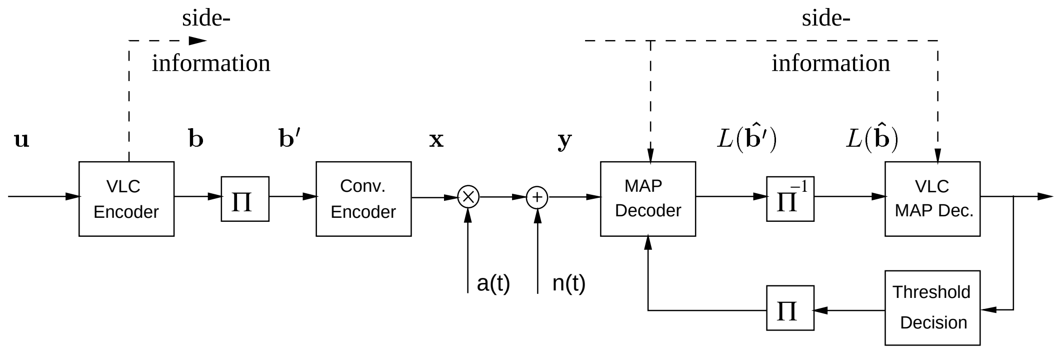

Fig. 4: Proposed transmission scheme

The structure of the system is sketched in Fig. 4. The outer variable length code could either be a Huffman code that is optimal in terms of average codeword length for given symbol probabilities $p(u)$ like code $\mathcal{C}_A$ in Table I. It also could be a variable length code that contains some explicit redundancy like the codes $\mathcal{C}_B$ and $\mathcal{C}_C$. Note that $\mathcal{C}_C$ is just the bit by bit inverse of $\mathcal{C}_B$. Variable length codes with explicit redundancy that provide error correcting capabilities have been studied by Bernard and Sharma in [6], [7] and by Buttigieg and Farrell in [8], [9]. Also [10] proposes codes with these properties.

In our system we also apply a new approach in variable length coding. We constructed a simple time variant variable length code which we denote as TVVLC. This TVVLC is generated by selecting codewords from the codes $\mathcal{C}_B$ and $\mathcal{C}_C$ in an alternating way from symbol period to symbol period. It turned out that this TVVLC combined with VLC-MAP decoding results in a superior performance compared with a standard time invariant variable length codes.

For transmission a sequence of K source symbols $\mathbf{u}$ is encoded by the variable length encoder. The output of the VLC encoder is a sequence $\mathbf{b}$ that consists of N bits. If we concatenate P of those packets to obtain a larger interleaver depth the resulting bit sequence $\mathbf{b}$ then consists of $N' = \sum_{i=1}^{P} N_i$ bits. $\mathbf{b}$ is permuted by an interleaver Π and passed to a convolutional encoder with code rate R. The applied block interleaver is depicted in Fig. 5. At the output of the convolutional encoder we obtain a bit sequence $\mathbf{x}$ consisting of $(N' + \mu)/R$ bits where μ is the memory of the convolutional encoder. This sequence is transmitted across the channel.

TABLE I
Huffman codes used in the simulations

		$\mathcal{C}_A$	$\mathcal{C}_B$	$\mathcal{C}_C$
u	$P(U=u)$	$\mathbf{c}(u)^{(A)}$	$\mathbf{c}(u)^{(B)}$	$\mathbf{c}(u)^{(C)}$
0	0.5	0	0	1
1	0.25	10	11	00
2	0.125	110	101	010
3	0.125	111	1001	0110

As we need the bit-length N of each variable length encoded packet for the VLC-MAP decoder this information has to be transmitted as a side information and must be protected by a powerful code to ensure that the decoder gets this information error free. We need $\lambda = \lceil \mathrm{ld}[\mathrm{K}(\mathrm{l}_{\max} - \mathrm{l}_{\min})] \rceil$ bits to represent this side information where $\lceil a \rceil$ is defined as the smallest integer value that is larger or equal to a.

