Synchronization recovery and state model reduction for soft decoding of variable length codes
Simon Malinowski, Hervé Jégou, Christine Guillemot

To cite this version:

HAL Id: inria-00119600
https://inria.hal.science/inria-00119600v2
Submitted on 12 Dec 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract—Variable length codes (VLCs) exhibit de-
synchronization problems when transmitted over noisy
channels. Trellis decoding techniques based on Maximum A
Posteriori (MAP) estimators are often used to minimize the
error rate on the estimated sequence. If the number of symbols
and/or bits transmitted are known by the decoder, termination
constraints can be incorporated in the decoding process. All the
paths in the trellis which do not lead to a valid sequence length
are suppressed. This paper presents an analytic method to assess
the expected error resilience of a VLC when trellis decoding
with a sequence length constraint is used. The approach is
based on the computation, for a given code, of the amount of
information brought by the constraint. It is then shown that
this quantity as well as the probability that the VLC decoder
does not re-synchronize in a strict sense, are not significantly
altered by appropriate trellis states aggregation. This proves
that the performance obtained by running a length-constrained
Viterbi decoder on aggregated state models approaches the one
obtained with the bit/symbol trellis, with a significantly reduced
complexity. It is then shown that the complexity can be further
deprecated by projecting the state model on two state models of
reduced size.

I. INTRODUCTION

VLCs are widely used in compression systems due to their
high compression efficiency. One drawback of VLCs is their
high sensitivity to errors. A single bit error may lead to the
de-synchronization of the decoder. Nevertheless, many VLCs
exhibit self-synchronization properties. The authors in [1]
show such properties for some binary Huffman codes. The
error recovery properties of VLCs have also been studied in [2],
where a method to compute the so-called expected
error span \(E_s\) (i.e., the expected number of source symbols on
which a single bit error propagates), has been proposed. The
same quantity has been called mean error propagation length
(MEPL) in [3]. The authors in [4] consider the variance of the
error propagation length (VEPL) to assess the resilience of a
code with hard decoding techniques. In [5], the method in [2]
is extended to compute the so-called synchronization gain/loss,
i.e., the probability that the number of symbols in the transmit-
ted and decoded sequences differ by a given amount \(\Delta S\) when
a single bit error occurs during the transmission. Note that
various VLC constructions have also been proposed to improve
the self-synchronization properties of the codes [6], [7], [8].
The author in [7] introduces a method to construct prefix-free
self-synchronizing VLCs called T-codes. The synchronization
property of these codes is analyzed in [9] in terms of the
expected synchronization delay.

VLC soft decoding techniques based on MAP (or MMSE)
estimators have also been considered to minimize the error
rates (or distortion) observed on the decoded sequences. The
approaches essentially differ in the optimization metrics as
well as in the assumptions made on the source model and on
the information available at the decoder. These assumptions
lead to different trellis structures on which the estimation or
soft-decoding algorithms are run. Two main types of trellises
are considered to estimate the sequence of emitted symbols
from the received noisy bitstream: the bit-level trellis proposed
in [10] and the bit/symbol trellis. The bit-level trellis leads to
low decoding complexity. However, it does not allow the
exploitation of extra information, such as the number of
 emitted symbols. It hence suffers from some suboptimality. If
the knowledge of the number of emitted symbols is available
at the decoder, the problem is referred to as soft decoding with
length constraint and is addressed, e.g., in [11][12][13][14].
This problem has led to the introduction of the bit/symbol
trellis in [15]. This trellis can optimally exploit such con-
straints, leading to optimal performance in terms of error
resilience. Nevertheless, the number of states of the bit/symbol
trellis is a quadratic function of the sequence length. The
 corresponding complexity is actually not tractable for typical
sequence lengths. In order to overcome this complexity hurdle,
most authors apply suboptimal estimation methods on this
optimal state model such as sequential decoding [14][16][17].

This paper presents a method to assess the error resilience
of VLCs when trellis decoding with length constraint is used
at the decoder side. The approach is based on the concept of
gain polynomials defined on error state diagrams introduced
in [2] and [5]. The method introduced in [5] to compute the
synchronization gain/loss is first recalled. This method is then
extended to the case of a symbol sequence of length \(L(S)\)
being sent over a binary symmetrical channel (BSC) of a
given crossover probability. The derivation is inspired from
the matricial method described in [3]. It has been shown in
[18][19] that the Markovian property of a source can be easily
integrated in the source model by expanding the state model by
a constant factor. We thus restrict the analysis to memoryless
sources. It is shown that for VLCs, the probability mass func-
tion (p.m.f.) of the synchronization gain/loss is a key indicator
of the error resilience of such codes when soft decoding with
length constraint is applied at the decoder side. The p.m.f.
of the gain/loss allows the computation of the probability that
the symbol length of the decoded sequence is equal to \(L(S)\), i.e.
the probability that the decoder resynchronizes in the strict-
sense (no gain nor loss of symbols during the transmission).
This quantity is given by \(P(\Delta S = 0)\). The length constraint
is used to discard all decoded sequences which do not satisfy
the constraint \(\Delta S = 0\). If \(P(\Delta S = 0)\) is high, the number of
 "de-synchronized” sequences which will be discarded will be

Synchronization recovery and state model reduction
for soft decoding of variable length codes

Simon Malinowski, Hervé Jégou, and Christine Guillemot
high. This results in increasing the likelihood of the correct sequence, hence in decreasing the decoding error rate. The entropy of the p.m.f. of the gain/loss represents the amount of information that the length constraint brings to the decoder. These two quantities (\(P(\Delta S = 0)\) and \(H(\Delta S)\)) are shown to better predict the relative decoding performance of VLCs when soft decoding with a length constraint is used, than the MEPL and VEPL measures (these measures are appropriate when hard decoding is used). Note that, in the following, the term MEPL will be used to refer to the expectation of the error propagation length.

This analysis is then used in Section III to assess the performance of MAP decoding on the aggregated state models proposed in [20], for jointly typical source/channel realizations. The aggregated state model is defined by both the internal state of the VLC decoder (i.e., the internal node of the VLC codetree) and the remainder of the Euclidean division of the symbol clock values by a fixed parameter called \(T\). This model aggregates states of the bit/symbol trellis which differs by multiple of \(T\) symbol clock instants. The parameter \(T\) controls the trade-off between estimation accuracy and decoding complexity. The choice of this parameter has indeed an impact on the quantity of information brought by the length constraint on the corresponding trellis. It is shown that the probability that the VLC decoder does not re-synchronize in a strict sense, as well as the entropy of the constraint, are not significantly altered by aggregating states, provided that the aggregation parameter \(T\) is greater than or equal to a threshold. An upper bound of this threshold is derived according to the analysis of Section III. This proves that the performance obtained by running a length-constrained Viterbi decoder on the aggregated trellis closely approaches the performance obtained on the bit/symbol trellis, with a significantly reduced complexity. Finally, it is shown in Section IV that the decoding complexity can be further reduced by considering separate estimations on trellises of smaller dimensions, whose parameters \(T_1\) and \(T_2\) are relatively prime. If the two sequence estimates are not equal, the decoding on a trellis of parameter \(T_1 \times T_2\) is then computed. The equivalence in terms of decoding performance between this approach, referred to as combined trellis decoding, and the decoding on a trellis of parameter \(T_1 \times T_2\) is proved for the MAP criterion, i.e. for the Viterbi algorithm [21].

