

Robust source coding with first order Multiplexed Codes : construction and soft decoding

Hervé Jégou, Christine Guillemot

► **To cite this version:**

Hervé Jégou, Christine Guillemot. Robust source coding with first order Multiplexed Codes : construction and soft decoding. [Research Report] RR-4942, INRIA. 2003. inria-00071637

HAL Id: inria-00071637

<https://hal.inria.fr/inria-00071637>

Submitted on 23 May 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.

***Robust source coding with first order Multiplexed
Codes: construction and soft decoding***

Hervé Jégou — Christine Guillemot

N° 4942

October 2003

THÈME 3



R
**apport
de recherche**

Robust source coding with first order Multiplexed Codes: construction and soft decoding

Hervé Jégou , Christine Guillemot

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet TEMICS

Rapport de recherche n° 4942 — October 2003 — 29 pages

Abstract: this paper addresses the problem of design of variable length codes (VLC) that would allow to exploit higher-order source statistics while still being resilient to transmission errors. The codes introduced extend and generalize the family of codes called multiplexed codes, referred to as stationary multiplexed codes. The design principle consists in creating fixed length codes (FLCs) for high priority information and in using the inherent redundancy to describe low priority data, hence the name “multiplexed codes”. The FLC codebooks are partitioned into equivalence classes according to high priority source statistics, focusing here on first-order statistics. The key aspect to limit error propagation inherent to the use of conditional probabilities relies on the choice of some appropriate partitions (or index assignment) of the set of codewords. Several index assignment methods are described. A crossed-Index Assignment method leads to the construction of a kernel made of codewords offering synchronization properties. The decoder re-synchronization capability can be further increased by periodic use of stationary multiplexed codes. Soft decoding algorithms using MAP, MPM and MMSE criteria are also described. Theoretical and practical performances in terms of error resilience and compression efficiency are analyzed.

Key-words: source coding, variable length codes, data compression, data communication, entropy codes

Codage de source robuste avec les codes multiplexés d'ordre 1: construction et décodage souple

Résumé : ce document traite de la construction de codes à longueur variable permettant d'exploiter des statistiques de sources non stationnaires tout en étant robustes aux erreurs de transmission. Ces codes étendent et généralisent la famille de codes appelés codes multiplexés. Le principe de construction consiste à créer des codes à longueur fixe (FLC) pour la source d'information prioritaire, et d'utiliser la redondance intrinsèque pour décrire l'information de priorité moindre, d'où le nom "codes multiplexés". L'ensemble des mots de codes est partitionné en classes d'équivalence, dont le cardinal dépend des statistiques de la source prioritaire. Nous nous concentrons ici sur des chaînes de Markov d'ordre 1. Un aspect primordial pour limiter la propagation des erreurs inhérente à l'utilisation de probabilités conditionnelles est le choix de partitions appropriées (ou l'assignement d'étiquetage binaire). Plusieurs méthodes d'étiquetage binaire sont décrites. Une méthode d'étiquetage binaire croisée conduit à la construction d'un noyau de mots de codes offrant des propriétés de synchronisation. La capacité de resynchronisation du décodeur peut encore être améliorée par l'utilisation périodique de codes multiplexés qui n'utilisent que la loi de probabilité stationnaire de la source. Des algorithmes de décodage souple selon les critères MAP, MPM et MMSE sont également décrits dans le cadre de ces codes. Les performances théoriques et expérimentales sont analysées.

Mots-clés : codage de source, codes à longueur variable, compression de données, communications numériques, codage entropique

1 Introduction

Entropy coding, producing variable length codewords, is a core component of any data compression scheme. Unlike fixed length codes (FLCs), variable length codes (VLCs) are designed to exploit the inherent redundancy of a symbol distribution. The main drawback of VLCs is their high sensitivity to channel noise: when some bits are altered by the channel, synchronization losses can occur at the receiver, the position of symbol boundaries are not properly estimated, leading to dramatic symbol error rates. This observation has motivated studies of the synchronization ability of VLCs [15] as well as the design of codes with better synchronization properties such as self-synchronizing Huffman codes [6][14], or reversible variable length codes (RVLC) [21][23][2]. The increased re-synchronization capability is however often obtained at the expense of redundancy, hence of decreased compression efficiency. MAP or MMSE estimators, capitalizing on coders sub-optimality by exploiting residual redundancy [20] (the so-called “excess-rate”) have also been shown to reduce the “de-synchronization” effect [17][3][2][7], however at the expense of high decoding complexity.

The decoder de-synchronization problem comes from the difficulty to properly segment the noisy bitstream into symbols. This segmentation problem can be addressed by introducing a-priori information in the bitstream, taking often the form of synchronization patterns, or of forbidden symbols or intervals (e.g. [18]), but again at the expense of loss in compression efficiency. One can alternately, by a proper structuring of the bitstream, reveal and exploit constraints on some bit positions. This idea is first applied in [19] to blocks within an image, leading to a bitstream structure called EREC (Error-Resilient Entropy Codes). It is shown in [10] that the bitstream structure used to transmit VLCs has a major impact on the error resilience. For a given codetree code, the error resilience performance bounds can be obtained with a family of codes called “binary multiplexed codes”, special case of so-called “multiplexed codes” introduced in [9]. The design principle of “multiplexed codes” is based on the fact that compression systems of real signals generate sources of information with different levels of priority (e.g. texture and motion information for a video signal). The codes designed are then such that the risk of “de-synchronization” is confined to the low priority information. Unlike most works on Unequal Error Protection (UEP) based on the use of different channel codes rates (e.g. [8]), UEP is intrinsically provided by the multiplexed code, without any redundancy. However, the family of codes presented in [9] allows to achieve maximum compression efficiency for the low priority source (e.g. the source before multiplexing can be pre-encoded by using an arithmetic code coupled with higher-order source statistical models) but only allows to exploit stationary statistical distributions for the high priority source \mathbf{S}_H . In the sequel, we will refer to this family of codes as *stationary multiplexed codes*.

In this paper, we address the problem of design of a family of multiplexed codes exploiting higher-order statistics for the high priority source, in order to achieve higher compression performances. We focus on first-order statistical models. Similarly to *stationary multiplexed codes*, a set of FLCs is first created for the high priority source (\mathbf{S}_H). The inherent redundancy of this code is then exploited to represent or store information of the low priority source (\mathbf{S}_L) which is assumed to be pre-encoded with a classical entropy code (e.g. arith-

metic codes). The set of FLCs is then partitioned into classes of equivalences according to \mathbf{S}_H conditional probabilities allowing to reach conditional entropy bounds. Theoretical performance *upper bounds* in terms of symbol error rate (SER) and mean-square error (MSE) are then derived for discrete memoryless channels. These bounds provide the performance expectations of SER and MSE when hard decoding is processed and are outperformed when a soft decoder is used instead. As for Vector Quantizers [12] [4], it is shown that the efficiency is dependent on the codeword index assignment strategies, i.e., on the construction of the classes of equivalence and on the index assignment in the different classes. Index assignment (IA) strategies for both criteria (SER and MSE) are described. A *crossed-IA* strategy leads to the construction of a set of codewords offering synchronization properties and called the code *kernel*. These codes can be decoded at low computational cost by using either hard or soft decision decoding techniques. In contrast with classical VLCs (e.g. Huffman, arithmetic), the complexity of the soft decision decoding method remains low, i.e. in the order of $O(K)$, where K is the length of the sequence of symbols to be decoded. MAP or MMSE decoding criteria can then be used. The decoder re-synchronization capability can be further increased by a periodic and adaptive use of stationary multiplexed codes to code some symbols of the sequence. An analysis of the trade-off between compression efficiency and resynchronization capability is given. Theoretical and practical performances both in terms of error resilience and compression efficiency are discussed for different index assignment strategies and for different decoding criteria.

2 Problem statement and Notations

Let $\mathbf{S}_H = (S_1, \dots, S_t, \dots, S_{K_H})$ be a sequence of source symbols of high priority taking their values in a finite alphabet \mathcal{A} composed of Ω symbols, $\mathcal{A} = \{a_1, \dots, a_i, \dots, a_\Omega\}$. Let $\mathbf{S}_L = (S'_1, \dots, S'_{i'}, \dots, S'_{K_L})$ be a sequence of source symbols of lower priority taking their values in a finite alphabet \mathcal{A}' . The source \mathbf{S}_H is assumed to be a first order Markov process. Note that, in the following, we reserve capital letters to random variables, and small letters to values of these variables. Bold face characters will be used to denote vectors or sequences. The stationary and conditional probabilities of the source \mathbf{S}_H are respectively denoted $\boldsymbol{\mu} = (\mu_1, \dots, \mu_i, \dots, \mu_\Omega)$ and $\boldsymbol{\nu} = (\nu_{1,1}, \dots, \nu_{i,i'}, \dots, \nu_{\Omega,\Omega})$, where μ_i stands for the probability $\mathbb{P}(S_t = a_i)$ and $\nu_{i,i'}$ stands for the conditional probability $\mathbb{P}(S_t = a_i / S_{t-1} = a_{i'})$. Emitted and received codewords are respectively denoted X_t and Y_t . The channel is assumed to follow a discrete memoryless channel (DMC) model with transition probabilities $P(Y_t|X_t)$ denoted $R(\cdot, \cdot)$.

3 First order Multiplexed Codes

3.1 Stationary Multiplexed Codes: a review

The design principle of *multiplexed codes* consists in creating a fixed length code (FLC) for the high priority source \mathbf{S}_H , in partitioning the set of FLCs in *classes of equivalence* and in

class \mathcal{C}_i	codeword $c_{i,q}$	a_i	$N_i = \text{card}(\mathcal{C}_i)$	μ_i	index q
\mathcal{C}_1	000	a_1	2	0.30	0
	001				1
\mathcal{C}_2	010	a_2	3	0.43	0
	011				1
	100				2
\mathcal{C}_3	101	a_3	1	0.25	0
	110				1
\mathcal{C}_4	111	a_4	1	0.02	0

Table 1: An example of stationary multiplexed codes ($c = 3$).

exploiting the redundancy inherent of each class to represent or store information of the low priority source \mathbf{S}_L . Thus, the fixed length codewords represent jointly a symbol of the high priority source and a set of symbols of the low priority source.