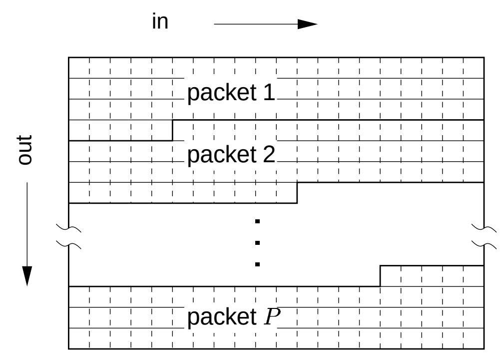

Fig. 5: Interleaving of a multi packet frame

In order to make a fair comparison between the system with the VLC-MAP decoder and a system with a conventional VLC decoder we fix the overall channel code rate. This results in a better protection level for the VLC sequence of the conventional system because in the scheme with VLC-MAP decoding additional redundancy has to be spent for the side information and the appropriate protection of it. The conventional system does not need any side information and therefore all redundancy can be spent for the protection of the variable length coded sequence itself. In [4] the rate loss is neglected and the length information is assumed to be adequately protected. Nevertheless we perform our simulations with this overhead and accept the performance loss due to the increased code rate. Therefore the presented gain of the proposed system compared with the conventional Huffman decoding is a kind of worst case situation. By using some more sophisticated schemes in transmitting the length information the results should be even better.

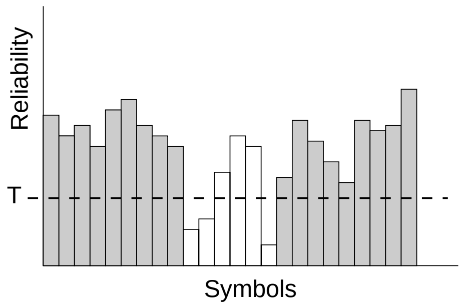

Fig. 6: Calculation of bit reliability values

At the receiver side the inner convolutional code is decoded using a symbol by symbol MAP decoder that provides the outer VLC-MAP decoder with soft-values for the estimate of the variable length encoded sequence $\hat{\mathbf{b}}$. The output of the VLC-MAP decoder is a hard decision of the source symbols on the one hand and a per symbol reliability value on the other hand. We use this reliability values to improve the performance of the inner convolutional code in a second iteration step. For this issue we generate a very rough estimate for the bit reliability of the sequence $\hat{\mathbf{b}}$ in the following way. For a given threshold T we consider all symbols above this threshold as reliably decoded whereas all symbols below are thought of being not reliable. This procedure is illustrated in Fig. 6. Starting at the beginning of the packet we can assign all bits corresponding to the symbols in the shaded area a reliability value of ∞. The same is done for the consecutive symbols at the end of the packet. All the remaining bits are assigned with an reliability value of zero. These reliability values are fed back to the inner MAP decoder as a priori information for the next iteration step.

VI. Simulation results

For the simulations we built packets of K=100 source symbols which were coded by the variable length code. We always used a concatenated system with outer VLC and inner convolutional code. The reference system was the Huffman code $\mathcal{C}_A$ followed by a convolutional code with rate R_1=4/9 which was generated by puncturing a memory μ=4 and rate R=1/4 mother code. We used a recursive systematic convolutional code and puncturing patterns from [11]. The overall average code rate of the reference system therefore is

$$R_{ref} = \frac{K \cdot H(U)}{(K \cdot l_{av}^{(A)} + \mu)/R_1}, \qquad (13)$$

where $H(U)$ is the entropy of U. With K=100 and code $\mathcal{C}_A$ we obtain R_{ref} = 0.4345.

For the new scheme we also built packets of K = 100 symbols. For variable length encoding we used $\mathcal{C}_B$ and $\mathcal{C}_C$ which both have an average codeword length $l_{av}^{(B)}$ = 1.875.

