

# Décodage séquentiel des codes convolutionnels de taux de codage élevés (1)

Sequential decoding of high rate convolutional codes



**David HACCOUN**

Département de Génie Électrique, École Polytechnique de Montréal, Montréal, CANADA, H3C 3A7.

David Haccoun est diplômé en Génie Physique de l'École Polytechnique de Montréal (1965), a obtenu une Maîtrise en Sciences (Génie Électrique) du Massachusetts Institute of Technology, Cambridge, USA en 1966, et un Doctorat es Sciences (Ph. D.) de l'Université McGill, Montréal, en 1974.

Depuis 1966, il est professeur au Département de Génie Électrique de l'École Polytechnique de Montréal, où il est chef de la section Communication (1987-...), et où il fut chef de la section Communication et Informatique (1980-1982). En 1984-1985, il fut Professeur de Recherche Invité à l'Université Concordia (Montréal). Ses recherches portent sur la théorie des communications, la théorie de l'information et du codage, les communications mobiles et les communications par satellite. Il est le co-auteur de l'ouvrage **Digital Communications by Satellite**, John Wiley, New York, 1981 (ouvrage traduit en Japonais et publié à Tokyo en 1984).

M. Haccoun agit comme consultant auprès d'industries et d'agences gouvernementales canadiennes, est membre de l'Ordre des Ingénieurs du Québec, de Sigma Xi et membre sénior de IEEE. Il fut membre du comité d'organisation et trésorier du Symposium International de Théorie de l'Information IEEE (Saint-Jovite, Canada, 1983), et président fondateur de la Société Canadienne de Théorie de l'Information (Montréal 1986-87).

## RÉSUMÉ

Une variante de l'algorithme à pile du décodage séquentiel des codes convolutionnels de taux élevés est présentée. La procédure consiste essentiellement à établir et à utiliser un seuil de rejet pour éliminer certaines branches inutiles à chaque cycle d'extension d'un nœud. Des résultats de simulation à l'ordinateur avec des codes de taux  $R=2/3$  et  $3/4$  montrent que des économies substantielles d'espace mémoire du décodeur peuvent être réalisées sans porter atteinte à la performance d'erreur ou à l'effort de calcul du décodeur.

## MOTS CLÉS

Codage convolutionnel, décodage séquentiel, algorithme à pile, Zigangirov-Jelinek, seuils de rejets.

## SUMMARY

*A modified version of the stack algorithm of sequential decoding is presented for the decoding of high rate convolutional codes. It is based on the use of a rejection threshold to discard some of the unlikely branch extensions at each node extension of the algorithm. Using computer simulation with the best known codes of rates  $2/3$  and  $3/4$ , we show that large savings in the stack memory of the decoder may be achieved at hardly any degradation of either the error performance or the computational effort.*

## KEY WORDS

*Convolutional encoding, sequential decoding, stack algorithm, Zigangirov-Jelinek, rejection threshold.*

(\*) Cette recherche a été subventionnée en partie par le Conseil National de Recherche en Sciences Naturelles et en Génie du Canada et par le Fonds FCAR du Québec.

### 1. Introduction

Dans les systèmes de communications numériques avec codage correcteur d'erreur, un problème important consiste à développer des techniques de codage et de décodage pouvant délivrer une performance d'erreur donnée avec des décodeurs de complexité acceptable.

Dans les canaux de communication sans mémoire comme ceux employés dans les liaisons par satellite, les systèmes utilisant le codage convolutionnel avec décodage probabiliste sont parmi les plus intéressants tant du point de vue de leur performance d'erreur (s'approchant le plus des performances ultimes prévues par la théorie de Shannon) que du point de vue de leur réalisation et implantation matérielle. Les deux principales techniques de décodage probabiliste des codes convolutionnels sont le décodage séquentiel [1] et le décodage de Viterbi [2, 3]. Chacune de ces techniques consiste à trouver un chemin particulier (le message transmis) dans un graphe orienté où on assigne aux branches des « métriques » ou valeurs de vraisemblance entre les données reçues du canal de transmission et les données qui auraient pu être transmises. L'objectif général du décodeur se résume donc à déterminer avec la plus grande fiabilité et le minimum d'efforts le chemin de métrique maximale. Ce chemin est la séquence décodée.

Au taux de codage  $R = 1/2$  le décodage de Viterbi [2] peut fournir un gain de codage de quelque 5 dB à une probabilité d'erreur par bit de  $10^{-5}$ , alors que le décodage séquentiel avec quantification pondérée permet d'atteindre aisément des gains de 7 à 8 dB [3]. Ces gains de codage peuvent se traduire soit par une diminution équivalente de la puissance de transmission de l'émetteur, soit par une augmentation du taux de transmission des données, ou bien sûr, par une combinaison des deux. Dans certaines applications telles les communications numériques par satellite, l'utilisation de ces techniques de codage et de décodage peut s'avérer particulièrement attrayante [3, 4].

