MAP Channel Decoding by exploiting Multilevel Source A Priori Knowledge

Stefan Heinen, Axel Geiler, Peter Vary
Institute of Communication Systems and Data Processing
RWTH Aachen, Templergraben 55, D-52056 Aachen, Germany
Phone: +49-241-806964, Fax: +49-241-8888186,
Email: heinen@IND.RWTH-Aachen.DE

Abstract

In digital transmission systems source encoders are used to remove redundancy from the source's output data in order to save transmission bandwidth. As the compressed data is sensitive against transmission errors channel encoding is used to perform forward error correction.

For practical reasons, source coding schemes produce multilevel parameters with nonzero residual redundancy which manifests itself in a nonuniform p.d.f. of a parameter and/or in the correlation of subsequent parameter values.

The objective of this paper is to derive methods to exploit *multilevel* a priori knowledge within the detection process of channel decoding. Joint a priori probabilities of bit groups are taken into consideration within the path metrics of the Viterbi algorithm.

1 Introduction

In data transmission systems source and channel coding are usually treated separately. This may be justified by Shannon's information theory which states that within the bounds of channel capacity a coding scheme working at arbitrary low bit error rates can be found by separately optimizing source and channel coding. This would imply to realize a perfectly working source encoder which outputs independent uniformly distributed bits or parameters. However, with practical source encoders, in most cases residual redundancy will be left. Shannon already mentioned [1]: "However, any redundancy in the source will usually help if it is utilized at the receiving point. In particular, if the source already has redundancy and no attempt is made to eliminate it in matching to the channel, this redundancy will help combat noise." In [2] Hagenauer uses this idea for "source-controlled channel decoding". This approach exploits the correlation of single bits of subsequent source encoded frames; the intraframe correlations between bits remain unused.

In this contribution we want to generalize this concept to parameters consisting of groups of *statistically dependent* bits. It is obvious that we will gain more if we exploit multilevel parameter statistics instead of bit statistics. Since the modifications will only affect the channel decoder the proposed approach is an attractive means to improve the transmission quality of a practical communication system where the decoding algorithm in the receiver is not part of a standard (e.g. GSM).

2 MAP Decoding Exploiting A Priori Knowledge on Single Bit Level

Fig. 1 shows a standard transmission system with an AWGN channel model. We assume a source encoder producing frames of N bits (for notational simplicity a sequence extending from p to q is denoted by $\mathbf{x}_p^q$)

$$\mathbf{x}_1^N := \{x_1, .., x_N\}$$

that are channel encoded resulting in M bits

$$\mathbf{y}_1^M := \{y_1, .., y_M\}$$

with $y_j \in \{0, 1\}$ which are sequentially transmitted via the channel. At the receiver the noisy sequence

$$\mathbf{z}_1^M := \{z_1, .., z_M\}$$

with $z_j \in \mathbf{R}$ is discretized with sufficient amplitude resolution to support soft decision and then forwarded to the channel decoder.

In the sequel we want to restrict our investigations to convolutional channel codes. Furthermor to simplify the formalism only encoders of rate $1/r$ are considered.

The Viterbi algorithm has been shown to be a powerful means for the decoding of convolutional codes. The classical detection approach is the "Maximum Likelihood Sequence Estimation" (MLSE) which is based on the criterion

$$C_{\text{MLSE}}: \quad p\left(\mathbf{z}_1^M | \hat{\mathbf{x}}_1^N\right) = \max_{\hat{\mathbf{x}}_1^N}!, \qquad (1)$$

where

$$\hat{\mathbf{x}}_1^N := \{\hat{x}_1, .., \hat{x}_N\}$$

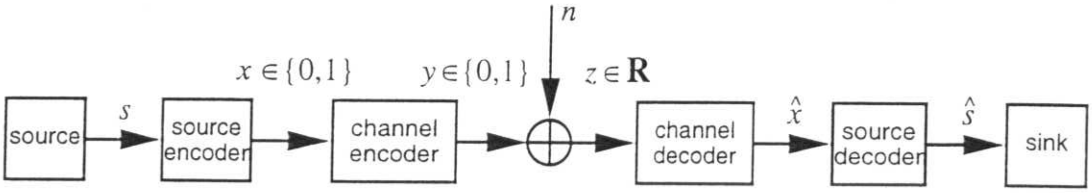

Fig. 1: Standard transmission system

is a tentative bit sequence. The conditional probability density function $p(\cdot|\cdot)$ describes the channel statistics. If the channel is assumed to be disturbed by additive noise n, then $p(\cdot|\cdot)$ can be identified as the noise's density function $p_n(\xi)$. In the case of white Gaussian noise, this formula leads to the well known "minimum Euclidian distance criterion". For the ML approach no knowledge about the distribution of the information bits is needed.

