

COLLOQUE NATIONAL SUR LE TRAITEMENT DU SIGNAL ET SES APPLICATIONS

NICE du 26 au 30 AVRIL 1977

RECEPTION DE DONNEES PAR UN CODEUR DE COMPLEXITE
REDUITE EN PRESENCE DE BRUIT ET D' INTERFERENCE ENTRE SYMBOLES

Ghassan KAWAS - KALEH

Ecole Nationale Supérieure des Télécommunications 46, rue Barrault 75 634 PARIS CEDEX 13

RESUME

La réception de données par l' algorithme de Viterbi en présence de bruit et d' interférence entre symboles demande un volume de mémoire et de calcul qui rend difficile sa réalisation.

Nous présentons une méthode de simplification en combinant des stratégies utilisées par l' algorithme de Viterbi et l' algorithme de pile. L' égaliseur proposé ne conserve en mémoire qu' un nombre réduit de survivants. A chaque étape, les survivants conservés sont les plus vraisemblables parmi ceux qui prolongent les survivants de l' étape précédente. La simulation effectuée sur un canal à haute distortion montre une nette supériorité de l' égaliseur proposé par rapport à d' autres égaliseurs.

SUMMARY

Maximum likelihood data reception in the presence of noise and intersymbol interference is of unacceptable complexity.

We propose a simplification method by combining strategies used by the Viterbi algorithm and Stack algorithm. Only a reduced number of survivors are stored in memory. They are the most likely among those which extend the survivors of the previous instant. Simulation results for a high distortion channel show a net superiority of the proposed equaliser over others.



RECEPTION DE DONNEES PAR UN CODEUR DE COMPLEXITE
REDUITE EN PRESENCE DE BRUIT ET D'INTERFERENCE ENTRE SYMBOLES

I. INTRODUCTION

Dans un système de transmission de données une suite d'impulsions, ou symboles, émise par une source, est transmise au destinataire à travers un canal dispersif qui introduit un bruit additif et des perturbations déterministes sous la forme d'interférence entre symboles. Pour des raisons de commodité, nous traitons le système de modulation d'impulsions en amplitude à bande latérale double. La généralisation à d'autres systèmes peut être faite aisément.

La source d'information émet une suite de symboles a_i séparés par des intervalles de temps égaux à T . Les a_i prennent des valeurs entières, équidistantes, représentant les symboles d'un alphabet Q de q éléments. On prend $Q = \{\pm 1, \pm 3, \dots, \pm(q-1)\}$. Le canal équivalent en bande de base est modélisé par une source de bruit aléatoire $b(t)$ et par un filtre linéaire de réponse impulsionnelle réelle $c(t)$, cf. Fig.1. $b(t)$ est supposé indépendant du signal, gaussien, blanc et centré.

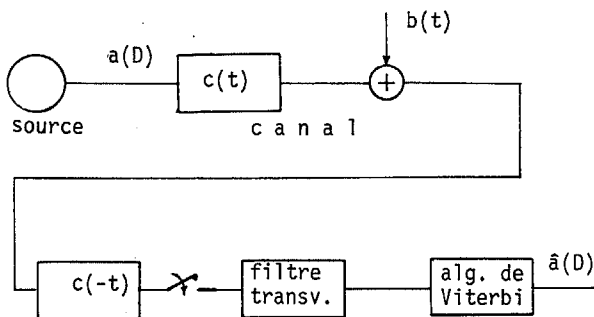


Fig 1. Chaîne de transmission de données avec récepteur optimal.

Forney [1] a proposé un récepteur, ou égaliseur, optimal composé d'un filtre adapté, suivi d'un échantillonneur, d'un filtre transversal et d'un décodeur optimal utilisant l'algorithme de Viterbi, cf. Fig. 1. La partie de la chaîne de transmission comprise entre la sortie de la source d'information et l'entrée du décodeur est équivalente au modèle discret de la Fig.2. Ce modèle se compose d'une machine séquentielle H , linéaire, déterministe, causale, de coefficients h_0, h_1, \dots, h_L , et d'une source de bruit additif produisant des variables aléatoires b_k indépendantes, gaussiennes, centrées et de variance σ^2 . La suite reçue traitée par le décodeur s'écrit :

$$z(D) = y(D) + b(D) \tag{1.a}$$

où :

$$y(D) = a(D) h(D) \tag{1.b}$$

$x(D)$ étant la transformée en D [1] de la suite $\{x_i\}$.