Cet article ne traite que de décodage séquentiel. On y présente le décodage de codes convolutionnels de taux élevés,  $R = (V-1)/V$ , ( $R = 2/3, 3/4, \dots$ ), plutôt que de taux faibles  $R = 1/V$ , ( $1/2, 1/3, \dots$ ) habituellement considérés. L'utilisation de ces codes de taux

élevés s'avère particulièrement indiquée lorsque la largeur de bande de transmission du canal est quelque peu limitée, puisque l'expansion de largeur de bande requise n'est que  $V/(V-1)$  au lieu d'être égale à  $V$  pour les codes de taux  $R = 1/V$ .

On présente des variantes de l'algorithme à pile de Zigangirov-Jelinek du décodage séquentiel qui permettent de diminuer de façon substantielle la complexité du décodage ainsi que les exigences de mémoire du décodeur, au prix d'une faible dégradation de la performance d'erreur. La technique est évaluée par simulation à l'ordinateur sur des codes de taux  $2/3$  et  $3/4$ .

On rappelle au lecteur les notions de base du codage convolutionnel et de la structure en arbre, notions essentielles aux opérations de décodage séquentiel. Après une brève introduction à la section 2 de l'algorithme à pile du décodage séquentiel (algorithme particulièrement bien adapté au décodage des codes de taux  $R = 1/V$ ), des modifications de cet algorithme pour le rendre efficace au décodage des codes de taux  $b/V$  sont présentées à la section 3. Les résultats de simulation sont données à la section 4 et démontrent les économies d'espace mémoire du décodeur que l'on peut réaliser avec des codes de taux  $2/3$  et  $3/4$ . Enfin, une simplification permettant d'améliorer d'avantage la technique est proposée à la section 5.

### 2. Codage convolutionnel et décodage séquentiel

CODAGE CONVOLUTIONNEL DE TAUX  $R = b/V$

Un codeur convolutionnel binaire de taux  $R = b/V$  peut être représenté par une machine séquentielle linéaire à états finis composée d'un registre à décalage de longueur  $K$  bits et de  $V$  additionneurs modulo-2 connectés à certaines cellules du registre à décalage. L'ensemble des connexions entre le registre à décalage et les additionneurs spécifie le code. Les données à encoder sont introduites  $b$  bits à la fois, provoquant un décalage de  $b$  positions du contenu du registre à décalage, et produisant  $V$  symboles codés,  $V > b$ , qui sont recueillis à la sortie des  $V$  additionneurs modulo-2. Le taux de codage est donc  $R = b/V$  bits/symbole.

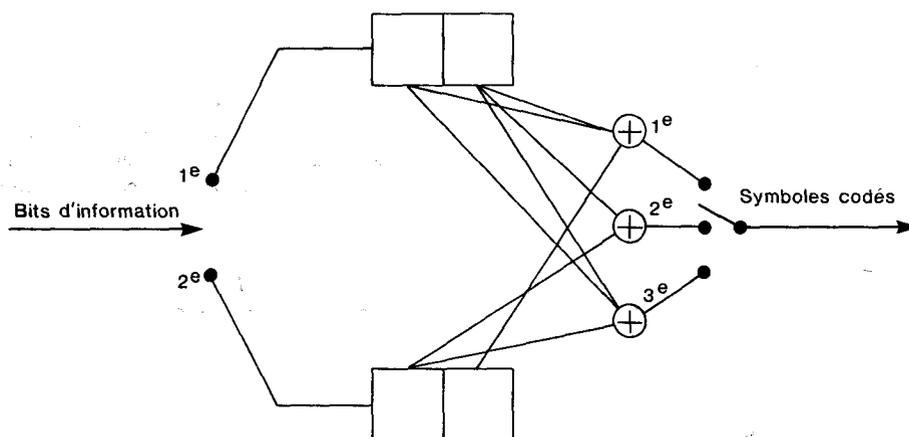


Fig. 1. — Schéma de codeur convolutionnel de taux  $R = 2/3$ .  $K = 4$ ,  $V = 3$ ,  $b = 2$ ,  $k_1 = k_2 = 2$ .

Dans cette représentation par registre à décalage, la longueur de contrainte du code (en bits d'information) est définie comme étant le nombre total de bits dans le registre qui peuvent influencer la sortie du codeur; c'est donc la longueur totale  $K$  du registre à décalage, L'état du codeur est spécifié par le contenu des  $(K - b)$  premières cellules du registre à décalage [2, 3].