In contrast, the "Maximum A Posteriori Sequence Estimation" (MAPSE) requires a priori knowledge about the sent information sequence. It is based on the criterion

$$C_{\text{MAPSE}} : \Pr\left(\hat{\mathbf{x}}_1^N = \mathbf{x}_1^N | \mathbf{z}_1^M\right) = \max_{\{\hat{\mathbf{x}}_1^N\}}! \qquad (2)$$

which means that we have to find the most probably sent sequence $\{x_1, .., x_N\}$ given the observed sequence $\{z_1, .., z_M\}$. It is well known that the MLSE and the MAPSE approach are formally equivalent if the information bits are identically distributed.

In a first approach, we want to characterize residual redundancy in the source encoded bits by stating that the a priori probabilities $\Pr(x_k = 0)$ and $\Pr(x_k = 1)$ of at least some bits x_k of the information sequence $\{x_1, .., x_N\}$ are unequal

$$\Pr(x_k = 0) \neq \Pr(x_k = 1). \qquad (3)$$

If we assume statistical independence of the information bits x_k and white channel noise, it will be possible to formulate a metric that can efficiently be maximized using the Viterbi algorithm. With r the number of channel encoded bits produced per information bit we can write the MAPSE metric (4). The term enclosed by brackets describes the branch metric for the transition from one decoding step to the next corresponding to one information bit. Therefore the single bit a priori probability $\Pr(x_k = \hat{x}_k)$ can directly be combined with the branch metric. The single bit MAP metric has only a slightly increased complexity compared to the Maximum Likelihood (ML) metric as just one additional multiplication (or addition in the logarithmic domain) is necessary per decoder branch.

Equation (4) corresponds to the "A Priori Viterbi Algorithm" (APRI-VA) approach in [2].

A major drawback is that multilevel a priori knowledge, i.e. the non-uniform distribution density on parameter level, is not exploited.

3 MAP Decoding Exploiting A Priori Knowledge on Parameter Level

We now want to show that a similar and only slightly more complex formula can be derived if we no longer assume statistical independence of the source encoded bits within a frame of N bits $\{x_1, .., x_N\}$. Instead, we presume statistical independence amongst groups of bits within the frame.

Thus the frame of N bits can be subdivided into Q parameters X_l, $l = 1, .., Q$, each consisting of $D = N/Q$ bits according to

$$X_l = \sum_{i=1}^{D} b_l(i) \cdot 2^{i-1}, \quad b_l(i) := x_{(l-1)D+i}. \qquad (5)$$

For the time being statistical dependence of subsequent values X_l is not considered. However, this can as well be taken into consideration by modelling the source encoder as a Markov source and estimating the a priori parameter knowledge with respect to previously decoded parameters [3].

With $Q = N/D$ and $\hat{b}_l(i) := \hat{x}_{(l-1)D+i}$ we obtain

$$\Pr\left(\mathbf{x}_1^N = \hat{\mathbf{x}}_1^N\right) = \prod_{l=1}^{Q} \Pr\left(X_l = \hat{X}_l\right) \qquad (6)$$

$$= \prod_{l=1}^{Q} \Pr\left(\{b_l(i) = \hat{b}_l(i)\}_{i=1}^{D}\right). \qquad (7)$$

Now the residual redundancy can be described by the non-identical probability of the source coded parameters. This means that for at least some pairs of parameter values $m, n \in \{0, .., 2^D - 1\}$ with $m \neq n$ the inequality

$$\Pr(X_l = m) \neq \Pr(X_l = n) \qquad (8)$$

holds. In this case the MAPSE metric results in (9).

In general, we cannot expect that the dependent bits are fed into the channel coder subsequently.