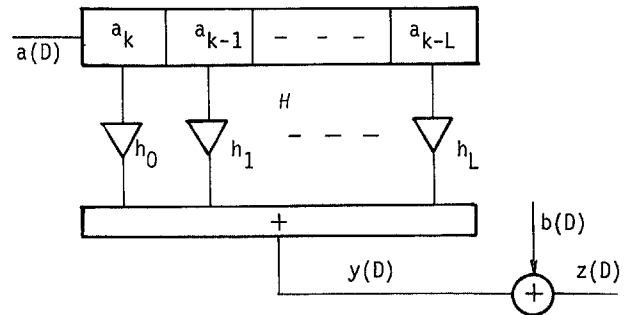


Fig.2. Modèle discret du canal

Le volume de calcul et de mémoire requis par l'algorithme de Viterbi est proportionnel au nombre d'états q^L [1]. Si on tient compte du fait que dans un canal téléphonique bien utilisé L peut varier entre 20 et 200, on peut entrevoir la difficulté de la réalisation du décodeur optimal. Il est donc indispensable de chercher un moyen pour alléger la complexité de l'égaliseur optimal sans subir une perte sensible d'optimalité. Ceci est l'objectif de l'égaliseur que nous proposons ici sous le nom de *décodeur de suite*. Nous avons jugé utile de le faire apparaître comme une modification de l'algorithme de pile de Jelinek-Zigangirov [8,9], de façon à montrer les liens entre cet algorithme et l'algorithme de Viterbi.

De nombreux auteurs [2-7] ont tenté de simplifier l'égaliseur optimal. Fredricsson [4] a étudié un égaliseur à deux survivants pour un système q -aire à réponse partielle $h(D) = 1 + D^n$. Par ailleurs, Vermeulen et Hellman [5] ont proposé un égaliseur de même complexité pour un canal quelconque, et précisé les conditions sur la réponse impulsionnelle pour que cet égaliseur soit asymptotiquement optimal. L'idée est encore développée dans [6] où Vermeulen propose de rejeter les survivants séparés de la suite reçue par une distance plus grande que $d_{min}/2$, d_{min} étant la distance minimale entre les mots de code engendré par H . La même démarche est reprise par Foschini [7] qui donne une démonstration plus générale de l'optimalité asymptotique et introduit la notion de "détecteur rapide".

Dans les égaliseurs de [5,7], une difficulté pratique apparaît : la détermination de d_{min} . Vermeulen [6] propose de ne conserver en mémoire qu'un nombre M de survivants. Cependant, l'éventualité qu'au cours du décodage, la mémoire soit rem-

RECEPTION DE DONNEES PAR UN CODEUR DE COMPLEXITE
REDUITE EN PRESENCE DE BRUIT ET D'INTERFERENCE ENTRE SYMBOLES

plie de survivants identiques n'est pas complètement exclue. Une telle situation est systématiquement écartée par le décodeur de suite.

L'algorithme de pile, adapté à l'égalisation, est décrit au § II ; les modifications conduisant au décodeur de suite sont introduites au § III et une comparaison avec l'égaliseur optimal est présentée. Dans le § IV deux égaliseurs pratiques utilisant respectivement l'algorithme de Viterbi [3] et le décodeur de suite sont comparés et le paramètre qui influence sur leurs performances est mis en évidence. Les résultats des simulation confirment les avantages de l'égaliseur proposé par rapport à d'autres égaliseurs.

II. L'ALGORITHME DE PILE

A la différence de l'algorithme de Viterbi qui utilise un treillis, l'algorithme de pile est basé sur une représentation en arbre de l'opération de codage réalisée par le canal H . La Fig.3 est un arbre représentant le codage réalisé par le canal pour le cas de symboles binaires appartenant à $Q = \{1, -1\}$. Dans ce graphe, la progression du temps est figurée de la gauche vers la droite. A l'entrée d'un symbole a_k dans H , une bifurcation se produit en un noeud. De chaque noeud sont issues q branches associées aux q valeurs possibles de a_k . Sur la figure 3, une bifurcation montante correspond au symbole -1 , et descendante au symbole 1 . Chaque branche représente un contenu déterminé du registre de $H : (a_k, a_{k-1}, \dots, a_{k-L})$, et donc, une sortie déterminée y_k .