Let us denote \mathcal{X} the set of codewords of length c . A stationary multiplexed code can be constructed by partitioning the codeword space \mathcal{X} into Ω subsets \mathcal{C}_i of cardinal $|\mathcal{C}_i| = N_i$, called equivalence classes. Each equivalence class \mathcal{C}_i is associated to a symbol a_i of the alphabet \mathcal{A} . A codeword $x \in \mathcal{X}$ such that $x \in \mathcal{C}_i$ can hence be associated to a symbol a_i of the alphabet \mathcal{A} of the high priority source and to an index value q_i with $0 \leq q_i \leq N_i - 1$. Thus, a multiplexed code can be defined by the bijection

$$\begin{aligned} \mathcal{X} &\rightarrow \mathcal{A} \times [0..N_i - 1] \\ x &\mapsto (a_i, q_x). \end{aligned} \quad (1)$$

Hence, each symbol S_t of the sequence of symbols of high priority to be encoded, during the encoding process, is mapped into a pair (\mathcal{C}_i, Q) of two variables denoting respectively the *equivalence class* and the index of the codeword in the *equivalence class* \mathcal{C}_i . A symbol $S_t = a_i$ of the flow \mathbf{S}_H can thus be encoded with any c -bit codeword x belonging to the *equivalence class* \mathcal{C}_i (see example of Table 1). The variable Q is an N_i -valued variable, taking its value q between 0 and $N_i - 1$ (see Table 1) and representing the inherent redundancy of the c -bits fixed length codes. The description length of the variable Q_i is then $\log_2(N_i)$ bits. Assuming that this capacity is fully used to describe data of the low priority source, the average description length of a symbol of the high priority source \mathbf{S}_H is then given by $c - \log_2(N_i)$ bits which will be equal to the source stationary entropy h^0 if $\forall i, c - \log_2(N_i) = -\log_2(\mu_i)$, i.e. if $N_i = \mu_i |\mathcal{X}|$.

For example, let $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$ be the alphabet of the source \mathbf{S}_H with the stationary probabilities given by $\mu_1 = 0.30$, $\mu_2 = 0.43$, $\mu_3 = 0.25$, and $\mu_4 = 0.02$. Table 1 gives an example of partitioning of the set of $N = 2^c$ codewords into the 4 *equivalence classes* associated to the alphabet symbols.

Therefore, to the realization of the sequence of symbols \mathbf{S}_H one associates a sequence of N_i -valued variables $\mathbf{N} = N_1, \dots, N_i, \dots, N_{K_H}$, used to describe jointly the \mathbf{S}_H and \mathbf{S}_L data

flows. The sequence of low priority symbols is assumed to be pre-encoded with a classical VLC (e.g. Huffman or arithmetic code). The resulting low priority bitstream must then be converted into a sequence of n_t -valued variables (q_t), where $t = 1, \dots, K_H$. The sequence of pairs (n_t, q_t) provides entries in the multiplexed code table. In [11], two methods have been introduced for this conversion:

- The first approach regards the low priority bitstream as an integer which is then decomposed using an iterative Euclidean decomposition. This approach allows to reach the entropy bound.
- A second approach relies on a constrained partition of the FLCs into *equivalence classes* with cardinals belonging to a subset of integer values. This approach still leads to a good approximation of the high priority source distribution, hence is quasi-optimal.

In the following any of these two methods applies. We will then only focus on the design of the partition that will depend on higher-order statistics of the source \mathbf{S}_H .

3.2 First-order multiplexed codes: Definition

The stationary multiplexed codes above are constructed for the stationary distribution of the high priority source \mathbf{S}_H . However, it is well known that, for Markov sources, higher compression efficiency can be obtained by adapting the codes to higher-order source statistics. We focus here on first-order statistics, however, the approach can be easily extended to higher orders.

First-order multiplexed codes can thus be constructed for the conditional probability distribution $\mathbb{P}(S_t = a_i | S_{t-1} = a_{i'})$ of the high priority source. The partition of the set of fixed length codewords \mathcal{X} then becomes conditioned by the realization of the previous symbol $S_{t-1} = a_{i'}$ in the sequence to be encoded. Let $\mathcal{C}_i^{i'}$ and $N_i^{i'}$ respectively denote the equivalence class associated to the symbol a_i and its cardinal when the previous symbol realization is $S_{t-1} = a_{i'}$. The set of codes, for all conditional symbol values from \mathcal{A} , defines a so-called *first-order multiplexed code*, referred to as \mathcal{C}^* in the sequel. The mean description length (mdl) of the code \mathcal{C}^* (for the high priority source \mathbf{S}_H) is given by

$$\text{mdl}(\mathcal{C}^*) = \sum_{(a_i, a_{i'}) \in \mathcal{X}^2} \mathbb{P}(S_t = a_i | S_{t-1} = a_{i'}) (c - \log_2(N_i^{i'})) \quad (2)$$

$$= - \sum_{a_{i'} \in \mathcal{X}} \sum_{a_i \in \mathcal{X}} \mu_i \nu_{i, i'} \log_2 \left(\frac{N_i^{i'}}{|\mathcal{X}|} \right). \quad (3)$$

The mdl of the code \mathcal{C}^* will be equal to the first-order entropy h^1 of the source if and only if

$$\forall (a_i, a_{i'}) \in \mathcal{A}^2, N_i^{i'} = \nu_{i, i'} |\mathcal{X}|. \quad (4)$$

Note that the higher compression efficiency may not result in a shorter sequence for \mathbf{S}_H , as in classical VLCs, but rather in a higher multiplexing capacity to be used for describing the low priority source \mathbf{S}_L .

The conditioning symbol values are often called contexts in source compression. In practice, when the size of the alphabet \mathcal{A} is large, the number of contexts may be too high leading to a problem of context dilution. This number can be reduced by considering different conditioning values as the same context. The context can then be defined by a function σ of the previous symbol realization, taking its value in the set of contexts, denoted Σ :

$$\begin{aligned} \mathcal{A} &\rightarrow \Sigma \\ a_{i'} &\mapsto \sigma(a'_i) \end{aligned} \quad (5)$$

Together with the realization of the symbol $S_t = a_i$ to be encoded, this context identifies the equivalence class to be used, denoted $\mathcal{C}_i^{\sigma(a_{i'})}$. If σ is a constant function, the multiplexed code is obviously a stationary multiplexed code. In that case, the code is referred to as \mathcal{C}^\emptyset .

4 Error resilience analysis on a DMC

This section analyzes the impact of the use of higher-order source statistics on the error resilience of the code, expressed both in terms of symbol error rate (SER) and of mean square error (MSE) obtained for the high priority source \mathbf{S}_H . Let us consider first-order multiplexed codes designed for a Markov source S of transition probabilities $\nu_{i,i'}$. Note first that with first-order multiplexed codes, as with stationary multiplexed codes, the high priority source does not suffer from de-synchronization problem: the segmentation of the bitstream into high priority source symbols is deterministic. However, erroneous contexts (conditioning values) may lead to error propagation in the decoding of the symbol values.

4.1 Modeling the dependencies in the coding and transmission chain

In order to capture the error propagation phenomenon, the SER and MSE performance bounds must be expressed in terms of the probability distribution $\mathbb{P}(\hat{S}_t | S_t, \hat{S}_{t-1}, S_{t-1})$, or equivalently (with the appropriate re-normalization factor) in terms of $\mathbb{P}(\hat{S}_t, S_t | \hat{S}_{t-1}, S_{t-1})$. The analysis hence requires to consider the global transmission chain formed by the source, the multiplexed source coder, the transmission channel and the decoder. The different variables involved in the global chain and their dependencies are graphically depicted in Fig.1. The coding, transmission and decoding chain can be modeled by a process represented by the pair of variables $Z_t = (S_t, \hat{S}_t)$. The process $Z_t = (S_t, \hat{S}_t)$, $t = 1 \dots K$ is a Markov

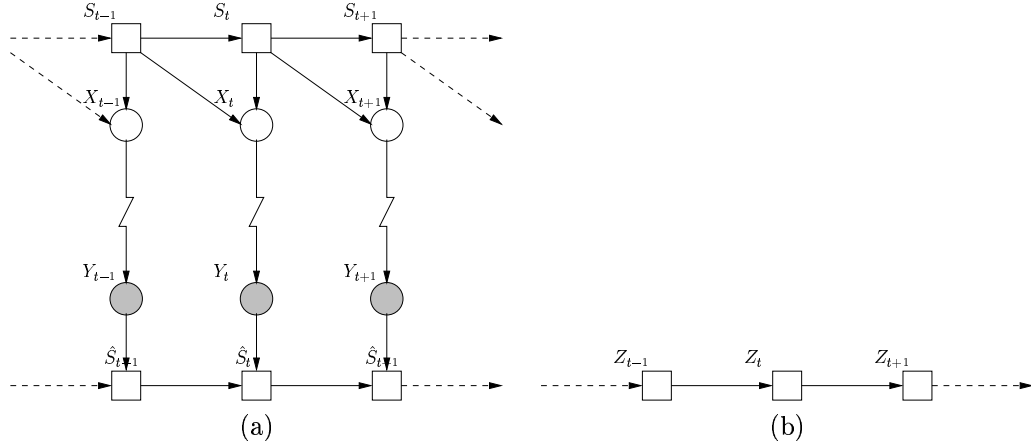


Figure 1: (a) Graphical representation of dependencies between the variables $(S_t)_t$, $(\hat{S}_t)_t$, $(X_t)_t$ and $(Y_t)_t$. (b) Product model: Markov process (Z_t) .