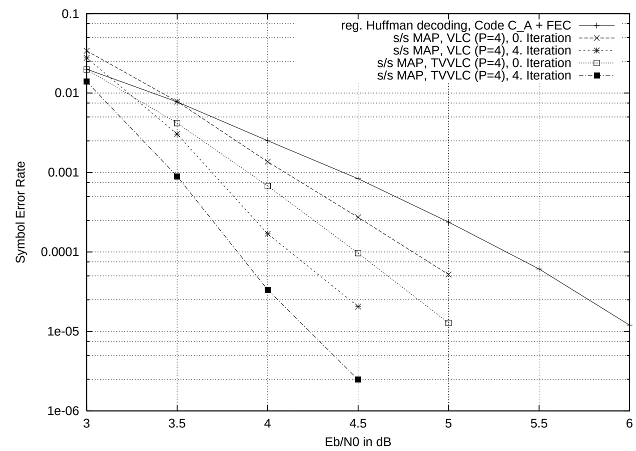

Fig. 7: Simulation results for AWGN channel

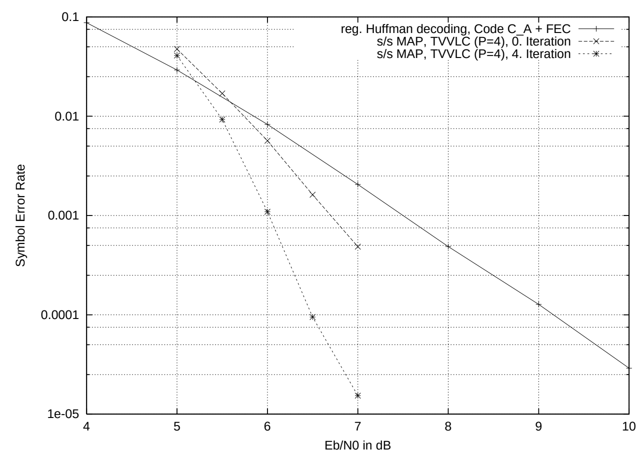

Fig. 8: Simulation results for Rayleigh channel

To use the algorithm introduced in section IV we need the number of bits N contained in a variable length coded packet. This number has to be transmitted as side information and has to be protected with a stronger code. In our simulations we used a rate 1/4 code to protect this side information. The number of bits required to represent the side information is denoted be λ as introduced above. We increased the interleaver depth by linking P=4 packets together. As interleaver we used the block interleaver from Fig. 5. For the new system we obtain an average code rate of

$$R_{\mathrm{VLC}} = \frac{P \cdot K \cdot H(U)}{(P \cdot \lambda + \mu) \cdot 4 + (P \cdot K \cdot l_{av}^{(B)} + \mu)/R_2}. \qquad (14)$$

We have chosen R_2 to be 8/15 so we obtain an average code rate for the new scheme of R_{vlc} = 0.4448.

The threshold T for the logarithmic (natural logarithm) a posteriori probabilities was set to $T = -5.0 \cdot 10^{-5}$ which corresponds to an APP of 0.99995. The interleaver size was 40 columns for the AWGN channel and 60 columns for the Rayleigh fading channel.

Fig. 7 shows simulation results for the AWGN channel. The solid curve shows the symbol error rate for the reference scheme with conventional decoding of the Huffman code $\mathcal{C}_A$. The other curves show the results for the iterative decoding approach with code $\mathcal{C}_B$ and with the TVVLC (code $\mathcal{C}_B$ and $\mathcal{C}_C$). For the iterative scheme the performance with no iteration and after the fourth iteration is shown. Using the TVVLC a gain of approximately 1.7 dB compared to the conventional approach is obtained at a symbol error rate of $1 \cdot 10^{-5}$.