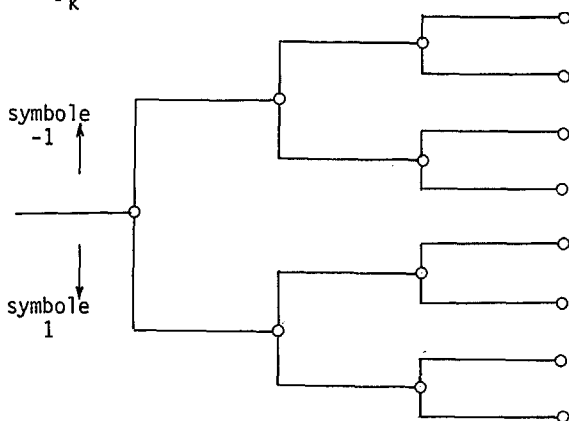


Fig 3. Représentation en arbre du codage réalisé par le canal H .

Comme dans le treillis, une suite d'entrée $a(D)$ donnée, trace dans l'arbre un chemin particulier. Le décodage consiste alors à retrouver ce chemin à partir de la suite reçue $z(D)$, $h(D)$ étant connue.

Un critère de décision doit être utilisé pour indiquer au décodeur le chemin à parcourir.

Le critère du maximum de probabilité a posteriori est adopté comme critère de décision : le décodeur doit choisir comme suite émise $a(D)$, celle dont la probabilité a posteriori $\text{Pr}[a(D)|z(D)]$ est la plus grande. Il est facile de voir [1] que ceci est équivalent à la règle de décision.

Choisir $a^{i'}$ (D) comme suite émise si :

$$\Gamma_i(1,k) < \Gamma_j(1,K) \text{ pour tout } j \neq i \quad (2.a)$$

où :

$$\Gamma_i(1,K) = \sum_{k=1}^K \gamma^{i'}(k) \quad (2.b)$$

$$\gamma^{i'}(k) = (z_k - y_k^{i'})^2 \quad i = 1, 2, \dots, q^k \quad (2.c)$$

$$y_k^{i'} = \sum_{m=0}^L h_m a_{k-m}^{i'} \quad (2.d)$$

$\Gamma_i(k,l)$ est la métrique de la partie du chemin i correspondant à a_k, a_{k+1}, \dots, a_l ; $\gamma(k)$ la métrique de la branche associée à a_k et $a_k^{i'}$ le symbole représenté par la $k^{\text{ème}}$ branche du chemin i ; K est la longueur de la suite $a(D)$. En attribuant à chaque chemin i une longueur égale à sa métrique, le décodage revient à chercher dans l'arbre, le plus court chemin.

Les différents chemins parcourus par le décodeur sont rangés dans une pile dans l'ordre croissant de leurs métriques, de sorte que le chemin le plus court, i.e. le plus vraisemblable occupe le sommet de la pile.

A chaque étape de l'algorithme, le seul chemin que l'on prolonge est celui qui occupe le sommet de la pile. Le prolongement est réalisé pour l'ensemble des q branches issus de son noeud terminal. Les nouveaux chemins ainsi obtenus sont à leur tour rangés dans la pile selon l'ordre de leurs métriques. La taille de la pile augmente donc de $q-1$ à chaque étape.

Au cours du décodage, le décodeur peut avoir à revenir en arrière dans l'arbre pour prolonger un chemin qui s'avère plus vraisemblable que celui qu'il vient de parcourir. Afin de minimiser le risque de s'engager dans des chemins incorrects, la métrique est biaisée de telle manière, qu'en moyenne, elle diminue le long d'un chemin correct et augmente dans le cas contraire.

