

DIXIEME COLLOQUE SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

507



NICE du 20 au 24 MAI 1985

SUR LE DECODAGE PONDERE DES CODES LINEAIRES

On Weighted Decoding of Linear Codes

G. Battail et Fang J.

E.N.S.T., Dép. SYC, 46 Rue Barrault, 75634 PARIS CEDEX 13

RESUME

On rappelle la représentation de l'ensemble des mots d'un code linéaire en blocs (n,k) par le diagramme du treillis. Il est dit "de forme normale" si la donnée des k premiers symboles détermine un mot du code unique.

On considère le décodage dans les trois cas où :

- la fiabilité individuelle des symboles est inconnue, manière classique de poser le problème qui conduit à prendre pour critère la distance de Hamming;
- le canal est à paquets d'erreurs localisables en ce sens que la fiabilité des symboles est nulle dans une zone connue avec certitude et absolue en dehors. Selon les dimensions de cette zone, le décodage est possible (et son résultat est alors exact) ou non;
- une mesure de la fiabilité de chaque symbole est disponible. Le décodage pondéré peut être interprété comme généralisant de façon probabiliste le cas précédent. Le treillis peut alors servir de support au décodage par l'algorithme de Viterbi.

Il est cependant possible de simplifier et on rappelle un algorithme qui le permet. Il comporte dans une première étape un reclassement des symboles par fiabilité non croissante. On crée ainsi dans chaque mot un paquet "justifié" à droite. Il comporte une version optimale au sens de la vraisemblance maximale et des variantes, obtenues notamment en limitant le nombre des chemins parcourus dans le treillis. La suite est consacrée à une ébauche d'analyse de la complexité de cet algorithme et à rendre compte de résultats de sa simulation.

Un modèle est introduit et les différentes probabilités pertinentes sont calculées en fonction de ses paramètres. On suppose d'abord le treillis de forme normale, mais on montre ensuite que l'essentiel des résultats subsiste si le treillis obtenu après le reclassement des symboles n'a pas cette forme.

Les résultats de simulation confirment qu'il suffit de parcourir un petit nombre des chemins du treillis pour obtenir une probabilité d'erreur résiduelle peu dégradée par rapport à l'optimum.

SUMMARY

The trellis diagram, as a means for representing the set of all words of a linear (n,k) block code, is first recalled. It is said "of normal form" if giving the first k symbols uniquely determines a codeword.

Decoding is contemplated in the following three cases:

- the reliability of each symbol is unknown; this is the conventional approach, which leads to use the Hamming distance as a criterion;

- the channel suffers error bursts which can be located i.e., the symbol reliability reduces to zero inside some known interval, whereas it is absolute outside. Depending on the length of this interval, decoding is possible (with an exact result) or not;

- a measure of each symbol reliability is available. Weighted decoding may then be interpreted as a probabilistic extension of the previous case. Decoding may be performed by the Viterbi algorithm using the trellis.

Simplifying is however possible and an algorithm for that purpose is first recalled. It begins in reordering the symbols according to their non-increasing reliability, thus creating in each word an error burst which is right "justified". An optimum version of this algorithm uses the maximum likelihood criterion, but simpler variants exist, especially resulting from limiting the number of paths considered in the trellis. The sequel is devoted to outline an analysis of this algorithm complexity and to report results from its simulation.

A model is proposed and the relevant probabilities are computed in terms of its parameters. The trellis is first assumed of normal form, but it is then shown that the results still essentially hold if the trellis resulting from the preliminary reordering has not this shape.

The simulation results confirm that only a very small number of paths need be considered in the trellis in order to get a residual error probability which entails only small degradation with respect to the optimum.



1 - Diagramme du treillis et décodage pondéré

La figure 1-a est le diagramme du treillis associé à un code linéaire en blocs de longueur n et de dimension k . Les abscisses, dites "niveaux de noeud", sont entières, de 0 à n , de même que les ordonnées, dites "états", qui représentent l'ensemble des vecteurs à $n-k$ composantes q -aires (ici $q = 2$). On appelle "branche" tout segment dans ce diagramme qui joint deux noeuds de niveaux consécutifs et "chemin" toute suite de branches joignant l'origine (niveau 0) à l'extrémité (niveau n). Chaque branche horizontale représente 0, chaque oblique 1. Le mode de représentation choisi assure une correspondance bijective entre chemins et mots du code /1/. Sur la figure, toutes les branches issues des noeuds de niveau 0 à $n-k-1$ divergent sans jamais converger et les branches aboutissant aux noeuds de niveau $k+1$ à n convergent sans jamais diverger. Cette forme de diagramme sera dite normale, traduction graphique de la forme systématique du code utilisé (les symboles d'information sont les k premiers) lorsque l'inégalité $2k \geq n$ est vérifiée, ce que nous supposons. Un treillis similaire peut être tracé pour une permutation quelconque des symboles du code, mais il n'est pas toujours de forme normale (les seuls codes représentés par des treillis de forme normale quelle que soit la permutation de leurs symboles sont optimaux, nécessairement non binaires, de distance minimale égale à sa borne supérieure $n-k+1$). La normalité du treillis entraîne qu'un chemin quelconque peut y être spécifié par le vecteur d'information formé des k premiers symboles d'un mot, noté u .