In some cases is advantageous to reorder the source encoded bits such that the most sensitive bits are positioned at the beginning and at the end of the transmitted block of length N where the error protection is better than in the middle.

Due to this reordering the dependent bits of a parameter will be spread over the transmitted block (see Fig. 2). For reasons of notational simplicity we assume that the parameter bits still are in the same order as indicated by 5, i.e. $b_l(i)$ preceeds $b_l(j)$ if $i < j$.

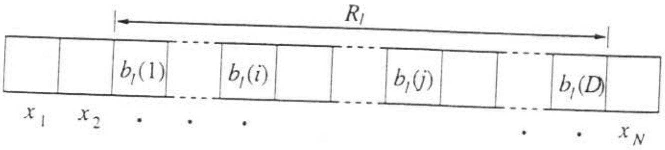

Fig. 2: Spreading of dependent bits

As we will see, this reordering will cause some important implications on our algorithm. In this context we have to classify the problem into two cases:

A. The range R_l over which the dependent information bits of parameter l are spread is greater than the constraint length $L + 1$ (L = number of shift registers) of the convolutional encoder.

B. The spreading range R_l is less or equal to $L + 1$.

Since case A is more general we will consider it first.

MAP metric on bit level:

$$M_{MAPSE,bit} = \prod_{k=1}^{N} \left[\prod_{\nu=1}^{r} p\left(z_{(k-1)r+\nu} | \{\hat{\mathbf{x}}_1^k\}\right) \right] \Pr(x_k = \hat{x}_k) \qquad (4)$$

MAP metric on parameter level:

$$M_{MAPSE,param} = \left(\prod_{k=1}^{N} \left[\prod_{\nu=1}^{r} p\left(z_{(k-1)r+\nu} | \{\hat{\mathbf{x}}_1^k\} \right) \right] \right) \cdot \left(\prod_{l=1}^{Q} \Pr\left(X_l = \hat{X}_l \right) \right) \qquad (9)$$

A. MAP decoding with $R_l > L+1$

The problem in formulating an appropriate metric for the Viterbi algorithm is how to split up the parameter probabilities $\Pr(X_l = m)$ such that a decision for the maximum metric can be made for every single decoding step k.

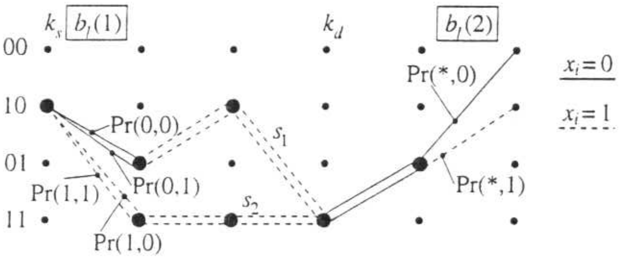

Fig. 3: Exact MAP decoding

To explain the difficulty consider Fig. 3 (solid lines represent transitions for information bits $b_l(i) = 0$ and dashed lines transitions for information bits $b_l(i) = 1$). In this example the parameter X_l consists of two bits $b_l(1)$ and $b_l(2)$ that are spread by $R_l = 5$. The constraint length of the encoder is $L + 1 = 3$. The depicted path splits up at decoding step k_s into an upper path s_1 and a lower path s_2 which merge at step k_d.

A MAP decision at step k_d requires to multiply the upper and lower path metrics by their particular parameter a priori probability $\Pr(b_l(1), b_l(2))$. Unfortunately at step k_s we do not know how the second bit $b_l(2)$ of the parameter will be decoded. Thus the final decision must be delayed by continuing the decoding process with two parallel paths whose metrics are multiplied by their respective a priori probabilities $\Pr(b_l(1), b_l(2))$. The first bit $b_l(1)$ is already determined by the transition from k_s to $k_s + 1$ for each path. At decoding step k_d the parallel branches of s_1 and s_2 merge. Therefore we have to handle two pairs of competing paths which requires two decisions, in particular

$$M_1 \cdot \Pr(0,0) \underset{b_l(1)=0}{\overset{b_l(1)=1}{\lessgtr}} M_2 \cdot \Pr(1,0), \text{ assumed } b_l(2) = 0$$

and

$$M_1 \cdot \Pr(0,1) \underset{b_l(1)=0}{\overset{b_l(1)=1}{\lessgtr}} M_2 \cdot \Pr(1,1), \text{ assumed } b_l(2) = 1.$$

M_1 and M_2 represent the metrics of the paths s_1 and s_2 without the a priori probabilities of parameter X_l. The relation $\underset{b_l(1)=0}{\overset{b_l(1)=1}{\lessgtr}}$ has to be understood such that if the lefthand side is less than the righthand side, decide $b_l(1) = 1$ and in case it is greater decide $b_l(1) = 0$.