La longueur  $K$  n'étant pas nécessairement un multiple de  $b$ , une représentation en parallèle du codeur est souvent préférable. Le codeur est alors composé de  $b$  registres à décalage de longueurs  $k_i, i=1, 2, \dots, b$ , connectés aux  $V$  additionneurs modulo-2. Les données sont alors introduites un bit à la fois mais simultanément dans les  $b$  registres à décalage. Naturellement, dans ce cas, chaque registre n'est décalé que d'un seul bit. Dans cette représentation, la longueur de contrainte du code est encore définie comme étant le nombre total de cellules des  $b$  registres à décalage.

A chaque cycle d'encodage, aux  $b$  bits d'information on fait correspondre  $2^b$  branches d'un arbre, chacune d'elle codée en  $V$  symboles. Par conséquent, une séquence de  $Lb$  bits d'information va déterminer dans l'arbre d'encodage un chemin particulier de longueur  $L$  branches et codé en  $LV$  symboles. Un exemple de codeur convolutionnel de longueur de contrainte  $K=4$  et de taux  $R=2/3$  est montré à la figure 1, et son arbre d'encodage à la figure 2.

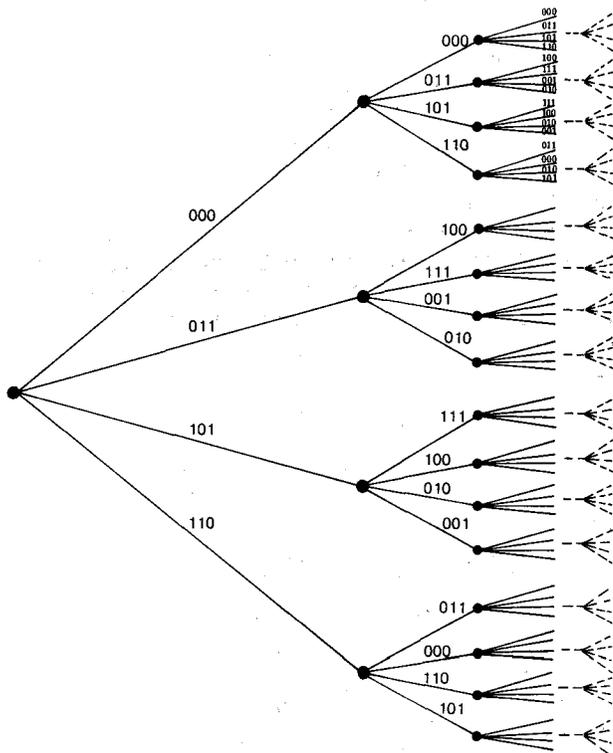


Fig. 2. — Arbre d'encodage du codeur de la figure 1.

**DÉCODAGE SÉQUENTIEL**

L'idée de base du décodage séquentiel consiste à décoder la séquence reçue du canal une branche à la fois, mais sans explorer tout l'arbre d'encodage. L'exploration de l'arbre est effectuée le long du chemin ayant la métrique ou vraisemblance courante maximale.

Partant du nœud origine de l'arbre le chemin devant être exploré d'un pas en avant est celui dont la métrique est la plus élevée parmi tous les chemins déjà examinés. Chaque pas en avant est une prolongation d'un nœud en tous ses  $2^b$  descendants, et le chemin qui atteint le premier un nœud terminal de l'arbre est accepté comme étant le chemin décodé.

En cours d'opération, le décodeur peut revenir en arrière et prolonger des chemins antérieurs, et possiblement incorrects. Afin de minimiser ces retours en arrière et l'exploration de chemins incorrects, la fonction de vraisemblance utilisée, appelée métrique de Fano croît en moyenne le long du chemin correct et décroît en moyenne sur tous les chemins incorrects [1, 6, 7].

La métrique de symbole est donnée par

$$(1) \quad \mu_i = \log_2 \frac{P(y_i | x_i)}{P(y_i)} - R$$

où  $x_i$  est le symbole codé émis dans le canal,  $y_i$  est le symbole reçu correspondant,  $P(y_i | x_i)$  la probabilité de transition du canal pour les symboles  $x_i$  et  $y_i$ , et  $R$  le taux de codage. Dans un canal sans mémoire la métrique est additive le long des symboles de branche d'un même chemin, de sorte que pour des séquences de longueur  $L$  branches ayant chacune  $V$  symboles codés, la métrique totale est donnée par

$$(2) \quad \Gamma = \sum_{i=1}^{LV} \mu_i$$

L'objectif du décodeur consiste donc à déterminer et à poursuivre le chemin de l'arbre d'encodage ayant la métrique (2) la plus élevée. Cette métrique s'applique aussi bien aux canaux à quantification ferme qu'à ceux à quantification pondérée permettant ainsi un gain additionnel de quelque 2 dB sur tout gain de codage obtenu sur un canal à quantification ferme [2, 3].

