DECODAGE PONDERE DES CODES CONCATENES AVEC CODE INTERIEUR CONVOLUTIF

G. Battail, G. Kawas-Kaleh, J.-C. Belfiore et R. Sfez

Ecole nationale supérieure des Télécommunications, Dépt SYC,
46 Rue Barrault, 75634 PARIS CEDEX 13

RESUME

On examine le décodage de codes concaténés, notamment avec un code intérieur convolutif. En supposant les codes constituants décodés l'un après l'autre, le décodage est globalement amélioré si la sortie du décodeur intérieur est pondérée. L'algorithme de Viterbi peut être modifié à cet effet. En élargissant la tâche du décodage à la détermination d'une distribution a posteriori sur les mots du code, étant donnée la distribution a priori des symboles en sortie du canal, on déduit de la règle de Bayes une formule générale mais complexe. Le principe de Kullback, pour une distribution a posteriori séparable par rapport aux symboles d'information, conduit à un système d'équations implicites qui est un cas particulier du résultat précédent. L'interprétation d'un code linéaire unique comme une sorte de produit de ses contrôles de parité aboutit à un algorithme par étapes où chacun de ces contrôles est décodé avec sortie pondérée par résolution de ce système. En revanche, avec le décodage séquentiel, le décodage du code concaténé global n'est pas plus compliqué que celui du code convolutif seul.

SUMMARY

This paper is intended to concatenated codes decoding, especially for a convolutional inner code. Assuming successive decoding of the associated codes, overall decoding is improved by weighting the inner decoder output. The Viterbi algorithm can be modified to this end. Redefining the part of decoding to determine a posterior probability distribution on the codewords, given the prior distribution of the symbols, Bayes rule results in a general, but complex, formula. Kullback principle, for a posterior distribution separable with respect to the information symbols, results in a system of implicit equations which is a particular case of the previous result. Interpreting a single linear code as a kind of product of its parity-checks results in a step-by-step algorithm where each check is decoded with weighted output by solving this system. When using sequential decoding, however, the overall decoding is no more complex than that of the convolutional code alone.

1 – INTRODUCTION

La concaténation de deux codes redondants est un procédé (fig. 1) qui réduit la complexité du décodage par rapport à celle d'un code unique, à probabilité d'erreur résiduelle et redondance égales [1]. On peut interpréter cette propriété en remarquant que, même avec un canal stationnaire, le résultat d'un décodage se présente sous la forme d'une suite de paquets d'erreurs: si le décodeur réussit le plus souvent à corriger les erreurs dues au canal, il substitue au mot correct un autre mot du code, qui en est au moins à la distance de Hamming minimale, lorsqu'il échoue; on a donc affaire à des erreurs groupées. Si l'on emploie un deuxième code spécialisé dans la correction des paquets d'erreurs (dit extérieur, le premier étant dit intérieur), on conçoit que l'on puisse atteindre des probabilités d'erreur très basses. Avec un code unique, le même résultat ne pourrait être obtenu qu'au prix d'une complexité supérieure.

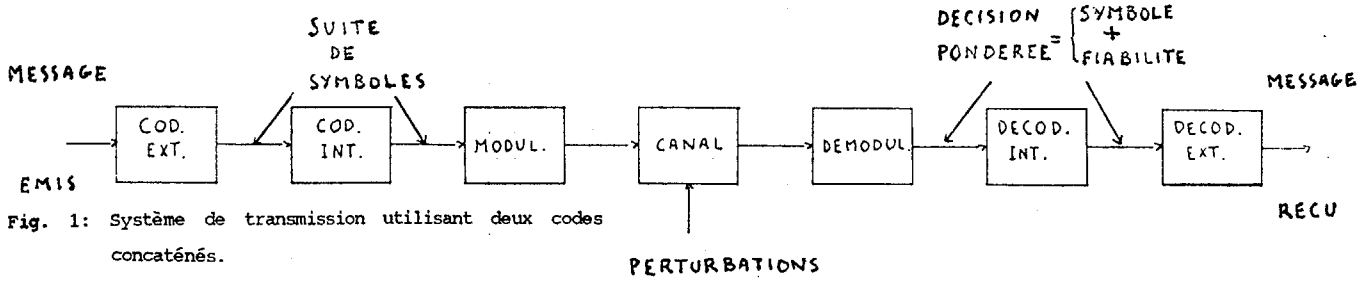

Fig. 1: Système de transmission utilisant deux codes concaténés.