Depending on the decisions on $b_l(1)$ made at step k_d the asterisk in the a priori probabilities $\Pr(*, 0)$ and $\Pr(*, 1)$ shown in Fig. 3 represents either 0 or 1. Note that it is possible that the parallel paths after step k_d decode $b_l(1)$ differently.

When extending the example to arbitrary parameter sizes $(D > 2)$ the decoding of each first bit $b_l(1)$ of a parameter X_l will increase the number of parallel paths by a factor of 2^{D-1}. After decoding any further bit $b_l(i)$, $i > 1$ the number of parallel paths shrinks by 2. Thus we can state that after decoding the i-th bit $b_l(i)$ each parameter X_l contributes with a factor of

$$F_i = 2^{D-i} \qquad (10)$$

to the number of parallel paths until $i = D$. It becomes obvious that the complexity of this decoding algorithm increases exponentially with the parameter length and linearly with the spread range R_l. Therefore it is not of practical interest and we should aim at an approximation with a complexity comparable to the standard Viterbi algorithm.

Approximate MAP decoding

In our approach we approximate the 2^{D-i} parallel paths by only one path whose metric is multiplied by an estimated a priori probability $\overline{\Pr}(b_l(1), .., b_l(i))$ that depends on the previously decoded parameter bits $b_l(1), .., b_l(i))$ but is independent of the future bits $b_l(i+1), .., b_l(D)$.

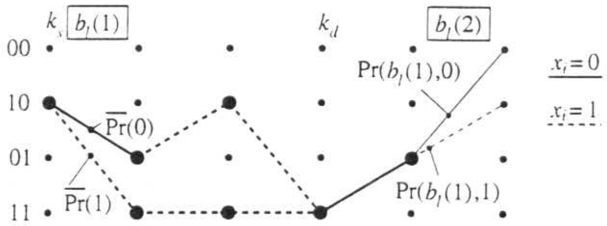

Fig. 4: Approximate MAP decoding

Of course, we introduce an inaccuracy into the path metric computation which will degrade the gain obtainable with optimal MAP decoding, but on the other hand we will decrease the complexity dramatically.

During the decoding process we take into consideration the already decoded bits of a parameter, thus improving the estimate until the whole parameter is decoded. This is achieved by multiplying the metric of the representative path by an enhancement factor

$$c_i^{(\mu)}, \quad i \in \{1, .., D\}, \quad \mu \in \{0, 1\}$$

whenever an information bit $b_l(i)$ of parameter X_l is decoded. The procedure described below comprises the necessary operations when decoding the i-th bit of a parameter X_l:

1. Compose two sets of a priori probabilities

   $$S_i^{(0)} = \{\Pr(X_l) | b_l(1), .., b_l(i-1) \text{ fixed}, \ b_l(i) = 0\}$$

   and

   $$S_i^{(1)} = \{\Pr(X_l) | b_l(1), .., b_l(i-1) \text{ fixed}, \ b_l(i) = 1\}$$

where the values of the already decoded bits $b_l(1), .., b_l(i-1)$ are fixed and shared by all probabilities in both subsets, while the values of $b_l(i+1), .., b_l(D)$ take on all possible values.

2. Compute two probability estimates
$$\overline{\Pr}(b_l(1), .., b_l(i-1), b_l(i) = 0)$$
$$\overline{\Pr}(b_l(1), .., b_l(i-1), b_l(i) = 1)$$
for each subset.

3. Compute the enhancemend factors $c_i^{(0)}$ and $c_i^{(1)}$ according to
$$c_i^{(\mu)} = \frac{\overline{\Pr}(b_l(1), .., b_l(i) = \mu)}{\overline{\Pr}(b_l(1), .., b_l(i-1))}, \qquad (11)$$
with $i \in \{1, .., D\}, \mu \in \{0, 1\}$ and multiply the metrics of the respective representative paths for transitions $b_l(i) = 0$ and $b_l(i) = 1$ by it.