La figure 1-b représente une pondération des symboles du code. Les abscisses y sont discrètes de 1 à n . On a porté en ordonnée le coefficient de vraisemblance

(1) $v_i \triangleq |a_i|,$

qui mesure la fiabilité de la décision optimale r_i prise sur le i -ème symbole à la démodulation, avec

(2) $\text{sgn}(a_i) = (-1)^{r_i},$ où

(3) $a_i \triangleq \log_e(\text{Pr}(r_i=0)/\text{Pr}(r_i=1)).$

Si les symboles reçus peuvent être représentés par l'addition modulo 2 d'un symbole certain c_i et d'une erreur binaire e_i (indépendante des symboles émis et des autres erreurs), de probabilité $p_{ei} \leq 1/2$, soit

(4) $r_i = c_i \oplus e_i,$ on a

(5) $v_i = \log_e((1-p_{ei})/p_{ei}).$

Les probabilités indiquées doivent s'entendre conditionnellement à l'utilisation qui est faite du signal reçu et nous distinguerons trois cas.

1 - La fiabilité individuelle des symboles est inconnue. On admet donc que la probabilité d'erreur est commune à tous, soit p_e (connue ou non). On utilise alors pour modèle de la voie de transmission le canal binaire symétrique. La fiabilité des symboles est représentée par la "courbe" 1 de la figure 1-b (les ordonnées n'y ont de signification qu'en des abscisses discrètes).

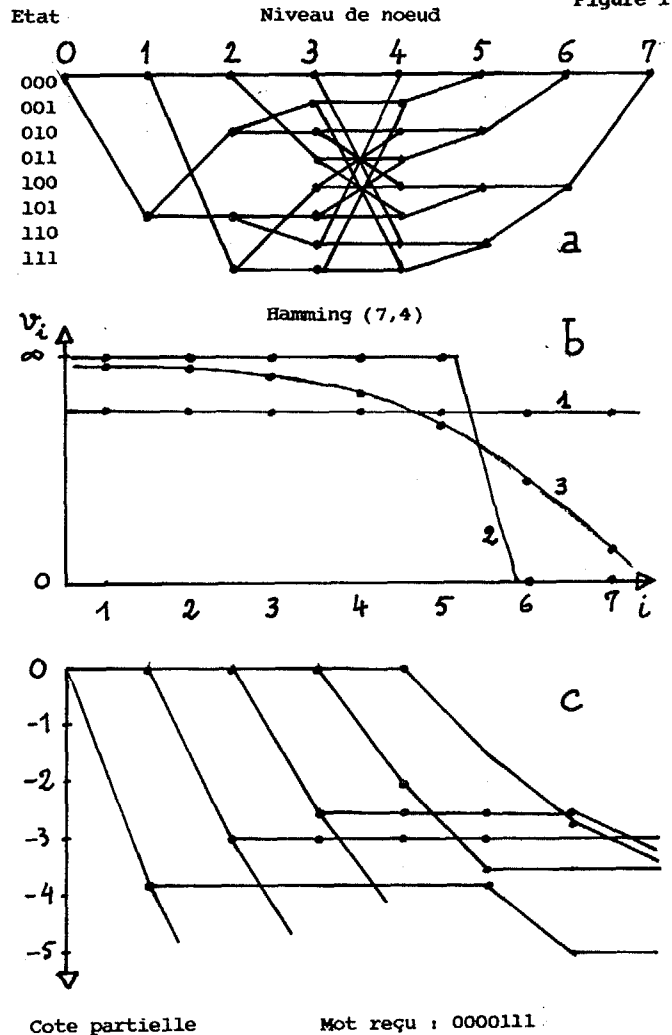
2 - On utilise au contraire le modèle du canal à paquets d'erreurs localisables, ainsi défini : toutes les erreurs sont groupées en des intervalles connus, dits "paquets d'erreurs", où la fiabilité est nulle. Elle est absolue extérieurement à eux. On utilise alors deux propriétés distinctes du canal : (a) une évaluation de la fiabilité des symboles, ici dichotomique; (b) la contiguïté des symboles de même fiabilité. La pondération est représentée par la "courbe" 2 de la figure 1-b en supposant le paquet localisé dans les dernières positions (pour un code cyclique, un décalage y ramène).

3 - La fiabilité des symboles individuels est connue, au moins approximativement, ne serait-ce que parce que

(3) est précisément la grandeur que le démodulateur optimal doit calculer pour prendre sa décision. (La disponibilité de l'information de fiabilité des symboles est donc la règle en pratique. Elle n'apparaît comme une exception que par rapport à une formulation du problème qui a permis en le simplifiant de bâtir la théorie des codes, mais dont la pertinence n'est qu'historique.) Pour tracer la "courbe" 3 de la figure 1-b, nous avons supposé qu'un reclassement préalable des symboles par fiabilité décroissante avait été effectué. Nous admettons provisoirement que ce reclassement est possible en conservant la forme normale du treillis.