2 – EXEMPLE DE DECODAGE DU CODE INTERIEUR AVEC SORTIE PONDEREE

2.1 – Code intérieur convolutif concaténé à un code de Reed-Solomon

Un schéma de concaténation efficace utilise pour code intérieur un code convolutif binaire de faible longueur de contrainte, décodé par l'algorithme de Viterbi (optimal et pouvant exploiter la pondération disponible en sortie d'un démodulateur) et pour code extérieur un code de Reed-Solomon construit sur une extension du corps binaire, spécialisé dans la correction de courts paquets d'erreurs puisque ses symboles sont représentés par des blocs de symboles binaires [2-4]. Un entrelacement permet d'adapter la taille des paquets d'erreurs résiduelles à celle des symboles du code de Reed-Solomon.

2.2 – Pondération en sortie du décodeur de Viterbi

L'un des auteurs a cherché à déduire du fonctionnement du décodeur intérieur une pondération utilisable pour améliorer le décodage extérieur. Il a proposé pour cela une modification de l'algorithme de Viterbi, relativement simple mais non optimale [5].

Nous allons décrire cette modification dans le cas d'un code convolutif binaire où un unique symbole de contrôle est associé à chacun des symboles d'information, c'est-à-dire de taux 1/2. La fig. 2 représente l'exemple classique d'un codeur engendrant un tel code. Ici comme dans la suite, nous supposons le canal sans mémoire et donc les perturbations sur les symboles reçus indépendantes. Nous supposons disponible à l'entrée du décodeur une grandeur réelle a correspondant à la réception de chaque symbole binaire b (bit), dite sa "valeur relative" (v.r.) a priori, telle que son signe représente la meilleure décision b* quant à b (+ pour 0 et – pour 1) et que son module en mesure la fiabilité. Elle est définie ainsi:

$$a = \log \frac{\text{Prob}\,(b=0)}{\text{Prob}\,(b=1)} = (-1)^{b^*} \log \frac{\text{Prob}\,(b^*=b)}{\text{Prob}\,(b^* \neq b)}; \qquad (1)$$

les probabilités s'entendent conditionnellement aux signaux reçus.

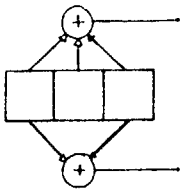

Fig. 2: Exemple de codeur convolutif.

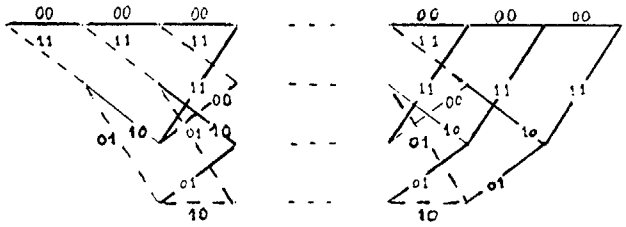

Fig. 3: Diagramme du treillis associé au codeur de la fig. 2.

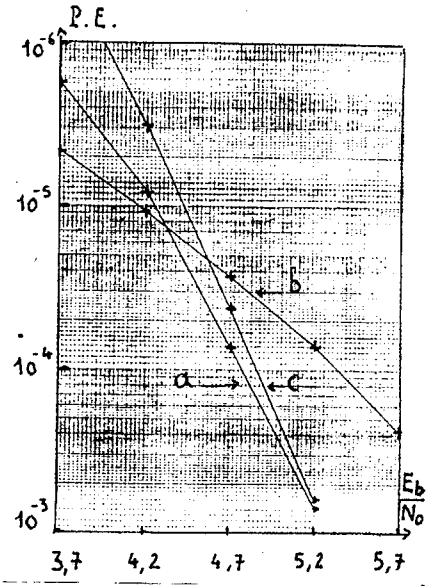

Fig. 4: Résultats du décodage d'un code concaténé (probabilité d'erreur en fonction du rapport de l'énergie reçue par symbole binaire d'information à la densité spectrale unilatérale du bruit, supposé additif gaussien et blanc):