chain. Its transition probability is given by

$$\boldsymbol{\pi} = \begin{pmatrix} \pi(z_0, z_0) & \pi(z_0, z_1) & \dots & \pi(z_0, z_{\Omega^2}) \\ \pi(z_1, z_0) & \pi(z_1, z_1) & \dots & \pi(z_1, z_{\Omega^2}) \\ \vdots & \vdots & \ddots & \vdots \\ \pi(z_{\Omega^2}, z_0) & \pi(z_{\Omega^2}, z_1) & \dots & \pi(z_{\Omega^2}, z_{\Omega^2}) \end{pmatrix} \quad (6)$$

where

$$\pi(z, z') = \mathbb{P}(\hat{S}_t = a_j; S_t = a_i \mid \hat{S}_{t-1} = a_{j'}; S_{t-1} = a_{i'}) \quad (7)$$

$$= \sum_{(X_t, Y_t) \in \mathcal{X}^2} \mathbb{P}(\hat{S}_t = a_j; Y_t; S_t = a_i; X_t \mid \hat{S}_{t-1} = a_{j'}; S_{t-1} = a_{i'}) \quad (8)$$

$$= \sum_{(X_t, Y_t) \in \mathcal{X}^2} \mathbb{P}(S_t = a_i; X_t \mid \hat{S}_{t-1} = a_{j'}; S_{t-1} = a_{i'}) \\ \times \mathbb{P}(\hat{S}_t = a_j; Y_t \mid S_t = a_i; X_t; \hat{S}_{t-1} = a_{j'}; S_{t-1} = a_{i'}). \quad (9)$$

It is shown in Appendix 1 that the transition probability $\pi(z, z')$ is actually given by

$$\pi(z, z') = \frac{\nu_{i, i'}}{N_i^{\sigma(a_{i'})}} \sum_{(X_t, Y_t) \in \mathcal{C}_i^{\sigma(a_i)} \times \mathcal{C}_j^{\sigma(a_{j'})}} R(Y_t \mid X_t), \quad (10)$$

where $R(Y_t \mid X_t)$ represents the channel model, i.e., the probability of receiving the codeword Y_t if the codeword X_t has been emitted. The entity $\nu_{i, i'}$ represents the conditional probability

of the high priority source, and $\mathcal{C}_i^{\sigma(a_i')}$ the corresponding class of equivalence of cardinality $N_i^{\sigma(a_i')}$.

4.2 Asymptotic SER and MSE bounds

For a sequence of infinite length, the Markov process Z_t converges toward a stationary state of probability $\mathbb{P}(Z_t) = \mathbb{P}(S_t; \hat{S}_t)$. The corresponding asymptotic SER and MSE performance bounds are thus function of the transition probability distribution $\mathbb{P}(\hat{S}_t|S_t)$, or equivalently (with the appropriate renormalization factor) of the joint probability $\mathbb{P}(S_t; \hat{S}_t)$. The joint probability distribution $\mathbb{P}(S_t; \hat{S}_t)$ can be regarded as the stationary probability of the Markov process defined by the pair of variables $Z_t = (S_t, \hat{S}_t)$. According to the Perron-Frobenius theorem¹, this stationary distribution is given by the normalized eigenvector associated to the eigenvalue 1 of the transition matrix π . In the following, this eigenvector is denoted $\lambda_\infty = (\lambda_\infty(a_1, a_1) \dots, \lambda_\infty(a_1, a_\Omega), \dots, \lambda_\infty(a_i, a_j) \dots, \lambda_\infty(a_\Omega, a_\Omega))^T$. This pdf verifies the relation $\lambda_\infty = \pi \lambda_\infty$. The asymptotic value of the SER can be expressed in terms of the summation of probabilities over product states $z = (a_i, a_i)$, i.e.

$$\text{SER}_{\mathcal{C}^*} = 1 - \sum_{a_i \in \mathcal{A}} \lambda_\infty(a_i, a_i) \quad (11)$$

$$= 1 - \delta^T \lambda_\infty \quad (12)$$

where the vector δ denotes a vector such that $\delta(a_i, a_j) = 1$ if $a_i = a_j$, and $\delta(a_i, a_j) = 0$ otherwise. Similarly, the asymptotic performance of the MSE is given by

$$\text{MSE}_{\mathcal{C}^*} = \Delta^T \lambda_\infty \quad (13)$$

where the distortion vector Δ is defined such that $\Delta(a_i, a_j) = \|a_i - a_j\|^2$. These bounds have been verified experimentally: As the number of channel realizations increases, the experimental SER and MSE values asymptotically converge toward the theoretical bounds given in Eqn.12 and Eqn.13.

4.3 Impact of index assignment on error resilience

The assignment of multiplexed codewords to the different equivalence classes for all the possible contexts (i.e. finding the partition) as well as the assignment of binary representations to the different codewords are referred to in the sequel as *index assignment* (IA). In the context of Vector Quantization (VQ), it has been shown that Error-Resilience can be improved, by modifying the binary labeling of codewords. To improve an existing IA, the binary switching algorithm [12] allows to find a local optimum. Simulated annealing [4] aims at finding a global optimum, by avoiding local optimum in a statistical manner. It generally provides better results than the binary switching algorithm. It has also been shown

¹All probabilities are assumed to be greater than zero, which leads to have the matrix π positive.

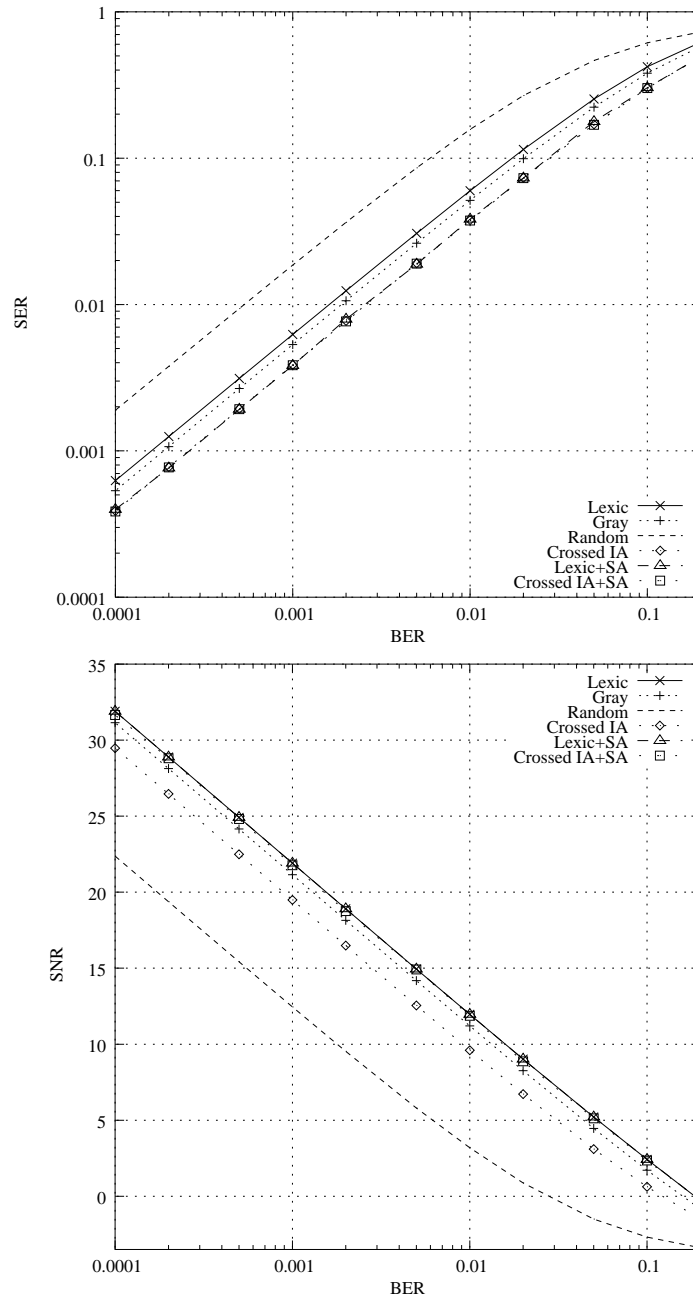


Figure 2: Theoretical SER and SNR performance, for a Gauss-Markov source ($\rho = 0.5$) quantized on 8 levels, with different index assignments. The parameters of the first-order multiplexed code used are $c = 5$ and 8 contexts (i.e. $\sigma(a_{i'}) = i'$).

that channel-optimized vector quantization [5] can improve the overall rate-distortion efficiency, but in that case the designed quantizer may not be optimum for noiseless channels. In this paper, the codebooks are assumed to be optimum, in terms of compression, i.e. they approximatively verify Eqn. 4. Consequently, the analysis is restricted to the IA of binary codewords. The IA is observed to have a major impact on the multiplexed codes error-resilience, as shown in Fig. 2. The optimal IA depends on channel and source properties. However, it may also depend on the performance measure used (e.g. SER or SNR). Fig. 2 shows that an IA based on Gray Codes has better SER performance than IA with lexicographical codes. However, the lexicographical IA outperforms the Gray code when the SNR measure is used.

5 Code design: crossed index assignment and code kernel

Thus, IA turns out to be a key issue. For stationary multiplexed codes, finding the optimal index assignment has a prohibitive computational cost: the number of possible partitions of \mathcal{X} into Ω subsets of cardinalities N_i ($1 \leq i \leq \Omega$) is given by $\frac{|\mathcal{X}|!}{\prod_{i=1}^{\Omega} N_i!}$. The proof is given in Appendix 2. For first-order multiplexed codes, the number of possible partitions is even higher and given by

$$\left(\frac{|\mathcal{X}|!}{\prod_{i=1}^{\Omega} N_i!} \right)^{|\Sigma|}, \quad (14)$$

where $|\Sigma|$ denotes the number of distinct contexts. The IA conditioned by a given context must take into account the IA conditioned by the other contexts, hence the name *crossed index assignment* (crossed IA).

The IA can in addition be designed in order to optimize different criteria, e.g. SER or MSE with performance bounds respectively given in Eqn. 12 and Eqn. 13, assuming hard-decision decoding. The optimal IA is hence given by the set of partitions (one per context) and corresponding codeword indexing leading to the minimum argument of these expressions. Each step of the IA optimization procedure according to Eqn. 12 and Eqn. 13 requires the computation of eigenvalues of matrices of size $\Omega^2 \times \Omega^2$. The resulting complexity is not easily tractable. One can design crossed index assignment strategies approaching the optimal performance with a much reduced complexity.