Le décodage pondéré exploite la structure du treillis en tenant compte de la pondération. Nous associons pour cela une grandeur réelle non positive à toute branche, dite sa "pente", nulle si le résultat de la décision sur le symbole reçu est identique au symbole représenté par cette branche; dans le cas contraire, égale à l'opposé du coefficient de vraisemblance correspondant. Nous associons à tout chemin sa "cote", égale à la somme des pentes de ses branches. On montre alors aisément que le chemin du treillis dont la cote est la plus grande représente le résultat du décodage à vraisemblance maximale, compte tenu de la fiabilité des symboles /2/. Les procédés de décodage peuvent être représentés graphiquement à l'aide de la figure 1-c, où l'axe des abscisses est commun avec celui de la figure 1-a et où l'on porte en ordonnée la "cote partielle" $Z_i(u)$, somme des pentes des branches du chemin spécifié par le vecteur d'information u entre l'origine et le noeud ayant pour niveau l'abscisse i , dont elle est fonction non croissante quel que soit u .

Figure 1



Cote partielle

Mot reçu : 0000111



SUR LE DECODAGE PONDERE DES CODES LINEAIRES

On Weighted Decoding of Linear Codes

Dans le cas 1, les branches qui correspondent à des symboles différents des symboles reçus sont pondérées par la même quantité $-v = \log_e(p/(1-p))$ et les autres par 0. Le décodage optimal consiste donc à déterminer le chemin du treillis ayant le plus petit nombre de symboles qui diffèrent de ceux du mot reçu, donc le mot le plus proche pour la métrique de Hamming.

Dans le cas 2, la fiabilité nulle des symboles au delà d'un niveau de noeud g entraîne que la pondération des $n-g-1$ dernières branches est nulle quel que soit le symbole reçu. Si $g \geq k$, un chemin de cote finale maximale (nulle) est désigné sans ambiguïté et le décodage est correct. Si $g < k$, plusieurs chemins de cote finale nulle subsistent et il est impossible de les départager (résultats bien connus /3/).

Nous considérons maintenant le cas 3, où la pondération n'est pas dégénérée comme dans les cas 1 et 2. L'algorithme de Viterbi consiste à calculer en parallèle les cotes partielles des chemins en chaque noeud de chaque niveau. Si le treillis est de forme normale, au sens défini ci-dessus, la convergence des branches à partir du niveau $n-k+1$ entraîne qu'un choix définitif (quoique partiel) devient possible, en faveur du meilleur des deux chemins qui convergent en chacun des noeuds des $n-k$ derniers niveaux. Seul subsiste à l'extrémité du treillis (niveau n) le résultat du décodage à vraisemblance maximale /4-6/.

2 - Algorithme de décodage pondéré

Des algorithmes exigeant un moindre volume de calcul exploitent le contexte fini des codes en blocs /7,2/. Nous consacrerons toute la suite à une analyse de l'algorithme décrit dans /2/ sous sa forme optimale pour le critère de vraisemblance maximale et à certaines de ses variantes, optimales ou non (l'une d'elles équivalente à l'algorithme de Dorsch /7/). Cet algorithme effectue une "exploration" qui consiste à essayer des chemins du treillis dans un ordre tel que le résultat soit le plus souvent trouvé après un petit nombre d'essais. On peut décrire ainsi l'algorithme optimal :

(a) Classer les symboles, en principe par fiabilité non croissante, mais en opérant le nombre minimal d'interversions nécessaires au maintien de la forme normale du treillis.

Calculer $Z_k(\underline{u}^0)$, où \underline{u}^0 est le vecteur d'information pour lequel on a $Z_k(\underline{u}^0) = 0$.

Fixer le seuil à $\sigma = Z_k(\underline{u}^0)$.

(b) Déterminer le vecteur d'information \underline{u} suivant dans l'ordre des $Z_k(\underline{u})$ non croissants (un algorithme effectuant cette opération est disponible).

Faire $i = 1$.

Si $Z_k(\underline{u}) > \sigma$, aller en (c); si $Z_k(\underline{u}) \leq \sigma$, aller en (e).

(c) Calculer $Z_{k+1}(\underline{u})$.

Si $Z_{k+1}(\underline{u}) > \sigma$, calculer $Z_k(\underline{u})$ et aller en (d). Si $Z_{k+1}(\underline{u}) \leq \sigma$ et si $i < n-k$, ajouter 1 à i et aller en (c); si $i = n-k$, aller en (b).

(d) Faire $\sigma = Z_k(\underline{u})$. Aller en (b).

(e) Le vecteur \underline{u} qui a déterminé le dernier seuil est le résultat définitif \underline{g} .

Cet algorithme devient équivalent à celui de Dorsch /7/ quand on remplace l'instruction "Faire $i = 1$ " à l'étape (b) par : "Faire $i = n-k$ ".

Nous présentons maintenant un bref commentaire de cet algorithme. La cote d'un chemin en son niveau de noeud k n'est qu'une mesure partielle de sa vraisemblance, mais elle est aisément calculée pour tout vecteur \underline{u} à k composantes binaires par le produit scalaire $Z_k(\underline{u}) = \underline{u} \cdot \underline{v}'$, où \underline{v}' est le vecteur ayant pour composantes les coefficients de vraisemblance des k premiers symboles. La cote finale (de niveau n) du mot spécifié par \underline{u} est le critère définitif, mais exige un volume de calcul plus important (celui même du codage d'un vecteur d'informa-

tion). La cote de niveau k sert ainsi de critère à une présélection de candidats plausibles, les seuls pour lesquels le calcul de la cote sera poussé au delà.