a) algorithme de Viterbi à sortie pondérée (codeur intérieur de la fig. 2, codeur extérieur de Reed-Solomon (7,3) sur le corps à 8 éléments);

b) décodage séquentiel pour le même code;

c) borne supérieure calculée de la probabilité d'erreur a).

Ici et dans la suite, la base des logarithmes utilisés est arbitraire, pourvu qu'elle soit aussi celle des exponentielles.

L'algorithme de Viterbi étant inchangé en ce qui concerne le calcul de la "métrique" et le choix du meilleur chemin dans le diagramme du treillis associé au codeur (fig. 3), la modification permettant d'en pondérer les sorties consiste à:

a) remplacer la décision ferme sur un symbole d'information prise en chaque noeud par le calcul d'une v.r. a posteriori provisoire égale à la différence des métriques des chemins y convergeant, mise en mémoire dans un registre particulier à ce noeud (au lieu de décisions fermes dans la mise en oeuvre habituelle de l'algorithme); la longueur de ce registre, infinie en principe, peut être raccourcie à 4 ou 5 fois la longueur de contrainte du codeur sans dégradation notable;

b) remplacer le transfert décalé du contenu du registre associé au noeud situé sur le meilleur chemin par un calcul qui tient compte des v.r. a posteriori mémorisées dans les registres des deux noeuds antécédents, en les traitant comme conditionnelles à la décision qui vient d'être prise. La v.r. a posteriori ainsi révisée doit être prise égale à

$$X = \log \frac{\exp(y) + \exp(x+y) + \exp(x+z) + \exp(x+y+z)}{1 + \exp(x) + \exp(z) + \exp(y+z)} \qquad (2)$$

où z est la dernière v.r. déterminée selon a) et où x et y sont les v.r. d'un même symbole présentes dans les registres associés respectivement aux antécédents correspondant aux choix de 0 et de 1 lors de la dernière décision. On peut utiliser l'approximation suivante de cette fonction:

$$X = \max(y, x+y, x+z, x+y+z) - \max(0, x, z, y+z); \qquad (3)$$

bien que grossière, elle paraît suffire en pratique et évite en outre les difficultés de calcul dues à la forte variation de la fonction exponentielle. Dans le cas où des v.r. de grand module et de signe contraire sont mémorisées dans l'étage de même rang des registres des noeuds antécédents, l'opération (2) (ou (3)) rend si nécessaire le module de la v.r. conservée inférieure à la marge avec laquelle la dernière décision est prise. Ainsi est résolue la contradiction possible entre deux estimations antérieures alors que le choix entre elles, par la dernière décision, est indécis.

La simulation du décodage concaténé utilisant cette pondération met en évidence une nette amélioration par rapport au cas où le décodage extérieur n'est pas pondéré, toutes choses égales par ailleurs (fig. 4). L'entrelacement s'avère plus efficace par bit que par symbole, et l'analyse de ces résultats remet partiellement en question l'interprétation simple du § 1.

3 – REDEFINITION DE LA FONCTION DU DECODAGE

D'une façon plus générale, on peut assigner au décodage des codes redondants la tâche de réévaluer la distribution de probabilité des mots pour tenir compte des contraintes du code, connaissant celle des symboles reçus [6] (au lieu de seulement désigner le mot le plus probable a posteriori).