5.1 Crossed Index Assignment strategy

Simplified optimization criteria controlling error propagation due to erroneous contexts and the reconstruction error of the current symbol can be considered. We first focus on reducing the error propagation effect by constructing the partition in such a way that the decoding of a symbol will result in a correct context even if the current context is erroneous.

5.1.1 Reduction of error propagation

the problem of IA at a given time instant is then of trying to maximize the probability that the next symbol will be decoded with a correct context. Let us denote $\mathbb{P}(\sigma_k)$ the probability of having the context σ_k at a symbol clock instant t . This probability is deduced from the source stationary probabilities by

$$\mathbb{P}(\sigma_k) = \sum_{a_i \in \sigma^{-1}(\sigma_k)} \mathbb{P}(S_t = a_i), \quad (15)$$

where $\sigma^{-1}(\sigma_k)$ is the set of symbols such that, $\forall a_i \in \sigma^{-1}(\sigma_k)$, $\sigma(a_i) = \sigma_k$. Now, let us denote $a_x^{\sigma_k}$ the symbol of \mathcal{A} represented by the codeword x if the current state of the decoder is the context σ_k . If the received codeword is correct but the context is erroneous, for example $\tilde{\sigma}_k$ instead of σ_k , the context used for the next symbol will be erroneous if and only if $\sigma(a_x^{\sigma_k}) \neq \sigma(a_x^{\tilde{\sigma}_k})$. Then, we can derive the following similarity measure, which applies to codewords:

$$d_{\mathcal{X}}(x) = \sum_{(\sigma_k, \tilde{\sigma}_k) \in \Sigma^2} \mathbb{P}(\sigma_k) \mathbb{P}(\tilde{\sigma}_k/\sigma_k) \delta(\sigma(a_x^{\sigma_k}), \sigma(a_x^{\tilde{\sigma}_k})), \quad (16)$$

where $\delta(.,.)$ denotes the Kronecker operator. The measure $d_{\mathcal{X}}(x)$ represents the probability that decoding the codeword x leads to a correct context for subsequent symbols, even if the current context is in error. As an approximation, we assume that $\mathbb{P}(\tilde{\sigma}_k/\sigma_k) \approx \mathbb{P}(\tilde{\sigma}_k)$, which avoids seeking eigenvectors of large matrices. This measure can be extended to the whole set of codewords, which leads to

$$d_{\mathcal{X}}(\mathcal{C}^*) = \sum_{x \in \mathcal{X}} \mathbb{P}(x) d_{\mathcal{X}}(x) \approx \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} d_{\mathcal{X}}(x). \quad (17)$$

The IA can then be chosen so that the measure $d_{\mathcal{X}}(\mathcal{C}^*)$ is optimized. The IA algorithm then proceeds as follows: the IA is first initialized by taking e.g. a lexicographical order. The algorithm then proceeds with permutations of the different symbols across the classes of equivalence, and this for the different contexts. If a permutation leads to an increase of $d_{\mathcal{X}}(\mathcal{C}^*)$, then it is retained. The algorithm stops when no further increase of $d_{\mathcal{X}}(\mathcal{C}^*)$ is observed. The underlying optimization principle is close to the binary switching algorithm [12], in so far as a local optimum of Eqn. 17 is searched using symbol permutations. A simulated annealing can also be applied. However, experimentally, we did not observe any significant improvement in terms of convergence for the final value $d_{\mathcal{X}}(\mathcal{C}^*)$.

5.1.2 Constrained optimization of the IA

the value $d_{\mathcal{X}}(\mathcal{C}^*)$ only depends on the assignment of symbols a_i of \mathcal{A} to the different classes, and this taking into account the different possible contexts. It does not take into account the channel characteristics, nor the binary representations of the codewords. The permutation

of the binary representations of two codewords² x_1 and x_2 does not modify the value $d_{\mathcal{X}}(\mathcal{C}^*)$. One can then choose the binary representations of the different codewords in order to reduce the SER (Eqn. 12) or the MSE (Eqn. 13). The above optimization of the crossed IA can be carried out using a simulated annealing (SA) algorithm, where the perturbations correspond to an arbitrary choice of the contexts and partition.

The performance of the SA algorithm strongly depends on the initial choices of the parameters. We first applied this SA algorithm using a lexicographical code as the initial state. However, with the SER criterion, the resulting codes were, most of the time, less error-resilient than the one obtained with the sequential approach described above in sections 5.1.1 and 5.1.2. Hence, in the sequel, we have chosen the code resulting from this approach, referred to as *crossed IA*, as the initialization of the SA algorithm.

5.2 Definition and properties of the kernel

The crossed IA strategy according to the error propagation and SER criterion actually amounts to constructing a subset of codewords, referred to as *the kernel* of the code with the following properties.

Definition 1: The kernel \mathcal{K} of a multiplexed code \mathcal{C}^* is the set of codewords of \mathcal{X} such that

$$x \in \mathcal{K} \text{ iff } \exists a_i \in \mathcal{A} / \forall \sigma_k \in \Sigma, x \in \mathcal{C}_i^{\sigma_k} \quad (18)$$

By definition, a codeword that belongs to the kernel always represents the same symbol of \mathbf{S}_H , whatever the context. Thus, the kernel can be partitioned into Ω sets of codewords, as

$$\mathcal{K} = \bigcup_{a_i \in \mathcal{A}} \mathcal{K}_i, \quad (19)$$

where \mathcal{K}_i is the set of codewords of the kernel that represents the symbol a_i for every context. For a stationary multiplexed code, we have obviously $|\mathcal{K}_i| = N_i$. For a first-order multiplexed code, the way a codeword of the kernel is decoded does not depend on past realizations. Thus, these codewords act as *hard synchronization* points. The expectation to encode a symbol with a codeword of the kernel is given by

$$\mathbb{E}(X_t \in \mathcal{K}) = \sum_{(a_i, a_{i'}) \in \mathcal{A}^2} \nu_{i, i'} \frac{|\mathcal{K}_i|}{N_i^{\sigma(a_{i'})}} \approx \frac{|\mathcal{K}|}{|\mathcal{X}|}. \quad (20)$$

The proposed approximation is deduced from the approximation of $N_i^{i'}$ given in Eqn. 4. It is valid if the probability distribution function $\mathbb{P}(x)$ over codewords is almost uniform (i.e. $\forall x, \mathbb{P}(x) \approx |\mathcal{X}|^{-1}$).

²which is equivalent to process, for each context σ_k , the permutation of symbols $a_{x_1}^{\sigma_k}$ and $a_{x_2}^{\sigma_k}$.

5.3 Maximum cardinality of the kernel

The cardinalities $N_i^{\sigma(a_{i'})}$ are first chosen in order to minimize the mdl given by Eqn. 3. Assuming that these values have already been found, one can then try to maximize the rate of *hard synchronization* points, i.e., to maximize the cardinality \mathcal{K} of the kernel. From Definition 1, we deduce that

$$\mathcal{K}_i = \mathcal{K} \bigcap_{a_{i'} \in \mathcal{A}} \mathcal{C}_i^{\sigma(a_{i'})}, \quad (21)$$

which leads to

$$|\mathcal{K}_i| \leq \min_{a_{i'} \in \mathcal{A}} N_i^{\sigma(a_{i'})}. \quad (22)$$

This means that, for each symbol value a_i of the high priority source \mathbf{S}_H , a maximum number of $\min_{a_{i'} \in \mathcal{A}} N_i^{\sigma(a_{i'})}$ codewords are assigned to classes $\mathcal{C}_i^{\sigma(a_{i'})}$, independently of the context values (or past symbol realizations). The equality can be obtained with some appropriate IA. In particular, we observed that the *crossed-IA* strategy with the SER criterion leads to maximize the kernel cardinality. The code kernel property can be used for random data access.

6 Soft decoding of multiplexed codes

Soft decoding techniques based on Bayesian estimation can be used to further improve the SER and SNR performances of multiplexed codes. The BCJR algorithm [1] directly applies provided that the forward and backward recursions are initialized in an appropriate manner as described below. This algorithm then allows to estimate the probabilities $\mathbb{P}(\mathbf{S}_H/\hat{S}_1, \dots, \hat{S}_{K_H})$ and $\mathbb{P}(S_t/\hat{S}_t)$ needed for deriving the symbol or sequence MAP or MMSE estimates.

6.1 Adaptation of the BCJR algorithm for multiplexed codes

Let us consider first the decoding of the high priority source which is assumed to be a Markov source. One could consider a joint decoding of both sources $(\mathbf{S}_H, \mathbf{S}_L)$, however at the expense of a prohibitive computational cost. Since \mathbf{S}_H defines a Markov process, a state of the trellis is fully identified by the source realization, as for the case of FLCs. Thus, the section of the trellis is of cardinality Ω . Therefore, unlike VLCs, an optimal³ trellis decoding based on the BCJR algorithm can be performed with a complexity in $O(K_H)$, which allows for soft decoding of very large sequences. A detailed analysis of this complexity is provided in section 6.1.2. The BCJR algorithm proceeds with the estimation of the probabilities $\mathbb{P}(S_t = a_i | Y_1; \dots, Y_K)$, knowing

³optimal in the sense that the so-called “excess-rate” is fully exploited: No pruning is done there.

- the Markov source transitions probabilities, i.e., $\nu_{i,i'} = \mathbb{P}(S_t = a_i | S_{t-1} = a_{i'})$,
- the probabilities $\mathbb{P}(X_t = x | S_{t-1} = a_{i'}; S_t = a_i)$, where X_t denotes the codeword emitted at the instant t corresponding to the symbol S_t . Since the distribution of the codewords of a given equivalence class is uniform, we have

$$\mathbb{P}(X_t = x | S_{t-1} = a_{i'}; S_t = a_i) = \begin{cases} \frac{1}{N_i^{\sigma(a_{i'})}} & \text{if } x \in \mathcal{C}_i^{\sigma(a_{i'})}, \\ 0 & \text{otherwise.} \end{cases} \quad (23)$$

- The channel transition probabilities $\mathbb{P}(Y_t = y | X_t = x)$, which are assumed to follow a DMC model.