II. LINK BETWEEN VLC SYNCHRONIZATION RECOVERY PROPERTIES AND SOFT DECODING PERFORMANCE WITH A LENGTH CONSTRAINT

Let \(S = S_1, \ldots, S_i, \ldots S_{L(S)}\) be a sequence of \(L(S)\) symbols. This sequence is encoded with a VLC \(C\), producing a bitstream \(X = X_1, \ldots, X_k, \ldots X_{L(X)}\) of length \(L(X)\). This bitstream is modulated using a binary phase shift keying (BPSK) modulation and is transmitted over an additive white Gaussian noise (AWGN) channel, without any channel protection. The channel is characterized by its signal to noise ratio, denoted \(E_b/N_0\) and expressed in decibels (dB). Note that we reserve block capital letters to represent random variables and small letters to represent their corresponding realizations. In this paper, the term polynomial refers to expressions of the form \(\sum_{i \in \mathbb{Z}} a_i x^i\), where \(x\) denotes the variable and \(a_i\) are polynomial coefficients. Hence, we include in this terminology either polynomial series (with an infinite number of non null coefficients) or finite length polynomials (such as \(\exists N \in \mathbb{N} \mid \forall n > N, a_{-1} = a_4 = 0\), both with negative powers.

A. The gain/loss of a variable length code

A method to compute the so-called expected error span \(E_s\) following a single bit error has been introduced in [2]. This method relies on an error state diagram which represents the states of the decoder when the encoder is in the root node. Hence, the error state diagram includes the internal states of the decoder, i.e. the internal nodes of the VLC, plus two states which represent the loss of synchronization state \(n_l\) and the return to synchronization state \(n_s\) respectively. Therefore, the set of states of the diagram is \(\{n_l, n_{a_1}, n_{a_2}, \ldots, n_s\}\), where the set \(\{a_1, a_2, \ldots\}\) represents the set of prefixes of the VLC. The state \(n_s\) of the error state diagram corresponds to a return of both encoder and decoder automata to the root node of the code tree. However, this state may not correspond to a strict sense synchronization. In other words, the number of decoded symbols may be different from the number of emitted ones. The branches of the error state diagram represent the transitions between two states of the decoder when a single source symbol has been emitted by the encoder. They are labeled by an indeterminate variable \(z\) which corresponds to the encoding of one source symbol. Hence, the gain along each edge is the probability of the transition associated with that edge multiplied by \(z\). In that case, the gain on the diagram from \(n_l\) to \(n_s\) (i.e. the transfer function between \(n_l\) and \(n_s\)) is a polynomial of the variable \(z\) such that the coefficient of \(z^i\) is the probability that the considered VLC resynchronizes after exactly \(i\) source symbols following the bit error. Evaluating the derivative of the gain polynomial at 1 provides the expected error span \(E_s\).

The branch labeling of the error state diagram has been extended in [5] so that the gain polynomial informs about the difference, caused by a single bit error, between the number of emitted and decoded symbols, after hard decoding of the received bitstream. This quantity, denoted \(\Delta S\), is referred to as the gain/loss. In order to evaluate the p.m.f. of the random variable \(\Delta S\), a new variable \(y^j\) is introduced in the branch labeling of the error state diagram. The exponent \(j\) represents the number of extra output symbols for each input symbol. Hence, the corresponding gain polynomial \(G(y, z)\) is function of both variables \(y\) and \(z\). Evaluating this polynomial at \(z = 1\) gives a polynomial in \(y\) only. For sake of clarity, we simply denote this polynomial as

\[
G(y) = G(y, z)|_{z=1}.
\]  

The coefficient of \(y^i\) in the polynomial \(G(y)\) gives the probability \(P(\Delta S = i)\) following one bit error. Note that \(i\) can be negative if the decoded sequence is longer than the encoded one. In this section, we focus on the behavior of the polynomial \(G(y)\). Since the variable \(z\) is not necessary, we compute directly the state diagram for \(z = 1\).
Let $\mathbf{H}$ be the transition matrix corresponding to the error state diagram.

$$
\mathbf{H} = \begin{pmatrix}
P(n_1|n_1) & P(n_0|n_1) & \cdots & P(n_s|n_1) \\
P(n_1|n_0) & P(n_0|n_0) & \cdots & P(n_s|n_0) \\
\vdots & \vdots & \ddots & \vdots \\
P(n_1|n_s) & P(n_0|n_s) & \cdots & P(n_s|n_s)
\end{pmatrix}
$$

(2)

where $P(n_{i_j}|n_{i_j})$ represents the probability to go to state $n_{i_j}$ from state $n_{i_j}$. Let us call $h^k_{i_j}(y)$ the element at row $n_{i_j}$ and column $j$ of the matrix $\mathbf{H}^k$. Note that $h^k_{i_j}(1)$ is the probability to go from state $n_{i_j}$ to state $n_{i_j}$ in $k$ stages, i.e. after the encoding of $k$ source symbols. The top right elements of the matrices $\mathbf{H}$ and $\mathbf{H}^k$ are respectively denoted $h(y) = P(n_s|n_l)$ and $h^k(y)$. The gain polynomial $G(y)$ can then be written as

$$
G(y) = \sum_{k \in \mathbb{N}^+} h^k(y).
$$

(3)

Hence, the gain polynomial $G(y)$ is obtained as the top right element of the matrix $(I - \mathbf{H})^{-1}$, where $I$ denotes the identity matrix of the same dimensions as $\mathbf{H}$. Note that this property holds if $(I - \mathbf{H})^{-1}$ exists.

Let us consider the 5-symbol source and the 16 VLCs used in [3] to illustrate these concepts. The probability of this source as well as the different codes are reproduced in Table I. These codes have the same mean description length of 2.2 bits per symbols.