3.1 – Utilisation de la règle de Bayes

L'un des auteurs, supposant les symboles d'information équiprobables avant la transmission, a calculé à cet effet par la règle de Bayes la probabilité de chacun des symboles d'information, conditionnellement aux autres et aux signaux reçus, en tenant compte des contraintes du code. Ce calcul a pu être mené à bien, dans le cas binaire, pour les codes convolutifs de taux 1/2 [7], ainsi que pour les codes linéaires en blocs (n,k). Dans ce dernier cas, il a pour résultat la v.r. a posteriori A_i du i-ème symbole du mot (définie par (1) mais avec les probabilités a posteriori):

$$A_i = \log \frac{1 + t(A_i)}{1 - t(A_i)}, \text{ avec} \qquad (4)$$

$$t(A_i) = \sum_{\underline{u}^{(i)}} t\Big[\sum_{j=1}^{n} g_i^j a_j (-1)^{\sum_{m=1, m\neq i}^{k} g_m^j u_m}\Big] \text{Prob}[\underline{u}^{(i)}], \qquad (5)$$

où $\underline{u}^{(i)}$ est le vecteur des symboles d'information autres que le i-ème, où la somme est effectuée pour tous les 2^{k-1} vecteurs $\underline{u}^{(i)}$ possibles et où t(.) est la fonction inverse de (4), soit

$$t(.) = [\exp(.) - 1]/[\exp(.) + 1]. \qquad (6)$$

La matrice génératrice G du code a pour élément général g_i^j, où i = 1, ... k est l'indice de la ligne et j = 1, ... n celui de la colonne.

Cette formule est générale mais très complexe, puisqu'elle implique de connaître la distribution de probabilité a posteriori des vecteurs $\underline{u}^{(i)}$, donc des 2^k mots du code. Celle-ci est déduite de la distribution a priori des n-uples en ne retenant que les probabilités des mots du code et en les normalisant. Puisque le canal est supposé sans mémoire, la probabilité a priori d'un n-uple est le produit des probabilités de chacun de ses bits, déduites de (1).

La formule homologue de (5) dans le cas d'un code convolutif de taux 1/2 est de même forme, à ceci près que les symboles d'information qui doivent être pris en compte au second membre sont tous ceux dont l'indice appartient à l'ensemble {i-L, i+L}, i excepté, L étant la longueur de contrainte du codeur convolutif.

3.2 – Utilisation du principe de Kullback

Une autre tentative visant au même but a utilisé le principe de Kullback, qui consiste à choisir pour distribution a posteriori q celle qui rend minimale son l'entropie mutuelle ("cross-entropy" ou "directed divergence") par rapport à la distribution a priori p, soit

$$H(q,p) = \sum_{i=1}^{n} q_i \log(q_i/p_i) \qquad (7)$$

avec $q = \{q_i\}$, $p = \{p_i\}$, et $\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} q_i = 1$. On applique ce principe à une distribution a posteriori séparable par rapport aux symboles d'information, c'est-à-dire de la forme

$$q_i = \prod_{j=1}^{k} Q_j^{b_j^i}(1-Q_j)^{1-b_j^i},\ i = 0, 1, \cdots 2^k-1, \qquad (8)$$

où b_j^i est le j-ème chiffre de la représentation binaire de i; Q_j peut alors être interprétée comme la probabilité que le j-ème bit vaille 0.

Dans le cas des codes linéaires en blocs, on ramène ainsi le décodage à la résolution du système d'équations non linéaires implicites suivant:

$$A_j = \sum_{i=1}^{n} g_j^i a_i \prod_{m=1}^{k} [t(A_m)]^{q_m^i}/t(A_j),\ j = 1, \cdots k, \qquad (9)$$

où g_j^i est l'élément général de la matrice génératrice du code et où t(x) a été définie par (6) [8]. Ce système d'équations est un cas particulier de la formule (5), dont il se déduit en y négligeant l'écart-type de la variable aléatoire en argument de la fonction t(.) du second membre [7]. La généralisation de ce résultat aux codes convolutifs, faisant apparaître au second membre les v.r. a posteriori dont l'indice appartient à l'ensemble {j-L,j+L}, j excepté, peut être envisagée pour fournir un moyen de pondération symbole par symbole, en sortie du décodeur intérieur, différent de la modification de l'algorithme de Viterbi du § 2.2. Son utilisation, même pour les codes en blocs, présente cependant des difficultés. En effet, si l'itération conduit rapidement (presque toujours) à des résultats stables, la résolution d'un système de la forme (9) est compliquée par l'existence possible de plusieurs solutions dont une seule satisfait au critère de Kullback. Nous verrons au § 4.3 un moyen de s'affranchir de cette difficulté.