In the following, the probabilities involved in the two recursions are denoted using the same notations as in [1], namely

$$\begin{aligned} \alpha_t(i) &= \mathbb{P}(S_t = a_i; Y_1 = y_1; \dots Y^t = y_t), \\ \beta_t(i) &= \mathbb{P}(Y_{t+1} = y_{t+1}; \dots Y_K = y_K | S_t = a_i), \\ \gamma_t(i', i) &= \mathbb{P}(S_t = a_i; Y_t = y_t | S_{t-1} = a_{i'}). \end{aligned} \quad (24)$$

The BCJR algorithm can also compute the probabilities $\mathbb{P}(\mathbf{s}_H / \hat{s}_1, \dots \hat{s}_{K_H})$. This probability is used by the MAP algorithm, but is not required for MPM (symbol MAP) and MMSE algorithms.

6.1.1 Initialization of forward and backward recursions

here, the initial and terminating states cannot be set to 0, as in [1]. In contrast, the probability $\alpha_1(m) = \mathbb{P}(S_t; Y_1)$, is computed from the DMC channel model and the stationary code used for the first symbol as

$$\begin{aligned} \alpha_1(i) &= \mathbb{P}(Y_1 = y_1 | S_1 = a_i) \mathbb{P}(S_1 = a_i) \\ &= \mu_i \sum_{x \in \mathcal{C}_{a_i}^{\sigma}} R(y_1 | x) \end{aligned} \quad (25)$$

Similarly, since for codes with an mdl close to the first-order entropy, the distribution of codewords is close to uniform, the probability $\beta_K(i)$, of each state i in the last section of the trellis, can be chosen as $\beta_K(i) = |\mathcal{X}|^{-1}$.

6.1.2 Complexity analysis

the computational cost of the BCJR algorithm applied to the decoding of the multiplexed codes is in $O(K_H)$ and can be expressed as $(C_m + C_a) K_H$, where C_m and C_a respectively denote the number of floating point multiplications and additions involved per symbol clock instant t . The factor C_m can be decomposed as $C_m = C_m(\gamma) + C_m(\alpha) + C_m(\beta) + C_m(\mathbb{P}(S_t | Y_1; \dots Y_{K_H}))$, where the variable $C_m(\square)$ denotes the computing cost per symbol

clock instant for each variable \square . The same decomposition applies to C_a . It is shown in annex 3 that the overall computational complexity is given by

$$C_m = \Omega(|\mathcal{X}| + 3\Omega + 1), \quad (26)$$

$$C_a = \Omega(|\mathcal{X}| + 2\Omega). \quad (27)$$

Note that, since floating numbers in machines are represented with finite precision, some re-normalization of probabilities α and β may be required, at the expense of a slight increase of the computational complexity. For example, the re-normalization processed such that $\forall t, \sum_{a_i \in \mathcal{A}} \alpha_t(i) = 1$ and $\forall t, \sum_{a_i \in \mathcal{A}} \beta_t(i) = 1$ leads to an increase of C_m and C_a by 2Ω . Note also that the computational cost above is slightly increased if the MAP algorithm is used, due to the additional estimation of $\mathbb{P}(S_{t-1} = m'; S_t = m; Y_1; \dots Y_{K_H})$ and the subsequent processing of the Viterbi algorithm [22].

6.2 Estimation criteria: MAP, MPM and MMSE

The MAP criterion (maximum *a posteriori*) corresponds to the optimal Bayesian estimation of a process S based on all available measurements Y : the MAP decoding problem consists then in computing

$$\hat{S} = \hat{S}_1 \dots \hat{S}_K = \arg \max_{s_1 \dots s_{K_H}} \mathbb{P}(S_1 = s_1; \dots S_{K_H} = s_{K_H} | Y_1 \dots Y_{K_H}). \quad (28)$$

where $s_1, \dots s_{K_H}$ are the realizations of the symbols $S_1, \dots S_{K_H}$ and take their values in the alphabet \mathbf{A} . The optimization is over all possible *sequences* s . For the SER criterion, the optimal sequence \hat{S} can be obtained by gathering local estimates \tilde{S}_t defined by

$$\tilde{S}_t = \arg \max_{a_i \in \mathcal{A}} \mathbb{P}(S_t = a_i | Y_1; \dots Y_{K_H}) \quad (29)$$

which are the MPM (maximum of posterior marginals) estimates of each hidden state of the Markov chain. However, if the mean square error (MSE) is the performance measure, the MPM criterion is sub-optimal. In this case, using marginal probabilities only (unlike sequence-based MMSE estimates, as in [16] [13]), the optimal decoder is the decoder that computes the conditional mean or minimum MSE (MMSE) as

$$\bar{S}_t = \sum_{a_i \in \mathcal{A}} \mathbb{P}(S_t = a_i | Y_1; \dots Y_{K_H}) \quad (30)$$

with $\mathbb{P}(S_t | Y_1, \dots Y_{K_H})$ sub-product of the BCJR algorithm. The decoder seeks a sequence of reproductions that will minimize the expected distortion, given the sequence of observations.

7 Hard and Soft Re-synchronization of contexts

The SER and SNR performances can be further improved at the expense of a slight decrease in compression efficiency, by encoding some symbols at known positions with a stationary

code rather than with a first-order code. The stationary codewords then act as *hard synchronization* points. One can also reduce the number of contexts in order to increase the source “excess-rate” which can then be exploited using the MMSE decoding algorithm. This method can be regarded as a soft synchronization technique. The last approach makes use of the kernel properties. At some a-priori known symbol clock instants, the encoder (resp. the decoder) are forced to represent the corresponding symbol with a codeword belonging to the kernel.

7.1 Error resilience analysis for finite length sequences

Let $\lambda_t(i, i')$ denote the probability $\mathbb{P}_t(S_t = a_i; \hat{S}_t = a_{i'})$ that the event $(S_t = a_i; \hat{S}_t = a_{i'})$ occurs at a given symbol clock instant t . Let $\boldsymbol{\lambda}_t$ be the vector of probabilities that the coder-decoder product model is in the different states of the state space \mathcal{A}^2 . For the first symbol, this vector is denoted $\boldsymbol{\lambda}_\emptyset$. Its components $\lambda_\emptyset(i, i')$ are given by the stationary pdf μ of the source \mathbf{S}_H .

$$\begin{aligned} \forall(i, i') / i = i', \lambda_\emptyset(i, i') = \mu_i, \\ \text{otherwise } \lambda_\emptyset(i, i') = 0. \end{aligned} \quad (31)$$

The first symbol can be encoded with a multiplexed code \mathcal{C}^\emptyset , constructed according to the stationary probability of the source \mathbf{S}_H ⁴. Let $\boldsymbol{\pi}_\emptyset$ denote the transition probability matrix of the coder-decoder product model associated to this code. For the symbol clock instant $t = 1$, the probability $\boldsymbol{\lambda}_1$ is given by $\boldsymbol{\pi}_\emptyset \boldsymbol{\lambda}_\emptyset$. Similarly, the general expression of the state for any given symbol clock instant t is given by

$$\boldsymbol{\lambda}_t = \boldsymbol{\pi}^{t-1} \boldsymbol{\pi}_\emptyset \boldsymbol{\lambda}_\emptyset. \quad (32)$$

Then, the SER of a sequence of K codewords is given by the average of symbol error rates over the sequence, i.e.

$$\text{SER}_{\mathcal{C}^*, K} = \frac{1}{K} \sum_{t=1}^K \boldsymbol{\delta}^T \boldsymbol{\lambda}_t \quad (33)$$

$$= \frac{1}{K} \boldsymbol{\delta}^T \left(\sum_{t=1}^K \boldsymbol{\pi}^{t-1} \right) \boldsymbol{\pi}_\emptyset \boldsymbol{\lambda}_\emptyset \quad (34)$$

Similarly, the average mean square error for the entire sequence is given by

$$\text{MSE}_{\mathcal{C}^*, K} = \frac{1}{K} \boldsymbol{\Delta}^T \left(\sum_{t=1}^K \boldsymbol{\pi}^{t-1} \right) \boldsymbol{\pi}_\emptyset \boldsymbol{\lambda}_\emptyset. \quad (35)$$

7.2 Synchronization using stationary multiplexed codes

In order to further reduce error propagation, one can encode symbols at a priori known positions⁵, with a stationary multiplexed code constructed from the stationary probability

⁴Using a specific code for the first symbol is obviously not mandatory. Note that, if the multiplexed code \mathcal{C}^\emptyset is used for the initialization, any given state vector $\boldsymbol{\lambda}$ that verifies $\|\boldsymbol{\lambda}\|_1 = 1$ can be chosen.

⁵equivalently, fixed bit positions

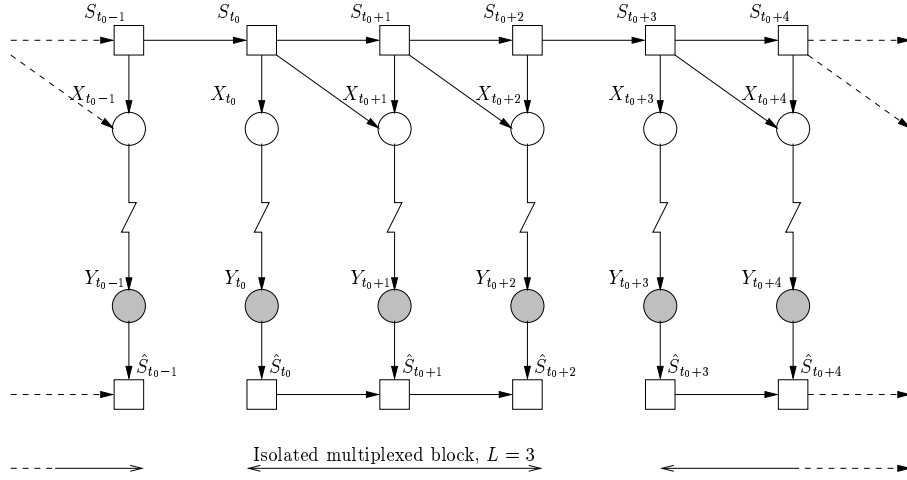


Figure 3: Hard synchronization of the decoder using the stationary multiplexed code \mathcal{C}^\emptyset every 3 symbols ($L = 3$). The decoded symbols \hat{S}_{t_0} and \hat{S}_{t_0+3} depend respectively only on the received codewords Y_{t_0} and Y_{t_0+3} .