Il est facile de voir que lorsque le rapport signal sur bruit est élevé, le décodeur avance rapidement dans l'arbre. En revanche, le volume de calcul,



RECEPTION DE DONNEES PAR UN CODEUR DE COMPLEXITE
REDUITE EN PRESENCE DE BRUIT ET D'INTERFERENCE ENTRE SYMBOLES

mesuré par le nombre moyen d'étapes de prolongement par symbole décodé, augmente lorsque ce rapport diminue ; le décodeur doit alors effectuer de nombreux retours en arrière et emprunter d'autres chemins. L'augmentation du nombre d'étapes entraîne également une augmentation de la taille de la pile qui devient rapidement extrêmement chargée.

III. MODIFICATION DE L'ALGORITHME

La particularité du problème de l'égalisation nous conduit à introduire dans l'algorithme de pile les modifications suivantes :

A. LIMITATION DE LA TAILLE DE LA PILE

La limitation pratique du volume de la mémoire a une dimension raisonnable nous contraint à réduire la taille de la pile et donc à ne conserver en mémoire qu'un nombre réduit M de chemins. La conséquence d'une telle réduction est évidemment une perte de performance, lorsque le chemin correct se trouve rejeté hors de la pile.

B. STABILISATION DU VOLUME DE CALCUL.

La différence essentielle entre l'algorithme de pile et l'algorithme de Viterbi est que dans ce dernier, le volume de calcul est constant et grand : à chaque instant le décodeur examine systématiquement tous les états du canal. Par contre, dans l'algorithme de pile, le volume de calcul est variable : il dépend, comme nous l'avons signalé, du rapport signal sur bruit. Or, dans l'égalisation, il est souhaitable d'imposer un volume de calcul constant. Les raisons en sont les suivantes :

1. En transmission de données, le rapport signal sur bruit est en général constant : les modifications éventuelles du canal se manifestent par des variations de sa réponse impulsionnelle ; c'est l'identificateur du canal qui tient compte de ces variations, le décodeur n'étant nullement affecté. Or, à rapport signal sur bruit constant, la stratégie spécifique de l'algorithme de pile, permettant une progression rapide pendant les périodes où ce rapport est élevé, n'offre plus guère d'intérêt.

2. Dans le décodage séquentiel d'un code récurrent, le parcours d'un chemin non correct est aisément décelé grâce aux grandes distances séparant les chemins possibles. En effet, la redondance introduite par le décodage récurrent entraîne une augmentation des distances séparant les mots de code. Ce n'est pas le cas, en revanche, pour le codage sans

redondance réalisé par un canal dispersif où ces distances sont petites. Pour éviter d'emprunter un chemin incorrect difficile à déceler, nous sommes alors amenés à avancer prudemment dans l'arbre, i.e. ne pas nous contenter de prolonger seulement le chemin occupant le sommet de la pile ; d'où l'idée de prolonger simultanément les M chemins ce qui conduit à un volume de calcul constant.

3. Enfin, un volume de calcul constant facilite la réalisation matérielle (hardware) de l'égaliseur.

Nous proposons donc de prolonger à chaque étape l'ensemble des M chemins présents dans la pile. Cette mesure garantit que tous les chemins soient "mis à jour" en les contraignant à comporter le même nombre de branches.

Signalons que ceci ne diminue pas la vitesse du décodage puisque les prolongements de tous les chemins peuvent être exécutés en parallèle.

Cette méthode de "mise à jour" a d'ailleurs été proposée par Jelinek [8] pour accroître la vitesse de son algorithme, ainsi que par Haccoun et Ferguson [10] pour réduire les variations du volume de calcul. Cependant, dans ces deux références, les prolongements simultanés concernent uniquement quelques chemins occupant le haut de la pile.

En prolongeant simultanément tous les chemins de la pile, nous perdons la liberté de retour en arrière dans l'arbre et nous acquérons la propriété de l'algorithme de Viterbi de progression systématique sur tous les chemins. Il n'est plus alors nécessaire d'introduire un biais dans le calcul de la métrique.

C. ELIMINATION DES CHEMINS REDONDANTS

Pour tirer le meilleur parti de la mémoire disponible, nous proposons d'exploiter la stratégie de l'algorithme de Viterbi consistant à ne pas garder en mémoire plusieurs chemins correspondant à un même état du canal, l'élimination de tous les chemins sauf celui dont la métrique est la plus petite n'entraînant aucune perte.