3.3 – Applications du décodage à sortie pondérée

Nous disposons en principe, avec l'algorithme du § 2.2 et plus généralement (5) et (9), de moyens d'obtenir une pondération en sortie du décodeur intérieur. Nous allons maintenant examiner une application de cette forme de décodage à un code linéaire unique.

4 – DECODAGE D'UN CODE LINEAIRE UNIQUE DECOMPOSE EN UN PRODUIT

4.1 – Interprétation d'un code linéaire comme un produit

Un code linéaire unique peut toujours être interprété comme obtenu par combinaison de codes plus simples, notamment par l'intersection (au sens des ensembles) des codes "de parité" associés séparément à chacune des lignes de sa matrice de contrôle (fig. 5). Ce type d'association est un cas particulier de concaténation, puisque le code linéaire considéré peut être obtenu par plusieurs codages successifs, chacun portant sur le résultat du codage précédent. On peut donc scinder son décodage en plusieurs étapes simples, chacune fournissant une pondération utilisable à l'étape suivante.

Fig. 5: a) Schéma du produit de codes de parité;
b) Schéma de même forme montrant la décomposition du code de Hamming (7,4) en fonction de ses contrôles de parité.
Les symboles d'information sont désignés par i et les symboles de contrôle par c.

4.2 – Décodage à sortie pondérée d'un code de parité

Pour un code de parité (k+1,k), le système d'équations (9) devient:

$$A_{u_j} = a_{u_j} + a_p \prod_{m=1,m\neq j}^{k} t(A_{u_m}), \qquad (10)$$

où u_j, avec j de 1 à k, désigne un symbole d'information et p le symbole de parité. La principale difficulté du décodage par résolution de (9), la multiplicité de ses solutions, est évitée ici si l'on remarque que tous les symboles du mot d'un code de parité jouent un rôle symétrique: n'importe lequel d'entre eux peut servir de symbole de parité. Le système (10) admettra une solution unique si l'on choisit pour symbole de parité celui auquel correspond la v.r. de plus faible module. Si celui-ci appartient aux symboles d'information initiaux, auxquels on souhaite se référer, par exemple de 1 à k, il suffit d'en réestimer la v.r. en fonction de celle des autres, par le moyen décrit maintenant.

Soit $\underline{c} = [c_1 \cdots c_{k+1}]$ un mot du code et supposons par exemple que la v.r. de plus faible module soit celle du premier symbole. Les symboles d'information utilisés pour la résolution du système sont alors $c_2, c_3, \cdots c_{k+1}$, mais on souhaite se référer à $c_1, c_2, \cdots c_k$. La résolution du système (10) a pour résultat les v.r. $A_2, A_3, \cdots A_{k+1}$. Les v.r. $A_2, A_3, \cdots A_k$, qui concernent des symboles communs aux deux jeux de symboles d'information, peuvent être considérées comme acquises, mais il faut déterminer une nouvelle v.r. a posteriori A_1 en fonction de A_{k+1}, que l'on cessera ensuite d'utiliser, et des autres v.r. a posteriori.

A cet effet, on écrit que la probabilité d'avoir $c_1 = 0$ est égale à la probabilité d'un nombre nul ou pair de uns dans l'ensemble des symboles $c_2, \cdots c_{k+1}$, d'où

$$A_1 = \log \frac{1 + \prod_{j=2}^{k+1} \tanh(A_j/2)}{1 - \prod_{j=2}^{k+1} \tanh(A_j/2)}, \qquad (11)$$

expression calculable par récurrence à l'aide de la fonction

$$f(x,y) \overset{\Delta}{=} \log \frac{1+\exp(x+y)}{\exp(x)+\exp(y)}; \qquad (12)$$