This may be clearified by the simple example depicted in Fig. 4, which again shows the decoding of a two bit parameter. We start with estimates for the subsets
$$S_1^{(0)} = \{\Pr(0,0), \Pr(0,1)\}$$
$$S_1^{(1)} = \{\Pr(1,0), \Pr(1,1)\}$$
yielding $\overline{\Pr}(0)$ and $\overline{\Pr}(1)$, respectively, as described below. In contrast to the exact MAP solution of Fig. 3, at step k_d we now have to make only *one* decision on the representative paths.

When decoding $b_l(2)$ we can improve the probability estimate by splitting up further the surviving path's probability subset into the subsets
$$S_2^{(0)} = \{\Pr(b_l(1), 0)\} \text{ and } S_2^{(1)} = \{\Pr(b_l(1), 1)\}.$$
Here the estimates coincide with the single set elements. The value of $b_l(1)$ is fixed and the same for both subsets. It depends on which representative path survived after the decision at step k_d.

In order to find a good estimation rule we need a measurement of the inaccuracy introduced by our approximation. A suitable measure is the total squared error caused by replacing all probabilities of one subset by only one estimated probability. For simplicity, let
$$\overline{\Pr}_i^{(\mu)} := \overline{\Pr}\left(b_l(1), .., b_l(i-1), b_l(i) = \mu\right)$$
be the estimated probability for the i-th bit. Then the total squared error entailed by replacing the parallel paths of the exact solution according to Fig. 3 by one representative path is
$$\left(\Delta_i^{(\mu)}\right)^2 = \sum_{P \in \mathbf{S}_i^{(\mu)}} \left(P - \overline{\Pr}_i^{(\mu)}\right)^2. \qquad (12)$$
It can be shown that by minimizing (12) the value of the enhancement factor (11) results in the conditional probability
$$c_i^{(\mu)} = \frac{\Pr\left(b_l(1), b_l(2), .., b_l(i) = \mu\right)}{\Pr\left(b_l(1), .., b_l(i-1)\right)}$$
$$= \Pr\left(b_l(i) = \mu \middle| b_l(1), .., b_l(i-1)\right). \quad (13)$$
Assuming that the information bit x_k is the i-th bit $b_l(i)$ of parameter X_l, the MAP metric can now be expressed as (14). This equation shows, that we have to keep in mind how the parameter bits $b_l(1), .., b_l(i-1)$ have been decoded for the current path. Actually the Viterbi algorithm's path memory contains this information. A non-optimized implementation could therefore perform a trace back for every parameter bit to find out how the previous bits have been decoded.

B. MAP decoding with $R_l \leq L+1$

We will now show that the decoding algorithm can even be simplified if the parameter spread range R_l is less or equal to the constraint length of the code.

Due to the design of a convolutional code the encoder's state directly corresponds to the last L encoded information bits. Therefore each state in the decoder trellis definitely determines the last L bits that would be decoded if the traceback started from this state.

To understand the implications for the MAP decoding algorithm consider the example in Fig. 5.

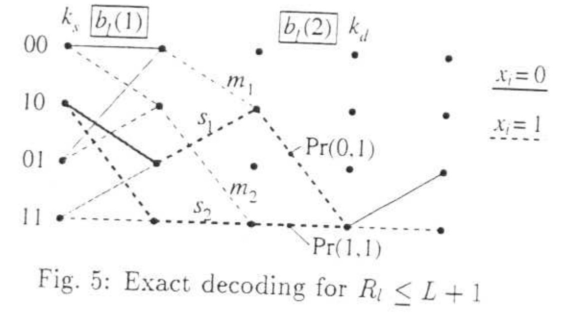

Fig. 5: Exact decoding for $R_l \leq L+1$

The second bit $b_l(2)$ of the parameter X_l is now placed at a distance of $L+1$ from $b_l(1)$. Again, to perform exact MAP decisions for the two bold paths s_1 and s_2 we have to multiply the metrics with the parameter a priori probabilities $\Pr(b_l(1), b_l(2))$. Actually we should do that for the transition from step k_s to k_s+1 since all subsequent path decisions will depend on the metric factor applied. But this can be postponed as shown in the example below.