<table>
<thead>
<tr>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.2</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$P(a_i) = 0.1$</th>
<th>$P(a_i) = 0.2$</th>
<th>$P(a_i) = 0.2$</th>
<th>$P(a_i) = 0.1$</th>
<th>$P(a_i) = 0.1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Example 1: Let us consider the code $C_5$. Its state diagram is depicted in Fig. 1. The transition matrix derived from the previous guidelines, i.e. by setting the variable $z$ to 1 in the extended diagram of [5], is given by

$$
\mathbf{H}_{C_5} = \begin{pmatrix}
0 & \frac{1}{11} & \frac{1}{11} & \frac{2y}{11} & \frac{7}{11} \\
0 & \frac{1}{5} & \frac{1}{5} & 0 & \frac{y}{5} \\
0 & 0 & \frac{1}{5} & 0 & \frac{3+2y}{5} \\
0 & \frac{1}{5} & \frac{1}{5} & 0 & \frac{2y}{5} \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

(4)

This leads to $G(y) = 0.0625y^{-1} + 0.8352 + 0.1023y$, which also means that

$$
\begin{align*}
P(\Delta S = -1) &= 0.1023 \\
P(\Delta S = 0) &= 0.8352 \\
P(\Delta S = 1) &= 0.0625.
\end{align*}
$$

(5)

B. Extension to the BSC

Let us recall that $\Delta S$ corresponds to the gain/loss engendered by a single bit error. We propose here to estimate $P(\Delta S = i)$ for a sequence of $L(S)$ symbols that has been sent through a BSC of crossover probability $p$ (equals to the bit error rate). Since in this section the VLC decoder is assumed to be a classical hard decoder, the analysis is also valid on an AWGN channel characterized by its signal to noise ratio $E_b/N_0$ by taking $p = \frac{1}{2} \text{erfc} \left( \sqrt{\frac{E_b}{N_0}} \right)$.

For a sequence of $L(S)$ symbols, the bitstream length $L(X)$ lies in the interval of integers $I = \{L(S) \times l_m, \ldots, L(S) \times l_M\}$, where $l_m$ and $l_M$ respectively denote the lengths of the shortest and longest codewords. Let $E$ denote the random variable corresponding to the number of errors after the hard decoding of the received bitstream $Y$. For $i \in \mathbb{Z}$, the probability $P(\Delta S = i)$ is given by

$$
P(\Delta S = i) = \sum_{e \in \mathbb{N}} P(\Delta S = i | E = e)P(E = e).
$$

(6)

For $e \in \mathbb{N}$, the probability $P(E = e)$ can be expressed as

$$
P(E = e) = \sum_{k \in I} P(E = e | L(X) = k)P(L(X) = k)
$$

if $e \leq L(S) \times l_M$

(7)

otherwise $P(E = e) = 0$,

(8)

where the quantities $P(E = e | L(X) = k)$ only depend on the signal to noise ratio and are equal to

$$
\begin{align*}
P(E = e | L(X) = k) &= \binom{k}{e} p^e (1-p)^{i-e} & \text{if } e \leq k \\
P(E = e | L(X) = k) &= 0 & \text{if } e > k.
\end{align*}
$$

(9)
For every \( k \in \mathcal{I} \), the probability \( P(L(X) = k) \) is calculated from the source statistics and the code \( C \) structure.

To calculate \( P(\Delta S = i) \) according to Eqn. \([3]\) we now need to compute the quantities \( P(\Delta S = i|E = e) \). For that purpose, let us now assume that the decoder has already recovered from previous errors when another error occurs. This assumption requires that the probability that an error occurs when the decoder has not returned to the synchronization state \( n_a \) is very low. The lower the error span is and the higher \( E_b/N_0 \) is, the more accurate this approximation is. Under this assumption, the quantity \( \Delta S \) is independently impacted by multiple errors.

Let us define

\[
G_e(y) = (G \ast \ldots \ast G)(y) = \sum_{i \in \mathbb{Z}} a_{i,e} y^i,
\]

where \( \ast \) denotes the convolution product. Note that the polynomial \( G_1 = G \) corresponds to the gain polynomial of Eqn. \([3]\). Under the previous assumption, the quantity \( a_{i,e} \) equals \( P(\Delta S = i|E = e) \). With Eqn. \([8]\) the resulting gain polynomial for this crossover probability can be expressed as

\[
\tilde{G}(y) = \sum_{e \in \mathbb{N}} G_e(y) P(E = e),
\]

where only the quantity \( P(E = e) \) depends on \( E_b/N_0 \). The coefficients \( \tilde{g}_i \) of \( \tilde{G} \) verify

\[
\tilde{g}_i = \sum_{e \in \mathbb{N}} a_{i,e} P(E = e)
\]

\[
= \sum_{e \in \mathbb{N}} P(\Delta S = i|E = e)P(E = e)
\]

\[
= P(\Delta S = i).
\]

Let \( \eta > 0 \) be a criterion of negligibility. For a given \( \eta \), the pseudo-degree \( d_\eta \) of the polynomial \( \tilde{G} \) is defined as

\[
d_\eta \triangleq \min_{N^*} \left| \sum_{i \in \mathbb{Z} \setminus \{-d, \ldots, d\}} \tilde{g}_i \right| < \eta.
\]

The pseudo-degree \( d_\eta \) of a polynomial is the degree beyond which the sum of the coefficients of this polynomial are below a given threshold \( \eta \).

**Example 2:** Let us determine the pseudo-degree such that \( \eta = 10^{-6} \) for the code \( C_5 \), \( E_b/N_0 = 6 \text{dB} \), and \( L(S) = 100 \). The estimates \( \tilde{g}_i \) obtained from Eqn. \([8]\) lead to

\[
\begin{align*}
P(\Delta S \leq -4) &= 0.0000002 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = -4) &= 0.00000235 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = -2) &= 0.00019218 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = 0) &= 0.98186664 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = 1) &= 0.0301524 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = 2) &= 0.0004930 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = 3) &= 0.0000053 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S \geq 4) &= 0.0000001.
\end{align*}
\]

Hence, according to the definition of the pseudo-degree in Eqn. \([3]\), \( d_\eta \) is 3. The values of \( P(\Delta S = i) \) obtained by simulation and averaged over \( 10^7 \) channel realizations are

\[
\begin{align*}
P(\Delta S \leq -4) &= 0.0000002 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = -3) &= 0.00000235 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = -2) &= 0.00019218 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = 0) &= 0.98186664 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = 1) &= 0.0301524 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = 2) &= 0.0004930 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S = 3) &= 0.0000053 \quad \text{lead to}
\end{align*}
\]

\[
\begin{align*}P(\Delta S \geq 4) &= 0.0000001.
\end{align*}
\]

and also lead to \( d_\eta = 3 \).