Il existe deux principaux algorithmes de décodage séquentiel : l'algorithme de Fano [1] et l'algorithme de Zigangirov-Jelinek appelé aussi algorithme à pile [5]. Seul l'algorithme à pile sera considéré ici. Avec cet algorithme le décodeur consiste essentiellement en une liste ou pile où sont stockés les chemins explorés. Le stockage des chemins est effectué par ordre décroissant des valeurs de leurs métriques. Le sommet de la pile contient le chemin de métrique maximale courante et sera donc prolongé en tous ses descendants sur une profondeur égale à une branche. L'algorithme se compose des trois étapes suivantes :

- 1° Calcul des métriques des successeurs du sommet de la pile et insertion des nouvelles extensions dans la pile.
- 2° Retrait de la pile du nœud sommet venant d'être prolongé.
- 3° Détermination du nouveau sommet de la pile. Si c'est un nœud terminal l'algorithme arrête. Sinon, retour à 1°.

Quel que soit l'algorithme utilisé, en décodage séquentiel le nombre de calculs effectués par bit décodé  $C$  est en moyenne très faible mais aussi très variable,

avec une fonction de répartition qui suit asymptotiquement une loi de type Pareto, c'est-à-dire

$$(3) \quad P(C \geq N) \approx AN^{-\alpha}, \quad N \gg 1$$

où  $A$  est une constante, et où le paramètre  $\alpha$ ,  $\alpha \geq 0$  appelé exposant de Pareto ne dépend que du canal et du taux de codage [1, 8]. Cette variabilité de calculs est sans contredit l'inconvénient principal du codage séquentiel, et certaines méthodes ont été élaborées pour la diminuer [9-11].

Pour des taux de codage  $R$  inférieurs à un taux limite dénoté  $R_{\text{comp}}$ , le nombre moyen de calculs par bit décodé  $C_{AV}$  est très faible, à peine supérieur à 1 pour l'algorithme à pile. Cependant à des taux  $R$  supérieurs à  $R_{\text{comp}}$ ,  $C_{AV}$  devient théoriquement non borné, entraînant un effort de calcul considérable, voire même une impossibilité à atteindre un nœud terminal de l'arbre. Le taux  $R_{\text{comp}}$  est appelé *taux de coupure* du décodage séquentiel et correspond à un exposant de Pareto  $\alpha$  égal à 1.  $R_{\text{comp}}$  ne dépend que du canal et se calcule facilement [3, 12]. Aussi d'un point de vue pratique, un paramètre important de fonctionnement de décodeur séquentiel est le rapport  $R/R_{\text{comp}}$  qu'on choisit légèrement inférieur mais non égal à 1.

L'effort moyen de calcul  $C_{AV}$  est quasi indépendant de la longueur de contrainte du code  $K$ , alors que la probabilité d'erreur décroît exponentiellement avec  $K$  [1, 5, 12, 13]. Par conséquent des codes de grandes longueurs de contrainte ( $K > 20$ ) peuvent facilement être utilisés en décodage séquentiel. ce qui n'est pas le cas du décodage de Viterbi où en pratique  $K$  est limité à des valeurs inférieures à 8. Avec l'algorithme à pile l'effort de calcul se traduit directement en exigences de stockage dans la pile. Chaque calcul effectué par le décodeur nécessite deux entrées dans la pile pour les codes de taux  $1/V$ , mais  $2^b$  entrées pour les codes de taux  $b/V$ . Par conséquent, ces exigences de stockage rendent très vite l'algorithme à pile inutilisable pour les codes de taux élevés. Des méthodes pour pallier à cette difficulté sont présentées ci-après.

### 3. Algorithme à pile avec seuils

Comme un décodeur séquentiel n'explore qu'une très petite partie de l'arbre d'encodage, on peut supposer que l'algorithme n'aura pas à examiner *toutes* les  $2^b$  extensions d'un même nœud. Utilisant la notion de seuil de rejet pour éliminer quelques-unes des branches les moins vraisemblables, on montre que la taille de la pile et le nombre moyen de calculs peuvent être substantiellement réduits sans conséquence appréciable sur la probabilité d'erreur. Avec cette modification à l'algorithme à pile, le décodage séquentiel devient attrayant même pour les codes de taux élevés.