Cet algorithme est optimal quant au critère, puisque l'exploration de nouveaux chemins ne cesse qu'après que la cote de niveau k est devenue inférieure au seuil σ , ce qui garantit que l'optimum a été trouvé. Des algorithmes non optimaux évitent cette vérification coûteuse en calcul, par exemple en limitant le nombre des chemins explorés à une valeur constante ou même en calculant systématiquement les cotes finales des J chemins ayant les meilleures cotes de niveau k , J étant très inférieur au nombre total des chemins du treillis, choisi pour limiter la complexité des calculs avec une dégradation de la probabilité d'erreur limitée par rapport à l'optimal. On bénéficie alors de la possibilité d'un calcul en parallèle, donc rapide, qui permet de plus une évaluation de la fiabilité du décodage de chaque mot.

Le classement préalable des symboles, à l'étape (a) de l'algorithme, a pour effet d'augmenter la part de la cote de niveau k dans la cote finale, rendant le critère provisoire qui sert au classement aussi voisin que possible du critère définitif. Il apparaît aussi comme une généralisation probabilisée du cas 2 (paquets d'erreurs localisables) où la cote finale est déterminée seulement par les g premiers symboles (elle ne peut être que nulle puisque ces symboles sont supposés certains). Le classement préalable équivaut bien à localiser les erreurs à la droite des mots, en ce sens que la probabilité d'une erreur est d'autant plus grande que l'on considère un symbole situé plus à droite dans le mot.

3 - Ebauche d'une analyse de l'algorithme

3.1 - Introduction d'un modèle

Supposons provisoirement les coefficients de vraisemblance certains et connus. Soit $-y_j$ la cote finale du chemin spécifié par le vecteur d'information \underline{u}^j . On a

$$y_j = g_j + x_j,$$

où $g_j \triangleq -Z_k(\underline{u}^j)$ sera dite profondeur et où x_j est la perte de cote entre les niveaux de noeud k et n . D'après les définitions (1) ou (5), la probabilité que la configuration d'erreurs réalisée dans les k premières positions soit \underline{u}^j est fonction exponentielle de g_j :

$$(6) \quad P(i) = Q_I \exp(-g_i),$$

où Q_I est la probabilité qu'aucune erreur ne se soit produite dans les k premiers symboles, c'est-à-dire

$$(7) \quad Q_I = \prod_j (1 + \exp(-v_j)) = \prod_i \exp(-g_i)$$

le produit étant effectué pour j de 1 à k et la somme pour i de 0 à $2^k - 1$. Nous supposons les vecteurs d'information numérotés dans l'ordre des profondeurs non décroissantes (celui même de leur exploration).

En ce qui concerne la perte de cote, il faut distinguer deux cas :

- si le vecteur d'information exploré \underline{u}^j est identique à la configuration d'erreurs réalisée, la perte de cote n'est due qu'aux erreurs dans les $n-k$ dernières positions et sa probabilité est de même forme que (6) :

$$(8) \quad P_c(x_j) = Q_c \exp(-x_j),$$

où Q_c est la probabilité qu'aucune erreur n'affecte les $n-k$ derniers symboles et s'exprime comme (7), j variant cette fois de $k+1$ à n et i de 0 à $2^{n-k} - 1$;
- si \underline{u}^j diffère de la configuration d'erreurs vraie, la perte de cote dépend de manière complexe du code employé et de la permutation obtenue lors du classement préalable. Nous admettrons à titre d'hypothèse simpli-



ficatrice que les 2^{n-k} pertes de cote possibles sont également probables, soit

$$(9) P_e(x_j) = 2^{-(n-k)}$$

quel que soit j et quel que soit le code. Les indices c et e repèrent respectivement les cas où le vecteur exploré et la configuration d'erreurs réalisée sont identiques ou différents.

Nous allons maintenant, pour nous affranchir de la connaissance des coefficients de vraisemblance, représenter la suite des profondeurs par une fonction certaine de leur indice et la perte de cote par une variable aléatoire.

3.2 - Représentation de la suite des profondeurs

En supposant k et n grands, avec $k/n = R$ maintenu constant, et en utilisant des approximations bien connues, on peut traduire la forme générale de variation de g_i en fonction de son indice (donc abstraction faite de la donnée d'un jeu particulier de coefficients de vraisemblance) par l'expression moyenne :

$$(10) g_i = v_u k H^{-1}(\log_2(i+1)/k), \quad i < 2^{k-1}, \\ = g_{i'}, \quad \text{avec } i' \cong 2^k - i - 1 \text{ si } i \geq 2^{k-1},$$

où $H^{-1}(\cdot)$ est la fonction inverse de l'entropie binaire $H(x) \cong -x \log_2(x) - (1-x) \log_2(1-x)$ et v_u le coefficient de vraisemblance moyen des k premiers symboles après reclassement. Une expression beaucoup plus simple, mais grossièrement approchée, est :

$$(11) g_i = v_0 \log_e(i+1),$$

où v_0 est une constante. La validité de (11) est restreinte à des indices $i \ll 2^k$.