μ . This code has already been introduced in section 7.1, and is referred to as \mathcal{C}^\emptyset . It asymptotically reaches, for high values of c , the stationary entropy of the source. Here, we restrict the analysis to the case where this code is used at symbol clock positions t such that $t \in L\mathbb{N}$, where $L \in \mathbb{N}^+ - \{0\}$. The corresponding code is referred to as $\mathcal{C}^{*,L}$. Every L symbols, the code \mathcal{C}^\emptyset acts as a *hard re-synchronization* point as depicted in figure 3. The rate of symbols encoded using the conditional probabilities is given by $\frac{L-1}{L}$, which leads to the following compression efficiency for $\mathcal{C}^{*,L}$:

$$\text{mdl}(\mathcal{C}^{*,L}) = \frac{1}{L} (\text{mdl}(\mathcal{C}^\emptyset) + (L-1) \text{mdl}(\mathcal{C}^*)) \quad (36)$$

Since some storage capacity can be lost during the conversion of \mathbf{S}_L into a finite sequence of states \mathbf{q} , the conversion of the source \mathbf{S}_L into a sequence of states is still processed for the whole sequence, according to the values N_t deduced from the code $\mathcal{C}^{*,L}$. Now, let us recall that the zero order and first order entropies of the source \mathbf{S}_H are respectively denoted h^0 and h^1 . If we assume that the codeword length c allows for a good precision in the stationary and conditional pdf approximations, the compression efficiency closely verifies

$$\text{mdl}(\mathcal{C}^{*,L}) = \frac{h^0 + (L-1)h^1}{L}. \quad (37)$$

Each codeword block of length L can be seen as a sequence of multiplexed codewords of finite length L : the vector λ_\emptyset provides the probabilities to observe the event ($S_t =$

$a_i) \wedge (\hat{S}_t = a_{i'})$ for symbols encoded/decoded with \mathcal{C}^\emptyset . Similarly, the vector λ_t gives the probabilities for all the symbol clock instant t' such that $t = t' \bmod L$. Thus, the SER and MSE expressions for the code $\mathcal{C}^{*,L}$ are directly given by, respectively, Eqn. 34 and Eqn. 35 with $K = L$.

Soft decoding methods also apply when a stationary multiplexed code is periodically used. Only one slight modification of the BCJR algorithm is required. For symbol clock instants for which such a code is used, the probability $\alpha_t(m)$ is computed using the initialization of Eqn. 25.

7.3 Synchronization using codewords belonging to the kernel

The codewords belonging to the kernel provide naturally (and for free) decoder re-synchronization capabilities, by limiting error propagation due to erroneous contexts. However, one can in addition constrain the use of kernel codewords in order to further increase the decoder re-synchronization capability. Obviously, this constraint for a symbol clock instant t has a cost in terms of mdl that depends on the number of codewords belonging to the kernel. The mdl for such a codeword is thus given by

$$\sum_{a_i \in \mathcal{X}} -\mu_i \log_2 \left(\frac{|\mathcal{K}_i|}{|\mathcal{X}|} \right), \quad (38)$$

instead of being given by Eqn. 3. If the decoder receives a symbol that does not belong to the kernel for pre-defined symbol clock instants, an error is detected. This a-priori information can be exploited by the soft decoding algorithms. However, we observed that the trade-off between compression efficiency and error resilience was not as good as the one obtained by a periodic use of stationary multiplexed codes (see Fig. 6).

8 Simulation Results

The SER and SNR performance of first-order multiplexed codes has been evaluated considering order-1 Gauss-Markov sources, with zero-mean, unit-variance and different correlation factors ($\rho = 0.5$ and $\rho = 0.9$). The source is quantized with an 8 levels uniform quantizer. All the simulations have been performed assuming a Binary Symmetric Channel. All the experiments reported in this section have been performed with sequences of 1000 symbols and the results are averaged over 1000 channel realizations. The first-order entropies of the two sources of correlation factors $\rho = 0.5$ and $\rho = 0.9$ are respectively $h^1 = 2.29$ and $h^1 = 1.52$. Their stationary entropy is $h^0 = 2.47$. The curves given here show the performance obtained for the high priority source (\mathbf{S}_H). The low priority source is supposed to be pre-encoded with a classical arithmetic code.

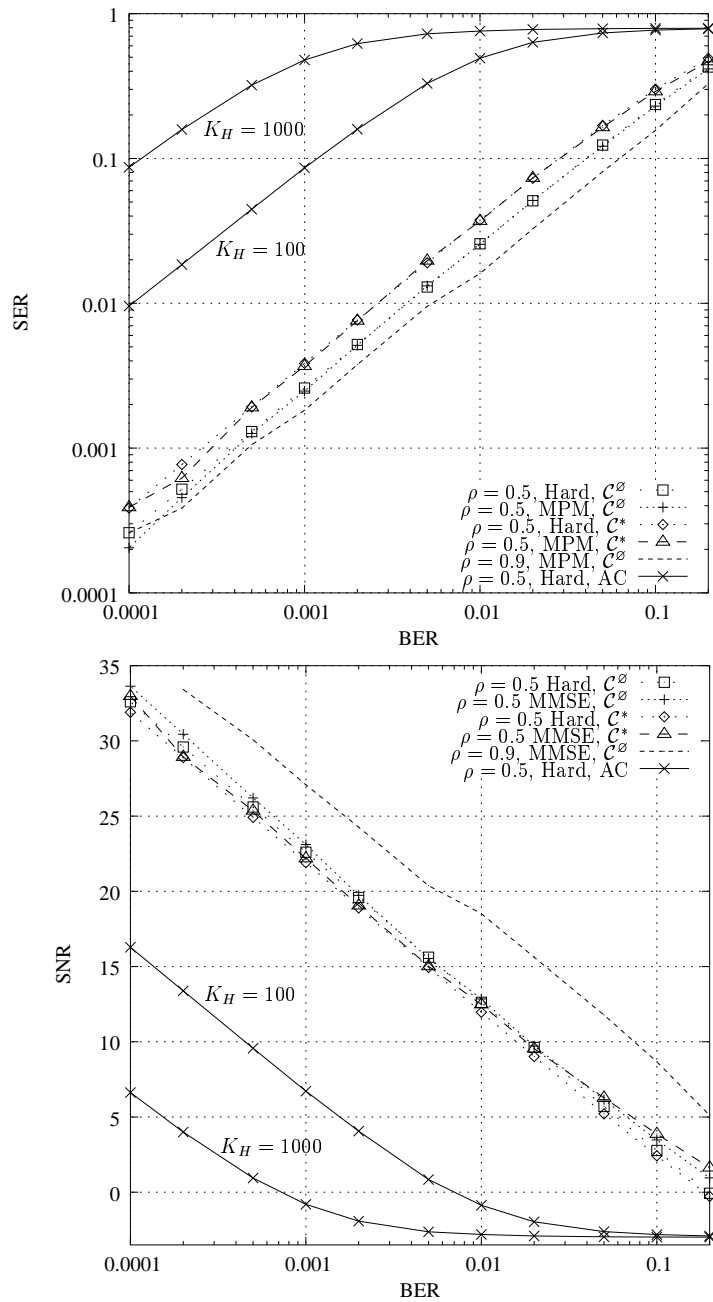


Figure 4: SER and SNR performance of stationary and first-order multiplexed codes (respectively constructed with SA and crossed IA+SA index assignments, in comparison with an arithmetic coder (AC).) with Hard, MPM and MMSE decoding techniques.

8.1 SER and SNR performance of stationary and first-order multiplexed codes

The first set of experiments aimed at showing the respective performance of stationary and first-order multiplexed codes in presence of bit errors and with different decoding methods, as well as showing how they compare with classical codes (i.e. Huffman and arithmetic codes). We consider stationary and first-order multiplexed codes with a codeword length $c = 5$ and constructed with respectively an index assignment based on the SA and crossed-IA + SA techniques. The mean description lengths obtained with the stationary and first-order multiplexed codes are respectively given by $mdl = 2.50$ and $mdl = 2.35$ for the source with a correlation factor of $\rho = 0.5$. For a correlation factor of $\rho = 0.9$, the first-order multiplexed code leads to a mean description length of $mdl = 1.70$.

Fig. 4 shows the SER and SNR curves obtained with hard decoding, MPM and MMSE decoding of stationary (one context) and first-order (with 8 contexts) multiplexed codes. Notice that the first-order multiplexed codes provide higher compression efficiency for a small loss in terms of error resilience (the SNR performance for a range of bit error rates from 10^{-4} to 10^{-1} are close for both codes). These curves show the strong advantage of first-order multiplexed codes versus arithmetic codes in presence of bit errors for similar compression efficiency. As expected, when the source correlation is low ($\rho \leq 0.5$), the gain brought by soft decoding techniques with respect to hard decoding is small. In particular, similar results have been obtained with the MAP and MPM estimation criteria. However, for sources with high correlation, MPM and MMSE decoding allow to improve respectively the SER and SNR performances.

8.2 Respective SER and SNR performance of multiplexed and arithmetic codes in a classical tandem source-channel coding scheme

A second set of experiments aimed at comparing the performance of multiplexed and arithmetic codes when used in classical tandem source-channel coding chain. The arithmetic source coder allows also to exploit higher-order source statistics. The source coders are followed by a rate $\frac{1}{2}$ convolutional code for protection and error correction. The overall rates (source+channel) of both chains are similar. Two constraint lengths have been considered for the convolutional code: 5 and 9. Here, to have a comparison as fair as possible, we assume that the source multiplexed codes have been designed without knowing the channel characteristics (the bit error rate). In particular, a simple lexicographical IA strategy is used instead of the more efficient crossed-IA+SA strategy which is adapted to the channel characteristics. We consider a source with a correlation factor $\rho = 0.5$. For both schemes, hard source decoding has been considered. Notice that, in both chains, soft outputs of the ECC could be used to improve the source decoder performance.