#### DÉTERMINATION DES SEUILS

Le principe de base consiste à établir un seuil de rejet de branches afin d'obtenir une grande réduction du nombre d'entrées dans la pile, mais sans porter atteinte à la performance d'erreur. Après l'extension d'un nœud en ses  $2^b$  branches, la métrique de chacune

de ces  $2^b$  extensions est comparée à ce seuil. Seules les extensions dont la métrique est supérieure au seuil sont stockées dans la pile; les autres extensions sont éliminées. Naturellement la valeur du seuil doit être choisie pour éliminer le plus grand nombre de branches possibles sans éliminer celles correspondant au chemin correct. Cette valeur dépend du canal, du taux de codage est d'un critère de rejet. La réjection étant basée sur les valeurs des métriques de branches, on calcule d'abord toutes les valeurs des accroissements de métriques de branches et leurs probabilités correspondantes. Soit  $T$  la valeur d'un seuil de rejet et soit  $P_T$  un critère de rejet, c'est-à-dire la probabilité que la métrique  $j$  d'une extension soit inférieure à  $T$ . Alors toute branche  $j$ ,  $j=1, 2, \dots, 2^b$ , sera éliminée si  $\mu_j < T$ .

Pour les canaux à quantification pondérée le nombre de valeurs de métriques de branches est en général élevé, et s'étend sur une grande plage de valeurs. Ce nombre dépend du rapport signal à bruit  $E_b/N_0$ , du nombre de niveaux de quantification et du taux de codage  $R$ , et donc du rapport  $R/R_{\text{comp}}$  où  $R_{\text{comp}}$  est le taux de coupure du canal. Par exemple pour un code de taux  $R=2/3$  et un canal quantifié à 3 bits avec  $E_b/N_0=3,36$  dB correspondant à  $R/R_{\text{comp}}=0,983$ , on compte 75 valeurs différentes de métriques de branche s'étendant de +9 à -225. Pour un code de taux  $R=3/4$ , un canal encore quantifié à 3 bits avec  $E_b/N_0=3,75$  dB et  $R/R_{\text{comp}}=0,988$ , le nombre de métriques de branche atteint 170 valeurs allant de +8 à -340. Quant au critère de rejet, l'expérience a montré que des valeurs de  $P_T$  comprises entre  $10^{-4}$  et  $10^{-6}$  sont les plus appropriées, les valeurs précises de seuil  $T$  à utiliser étant déterminées par simulation.

Pour les canaux à quantification ferme, ce problème est considérablement simplifié, puisque le nombre de valeurs de métriques de branches différentes est très faible. De plus, les meilleurs codes connus de taux  $R=b/V$  présentent une propriété qui les rend particulièrement attrayants pour notre procédure de rejet [14, 15]. En effet, les sorties des codeurs peuvent être partitionnées en deux ensembles disjoints, correspondant à des branches de poids de Hamming de valeurs paires ou impaires. Les  $2^b$  extensions d'un nœud sont donc toutes de poids pair ou toutes de poids impair. Quant au nombre de branches de poids  $d$  issues d'un

même nœud, il est égal à  $\binom{V}{d}$ . Supposant la séquence nulle transmise (formée de zéros uniquement), les deux ensembles de branches peuvent donc être considérés comme deux patrons d'erreurs possibles pour chaque branche reçue du canal : un patron d'erreurs paires (y compris aucune erreur), et un patron d'erreurs impaires. Naturellement ces patrons d'erreurs vont simplifier considérablement la détermination des seuils de rejet et le déroulement de l'algorithme de décodage.

Par exemple pour les codes de taux  $2/3$  et  $3/4$  on voit sur la figure 3 les deux patrons d'erreurs possibles. Les valeurs de seuils peuvent donc être choisies pour éliminer les erreurs simples, doubles, triples, etc., et leurs probabilités correspondantes facilement calculées. Par exemple, considérons un code de taux  $R=2/3$  et un canal binaire symétrique de probabilité