Nous nous proposons maintenant de déterminer la constante v_u dans (10). La densité de probabilité du i -ème coefficient de vraisemblance après reclassement s'écrit

$$(12) p_i(x) = n C_{n-1}^{i-1} P_V(x) F^{n-i}(x) G^{i-1}(x),$$

où $p_i(x)$ est la densité de probabilité commune des coefficients de vraisemblance et $F(x) = 1 - G(x)$ leur fonction de répartition. Sa moyenne m_i s'obtient par intégration de $x p_i(x)$. La constante v_u est la moyenne de m_i pour i de 1 à k .

Il semble même que l'on puisse calculer m_i par

$$(13) m_i = G^{-1}(i/n - 1/2n),$$

cette expression étant au moins une approximation valable asymptotiquement pour k grand.

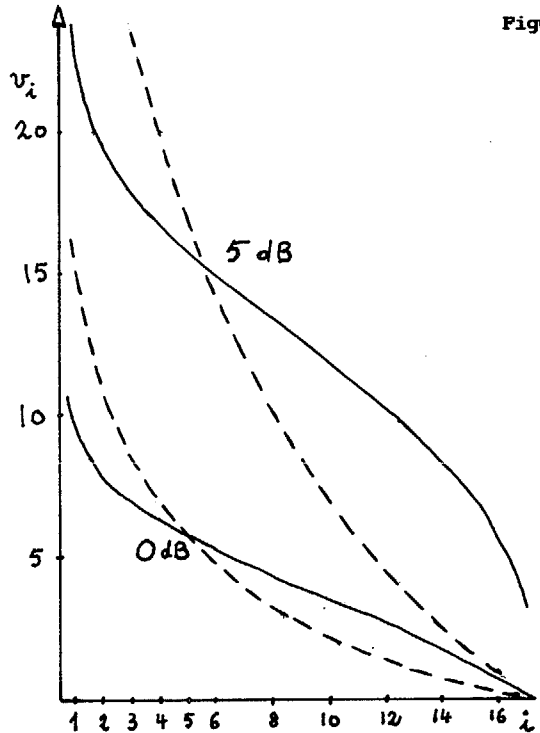
La fig. 2 illustre la relation (13) en représentant la fonction $G^{-1}(x)$ dans le cas de bruit gaussien additif pour diverses valeurs du rapport signal à bruit, l'amplitude du signal étant soit constante (trait plein), soit à distribution de Rayleigh (tirets).

La constante v_0 dans (11) peut être déterminée en remplaçant la somme finie des probabilités $P(i)$ données par (6) par la série de Riemann déduite de (11), d'où

$$(14) v_0 = \zeta^{-1}(1/Q_T),$$

où $\zeta(\cdot)$ est la fonction de Riemann. On peut ainsi déterminer v_0 en fonction d'une valeur de Q_T déduite de données de simulation. Ce calcul n'a de sens que si les termes de la série deviennent négligeables au delà d'une petite fraction des 2^k profondeurs, ce qui implique que v_0 , et donc le rapport signal à bruit soient suffisants.

Figure 2



3.3 - Représentation des pertes de cote

Nous représenterons la perte de cote par une variable aléatoire indépendante de la profondeur et de l'indice j , mais dont la densité de probabilité est différente selon que u^j est identique ou non à la configuration d'erreurs réalisée. Nous caractériserons la distribution de cette variable par sa fonction de répartition complémentaire, complément à 1 de sa fonction de répartition. Dans le premier cas, elle est l'homologue continu de (8) :

$$(15) G_c(x) = 1 \quad \text{si } x \leq 0 \\ = \exp(-x) \quad \text{si } x > 0$$

Dans le second cas, nous la désignerons par $G_e(x)$. Nous n'en connaissons pas l'expression exacte pour k fini. Asymptotiquement pour k devenant grand, $G_e(x/(n-k)v_0)$ tend vers une transition brusque de 1 à 0 lorsque son argument vaut $1/2$. v_0 désigne la moyenne des coefficients de vraisemblance moyens des $n-k$ derniers symboles après reclassement, que l'on déduit de (13). L'abscisse de cette transition croît donc indéfiniment avec k . Pour des valeurs finies de k , nous proposerons plus loin des expressions approchées de $G_e(x)$.

3.4 - Probabilités déduites du modèle

La figure 3-a représente le modèle utilisé. On y a porté en ordonnée l'indice j du vecteur exploré et en abscisse la somme y_j de la profondeur g_j et de la perte de cote x_j . g_j est la fonction (10) ou (11), certaine; x_j est une réalisation d'une variable aléatoire X_j dont la densité de probabilité est représentée sur la figure 3-b dans les deux cas $j \neq i$ et $j = i$, i étant l'indice de la configuration d'erreurs effective.