The simulated values of \( P(\Delta S = i) \) are close to the estimated ones for a large set of \( E_b/N_0 \) values, which validates the approximation. The pseudo-degrees for \( \eta = 10^{-6} \) of the codes introduced in Table \([1]\) have been computed and are given in Table \([1]\).

**C. Code selection criteria**

Let us consider a MAP estimation run on the bit/symbol trellis, with an additional constraint on the length of the decoded sequence. This length constraint is used to discard all decoded sequences having a number of symbols which differs from the number of transmitted symbols, that is, which does not satisfy the constraint \( \Delta S = 0 \). On the bit/symbol trellis, the decoder has two kinds of information to help the estimation: the excess rate of the code and the information brough by the length constraint. The excess rate of a VLC (residual redundancy in the encoded bitstream) is given by the difference between the mean description length (mdl) of the code and the entropy of the source. The information brough by the length constraint on the bit/symbol trellis is given by the entropy of the p.m.f. of the gain/loss measure (\( \Delta S \)). For the considered set of codes, the excess rate is equal to 0.0781 bits of information and is the same for all codes of Table \([1]\).

From the p.m.f. of \( \Delta S \) the following two quantities can be computed:

- the probability \( P(\Delta S = 0) \) to have a strict sense resynchronization
- the entropy \( H(\Delta S) \).

If the probability \( P(\Delta S = 0) \) is small, the number of “de-synchronized” sequences which will be discarded will be high, then the probability of detecting and correcting errors increases. This results in increasing the likelihood of the correct sequence, hence in decreasing the decoding error rate. As explained below, the entropy \( H(\Delta S) \) measures the amount of information brought by the length constraint on the bit/symbol trellis. To design performance criteria for VLCs, we consider codes having the same mdl so that their performance can be fairly compared. Hence, the values \( P(\Delta S = 0) \) and \( H(\Delta S) \), computed from the p.m.f. of the gain/loss measure, are indicators of the performance of a VLC when soft decoding with length constraint is applied at the decoder side. Table \([1]\) shows the values of these two quantities for the codes of Table \([1]\) together with the MEPL and the VEPL of \([3]\). The corresponding decoding performance in terms of the normalized Levenshtein distance (NLD) \([22]\), BER and frame error rate (FER) obtained with the bit/symbol trellis, for \( E_b/N_0 = 6 \text{dB} \) and \( L(S) = 100 \), are also given. It can be observed that
These three codes have the same mean description length and made for longer sequences (see Table III for NLD. Similarly, the code one could expect this code to lead to the worst decoding H.

Simulations have also been performed with a larger source

<table>
<thead>
<tr>
<th>Code</th>
<th>(d_q)</th>
<th>(P(\Delta S = 0))</th>
<th>(H(\Delta S))</th>
<th>MEPL[3]</th>
<th>VEPL[3]</th>
<th>NLD</th>
<th>BER</th>
<th>FER</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C_1) 3</td>
<td>0.9185</td>
<td>0.499</td>
<td>3.89256</td>
<td>34.721</td>
<td>0.00877</td>
<td>0.00193</td>
<td>0.34053</td>
<td></td>
</tr>
<tr>
<td>(C_2) 4</td>
<td>0.9065</td>
<td>0.578</td>
<td>2.02273</td>
<td>2.003</td>
<td>0.00632</td>
<td>0.00191</td>
<td>0.33641</td>
<td></td>
</tr>
<tr>
<td>(C_3) 4</td>
<td>0.8971</td>
<td>0.595</td>
<td>2.06061</td>
<td>2.107</td>
<td>0.00626</td>
<td>0.00192</td>
<td>0.33636</td>
<td></td>
</tr>
<tr>
<td>(C_4) 4</td>
<td>0.8913</td>
<td>0.608</td>
<td>4.07692</td>
<td>27.800</td>
<td>0.00759</td>
<td>0.00177</td>
<td>0.31548</td>
<td></td>
</tr>
<tr>
<td>(C_5) 3</td>
<td>0.9187</td>
<td>0.497</td>
<td>1.71023</td>
<td>1.200</td>
<td>0.00586</td>
<td>0.00194</td>
<td>0.34296</td>
<td></td>
</tr>
<tr>
<td>(C_6) 4</td>
<td>0.8996</td>
<td>0.578</td>
<td>3.54546</td>
<td>18.854</td>
<td>0.00758</td>
<td>0.00182</td>
<td>0.32638</td>
<td></td>
</tr>
<tr>
<td>(C_7) 5</td>
<td>0.7088</td>
<td>1.287</td>
<td>1.55556</td>
<td>0.370</td>
<td>0.00619</td>
<td>0.00154</td>
<td>0.21849</td>
<td></td>
</tr>
<tr>
<td>(C_8) 10</td>
<td>0.7006</td>
<td>1.553</td>
<td>2.34861</td>
<td>2.045</td>
<td>0.00646</td>
<td>0.00134</td>
<td>0.19543</td>
<td></td>
</tr>
<tr>
<td>(C_9) 9</td>
<td>0.6703</td>
<td>1.632</td>
<td>1.95707</td>
<td>1.025</td>
<td>0.00571</td>
<td>0.00123</td>
<td>0.16739</td>
<td></td>
</tr>
<tr>
<td>(C_{10}) 36</td>
<td>0.6401</td>
<td>2.267</td>
<td>6.18182</td>
<td>36.231</td>
<td>0.00483</td>
<td>0.00074</td>
<td>0.10354</td>
<td></td>
</tr>
<tr>
<td>(C_{11}) 8</td>
<td>0.8797</td>
<td>0.655</td>
<td>1.85227</td>
<td>2.233</td>
<td>0.00614</td>
<td>0.00183</td>
<td>0.32219</td>
<td></td>
</tr>
<tr>
<td>(C_{12}) 8</td>
<td>0.8882</td>
<td>0.620</td>
<td>1.71678</td>
<td>1.506</td>
<td>0.00617</td>
<td>0.00187</td>
<td>0.32951</td>
<td></td>
</tr>
<tr>
<td>(C_{13}) 8</td>
<td>0.8860</td>
<td>0.634</td>
<td>1.79798</td>
<td>1.914</td>
<td>0.00615</td>
<td>0.00182</td>
<td>0.32142</td>
<td></td>
</tr>
<tr>
<td>(C_{14}) 8</td>
<td>0.8957</td>
<td>0.599</td>
<td>2.03104</td>
<td>2.952</td>
<td>0.00666</td>
<td>0.00186</td>
<td>0.32698</td>
<td></td>
</tr>
<tr>
<td>(C_{15}) 8</td>
<td>0.8941</td>
<td>0.610</td>
<td>2.20321</td>
<td>4.144</td>
<td>0.00685</td>
<td>0.00189</td>
<td>0.33244</td>
<td></td>
</tr>
<tr>
<td>(C_{16}) 6</td>
<td>0.9044</td>
<td>0.564</td>
<td>1.96086</td>
<td>2.615</td>
<td>0.00672</td>
<td>0.00193</td>
<td>0.33829</td>
<td></td>
</tr>
</tbody>
</table>