A_{ci} étant la v.r. associée à la somme modulo 2 des i symboles $c_2 \cdots c_{i+1}$, on a

$$A_{c(i+1)} = f(A_{ci}, a_{i+2}),\ A_{c0} = +\infty \qquad (13)$$

et le résultat définitif de (11) est donc $A_1 = A_{c(k-1)}$. Le calcul de la fonction f(.,.) peut n'être qu'approché, par exemple par:

$$f(x,y) \sim \max(0,x+y) - \max(x,y) = \mathrm{sgn}(xy)\min(|x|,|y|), \qquad (14)$$

homologue de (3). L'approximation de A_1, obtenue est alors égale au plus petit module des réels $A_2, \cdots A_{k+1}$ affecté du signe de leur

produit.

4.3 – Décodage pondéré d'un code linéaire

Le décodage d'un code linéaire (n,k) quelconque C, interprété selon le § 4.1, peut alors être effectué par étapes. En supposant sa matrice génératrice de forme systématique avec les symboles d'information dans les k premières positions, chaque ligne de la matrice de contrôle correspondante ne possède qu'un symbole de parité de C; elle définit donc un code de parité dont les symboles d'information appartiennent à ceux de C.

On résout le système de la forme (10) associé à une ligne de cette matrice (par exemple par itération) en choisissant pour symbole de contrôle celui auquel correspond la plus petite vraisemblance. On révise si c'est nécessaire sa v.r. par (13) ou (14) et on substitue les v.r. a posteriori ainsi obtenues aux v.r. a priori initiales. On décode de la même façon un autre contrôle de parité avec des v.r. a priori dont certaines, qui correspondent aux symboles d'information communs avec le précédent, viennent d'être modifiées par le décodage. On poursuit ainsi avec les autres contrôles. La mise en oeuvre de ce procédé ne donne de résultat satisfaisant que si les contrôles les plus fiables sont décodés les premiers. Leur fiabilité peut être mesurée, après une tentative de décodage, par le plus petit des accroissements en module des v.r. d'information.

Une autre façon de décoder chacun des contrôles consiste à chercher toutes les solutions du système d'équations (10) pour les symboles d'information qui servent de référence (de 1 à k ici) et d'en pondérer les différentes solutions en fonction des probabilités a posteriori des mots correspondants. Cela revient en fait, si l'on suppose les v.r. a posteriori de module suffisamment grand, à utiliser la relation (5) en ne l'appliquant qu'aux solutions de (10).

La simulation montre que les deux procédés donnent des résultats voisins, avec une dégradation par rapport au décodage pondéré optimal [9] qui n'est que de quelques dizièmes de dB.

5 – DECODAGE SEQUENTIEL GLOBAL DU CODE CONCATENE

On peut cependant s'interroger sur l'utilité d'un décodage séparé de chacun des codes concaténés. La concaténation restreint aux mots du code extérieur les suites d'information possibles à l'entrée du codeur convolutif intérieur, donc les chemins possibles dans sa représentation par un arbre. Par exemple, si le code extérieur est donné sous forme systématique avec les k symboles d'information en tête, les k premières branches de l'arbre du code intérieur sont communes avec celles du code résultant de la concaténation, mais les (n-k) noeuds suivants sont "désaffectés" par le codage extérieur puisque les branches correspondantes sont déterminées par les k premières. On obtient ainsi un arbre avec une bifurcation à chacun des k premiers noeuds, prolongé par des suites de (n-k) branches sans bifurcation déterminées par le noeud atteint au niveau k, puis où les bifurcations reprennent sur une profondeur de k branches et cessent pour les (n-k) suivantes, etc (fig. 6). Le décodage séquentiel du code résultant de la concaténation est possible: il suffit, pour suivre un chemin au delà du noeud qui précède les noeuds désaffectés, de calculer la suite des (n-k) branches déterminée par le chemin qui y mène (en effectuant le codage extérieur) et la "métrique" de cette suite.