Nous supposons que l'exploration des vecteurs u^i est limitée aux J premiers, afin de traiter à la fois le cas optimal $J = 2^k$ et le cas $J < 2^k$ où l'on simplifie l'algorithme en limitant leur nombre. La probabilité d'obtenir le résultat I , indice de la plus grande cote finale, en explorant le vecteur u^m , conditionnellement à la réalisation de la configuration d'erreurs u^i , a pour expression

$$(16) P(I=m|i) = - \int_{g_m}^{\infty} G_{\min}(x;J|i) D_{l,m} G(x-g_m) dx,$$

SUR LE DECODAGE PONDERE DES CODES LINEAIRES

On Weighted Decoding of Linear Codes

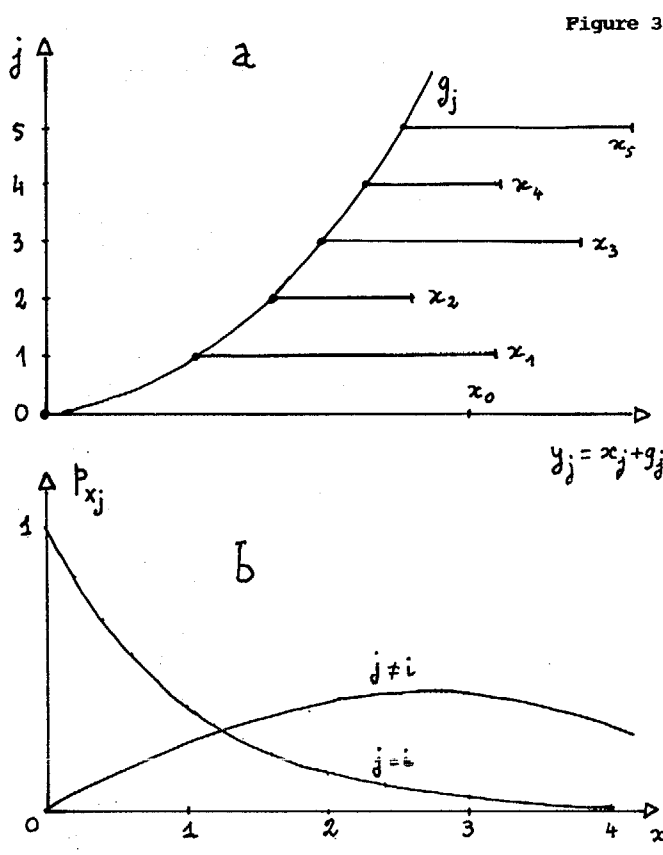


Figure 3

$$(21) P_g(m; J|i) = G_{\min} (g_m; J|i) - G_{\min} (g_{m+1}; J|i).$$

Les diverses probabilités pertinentes, abstraction faite de la configuration d'erreurs réalisée, se déduisent des expressions conditionnelles qui viennent d'être obtenues en calculant leur espérance par rapport à i , dont la distribution est donnée par (6). La décroissance de $P(i)$ est un autre argument en faveur de la limitation du nombre des chemins explorés.

3.5 - Complexité asymptotique

Définissons $C(k, m) \triangleq \sum_{i=0}^m C_k^i$. La profondeur g_0 est nulle, les profondeurs g_i sont de l'ordre de v pour i de 1 à $k \dots$, de l'ordre de $m v$ pour i de $C(k, m-1)$ à $C(k, m)$. Selon l'approximation pour k grand déjà utilisée en (10),

$C(k, m) = 2^{kH(m/k)} + O(\log k)$, $m \leq k/2$, où $H(x) \triangleq -x \log_2(x) - (1-x) \log_2(1-x)$ est l'entropie. La profondeur $kv_c(1-R)/2R$, au delà de laquelle il est assurément inutile de poursuivre l'exploration, est donc atteinte pour un indice i de l'ordre de

$$(22) i_t \approx 2^{kH(v_c(1-R)/2Rv_u)}$$

le nombre des chemins qu'il faut explorer pour le décodage optimal varie donc exponentiellement avec k , mais il est réduit par rapport à l'application pure et simple de l'algorithme de Viterbi si l'exposant est inférieure à $k(1-R)/R$. Il en est toujours ainsi si $R = 1/2$. Sinon, il faut que v/v_u soit assez petit, donc le rapport signal à bruit assez grand (et d'autant plus que R est grand). On a d'ailleurs $v/v_u < (1-R)/R$.

Par ailleurs, la probabilité d'erreur résiduelle ne devient petite pour k grand que si $P(i)$ est très petite lorsque g_i est de l'ordre de $kv_c(1-R)/R$, donc i donné par (22). Soit $P(i_t) = \epsilon$. Si l'on évalue $P(i)$ à l'aide de l'approximation (11), on a compte tenu de (14)

$$P(i) \approx \epsilon (i_t/i)^{v_0}$$

la probabilité $P(i)$ reste donc petite pour $i < i_t$, en particulier si v_0 est petit ((14) entraîne $v_0 > 1$). La limitation à $J < i_t$ du nombre des chemins est de ce seul fait possible avec une dégradation faible.

3.6 - Ebauche d'une analyse pour k fini

Pour des valeurs de k pratiques, même grandes, $G_e(x)$ est loin d'avoir une transition brusque. La remarque qui suit l'inégalité (20) s'applique donc et suggère une réduction du nombre de chemins au delà de celle que l'on peut déduire de l'analyse précédente.