The above analysis is used to assess the conditions for optimality of MAP decoding with length constraint on the aggregated state model described in [20]. This model keeps track of the symbol clock values modulo a parameter \(T\) instead of the symbol clock values as on the classical bit/symbol trellis. The state aggregation leads to a significantly reduced decoding complexity, as detailed in Section III-B. In this section, it is shown that, from \(d_q\) (the pseudo-degree of the polynomial representation of \(\Delta S\)), one can derive the minimal value of \(T\) required to have nearly optimum decoding performance (i.e. which closely approaches the performance obtained with the bit/symbol trellis). For these values of \(T\), we show that the amount of information conveyed by the length constraint is not significantly altered by state aggregation.

A. Optimal state model

The sequence of transmitted bits can be modeled as a hidden markov model with states defined as \(X_k\), where \(k\) represents the bit clock instants, \(1 \leq k \leq L(\mathbf{X})\). Let \(N_k\) denote the random variable corresponding to the internal state of the VLC (i.e. the internal node of the VLC codetree) at the bit clock instant \(k\). For instance, the possible values of \(N_k\) for the code \(C_0 = \{0, 10, 11\}\) are \(n_k\) and \(n_{k+1}\), where \(n_k\) represents the root node of the VLC codetree. In the bit-level trellis [10], the decoder state model is defined by the random variable \(N_k\) only. The internal states of the automaton associated with a given VLC are defined by the internal nodes of the codetree, as depicted in Fig. **II-a** for the code \(C_0\). The corresponding decoding trellis is given in Fig. **II-a**.
TABLE IV  
SOURCE AND CODES FOR THE ENGLISH ALPHABET USED IN THIS PAPER.

<table>
<thead>
<tr>
<th>ASCII Code</th>
<th>Symbol probability</th>
<th>C_{17}[2]</th>
<th>C_{18}[2]</th>
<th>C_{19}[5]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.08833733</td>
<td>0000</td>
<td>0100</td>
<td>0100</td>
</tr>
<tr>
<td>B</td>
<td>0.01267680</td>
<td>011111</td>
<td>111101</td>
<td>000101</td>
</tr>
<tr>
<td>C</td>
<td>0.02081665</td>
<td>11111</td>
<td>11100</td>
<td>01100</td>
</tr>
<tr>
<td>D</td>
<td>0.04765344</td>
<td>00010</td>
<td>10110</td>
<td>01101</td>
</tr>
<tr>
<td>E</td>
<td>0.1487569</td>
<td>001</td>
<td>000</td>
<td>100</td>
</tr>
<tr>
<td>F</td>
<td>0.02455297</td>
<td>11100</td>
<td>11010</td>
<td>00011</td>
</tr>
<tr>
<td>G</td>
<td>0.01521216</td>
<td>011101</td>
<td>111101</td>
<td>001100</td>
</tr>
<tr>
<td>H</td>
<td>0.05831331</td>
<td>1000</td>
<td>1000</td>
<td>1100</td>
</tr>
<tr>
<td>I</td>
<td>0.05644515</td>
<td>1001</td>
<td>1001</td>
<td>1111</td>
</tr>
<tr>
<td>J</td>
<td>0.00808964</td>
<td>11101011</td>
<td>11111110</td>
<td>00111010</td>
</tr>
<tr>
<td>K</td>
<td>0.0867360</td>
<td>111100</td>
<td>111110</td>
<td>0011100</td>
</tr>
<tr>
<td>L</td>
<td>0.04123298</td>
<td>00011</td>
<td>10111</td>
<td>00100</td>
</tr>
<tr>
<td>M</td>
<td>0.02361889</td>
<td>11110</td>
<td>11011</td>
<td>01110</td>
</tr>
<tr>
<td>N</td>
<td>0.06498532</td>
<td>0111</td>
<td>0111</td>
<td>0101</td>
</tr>
<tr>
<td>O</td>
<td>0.07245796</td>
<td>0101</td>
<td>0101</td>
<td>1101</td>
</tr>
<tr>
<td>P</td>
<td>0.0575393</td>
<td>10111</td>
<td>11001</td>
<td>01111</td>
</tr>
<tr>
<td>Q</td>
<td>0.00808964</td>
<td>111010100</td>
<td>111111110</td>
<td>001110100</td>
</tr>
<tr>
<td>R</td>
<td>0.06872164</td>
<td>0101</td>
<td>0101</td>
<td>1101</td>
</tr>
<tr>
<td>S</td>
<td>0.05537763</td>
<td>1010</td>
<td>1010</td>
<td>1110</td>
</tr>
<tr>
<td>T</td>
<td>0.09354149</td>
<td>110</td>
<td>001</td>
<td>101</td>
</tr>
<tr>
<td>U</td>
<td>0.02762299</td>
<td>10110</td>
<td>11000</td>
<td>00101</td>
</tr>
<tr>
<td>V</td>
<td>0.01160928</td>
<td>11101</td>
<td>11110</td>
<td>001111</td>
</tr>
<tr>
<td>W</td>
<td>0.01885161</td>
<td>01110</td>
<td>11010</td>
<td>001101</td>
</tr>
<tr>
<td>X</td>
<td>0.00046784</td>
<td>111011</td>
<td>1111110</td>
<td>00111011</td>
</tr>
<tr>
<td>Y</td>
<td>0.01521216</td>
<td>011110</td>
<td>111100</td>
<td>000100</td>
</tr>
<tr>
<td>Z</td>
<td>0.00053376</td>
<td>111010100</td>
<td>111111110</td>
<td>001110110</td>
</tr>
</tbody>
</table>

TABLE V  

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C_{17}</td>
<td>0.7312</td>
<td>1.376</td>
<td>5.456</td>
<td>5.868</td>
<td>0.0020986</td>
<td>0.53768</td>
</tr>
<tr>
<td>C_{18}</td>
<td>0.8388</td>
<td>0.861</td>
<td>3.863</td>
<td>3.960</td>
<td>0.002094</td>
<td>0.56607</td>
</tr>
<tr>
<td>C_{19}</td>
<td>0.8433</td>
<td>0.844</td>
<td>1.915</td>
<td>1.192</td>
<td>0.002105</td>
<td>0.56909</td>
</tr>
</tbody>
</table>

Let us assume that the number of transmitted symbols is perfectly known on the decoder side. To use this information as a termination constraint in the decoding process, the state model must keep track of the symbol clock (that is of the number of decoded symbols). The optimal state model is defined by the pair of random variables $(N_k, M_k)$ [13] [15], where $T_k$ denotes the symbol clock instant corresponding to the bit clock instant $k$. Since the trellis corresponding to this model is indexed by both the bit and the symbol instants, it is often called the bit/symbol trellis. This trellis is depicted in Fig. 2-b for the code $C_0$. The number of states of this model is a quadratic function of the sequence length (equivalently the bitstream length). The resulting computational cost is thus not tractable for typical values of the sequence length $L(S)$.