At step $k_d - 1$ the path m_1 merges with the survivor path s_1. It can be observed that the paths m_1 and s_1 both decode $b_l(1) = 0$ (solid line). Consequently the metrics of m_1 and s_1 have to be multiplied by the same a priori probabilities. The same is valid for paths m_2 and s_2. Finally, as for the path decisions only the quotient of the path metrics is of interest, we can delay the multiplication of the a priori probability until the transition from step $k_d - 1$ to k_d. This is advantageous because with this transition we also decide on the second parameter bit $b_l(2)$ and therefore we know which parameter probability has to be applied on the respective path metric.

Our considerations lead to the following decoding algorithm in the case $R_l \leq L+1$:

1.) Perform the Viterbi algorithm with the standard ML metric.

2.) Whenever the last bit of a parameter is decoded, extract the previously decoded bits of the parameter from the current state and use them together with the currently decoded bit to select the respective a priori probability the path metric has to be multiplied with.

This shows that the proposed MAP decoding algorithm will only slightly increase the complexity compared to a Viterbi algorithm using the standard ML metric.

Comparing the complexity of our approach with that of a recent publication [4], we find that our algorithm in general is less complex. To make this clear, let us consider the following example.

In [4], a rate 3/4 convolutional code taken from [5] is applied to encode 3-bit speech encoder parameters. The

Metric on parameter level for approximate MAP decoding:

$$M_{MAPSE,param} = \prod_{k=1}^{N} \left[\prod_{\nu=1}^{r} p\left(z_{(k-1)r+\nu} | \{\hat{\mathbf{x}}_1^k\} \right) \right] \cdot \Pr\left(b_l(i) = \hat{x}_k \,\middle|\, b_l(1), .., b_l(i-1) \right) \tag{14}$$

trellis of this code has eight transitions emerging from each state. Since such a transition decodes the three bits of a parameter its path metric can be directly combined with the particular a priori probability. The code has a memory of $L = 5$ and a free distance of $d_{free} = 5$.

We choose a significantly less complex punctured rate 3/4 code derived from a rate 1/2 code. From [6] we can adopt a code with comparable (in fact slightly higher) performance ($L = 6$, $d_{free} = 5$). Even though the trellis of this code has more states than the code applied in [4] it can be shown that its trellis complexity [7] is lower. For the unpunctured code we obtain a complexity of

$$\frac{4}{3} \cdot 2^{5+3} = 341.3$$

symbol metric calculations per information bit while the punctured code has a complexity of

$$\frac{4}{3} \cdot 2^{6+1} = 170.7$$

symbol metric calculations per information bit, which is a saving of 50 percent. Moreover our concept allows a higher flexibility when choosing an appropiate channel code since the denominator of the code rate needs not to be a multiple of the parameter size M.

The question how to provide the a priori knowledge about source parameters is beyond the scope of this paper but was already covered in several recent publications [3, 8]. The principles derived there can directly be adopted to our algorithm.

4 Results

To study the behavior of our algorithm let us first consider a simple experiment. We model the source encoder output as a zero mean white Gaussian noise source with variance $\sigma^2 = 1$. The noise signal is linearly quantized with 16 levels using a midtread quantization characteristic with minimum value -8.5 and maximum value 7.5 respectively, thus yielding parameters of 4 bits. The entropy of this source is about 2 bit thus generating a statistical dependence between the parameter bits. The channel encoder applies a rate 1/2 code with octal generators $G_1 = 15$, $G_2 = 17$ taken from [9].

Fig. 6 depicts the simulation results employing an AWGN channel, for the ML decoding, the APRI-VA [2] using single bit a priori probabilities and the new MAP approaches, both the approximate and the exact one. Except for the exact MAP decoding in all other simulations the source bits were interleaved by a 4×6 block interleaver to simulate a spread range $R_l > L + 1$.

The performance of the single bit MAP decoder in this example is only slightly better than ML decoding. This is due to the almost symmetrical distribution density function of the quantized source.