Pour aborder quantitativement ce problème, nous avons essayé de représenter par des densités de probabilité connues la distribution des pertes de cote déduite des simulations. En particulier, une analogie de forme (d'ailleurs explicable en théorie) nous a conduit à représenter approximativement cette densité par la loi de Rice

$$(23) p(x) = (x/\sigma^2) \exp(-x^2/2\sigma^2 - r) I_0(x(2r)^{1/2}/\sigma),$$

les paramètres σ et r étant ajustés au mieux aux données de simulation. Pour un code court et un rapport signal à bruit petit, la meilleure approximation est obtenue pour $r = 0$, cas où la loi de Rice dégénère en celle de Rayleigh. On peut alors calculer à l'aide de ce modèle qu'une limitation à J du nombre des chemins dégrade la probabilité d'erreur résiduelle d'un facteur inférieur à 2, J ne variant que proportionnellement à k^2 . Pour des codes de longueur moyenne ou grande, r est différent de 0 et la complexité de la fonction de répartition interdit un calcul du même type. Les

où $G_{\min}(x; J|i)$ est la fonction de répartition complémentaire de la plus petite des y_j , $j = 0, 1, \dots, J-1$, égale au produit de leurs fonctions de répartition complémentaires. Puisque la configuration d'erreurs u^i a été réalisée, la fonction de répartition complémentaire de la j -ème somme est $G_e(x-g_j)$ pour j différent de i et $G_c(x-g_i)$ pour $j = i$; on peut donc l'écrire :

$$(17) G_{\min}(x; J|i) = \prod_{j=0}^{J-1} G_e(x-g_j) R(x-g_i), \text{ avec}$$

$$(18) R(x) \triangleq G_c(x)/G_e(x).$$

Par ailleurs, $D_{g_m} G(\cdot)$ désigne la dérivée logarithmique par rapport à x du $(m+1)$ ème facteur de $G_{\min}(x; J|i)$, soit $G_e(x-g_m)$ pour $m \neq i$ et $G_c(x-g_i)$ pour $m = i$, auquel cas elle vaut -1 d'après (15).

On tire en particulier de (16) la probabilité de décodage correct, conditionnelle à la réalisation de u^i :

$$(19) P_c(i) \triangleq P(I=i|i) = \int_{g_i}^{\infty} G_{\min}(x; J|i) dx.$$

On notera qu'elle décroît avec i , puisque l'intégrande est une fonction positive et non croissante de x et de i , et que les g_i forment une suite non décroissante. Compte tenu de (19) et (17), on en déduit la majoration :

$$(20) P_c(i) < \prod_{j=0}^{i-1} G_e(g_i - g_j).$$

Le second membre décroît, donc la probabilité d'erreur résiduelle conditionnelle à i croît avec i : plus on pousse loin l'exploration et plus grande est la probabilité d'erreur de la décision optimale. On dégradera donc peu la probabilité d'erreur moyenne en arrêtant l'exploration des chemins à un indice où cette probabilité devient grande.

On déduit aussi de (16) la probabilité (conditionnelle à u^i) de garantir, lors de l'exploration de u^m , que l'optimum a été atteint à l'une des étapes précédentes :



SUR LE DECODAGE PONDERE DES CODES LINEAIRES

On Weighted Decoding of Linear Codes

résultats de simulation présentés dans le paragraphe 4 montrent que le nombre des chemins explorés peut être limité à J raisonnablement petit sans dégradation notable par rapport à l'optimum.

3.7 - Hypothèse de normalité

Nous avons présenté cette analyse en supposant que le classement préalable par fiabilité non croissante et la normalité du treillis étaient compatibles. La théorie et la simulation montrent que des interversions en très petit nombre suffisent toujours à rétablir la normalité; elles ne concernent en outre que des symboles de vraisemblance moyenne. Leur effet peut donc être négligé et les conclusions ci-dessus subsistent.

4 - Résultats de simulation

4.1 - Taux d'erreur/rapport signal à bruit

On a simulé la transmission par différents codes linéaires en blocs en présence de bruit additif, stationnaire, gaussien et blanc. Le taux d'erreur moyen par mot a été déterminé en fonction du rapport E_b/N_0 (énergie reçue par symbole d'information, divisée par la densité spectrale unilatérale du bruit) pour diverses valeurs de J et divers codes. La figure 4 est un exemple de résultats ainsi obtenus. On y a aussi représenté la probabilité d'erreur par bloc de k symboles transmis sans codage, pour la comparaison.