**B. Aggregated State model: a brief description**

The aggregated state model proposed in [20] is defined by the pair of random variables $(N_k, M_k)$, where $M_k = T_k \mod T$ is the remainder of the Euclidean division of $T_k$ by $T$. The corresponding realization of $M_k$ is denoted $m_k$. Note that $T = 1$ and $T = L(S)$ amounts to considering respectively the bit-level trellis and the bit/symbol trellis. The automaton and decoding trellis of parameter $T = 2$ correspond to this state model are depicted for the code $C_0$ in Figs. 2-b and 2-c respectively. The transitions which terminate in the state $n_k$, that is corresponding to the encoding/decoding of a symbol, modify $M_k$ as $M_k = M_{k-1} + 1 \mod T$. Hence, the transition probabilities on this automaton are given by

$$P(N_k = n_k, M_k = m_k | N_{k-1} = n_{k-1}, M_{k-1} = m_{k-1}) = \begin{cases} P(N_k = n_k | N_{k-1} = n_{k-1}) & \text{if } n_k \neq n_{k-1} \\

P(N_k = n_k | N_{k-1} = n_{k-1}) & \text{if } n_k = n_{k-1} \\

0 & \text{otherwise} \end{cases}$$

(18)
where the probabilities $P(N_k = n_k | N_{k-1} = n_{k-1})$ are deduced from the source statistics. Note that the transition probabilities $P(N_k | N_{k-1})$ are the ones used in the bit-level trellis.

The proposed state model keeps track of the symbol clock values modulo $T$ during the decoding process. In order to exploit this information, the decoder has to know the value $m_{L(X)} = L(S) \mod T$. This information can be used as a termination constraint, as depicted in Fig. 3. If this value is not given by the syntax elements of the source coding system, it has to be transmitted. The transmission cost of $m_{L(X)}$ is greater than or equal to $\log_2(T)$ bits. Note that the knowledge of this value has a lower cost than the one of transmitting the exact number of emitted symbols in the bit/symbol trellis. In the following, the quantity $m_{L(X)}$ is assumed to be known by the decoder. The estimation is performed using the Viterbi algorithm [21], hence minimizing the FER. In the sequel, the error resilience will be measured according to this criterion. For the estimation, the paths which do not satisfy the appropriate boundary constraints, i.e. the paths that do not terminate in states of the form $(n_e, m_{L(X)})$, are discarded. The number of states of the trellis of parameter $T$ satisfies

$$\nu_T \leq T \times L(X) \times \Gamma,$$

where $\Gamma$ represents the number of internal nodes of the code. The inequality in Eqn. [19] results from the fact that some pairs $(n_k, t_k)$ are not reachable according to the code structure. Such states are mostly located at the first and last bit clock instants of the trellis. However, for some particular codes, some states are not reachable all along the trellis. For example, for the set of codewords $\{0, 100, 101, 110, 111\}$, the states $(n_e, 2q), q \in \mathbb{N}$ are not reachable for any bit clock instants. To approximate the complexity on a trellis of parameter $T$, the worst case in terms of the number of states is considered, i.e. we assume that

$$\nu_T \approx T \times L(X) \times \Gamma,$$

Hence, as the number of states of the bit-level trellis is equal to $L(X) \times \Gamma$, the computational cost $D_T$ corresponding to the trellis of parameter $T$ can be approximated as

$$D_T \approx T \times D_{bal},$$
where $D_{\text{bal}}$ denotes the computational cost of the bit-level trellis. This computational cost is approximatively linear in the sequence length and in $T$.

C. Aggregated state model: analysis

According to the definition of the pseudo-degree (Eqn. [34] of the polynomial $G(y)$, the probability that $\Delta S$ belongs to the interval $\{-d_\eta, \ldots, d_\eta\}$ is greater than or equal to $1 - \eta$. This leads to the following property. Property I: A value of $T$ such that $T = d_\eta$, and all the more such that $T > d_\eta$, ensures that the Viterbi algorithm run on the aggregated trellis selects, with at least probability $1 - \eta$, a sequence with the correct number of symbols.

However, this property does not mean that the algorithm will offer similar results as the ones on the bit/symbol trellis. To analyze the respective performance of both models, the amount of information conveyed by the termination constraint in both cases must be quantified. These quantities are respectively given by the entropies of the random variables $\Delta S \mod T$ and $\Delta S$. They depend on the sequence length and $E_b/N_0$, which are assumed to be fixed. Here, we show that by setting the aggregation parameter $T$ to $T = 2d_\eta + 1$, the information brought by the length constraint on the aggregated trellis ($H(\Delta S \mod T)$) tends towards the one available on the bit/symbol trellis ($H(\Delta S)$).

For a trellis of parameter $T$ and following the analysis of Section II-B, the quantity

$$ \tilde{g}^T \triangleq P(\Delta S \mod T = i) $$

(22)

can be computed from the quantities $\tilde{g}_i$ as

$$ \tilde{g}^T_i = \sum_{j \in Z} \tilde{g}_{i+jT+i}. $$

(23)

The entropy of the termination constraint on a trellis of parameter $T$ is then given by

$$ H(\Delta S \mod T) = - \sum_{i \in \{0, \ldots, T-1\}} \tilde{g}^T_i \log_2 \tilde{g}^T_i $$

(24)

$$ \geq H(\Delta S) + \sum_{i \in \{0, \ldots, T-1\}} \tilde{g}_i \log_2 \tilde{g}_i. $$

(25)

When $T = 2d_\eta + 1$, (23) can be re-written as

$$ H(\Delta S \mod 2d_\eta + 1) \geq H(\Delta S) + \sum_{i \in \{-d_\eta, \ldots, d_\eta\}} \tilde{g}_i \log_2 \tilde{g}_i. $$

(26)

Let us now assume that $\eta < \frac{1}{2}$. Then the function $x \mapsto x \log(x)$ decreases on the interval $[0, \eta]$ and since $\forall i \notin \{-d_\eta, \ldots, d_\eta\}, \tilde{g}_i \leq \eta$, we have

$$ \sum_{i \notin \{-d_\eta, \ldots, d_\eta\}} \tilde{g}_i \log_2 \tilde{g}_i \geq \left| \left\{ i \notin \{-d_\eta, \ldots, d_\eta\}, \tilde{g}_i > 0 \right\} \right| \eta \log_2 \eta $$