To explain the effect consider the following simple example: Let a source produce four different symbols A, B, C and D with the probabilties $\Pr(A) = \Pr(B) = 0.2$ and $\Pr(C) = \Pr(D) = 0.3$. If we encode the symbols with two bits and assign

$$A \rightarrow 00, B \rightarrow 10, C \rightarrow 01, D \rightarrow 11$$

then for both the first and the second bit a probability of $\Pr(0) = \Pr(1) = 0.5$ results thus pretending at the single bit MAP decoder that the source contains no redundancy. In contrast the assignment

$$A \rightarrow 00, B \rightarrow 01, C \rightarrow 11, D \rightarrow 10$$

results in $\Pr(0) = 0.4$ for the first and $\Pr(0) = 0.5$ for the second bit. This example makes clear that the single bit MAP decoder is very sensitive to the chosen bit pattern assignment or the shape of the parameter density function, respectively.

As further simulations have shown, the approximate parameter MAP decoder also slightly depends on the bit pattern assignment but its performance is always better than that of the single bit MAP decoder. The performance of the exact parameter MAP decoder according to case B is completely independent of the bit pattern assignment, as expected.

In order to test our concept in a practical communication system we choose 32kbit/s ADPCM as the source encoder. This codec produces one 4-bit output parameter per input sample. For speech input signals it can be shown that some parameter values are much more probable than others. Fig. 7 shows the normalized histogram of the ADPCM output parameter for a long representative speech sequence. Moreover the correlation between subsequent parameter values is almost zero.

Therefore, we modelled the ADPCM parameter as a white source with the measured distribution of Fig. 7.

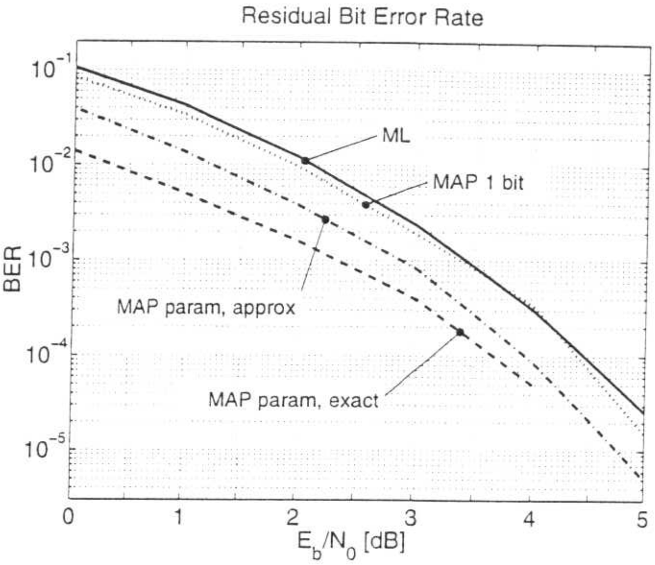

Fig. 6: Residual bit error rate for ML, single bit MAP (APRI-VA), approximate parameter MAP (case B) and exact parameter MAP decoding (case A)