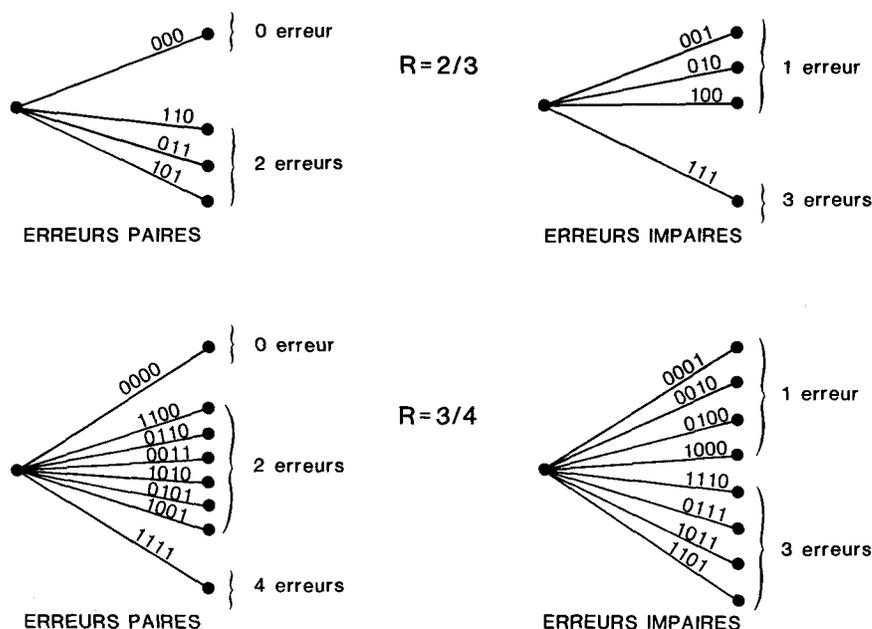


Fig. 3. — Patrons d'erreurs possibles pour les codes de taux 2/3 et 3/4 dans un canal binaire symétrique.

de transition  $p$ . Les probabilités correspondant au schéma de branches de poids pairs sont  $(1-p)^3$  et  $3p^2(1-p)$ , alors que pour les branches de poids impairs ces probabilités sont  $3p(1-p)^2$  et  $p^3$ , correspondant à une et trois erreurs respectivement. Il y a donc pour ce canal et des codes de taux  $R=2/3$  trois seuils de rejet possibles correspondant à l'élimination des erreurs simples, doubles ou triples. Un choix raisonnable du seuil correspond à éliminer les branches portant deux erreurs ou plus, et donc  $P_T = 3p^2(1-p) + p^3$ . Par conséquent, pour un canal à quantification ferme et un code de taux 2/3 un nœud de l'arbre sera prolongé soit selon une seule branche (élimination de trois branches), soit selon trois branches (élimination d'une seule branche). Pour des codes de taux de codage très élevé (par ex.  $R=7/8$ ) on voit que cette technique très simple permet l'élimination d'un nombre considérable de branches et donc de diminuer substantiellement l'effort de décodage ainsi que l'espace mémoire requis par la pile du décodeur séquentiel.

#### 4. Résultats de simulations

Des simulations extensives avec des codes de taux  $R=(V-1)/V$  variant de 2/3 à 7/8 ont été effectuées à l'ordinateur sur des canaux à quantification ferme, et à quantification pondérée à quatre et huit niveaux [16, 17]. Cependant, seuls les résultats obtenus avec les meilleurs codes connus de taux  $R=2/3$  et 3/4 [14, 15] sont présentés ici. Les progiciels utilisés ont été écrits en FORTRAN et exécutés sur un ordinateur IBM 4341 de l'École Polytechnique de Montréal.

Dans ces simulations les données transmises sont groupées en blocs de 500 et 750 bits d'information pour les codes de taux 2/3 et 3/4 respectivement. Chaque bloc est suivi d'une séquence connue du déco-

deur (séquence de zéros) afin de remettre à zéro les registres à décalage du codeur local du décodeur et resynchroniser le système. En cas de débordement de la pile, le bloc courant est éliminé, le système est remis à zéro et le décodage peut se poursuivre pour les blocs suivants.

Au cours des simulations, les comportements de plusieurs paramètres d'intérêt ont été étudiés, en particulier :

$C_{AV}$  = nombre moyen de calculs par nœud décodé.

$N_{AV}$  = nombre moyen d'extensions de branche par nœud prolongé.

$S_{AV}$  = nombre moyen d'entrées dans la pile par bloc décodé.

La figure 4 donne un exemple des variations de ces paramètres en fonction de la valeur du seuil de rejet  $T$  pour un canal à quantification pondérée à huit niveaux avec  $E_b/N_0 = 3,76$  dB et un code de taux  $R=2/3$ . La figure 4 montre que les paramètres  $C_{AV}$ ,  $N_{AV}$  et  $S_{AV}$  passent tous par une valeur minimale pour une valeur de seuil  $T = -100$ , correspondant à une probabilité de rejet  $P_T = 5,5 \times 10^{-5}$ . Au-delà de cette valeur de seuil et jusqu'à la valeur  $T = -267$  correspondant à l'absence de seuil,  $C_{AV}$  ne change pas, indiquant que la présence du seuil de rejet ne provoque pas de calculs supplémentaires. De plus, il a été observé que le nombre d'erreurs de décodage diminue pour  $T$  variant de  $-80$  à  $-100$ , mais demeure inchangé au-delà de  $T = -100$ , démontrant que les branches éliminées par le seuil à la valeur  $-100$ , et *a fortiori* à des valeurs plus faibles, n'appartiennent pas au chemin correct. La conservation de ces branches utilise donc inutilement l'espace mémoire de la pile et alourdit les opérations de décodage sans contribuer à la performance d'erreur du décodeur. Par conséquent dans ce cas la valeur optimale du seuil de rejet est choisie égale à  $-100$ . Une procédure similaire est utilisée pour déterminer les valeurs opti-