Ainsi, lorsqu'un chemin ne représente pas un état décrit par un chemin déjà présent dans la pile, il est introduit dans celle-ci. Dans le cas contraire, les métriques sont comparées, et seul est conservé dans la pile le chemin possédant la métrique la plus petite. Cette mesure permet d'écarter la

RECEPTION DE DONNEES PAR UN CODEUR DE COMPLEXITE
 REDUITE EN PRESENCE DE BRUIT ET D'INTERFERENCE ENTRE SYMBOLES

situation rencontrée par Vermeulen [6] où toute la pile est occupée par des chemins identiques.

Ces modifications étant faites, il ne reste qu'à limiter à une longueur δ la taille de la mémoire réservée à chaque chemin. Ceci signifie que le décodeur prend ses décisions avec un retard δ .

Nous sommes maintenant en mesure de préciser l'algorithme exécuté par le décodeur de suite. Désignons par $\gamma_{m,j}(k)$ la métrique de la $k^{\text{ème}}$ branche du $m^{\text{ème}}$ chemin de la pile correspondant à $a_k = j$:

$$\gamma_{m,j}(k) = (z_k - j h_0 - \sum_{i=1}^L h_i a_{k-i}^{j'})^2 \quad (3)$$

où $m = 1, 2, \dots, M$ et $j \in Q$.

Lorsque le décodeur de suite reçoit z_k , les chemins $\{(\hat{a}_i, 1 < i < k-1)_m, m = 1, 2, \dots, M\}$ sont déterminés ainsi que leurs métriques $\{\Gamma_m(1, k-1), m = 1, 2, \dots, M\}$. La progression à l'étape suivante se fait selon l'algorithme suivant :

Initialiser : $k = 0, \Gamma_m(1,1) = 0$ pour tout $m = 1, 2, \dots, M$

"1" Incrémenter $k : k + 1$

"2" Calculer :

$$\Gamma_{m,j}(1,k) \triangleq \Gamma_m(1, k-1) + \gamma_{m,j}(k)$$

pour tout $j \in Q$ et tout $m = 1, 2, \dots, M$.

"3" Chercher les chemins qui représentent le même état du canal et les éliminer, à l'exception de celui qui possède la métrique la plus petite.

"4" Conserver les M chemins dont les métriques correspondent aux M valeurs les plus petites parmi les $\Gamma_{m,j}(1,k)$; éliminer les autres.

"5" Décider $\hat{a}_{k-\delta} = a_{k-\delta}^{j'}$ où j' désigne le chemin occupant le sommet de la pile.

"6" Si $k < K$ aller en "1" sinon FIN.

Comme nous l'avons signalé, les modifications apportées à l'algorithme de pile lui ôtent sa propriété de retour en arrière dans l'arbre. Ces modifications mettent en évidence le rapport entre les algorithmes de pile et de Viterbi. Remarquons qu'en prenant $M = q^L$, l'algorithme proposé n'est autre que l'algorithme de Viterbi lui-même.

Le décodeur de suite diffère de l'égaliseur optimal en ce qu'il ne conserve en mémoire que M survivants parmi les q^L . A chaque étape, les survivants conservés sont les plus vraisemblables parmi ceux qui prolongent les survivants de l'étape précédente.

Le calcul des performances de cet égaliseur étant malaisé, nous avons eu recours à la simulation. Pour chacun des points de la Fig. 4., 5 000 symboles binaires $\{-1, +1\}$ sont transmis à travers un canal de coefficients $(0,227; 0,46; 0,688; 0,46; 0,227)$. L'égaliseur optimal utilisant 16 survivants est à comparer à un décodeur de suite utilisant 4 survivants.