(27)

where the cardinal $\left| \left\{ i, \tilde{g}_i > 0 \right\} \right|$ of the set of possible non-zero values of $\tilde{g}_i$ is bounded by the bitstream length $L(X)$. Together with $L(X) \geq T$, this leads to

$$ \left| \left\{ i \notin \{-d_\eta, \ldots, d_\eta\}, \tilde{g}_i > 0 \right\} \right| \leq L(X) - 2d_\eta - 1. $$

(28)

Hence, for a given $\eta$, we have the following lower and upper bounds:

$$ H(\Delta S) + (L(X) - 2d_\eta - 1) \eta \log_2 \eta \leq H(\Delta S \mod 2d_\eta + 1) \leq H(\Delta S). $$

(29)

These bounds mean that for $\eta$ small enough, hence for $T = 2d_\eta + 1$ sufficiently high, the quantity of information brought by the length constraint on the aggregated trellis of parameter $T$ tends toward the one available on the bit/symbol trellis.

Example 3: Let us consider the same parameters as in Example 2 (i.e. code $C_5$, $d_\eta = 3$). From Eqn. 29 we deduce that

$$ H(\Delta S) - H(\Delta S \mod 2d_\eta + 1) \leq \left( -100 \log(5) - 5 \right) \eta \log_2(\eta) $$

$$ \leq 0.04900 \text{ bits} $$

(30)

The convergence of $H(\Delta S \mod T)$ is depicted in Fig. 4 for codes of Table I. In this figure, the arrows represent the values of $H(\Delta S)$ for the considered codes. Note that for $C_{10}$, $H(\Delta S \mod T)$ has not converged towards $H(\Delta S)$ yet for $T = 10$. For the other codes, the limit is reached for $T \leq 10$. According to Section II-C, the best codes are those with the highest values of $H(\Delta S)$. Such codes require a higher value of $T$ to approach the value $H(\Delta S)$ of the entropy of the termination constraint on the bit/symbol trellis, since the pseudo-degree of these codes is higher. Nevertheless, for the considered set of codes, the values of $T$ leading to the same performance as on the bit/symbol trellis are always lower than $L(X)$. Note that for the code $C_{13}$, $H(\Delta S \mod 2) = 0 = H(\Delta S \mod 1)$. This means that the decoding performance of code $C_{13}$ on a trellis of parameter $T = 2$ is the same as the one on the bit/level trellis ($T = 1$).

The previous analysis has been validated by simulation, for sequences of $L(S) = 100$ symbols. For each parameter set ($VLC, E_b/N_0$, and $T$), the FER is measured over $10^5$ channel realizations. The performance at different values of the parameter $T$ and for the codes $C_5, C_7, C_{10}$ and $C_{13}$ is given
TABLE VI
FER for soft decoding (Viterbi) with different values of the aggregation parameter $T$.

<table>
<thead>
<tr>
<th>$E_b/N_0$</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T = 1$</td>
<td>0.99120</td>
<td>0.92309</td>
<td>0.70404</td>
<td>0.38774</td>
<td>0.14558</td>
</tr>
<tr>
<td>$T = 2$</td>
<td>0.98805</td>
<td>0.90368</td>
<td>0.66193</td>
<td>0.34633</td>
<td>0.12452</td>
</tr>
<tr>
<td>$T = 3$</td>
<td>0.98698</td>
<td>0.89901</td>
<td>0.65527</td>
<td>0.34313</td>
<td>0.12388</td>
</tr>
<tr>
<td>$T = 4$</td>
<td>0.98665</td>
<td>0.89795</td>
<td>0.65457</td>
<td>0.34298</td>
<td>0.12396</td>
</tr>
<tr>
<td>$T = 5$</td>
<td>0.98652</td>
<td>0.89782</td>
<td>0.65419</td>
<td>0.34296</td>
<td>0.12396</td>
</tr>
<tr>
<td>$T = 10$</td>
<td>0.98651</td>
<td>0.89780</td>
<td>0.65448</td>
<td>0.34296</td>
<td>0.12386</td>
</tr>
</tbody>
</table>

in Table VI. In this table, the best decoding performance for each code, at different values of $E_b/N_0$ is written in italics. These values correspond to the performance obtained on the bit/symbol trellis. Note that these values are obtained for a value of $T$ which is considerably lower than $L(S)$. As predicted, the trellis of parameter $T = 2$ does not bring any improvement in terms of error resilience for the code $C_{13}$ compared to the bit-level trellis. These results validate the criteria described in Section III-L to select good codes in terms of error resilience. Indeed, according to these criteria and the simulation results, the best code among the ones proposed in Table III is the code $C_{10}$ and the worst is the code $C_5$.

IV. COMBINED TRELlis DECODING

A. Motivation

In this section, we propose an approach allowing further reduction of the decoding complexity without inducing any suboptimality in terms of decoding performance. The optimality of this approach is proved for the FER criterion. This approach is motivated by the following equivalence

$$L(S) \mod (T_1 \times T_2) = m$$

$$\iff \begin{cases} L(S) \mod T_1 = m \mod T_1 \\ L(S) \mod T_2 = m \mod T_2, \end{cases}$$

satisfied if $T_1$ and $T_2$ are relatively prime. Note that, if $T_1$ and $T_2$ are not relatively prime, the converse is not satisfied.

Property 2: Let us assume that $T_1$ and $T_2$ are relatively prime and that $T_3 \not\equiv T_1 \times T_2$. Let us denote by $\hat{S}_1$, $\hat{S}_2$ and $\hat{S}_3$ the estimates of $S$ provided by the Viterbi algorithm run on the trellises of parameters $T_1$, $T_2$ and $T_3$ respectively. Then, we have

$$\hat{S}_1 = \hat{S}_2 \Rightarrow \hat{S}_3 = \hat{S}_1 = \hat{S}_2.$$  

Proof: Let us first emphasize that the probability of a sequence, computed by the Viterbi algorithm on a trellis of parameter $T$ does not depend on $T$. Let us assume that if two sequences have the same probability, then a subsidiary rule is applied to select one sequence amongst the two. For instance, the lexicographical order can be chosen as a comparison rule. Such a rule ensures that the Viterbi algorithm behavior is deterministic. Let

$$\mathcal{S}_T = \{s/l(S') \mod T = L(s) \mod T\}$$

be the set of sequences satisfying the termination constraint for the trellis of parameter $T$. From Eqn. (31), we deduce that if $T_3 = T_1 \times T_2$ with $T_1$ and $T_2$ relatively prime, then

$$\mathcal{S}_{T_3} = \mathcal{S}_{T_1} \cap \mathcal{S}_{T_2},$$

hence,

$$\mathcal{S}_{T_3} \subseteq \mathcal{S}_{T_1}.$$  

Moreover, since we have assumed that $\hat{S}_1 = \hat{S}_2$, we get

$$\hat{S}_1 \in \mathcal{S}_{T_3}.$$  

The estimate $\hat{S}_1$ provided by the Viterbi algorithm applied on the trellis of parameter $T_1$ is then such that

$$\hat{S}_1 = \arg \max_{s' \in \mathcal{S}_{T_1}} P(s'|X)$$

$$= \arg \max_{s' \in \mathcal{S}_{T_3}} P(s'|X)$$

$$= \hat{S}_3,$$

where the subsidiary rule may be used in the selection of the maximum. This concludes the proof.