Mais le résultat de cette concaténation n'est qu'un code convolutif particulier; comme il existe de meilleurs codes de mêmes paramètres qui ne sont pas séparables en deux codes concaténés, des résultats meilleurs peuvent être espérés d'un code convolutif unique décodé séquentiellement. L'intérêt même de la concaténation apparaît ainsi lié au choix de l'algorithme de décodage du code convolutif intérieur: considérable quand la complexité du décodage dépend fortement de la longueur de contrainte du code (cas de l'algorithme de Viterbi), à peu près nul quand cette complexité n'en dépend à peu près pas (décodage séquentiel).

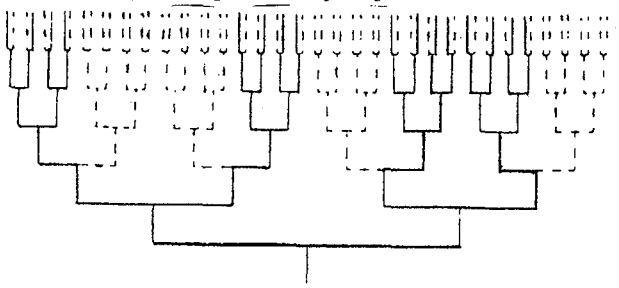

Fig. 6: Arbre associé au code obtenu par concaténation du code de la fig. 2 et du code de parité (3,2) (en trait fort; l'arbre du code convolutif seul est représenté en tirets).

Quelques résultats de simulation sont présentés sur la fig. 4 où ils peuvent être comparés à ceux du procédé décrit au § 2.2. L'impossibilité d'employer l'entrelacement explique probablement leur moindre qualité, alors qu'il contribue à celle des résultats mentionnés en 2.2.

6 – CONCLUSION

En conclusion, il apparaît que l'on peut envisager deux grands types de procédés de décodage: **globaux**, tels que le décodage séquentiel, opérant sur le code dans son ensemble même s'il est obtenu par la concaténation de plusieurs; et **analytiques**, comportant des décisions pondérées intermédiaires successives, même si le code utilisé est défini globalement. La possibilité même de ces derniers est une conséquence de la redéfinition proposée au § 3. La complexité moyenne des algorithmes du premier type est petite, mais le volume de calcul nécessaire est aléatoire, avec le risque de dépasser les possibilités du décodeur, qui doit être dimensionné pour des circonstances rarement réalisées. Avec un décodage du second type, le volume de calcul est constant; alors que la complexité du décodage optimal est prohibitive, même dans ses versions simplifiées, la décomposition en étapes simples la réduit très sensiblement, au prix d'une dégradation minime par rapport au décodage optimal.

Références

[1] G.D. FORNEY Jr, Concatenated codes, MIT Press, 1966

[2] D.D. FALCONER, A hybrid sequential and algebraic decoding scheme, Ph.D. dissertation, Dep. Elec. Eng., MIT, Cambridge, Mass., 1966

[3] J.P. ODENWALDER, Optimal decoding of convolutional codes, Ph.D. dissertation, Dep. Elec. Eng., Univ. California, Los Angeles, 1970

[4] G.W. ZEOLI, Coupled decoding of block-convolutional concatenated codes, IEEE Trans. on Comm., **COM-21**, mars 1973, pp. 219-226

[5] G. BATTAIL, Pondération des symboles décodés par l'algorithme de Viterbi, Annales des Téléc., à paraître

[6] G. BATTAIL, Le décodage pondéré en tant que procédé de réévaluation d'une distribution de probabilité, Colloque "3 journées sur le codage", Cachan, 24-26 novembre 1986; texte proposé aux Annales des Téléc.

[7] J.-C. BELFIORE, Un algorithme de décodage symbole par symbole, à décisions souples, de codes convolutifs binaires de taux 1/2, Colloque "3 journées sur le codage", Cachan, 24-26 novembre 1986

[8] G. BATTAIL, Décodage pondéré des codes linéaires: un nouvel algorithme, Colloque IPMU, Paris, 30 juin-4 juillet 1986

[9] G. BATTAIL & J. FANG, Décodage pondéré optimal des codes linéaires en blocs II.- Analyse et résultats de simulation, Annales des Téléc. **41** n° 11-12, nov.-déc. 1986, pp. 580-604