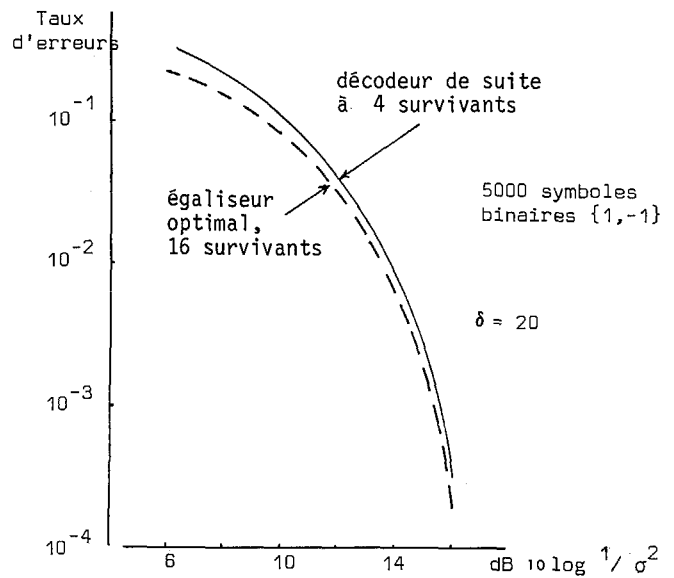


Fig 4. Performances comparées du décodeur de suite et de l'égaliseur optimal.

IV. STRUCTURE PRATIQUE

La Fig. 5 représente un schéma pratique [3] remplaçant le schéma théorique de la Fig. 1. Le préfiltre P est un filtre de mise en forme du spectre du canal équivalent vu par le décodeur. Les coefficients de P sont réglés de façon à minimiser l'erreur quadratique moyenne $\mathcal{E} = E(e_k^2)$ entre l'entrée du décodeur et la sortie d'un canal désiré H . Le décodeur utilise les coefficients de H pour, à partir de la sortie de P , fournir une estimation $\hat{a}(D)$ de la suite émise $a(D)$. L'entrée du décodeur s'écrit :

$$z(D) = a(D) h(D) + e(D) \quad (4)$$

$e(D)$ est considéré comme un bruit par le décodeur ; la minimisation de \mathcal{E} est donc équivalente à la maximisation du rapport signal sur bruit à l'entrée du décodeur.



RECEPTION DE DONNEES PAR UN CODEUR DE COMPLEXITE
REDUITE EN PRESENCE DE BRUIT ET D'INTERFERENCE ENTRE SYMBOLE

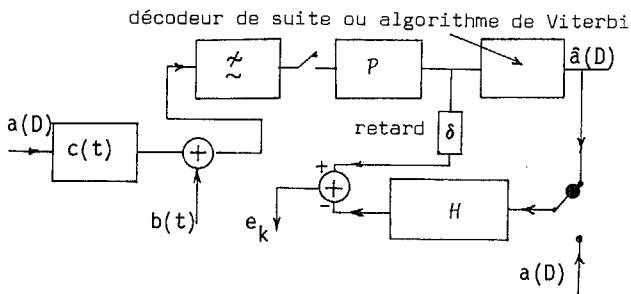


Fig 5. Schéma pratique

Falconer et Magee [3] ont calculé les coefficients optimaux de P et H et proposé d'imposer à H une longueur de mémoire L petite pour que le décodeur utilisant l'algorithme de Viterbi soit de complexité réduite. On montre [11] que \mathcal{E} est une fonction décroissante de L . La réduction de L entraîne donc l'augmentation de \mathcal{E} . L'égaliseur de Falconer et Magee [3], usuellement reconnu comme la réalisation pratique de l'égalisation par l'algorithme de Viterbi, pâtit donc de ce dilemme : L doit être à la fois grand, pour diminuer \mathcal{E} , et petit, pour réduire la complexité de l'algorithme de Viterbi.

La complexité réduite du décodeur de suite par rapport à celle de l'algorithme de Viterbi est obtenue en imposant au décodeur de ne conserver en mémoire qu'un nombre réduit de survivants. La situation est donc essentiellement différente de celle de l'égaliseur de Falconer et Magee ; le décodeur de suite permet d'augmenter L sans augmenter le nombre de survivants utilisés. Cependant, il n'est pas souhaitable de choisir L très grand car ceci entraînerait une complication du calcul des métriques, cf.(3).

La mise en forme spectrale réalisée par P entraîne l'augmentation de la distance minimale entre les mots de code engendrés par le canal, ce qui implique une amélioration des performances de l'égaliseur. Ceci est confirmé par les résultats de simulation reportés sur la Fig. 6 qui montre l'amélioration des performances du décodeur de suite due à l'utilisation de P à 15 coefficients tel que $L=5$.