4.2 - Effet de la limitation de J

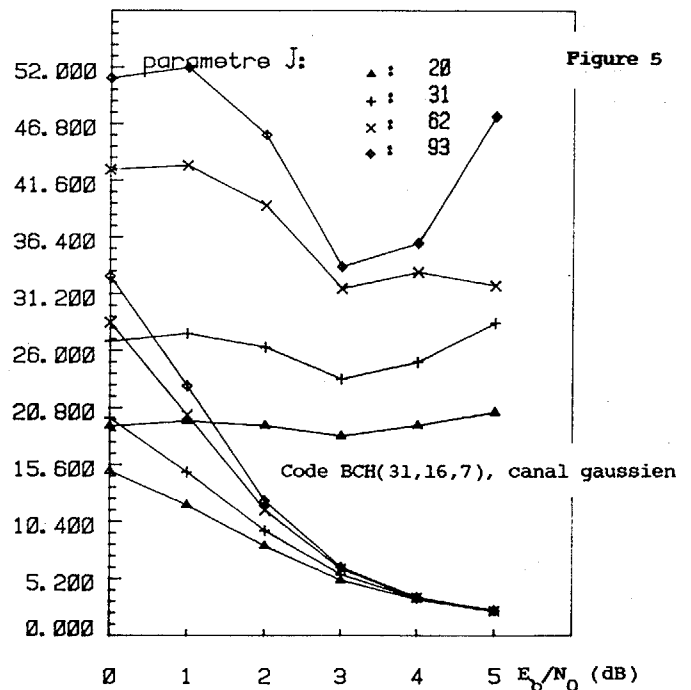
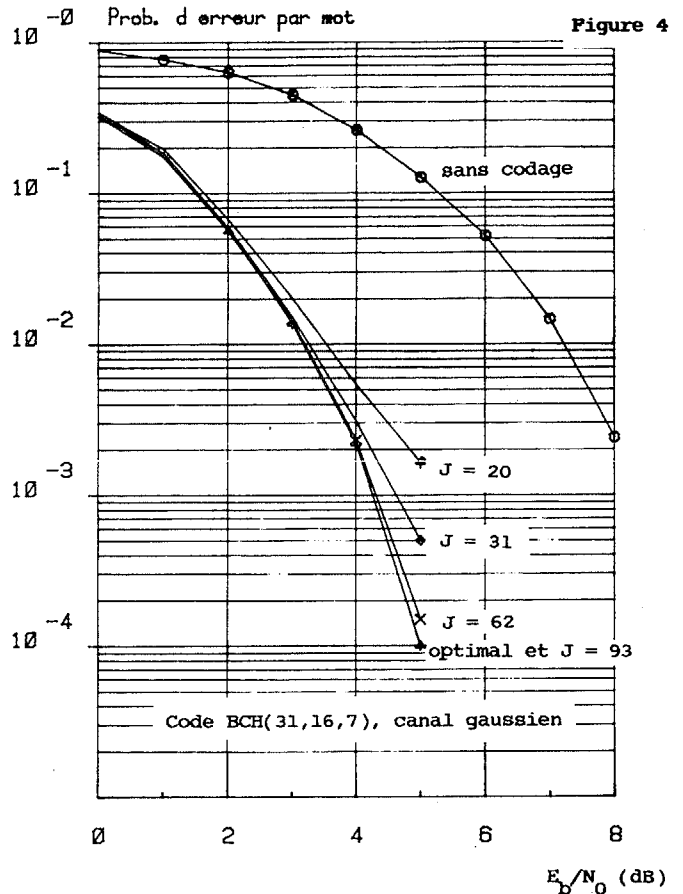
On a représenté en fonction du rapport signal à bruit, pour différents codes : (a) le nombre moyen des chemins explorés par mot décodé, quel que soit le résultat du décodage; (b) le nombre moyen des chemins explorés par mot décodé, mais exclusivement quand le résultat du décodage est erroné. Les résultats obtenus pour le même code ont été portés sur la figure 5. Le nombre (a) tend vers 1, comme on pouvait s'y attendre. Le nombre (b) dépend de J . Si J est petit, il en est voisin quel que soit le rapport signal à bruit; cependant, au delà d'une certaine valeur de J , ce nombre tend vers une valeur constante. Il suffit de prendre J de cet ordre pour limiter la probabilité d'erreur.

4.3 - Remarque

Les simulations font apparaître un comportement paradoxal dont l'analyse ébauchée ci-dessus rend compte. La dégradation de la probabilité d'erreur due à la limitation à J chemins est surtout importante lorsque cette probabilité est faible, donc lorsque le rapport signal à bruit est bon : le décodage exige en quelque sorte davantage de moyens (J plus grand) dans les conditions où l'on pourrait le croire le plus facile.

5 - Conclusion

La conclusion la plus importante que nous pouvons tirer de l'analyse et des résultats de simulation est sans doute la possibilité d'un décodage de probabilité d'erreur résiduelle peu dégradée par rapport à l'optimum, mais n'explorant qu'une très faible fraction des chemins du treillis. Ce résultat est dû dans une large mesure au reclassement des symboles qui crée, en quelque sorte, des paquets d'erreurs "justifiés" à droite (ce qui n'est concevable qu'avec la pondération). Le succès des procédés de codage sur les voies intrinsèquement affectées de paquets repose au contraire sur une répartition aussi égale que possible de la fiabilité des symboles entre les différents mots, grâce à des techniques auxiliaires comme la diversité, l'entrelacement ou l'étalement de spectre. Ces stratégies antagonistes coopèrent.



6 - Références

- /1/ BATAIL G., Annales Téléc. 38, 1-2, pp. 3-15
 - /2/ BATAIL G., Annales Téléc. 38, 11-12, pp. 443-459
 - /3/ FORNEY G.D., IEEE Trans. COM-19, 5, II, pp. 772-781
 - /4/ VITERBI A.J., IEEE Trans. IT-15, 6, pp.732-734
 - /5/ FORNEY G.D., IEEE Proc. 61, 3, pp.261-278
 - /6/ WOLF J.K., IEEE Trans. IT-24, 1, pp.76-80
 - /7/ DORSCH B.G., IEEE Trans. IT-20, 3, pp. 391-394
- Une publication plus détaillée sur l'analyse et les simulations présentées ici est en préparation.