The SER and SNR values obtained are shown in Fig. 5. It clearly appears that, in contrast with arithmetic codes, multiplexed codes do not suffer from dramatic de-synchronization problems if some residual bit errors occur, resulting in significantly higher SNR and SER performances. Multiplexed codes do not suffer from dramatic de-synchronization problems

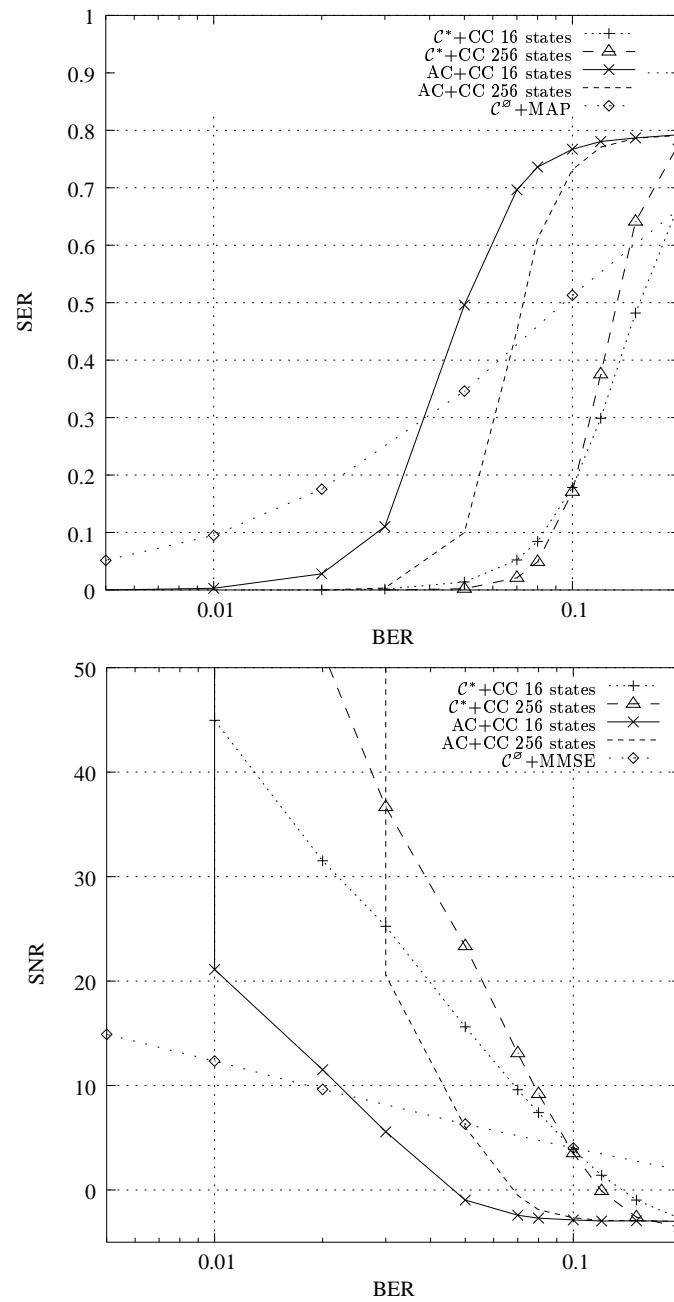


Figure 5: Respective SER and SNR performances of multiplexed and arithmetic codes when used in a classical tandem source-channel coding chain. The multiplexed code considered relies on a lexicographical IA. The correlation factor of the source is $\rho = 0.9$.

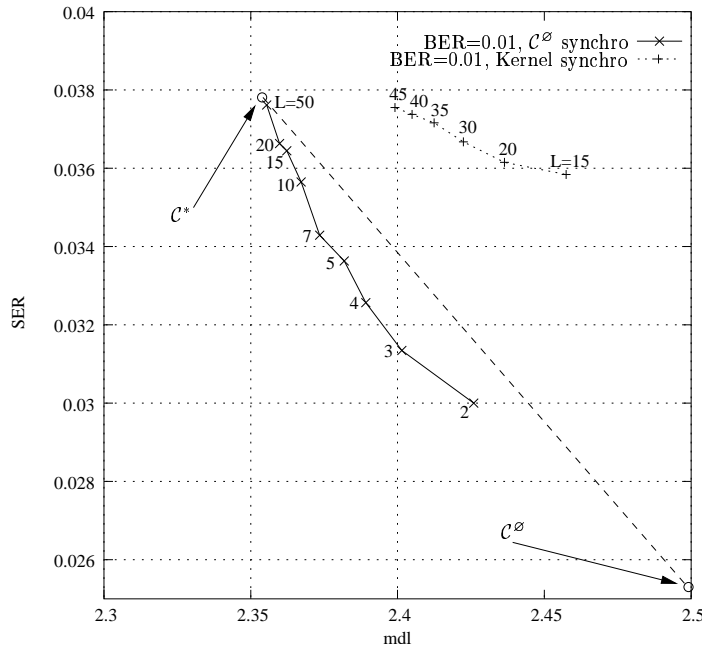


Figure 6: SER-mdl trade-off of two synchronization schemes: use of stationary codewords or of codewords belonging to the kernel every L symbols. The source considered is a Gauss-Markov source with a correlation factor of $\rho = 0.5$. The circular points are those obtained with respectively the stationary code \mathcal{C}^\emptyset and the first-order multiplexed code \mathcal{C}^* (without redundancy).

if some residual bit errors occur. They indeed appear to be an efficient way to give some “free” protection, i.e. at no cost in terms of compression efficiency. Notice that, when used in a classical tandem source-channel coding chain, the error correcting codes apply to both the high and the low priority source. When the two sources are encoded separately (as in the chain based on classical arithmetic codes), one has the additional freedom to use different channel codes for the two sources in the direction of unequal error protection. However, this requires the channel properties to be known at the time of encoding. The multiplexed codes offer an inherent unequal protection of both sources without requiring the channel properties to be known. Fig. 5 also shows the gain obtained with first-order multiplexed codes with respect to stationary codes (\mathcal{C}^\emptyset).

8.3 Synchronization of the codes

A third set of experiments aimed at evaluating the performance of two synchronization techniques: a use of stationary multiplexed codewords and of codewords belonging to the kernel every L symbols. Fig. 6 illustrates the corresponding mdl-SER trade-off. The decoding algorithm used is based on the soft decoding technique described above with the MPM estimation criterion. The synchronization strategy based on a constrained and periodic insertion of codewords belonging to the kernel leads to a slight decrease of the SER, however at the expense of a prohibitive cost in terms of mdl. In contrast, the use of a stationary code is clearly an efficient way to decrease the SER with a low cost in terms of mdl. The corresponding SER curve is convex and located below the dotted line curve linking the mdl-SER points of the stationary (\mathcal{C}^\emptyset) and first-order (\mathcal{C}^*) multiplexed codes. This curve corresponds to the case where part of the sequence would be coded with stationary codes and the rest with first-order codes in a proportion that varies along the mdl axis. The fact that the SER curve is below the dotted line curve evidences the effect of the periodic use of stationary codes on the resynchronization capability of the decoder.

9 Conclusion

In this paper, we have described a new family of codes that allow to exploit higher-order source statistics while still being error-resilient. The design principle is based on the assumption that coding systems generate sources of different priority, which has then to be protected differently. Performance bounds in terms of SER and MSE are given. With respect to classical source codes (i.e. Huffman and arithmetic codes), the stationary and first-order multiplexed codes are shown to have significantly higher error-resilience (i.e. higher SER and SNR performance in presence of bit errors) for the same compression efficiency, even with a simple lexicographical index assignment. However, the index assignment (IA) having an impact on the resulting efficiency of the codes, different IA strategies have been described. It is shown in particular that a significant SER and SNR performance gain can be obtained with an IA referred to as *crossed*-index assignment that takes into account the channel characteristics. A subset of codewords, the *kernel* of the code, is also shown to have properties allowing for improved error-resilience and random data access. Soft decoding principles with different estimation criteria (MAP, MPM and MMSE) are shown to apply in a rather straightforward (MAP, MPM and MMSE) to multiplexed codes, with a complexity which varies linearly with the length of the sequence of symbols to be decoded. Several techniques based on the code properties are then described to further improve the decoder re-synchronization capability. These codes, due to their high error resilience at almost no cost in terms of compression in comparison with arithmetic codes turn out to be very good candidates for future wireless multimedia communication systems.

References

- [1] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Inform. Theory*, pages 284–287, Mar. 1974.
- [2] R. Bauer and J. Hagenauer. Turbo fec/vlc decoding and its application to text compression. In *Proc. Conf. Inform. Theory and Systems*, pages WA6.6–WA6.11, Mar. 2000.
- [3] N. Demir and K. Sayood. Joint source-channel coding for variable length codes. In *Proc. IEEE DCC'98*, pages 139–148, Mar. 1998.
- [4] N. Farvadin. A study of vector quantization for noisy channels. *IEEE Trans. Inform. Theory*, 36(4):799–809, July 1990.
- [5] N. Farvadin and Vinay Vaishampayan. On the performance and complexity of channel-optimized vector quantizers. *IEEE Trans. Inform. Theory*, 37(1):155–160, Jan. 1991.
- [6] T.J. Ferguson and J. H. Rabinowitz. Self-synchronizing huffman codes. *IEEE Trans. Inform. Theory*, IT-30(4):687–693, July 1984.
- [7] A. Guyader, E. Fabre, C. Guillemot, and M. Robert. Joint source-channel turbo decoding of entropy coded sources. *IEEE J. Select. Areas Commun.*, 19(9):1680–1696, Sept. 2001.
- [8] J. Hagenauer. Rate-compatible punctured convolutional codes (rcpc codes) and their applications. *IEEE Trans. Commun.*, 36(Apr.):389–400, Apr. 1988.
- [9] H. Jégou and C. Guillemot. Source multiplexed codes for error-prone channels. In *Proc. IEEE ICC'03*, May 2003.
- [10] H. Jégou and C. Guillemot. Variable length codes over noisy channels: transmission framework and self-multiplexed codes. *submitted to IEEE Trans. Commun.*, Apr. 2003.
- [11] Hervé Jégou and Christine Guillemot. A new class of codes for transmission of heterogeneous data: Multiplexed codes, Aug. 2003. INRIA Research Report 4922.
- [12] A.Gersho K. Zeger. Pseudo-gray coding. *IEEE Trans. Commun.*, 38(12):2147–2158, Dec. 1990.
- [13] F. Lahouti and A. K. Khandani. An efficient mmse source decoder for noisy channels. In *Proc. International Symposium on Telecommunications*, pages 784–787, 9 2001. Teheran, Iran.
- [14] W.M. Lam and A.R. Reibman. Self-synchronizing variable-length codes for image transmission. In *Proc. IEEE ICASSP'92*, volume Mar., pages 477–480, Sept. 1992.