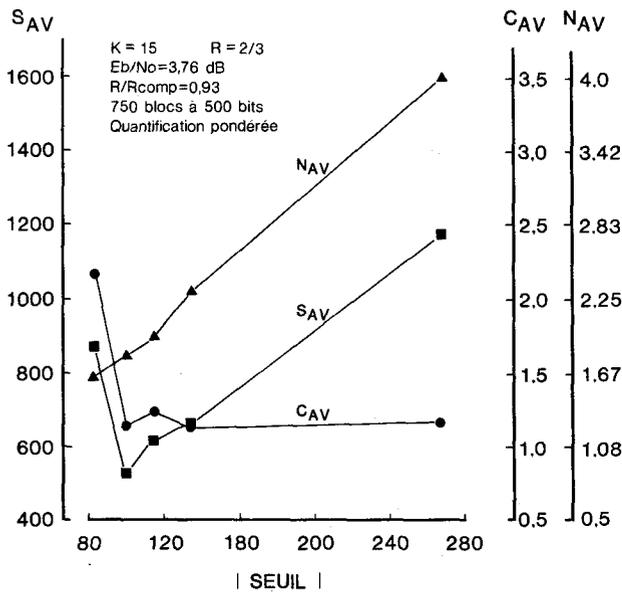


Fig. 4. — Variations des paramètres  $S_{AV}$ ,  $N_{AV}$  et  $C_{AV}$  en fonction du seuil de rejet pour un code  $K=15$ ,  $R=2/3$  sur un canal à quantification pondérée à huit niveaux, à  $E_b/N_0=3,76$  dB,  $R/R_{comp}=0,93$ .

males des seuils pour chaque taux de codage et chaque canal à quantification pondérée.

Les figures 5 et 6 comparent les fonctions de répartition du nombre de calculs effectués par nœud décodé avec et sans seuil de rejet, pour les codes de taux 2/3 et 3/4 respectivement. Tel que déterminé plus haut, le seuil du code de taux 2/3 est fixé à  $-100$ , alors que pour le code de taux 3/4, par une procédure similaire sa valeur optimale a été établie à  $-148$ . Tel que prévu le nombre moyen de calculs  $C_{AV}$  demeure pratiquement inchangé pour les fonctionnements avec et sans seuil, alors que  $N_{AV}$  le nombre d'entrées insérées dans la pile par nœud prolongé est considérablement réduit passant de 4 à 1,134 pour le code de taux 2/3, et de 8 à 2,945 pour le code de taux 3/4. Le

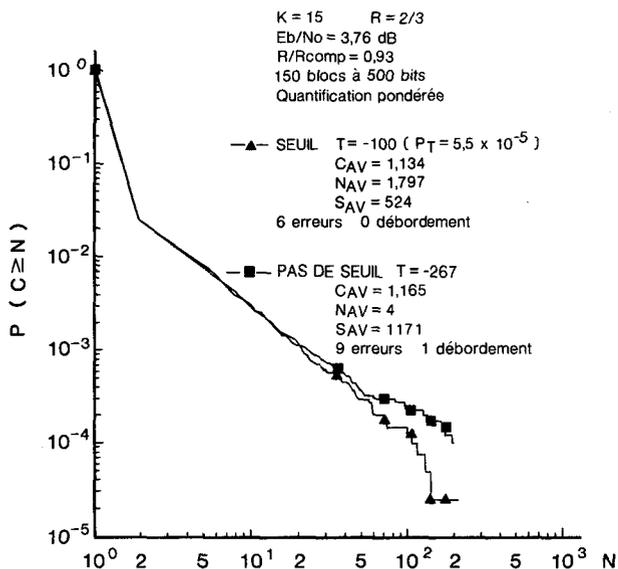


Fig. 5. — Fonctions de répartition empiriques du nombre de calculs par nœud décodé obtenues avec et sans seuil de rejet pour un code  $K=15$ ,  $R=2/3$ , à  $E_b/N_0=3,76$  dB,  $R/R_{comp}=0,93$  et un canal à quantification pondérée à huit niveaux.