La Fig. 6 permet également de comparer un décodeur de suite avec préfiltre utilisant 4 survivants, d'une part à un égaliseur de Falconer et Magee utilisant 16 survivants, i.e. quatre fois plus complexe, et d'autre part à un égaliseur récursif composé d'un préfiltre à 15 coefficients et d'un circuit de rebouclage à 5 coefficients. Les simula-

tions ont été effectuées pour un canal à haute distortion de coefficients (0,227 ; 0,46 ; 0,688 ; 0,46 ; 0,227). Des suites de 10 000 symboles appartenant à l'alphabet $\{ 3, 1, -1, -3 \}$ ont été utilisées. Contrairement à ce que l'on pourrait attendre, l'égaliseur de Falconer et Magee est ici moins performant que l'égaliseur récursif. La cause en est que le canal simulé est très distordu et qu'un canal désiré utilisé tel que $L=2$ n'est pas suffisant pour égaliser le zéro double présent dans le spectre. En revanche, l'égaliseur récursif dispose d'un canal désiré tel que $L=5$, qui permet de réduire sensiblement \mathcal{E} . Falconer et Magee ont constaté la même dégradation des performances de leur égaliseur par rapport à celles de l'égaliseur récursif dans une simulation faite pour un canal à haute distortion [12].

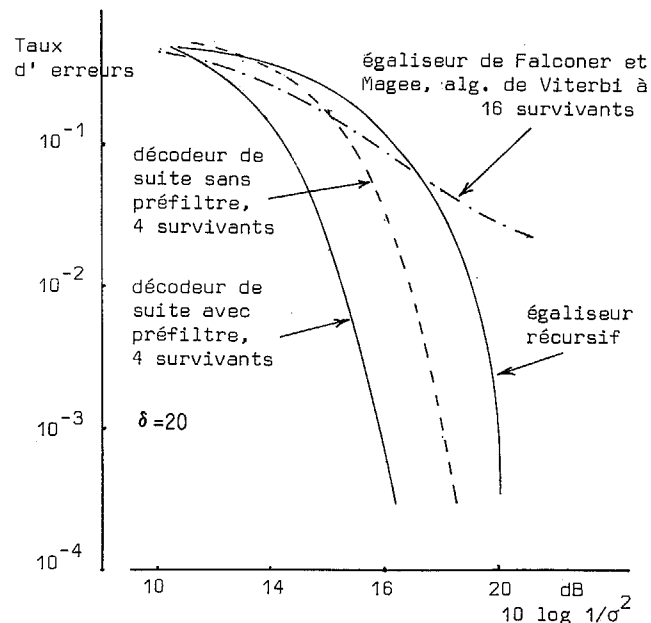


Fig. 6. Comparaisons

RÉFÉRENCES

- [1] Forney, IEEE, vol. IT-18, N° 3, pp. 363-378, May 1972.
- [2] Qureshi and Newhall, IEEE, vol. IT-19, N° 4, pp. 448-457, July 1973.
- [3] Falconer and Magee, BSTJ vol. 52, N° 9, pp. 1541 - 1562, Nov. 1973
- [4] Fredricsson, IEEE Conf. Inform. Theory, Suède 1976.
- [5] Vermeulen and Hellman, Intern. Conf. Commun., Minneapolis, Minn. June 1974.
- [6] Vermeulen, F. L., " Low Complexity Decoders for Channels with Intersymbols Interference ", Ph. D. Dissertation, Dept. Elect. Eng. Stanford Univ., August 1975.
- [7] Foschini, " Theory of Fast Detection of Digital Data ", à paraître.
- [8] Jelinek, IBM J. Res. Develop., vol. 13, pp. 675-685, Nov. 1969.
- [9] Zigangirov, Probl. Peredach. Inform., vol. 2, N° 4, pp. 13-16, 1966.
- [10] Haccoun and Fergusson, IEEE, Vol. IT-21, Nov. 1975.
- [11] Cantoni and Kwong, IEEE, vol. IT-20, Nov. 1974.
- [12] Falconer and Magee, IEEE, vol. COM-24, Nov. 1976.