-
- [15] J.C. Maxted and J.P. Robinson. Error recovery for variables length codes. *IEEE Trans. Inform. Theory*, IT-31(6):794–801, Nov. 1985.
 - [16] D. J. Miller and M. Park. A sequence-based approximate mmse decoder for source coding over noisy channels using discrete hidden markov models. *IEEE Trans. Commun.*, 46(2):222–231, 1998.
 - [17] A.H. Murad and T.E. Fuja. Joint source-channel decoding of variable length encoded sources. In *Proc. Inform. Theory Workshop*, pages 94–95, June 1998.
 - [18] B.D. Pettijohn, K. Sayood, and M.W. Hoffman. Joint source/channel coding using arithmetic codes. In *Proc. IEEE DCC'2000*, Mar. 2000. Snowbird, Utah.
 - [19] D.W. Redmill and N.G. Kingsbury. The errec: An error resilient technique for coding variable-length blocks of data. *IEEE Trans. Image Processing*, 5:565–574, Apr. 1996.
 - [20] K. Sayood and J.C Borkehagen. Use of residual redundancy in the design of joint source/channel coders. *IEEE Trans. Commun.*, 39(6):838–846, June 1991.
 - [21] Y. Takishima, M. Wada, and H. Murakami. Reversible variable length codes. *IEEE Trans. Commun.*, 43(2/3/4):158–162, Feb. 1995.
 - [22] A. Viterbi. Error bounds for convolution codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theory*, (13):260–269, 1967.
 - [23] J. Wen and J. D. Villasenor. Reversible variable length codes for efficient and robust image and video coding. In *Proc. IEEE DCC'98*, pages 471–480, Apr. 1998.

Appendix 1: Transition probability $\pi(z, z')$ of the coder-decoder product model

Given that $\mathbb{P}(S_t = a_i; X_t | \hat{S}_{t-1} = a_{j'}; S_{t-1} = a_{i'}) = \mathbb{P}(S_t = a_i; X_t | S_{t-1} = a_{i'})$, the first term on the right side of Eqn. 9 can be rewritten as

$$\mathbb{P}(S_t = a_i; X_t | \hat{S}_{t-1} = a_{j'}; S_{t-1} = a_{i'}) = \mathbb{P}(X_t | S_t = a_i; S_{t-1} = a_{i'}) \nu_{a_i, a_{i'}} \quad (39)$$

In addition, for given symbol realization $S_t = a_i$ and $S_{t-1} = a_{i'}$, the multiplexed codeword X_t is such that $X_t \in \mathcal{C}_i^{\sigma(a_{i'})}$. Therefore, assuming that all the codewords belonging to a class of equivalence have the same probability (which is valid if the low priority source is compressed and converted into a sequence of states q_t in a quasi-optimal manner, e.g. using an arithmetic coder)

$$\forall X_t \in \mathcal{C}_i^{\sigma(a_{i'})}, \mathbb{P}(X_t | S_t = a_i; S_{t-1} = a_{i'}) = \mathbb{P}(X_t | X_t \in \mathcal{C}_i^{\sigma(a_{i'})}) = \frac{1}{N_i^{\sigma(a_{i'})}}, \quad (40)$$

$$\forall X_t \notin \mathcal{C}_i^{\sigma(a_{i'})}, \mathbb{P}(X_t | S_t = i; S_{t-1} = a_{i'}) = 0. \quad (41)$$

From Eqn. 39 and Eqn. 40 we deduce

$$\forall X_t \in \mathcal{C}_i^{\sigma(a_{i'})}, \mathbb{P}(S_t = a_i; X_t | \hat{S}_{t-1} = a_{j'}; S_{t-1} = a_{i'}) = \frac{\nu_{a_i, a_{i'}}}{N_i^{\sigma(a_{i'})}} \quad (42)$$

The other probability of Eqn. 9 can be processed as follows, starting with the following simplification:

$$\mathbb{P}(\hat{S}_t = a_j; Y_t | S_t = a_i; X_t; \hat{S}_{t-1} = a_{j'}; S_{t-1} = a_{i'}) = \mathbb{P}(\hat{S}_t = a_j; Y_t | X_t; \hat{S}_{t-1} = a_{j'}). \quad (43)$$

Since for a given symbol value ($\hat{S}_t = a_j$) associated to a multiplexed codeword $Y_t \in \mathcal{C}_j^{\sigma(a_{j'})}$, one has the previous symbol value $\hat{S}_{t-1} = a_{j'}$,

$$\forall Y_t \in \mathcal{C}_j^{\sigma(a_{j'})}, \mathbb{P}(\hat{S}_t = a_j; Y_t | X_t; \hat{S}_{t-1} = a_{j'}) = \mathbb{P}(Y_t | X_t; S_{t-1} = a_{j'}) = \mathbb{P}(Y_t | X_t) = R(Y_t | X_t) \quad (44)$$

$$\forall Y_t \notin \mathcal{C}_j^{\sigma(a_{j'})}, \mathbb{P}(\hat{S}_t = a_j; Y_t | X_t; \hat{S}_{t-1} = a_{j'}) = 0 \quad (45)$$

Then from Eqn. 40, Eqn. 41, Eqn. 44 and Eqn. 45, one can finally deduce

$$\pi(z, z') = \frac{\nu_{a_i, a_{i'}}}{N_i^{\sigma(a_{i'})}} \sum_{(X_t, Y_t) \in \mathcal{C}_i^{\sigma(a_{i'})} \times \mathcal{C}_j^{\sigma(a_{j'})}} R(Y_t | X_t) \quad (46)$$

□

Appendix 2: Number N_P of partitions of \mathcal{X} that verify cardinalities constraints

Let us denote N_1, \dots, N_Ω the cardinalities of equivalence classes $\mathcal{C}_1, \dots, \mathcal{C}_\Omega$ deduced from the mean description length optimization. For the first equivalence class, there are $\binom{|\mathcal{X}|}{N_1}$ choices of N_1 codewords among $|\mathcal{X}|$. For the i^{th} equivalences, N_i codewords have to be chosen among the $|\mathcal{X}| - \sum_{k=1}^{i-1} N_k$ remaining codewords. Thus, the number of partition is given by

$$N_P = \prod_{i=1}^{\Omega} \binom{|\mathcal{X}| - \sum_{k=1}^{i-1} N_k}{N_i}. \quad (47)$$

Assuming that all the codewords of \mathcal{X} are assigned to an equivalence class, we have $|\mathcal{X}| - \sum_{k=1}^{\Omega-1} N_k = N_\Omega$. With $\binom{n}{k} = \frac{n!}{p!(n-p)!}$, Eqn. 47 leads to

$$N_P = \frac{|\mathcal{X}|!}{N_1!(|\mathcal{X}| - N_1)!} \frac{(|\mathcal{X}| - N_1)!}{N_2!(|\mathcal{X}| - N_1 - N_2)!} \dots \frac{(N_\Omega)!}{N_\Omega!0!}, \quad (48)$$

which can be simplified as

$$N_P = \frac{|\mathcal{X}|!}{\prod_{i=1}^{\Omega} N_i!}. \quad (49)$$

□

Appendix 3: Computational cost of the BCJR algorithm applied to multiplexed codes

To begin with, the probability $\gamma_t(i', i)$ is computed as

$$\gamma_t(i', i) = \nu_{i,i'} \sum_{x \in \mathcal{X}} \mathbb{P}(X_t = x | S_{t-1} = a_{i'}; S_t = a_i) R(Y_t | x). \quad (50)$$

According to Eqn. 23, the probability $\mathbb{P}(X_t = x | S_{t-1} = a_{i'}; S_t = a_i)$ is not null if and only if $x \in \mathcal{C}_i^{\sigma(i')}$. This allows to restrict the summation to the set of codewords $\mathcal{C}_i^{\sigma(i')}$ in Eqn. 50. The cardinal of this equivalence class being $N_i^{\sigma(i')}$, the numbers of multiplications and additions involved in the calculation of $\gamma_t(i', i)$ for each tuple $(a_i, a_{i'}) \in \mathcal{A}^2$ are respectively given⁶ by $N_i^{\sigma(i')} + 1$ and $N_i^{\sigma(i')}$. Since $\forall a_{i'}, \sum_{a_i \in \mathcal{A}} N_i^{\sigma(a_{i'})} = |\mathcal{X}|$, the numbers of multiplications and additions for the variable γ are given by

$$C_m(\gamma) = (|\mathcal{X}| + \Omega) \Omega, \quad C_a(\gamma) = |\mathcal{X}| \Omega. \quad (51)$$

and by

$$\begin{aligned} C_m(\alpha) &= \Omega^2, & C_m(\beta) &= \Omega^2, & C_m(\mathbb{P}(S_t | Y_1; \dots Y_K)) &= \Omega, \\ C_a(\alpha) &= \Omega^2, & C_a(\beta) &= \Omega^2, & C_a(\mathbb{P}(S_t | Y_1; \dots Y_K)) &= 0, \end{aligned} \quad (52)$$

for α , β and λ respectively. The overall computational complexity is then

$$C_m = \Omega (|\mathcal{X}| + 3\Omega + 1), \quad (53)$$

$$C_a = \Omega (|\mathcal{X}| + 2\Omega). \quad (54)$$

□

⁶In fact, an addition can be avoided with an appropriate implementation.



Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
<http://www.inria.fr>
ISSN 0249-6399