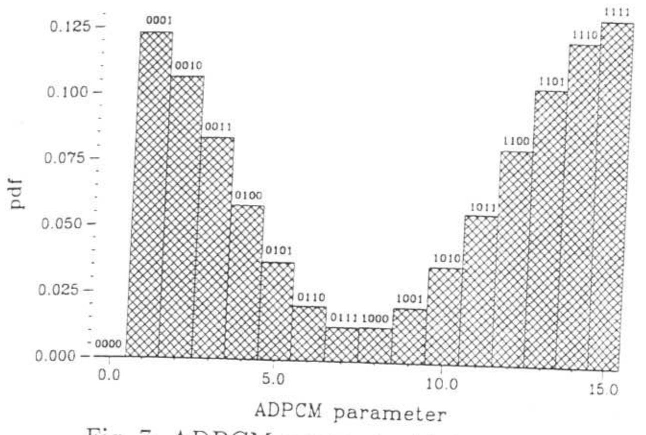

Fig. 7: ADPCM parameter histogram

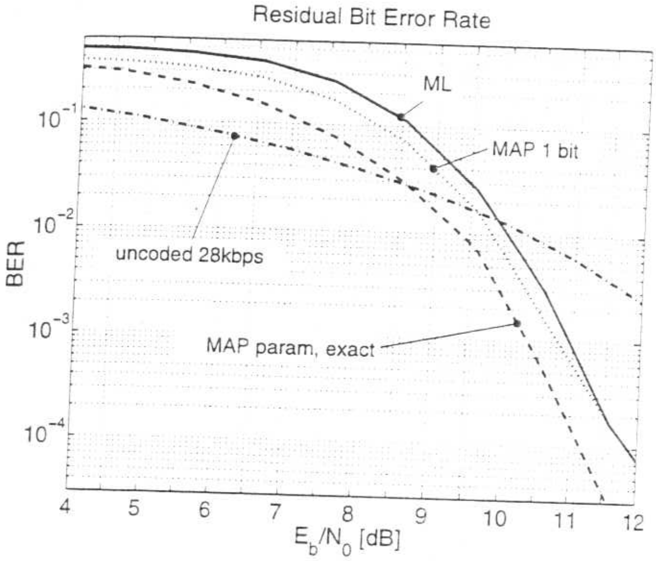

Fig. 8: Modified ADPCM, residual bit error rate for ML, single bit MAP (APRI-VA) and exact parameter MAP decoding (case B)

Howerver, as we did not want to change the gross bit rate of 32kbit/s on the channel with the applied forward error correction scheme we modified the ADPCM codec such that the bit rate is merely 28kbit/s by producing alternating parameters with 4 or 3 bits each (in this case we have to apply two parameter histograms, one for the 3-bit and one for the 4-bit parameter).

Thus 4kbit/s are available for a rate 7/8 punctured [6] channel code ($G_1 = 23$, $G_2 = 35$). Fig. 8 shows the residual bit error rates for the ML, the single bit MAP and the exact parameter MAP decoder again for transmission over an AWGN channel. In this case the gain achieved by channel coding is marginal due to the high rate code but it can be improved by MAP decoding exploiting parameter a priori knowledge especially at low signal to noise ratios. Note that the simulation speech sequence was not included in the training sequence applied to obtain the histogram in Fig. 7.

5 Conclusion

We have presented two new concepts of exploiting a priori knowledge on parameter level for the decoding of convolutional codes.

The first approach gives an approximate solution to the problem but is not restricted to a certain arrangement of the information bits before channel encoding.

The second approach performs an exact MAP sequence decoding, yet it is restricted to cases where the bits of one parameter are encoded subsequently and the size of the parameter does not exceed the constraint length of the convolutional code.

To simplify the formal description only white parameter sources were considered, i.e. we did not model correlations between subsequent parameter values. Yet, the principles proposed here also be extended to coloured sources. However, this requires the application of a soft output channel decoder, e.g. the SOVA [10].

Compared to ML decoding, both approaches do not significantly increase complexity.

References

[1] C. E. Shannon, "Collected Papers," in *IEEE Press, New York*, p. 40, N. J. A. Sloane and A. Wyner, Eds., 1993.

[2] J. Hagenauer, "Source-Controlled Channel Decoding," *IEEE Transactions on Communications*, vol. 43, pp. 2449–2457, Sept. 1995.

[3] T. Fingscheidt and P. Vary, "Robust Speech Decoding: A Universal Approach to Bit Error Concealment," in *ICASSP'97*, vol. 3, pp. 1667–1670, Apr. 1997.

[4] F. I. Alajaji, N. C. Phambo, and T. E. Fuja, "Channel Codes that Exploit the Residual Redundancy in CELP-Encoded Speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, pp. 325–336, Sept. 1996.

[5] S. Lin and J. Daniel J. Costello, *Error Control Coding*. Prentice-Hall, 1983.

[6] L. H. C. Lee, "New Rate-Compatible Punctured Convolutional Codes for Viterbi Decoding," *IEEE Transactions on Communications*, vol. 42, pp. 3073–3079, dec 1994.

[7] R. J. McEliece and W. Lin, "The Trellis Complexity of Convolutional Codes," *IEEE Transactions on Information Theory*, vol. 42, pp. 1855–1864, november 1996.

[8] C. Gerlach, "A Probabilistic Framework For Optimum Speech Extrapolation in Digital Mobile Radio," in *Proceedings ICASSP*, (Minneapolis), pp. II.419–II.422, Apr. 1993.

[9] J. G. Proakis, *Digital Communications*. McGraw-Hill, 1989.

[10] J. Hagenauer and P. Höher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications," *Proceedings GlobeCom*, pp. 1680–1686, Nov 1989.