This property means that if a sequence is selected by the trellises of parameters $T_1$ and $T_2$, then this sequence is also selected by the trellis of parameter $T_3$.

B. The decoding algorithm

The purpose of the algorithm described in this section is to exploit Property 2. The corresponding approach is referred to as combined trellis decoding. The rationale behind this approach is to use two trellises of parameters $T_1$ and $T_2$ instead of the trellis of parameter $T = T_1 \times T_2$ in order to reduce the overall decoding complexity. We will also assume that the greatest common divisor (gcd) of $T_1$ and $T_2$ is 1, i.e. that $T_1$ and $T_2$ are relatively prime. The decoding of a sequence proceeds as follows:

1. The Viterbi algorithm is applied to both trellises $T_1$ and $T_2$. They respectively provide the estimated sequences $\hat{S}_1$ and $\hat{S}_2$. 

2. The subsidiary rule is applied to select one sequence amongst the two. For instance, the lexicographical order can be chosen as a comparison rule.

3. The estimate $\hat{S}_1$ is then selected by the trellis of parameter $T_1$. 

4. The estimate $\hat{S}_3$ is then selected by the trellis of parameter $T_3$. 

5. The best estimate $\hat{S}$ is then the sequence selected by both trellises $T_1$ and $T_2$.
synchronisation recovery properties. For example, ∆ estimate. This quantity can hence be assumed to increase with
the trellises of parameter T_1 and T_2. Hence, the performance of the above 3-step decoding algorithm of parameters T_1 and T_2 is equivalent to the one obtained with a Viterbi decoder operating on the trellis of parameter T_1 × T_2.

C. Expected computational cost of the proposed algorithm

First, let us recall that if T = 1, the resulting trellis is equivalent to the bit-level trellis. If T is greater than or equal to L(S) − L(S) / mtd + 1 (hence greater than or equal to L(S)), the trellis is equivalent to the bit/symbol trellis. The intermediate values of T amount to considering trellis whose complexity is lower than the one of the bit/symbol trellis (see Section III-B). The expectation D_{mid}(T_1, T_2) of the computational cost of the proposed decoding scheme is then given by

\[ D_{mid}(T_1, T_2) = T_1D_{bal} + T_2D_{bal} + \rho T_1T_2D_{bal} \] (40)

where \( \rho = P(\hat{S}_1 \neq \hat{S}_2) \). In the following, D_{mid}(T_1, T_2) will be denoted D_{mid}. The proposed method is worthwhile in terms of computational cost if D_{mid} < T_1 × T_2 × D_{bal}, i.e. if

\[ \rho < \rho^* = 1 - \frac{T_1 + T_2}{T_1 \times T_2}. \] (41)

Therefore, the benefit of the proposed algorithm depends on the probability \( \rho \) that the two estimators return the same sequence estimate. The probability \( \rho \) decreases when the channel noise and/or the sequence length increases. Fig. 5 illustrates the complexity reduction brought by the combined trellis decoding algorithm for the same decoding performance. For the considered settings, a lower computational cost is obtained with this approach as long as E_b/N_0 is greater than 0.65 dB.

D. Constrained optimization of trellis parameters T_1 and T_2

Let T_c be a targeted decoding performance. According to the combined trellis decoding scheme described above, this level of performance can be reached using two trellises of parameters T_1 and T_2 such that T_1 × T_2 = T_c, T_1 and T_2 being relatively prime. Without loss of generality, let us assume that T_2 = T_1 + \Delta T, and T_1 = T_1 × (T_1 + \Delta T). Note that parsing the set N^+ × N^+ with the pairs (T_1, \Delta T) ensures to parse the set of attainable constraints. The probability \( \rho \) is a function of T_1 and T_2, hence a function of T_1 and \( \Delta T \). The computational cost D_{mid} of the combined trellis decoding algorithm of parameters T_1 and T_1 + \Delta T is given by

\[ D_{mid}(T_1, \Delta T) = \rho(T_1, \Delta T)D_{T_1} + D_{bal}(2T_1 + \Delta T) \] (42)

The quantity \( \rho(T_1, \Delta T) \) represents the probability that the trellises of parameter T_1 and T_1 + \Delta T do not provide the same estimate. This quantity can hence be assumed to increase with \( \Delta T \). This assumption may not be satisfied for codes having specific synchronisation recovery properties. For example, according to section III, even values of T are not appropriate for the code C_{13}. Indeed, for this code, a trellis of parameter T = 2q − 1, q ∈ N provides better decoding performance than a trellis of parameter T = 2q, q ∈ N. The previous assumption is not always satisfied for this specific code. Under the assumption that \( \rho(T_1, \Delta T) \) increases with \( \Delta T \), we deduce the following property from Eqn. 42.

Property 3: Let \( T_c ∈ N^+ \) and \( R_p \subseteq N^+ × N^+ \) be the subset of positive integers which are relatively prime. Then

\[ \arg \min_{T_1, T_2 ∈ R_p} D_{mid} = \arg \min_{T_1, T_2 ∈ R_p} |T_2 − T_1|. \] (43)

According to that property, the set of pairs (T_1, T_2) such that T_2 = T_1 + 1 is optimum.

V. Conclusion

This paper makes the link between re-synchronisation properties of VLCs and length-constrained MAP estimation techniques of these codes. This analysis is also used to assess conditions for optimality of state aggregation on the bit/symbol trellis widely used for soft decoding of VLC encoded sources. Nearly optimal decoding performance can be achieved with a reduced decoding complexity with respect to the classical bit/symbol trellis. A combined trellis decoding algorithm, further reducing the decoding complexity without inducing suboptimality, is then proposed. The aggregated trellises can easily be coupled with a convolutional code or a turbo-code in an iterative structure, as done in [13], and [23].
REFERENCES