In Fig. 8 the results for the fully interleaved Rayleigh fading channel and TVVLC are shown. In this simulation the frame also consists of four packets. For this channel we obtain a gain of approximately 3.1 dB at a symbol error rate of $2 \cdot 10^{-5}$.

We evaluated the symbol error probability by a simple symbol by symbol comparison of the decoded sequence with the original sequence. Doing this the self-synchronizing property of variable length codes is not considered. Although a deletion or insertion of a symbol in the decoded sequence is not possible in our approach long error bursts can be observed due to the rather bad distance profile of the variable length code. Nevertheless the proposed scheme results in significant gains in signal to noise ratio.

VII. Conclusions

In this paper we presented a new trellis representation for variable length coded sequences generated by i.i.d. sources. On this trellis either sequence estimation or symbol by symbol MAP decoding is possible. With a modified BCJR algorithm that can be applied to this trellis we can generate symbol by symbol reliability values of the decoded sequence. The soft output was used in an iterative system to improve the overall SER performance. By a more efficient coding of the additional length information even larger gains could be obtained. Several components and parameters of the system, i.e. the interleaver, the allocation of redundancy between the variable length code and the convolutional code or the calculation of bit reliability values from the symbol reliabilities at the output of the VLC-MAP decoder, are still not chosen in an optimal way. This is subject to further research on this topic. Note also that the VLC-MAP decoder may become very complex with increasing symbol alphabet and increasing sequence length. This problem could be overcome by suboptimal algorithms.

VIII. References

[1] L. R. Bahl, J. Cocke, F. Jelinek, J. Raviv, "Optimal decoding of linear codes for minimal symbol error rate," in IEEE Trans. on Inform. Theory, Vol. IT-20, pp. 284-287, March 1974

[2] N. Demir and K. Sayood, "Joint source/channel coding for variable length codes", in Proc. IEEE Data Compression Conference, Snowbird, Utha, March 1998, pp. 139-148

[3] M. Park and D. J. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximated MAP sequence estimation", in Proc. IEEE Int. Conf. on Acoustics Speech and Signal Processing, Phoenix, Arizona, March 1999

[4] M. Park and D. J. Miller, "Decoding entropy-coded symbols over noisy channels using discrete HMMs, in Proc. Conf. on Information Sciences and Systems (CISS), Princeton, USA, March 1998

[5] J. Wen, J. D. Villasenor, "Utilizing soft information in decoding variable length codes", in IEEE Data Compression Conference, Snowbird, Utha, March 1999

[6] M. A. Bernard and B. D. Sharma, "Some combinatorial results on variable length error correcting codes", in Ars Combinatoria, Vol. 25B, 1988, pp. 181-194

[7] M. A. Bernard and B. D. Sharma, "A lower bound on the average codeword length of variable length error correcting codes" in IEEE Trans. on Inform. Theory, Vol 36, No. 6, 1990, pp. 1474-1475

[8] V. Buttigieg and P.G. Farrell, "On variable-length error-correcting codes", in Proc. 1994 IEEE ISIT, Trondheim, Norway, p. 507, June 27.- July 1. 1994.

[9] V. Buttigieg and P.G. Farrell , "A maximum a-posteriori (MAP) decoding algorithm for variable-length error-correcting codes", in Codes and cyphers: Cryptography and coding IV, Essex, England, The Institute of Mathematics and its Applications, pp. 103-119, 1995.

[10] V. B. Balakirsky, "Joint source-channel coding with variable length codes", in Proc. 1997 IEEE ISIT, Ulm, Germany, p. 419, June 29 - July 4. 1997

[11] M. Bystrom, T. Stockhammer and O. Grimm, "Optimal combined source-channel rate allocation with applications to MPEG-4 compressed video", submitted to IEEE JSAC, Special Issue on Error-Resilient Image and Video Transmission

[12] P. Robertson, E. Villebrun and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain", in Proc. ICC '95, Seattle, USA, June 1995, pp. 1009-1013