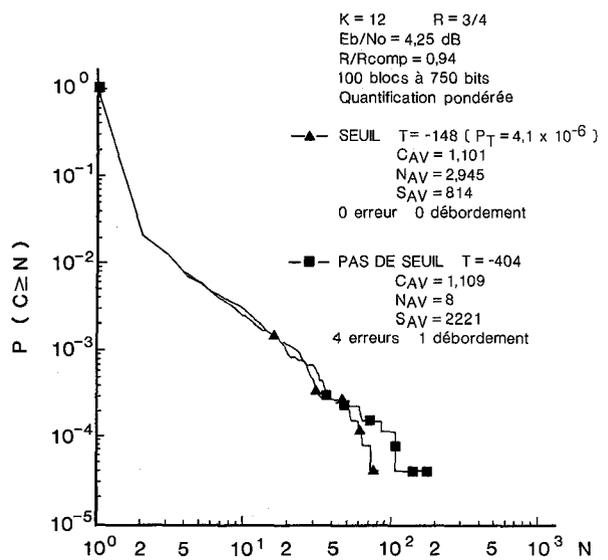


Fig. 6. — Fonctions de répartition empiriques du nombre de calculs par nœud décodé obtenues avec et sans seuil de rejet pour un code  $K=12$ ,  $R=3/4$ , à  $E_b/N_0=4,25$  dB,  $R/R_{comp}=0,94$  et un canal à quantification pondérée à huit niveaux.

nombre moyen d'entrées dans la pile  $S_{AV}$  est réduit dans les mêmes proportions passant de 1171 à 524, et de 2221 à 814 pour les codes de taux 2/3 et 3/4 respectivement. Sachant que chaque entrée dans la pile nécessite plus de 100 bits d'espace mémoire on voit que l'algorithme à seuil permet une économie appréciable d'espace mémoire de pile tout en allégeant les opérations de décodage.

L'examen des figures 5 et 6 montre que les fonctions de répartition  $P(C \geq N)$  sont très peu différentes dans les cas avec et sans seuil de rejet, indiquant que la variabilité de l'effort de calcul n'est pratiquement pas

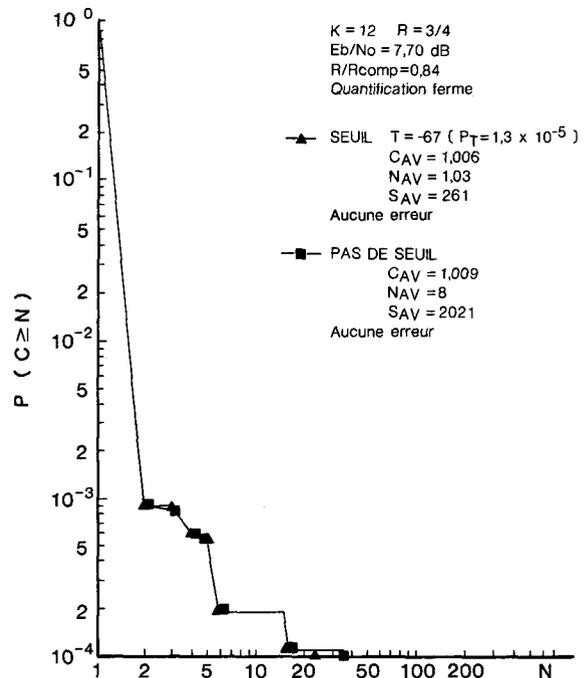


Fig. 7. — Fonctions de répartition empiriques du nombre de calculs par nœud décodé obtenues avec et sans seuil de rejet pour un code  $K=12$ ,  $R=3/4$ , à  $E_b/N_0=7,70$  dB,  $R/R_{comp}=0,84$  et un canal à quantification ferme.

affectée par l'utilisation des seuils de rejet. Quant à la performance d'erreur, de nombreuses simulations ont montré qu'elle est peu ou pas dégradée par le seuil. Enfin, nos simulations ayant été effectuées avec des piles de tailles identiques, le nombre de débordements de pile pour les opérations sans seuil avait tendance à être légèrement supérieur à celui des opérations avec seuil.

Pour des canaux avec quantification ferme la détermination des seuils de rejet est immédiate et les simulations ont été effectuées avec un seuil  $T$  correspondant à l'élimination de toute branche portant deux erreurs ou plus. Les résultats ont montré les mêmes tendances que pour les canaux avec quantification pondérée, mais les économies d'espace mémoire de pile sont encore plus marquées. Par exemple comme le montre la figure 7, pour un code de taux  $3/4$  le nombre moyen d'extensions par nœud est égal à 1,03 au lieu d'être égal à 8 pour un algorithme sans seuil, entraînant une réduction équivalente de la taille de la pile, qui n'utilise que 261 entrées au lieu de 2021. Quant aux fonctions de répartition de l'effort de calcul, encore là, les algorithmes avec seuil de rejet ne montrent pas de différence sensible avec ceux sans seuil.

## 5. Simplification des algorithmes

On observe que pour obtenir des performances d'erreur raisonnables, les valeurs de  $E_b/N_0$  requises pour les codes de taux  $R=b/V$  sont relativement élevées, même pour des rapports de  $R/R_{comp}$  près de 1. Les canaux sont donc de « bonne qualité » avec des probabilités de transition relativement faibles. Par conséquent les erreurs de transmission seront assez rares, et de là, les métriques de branches seront fréquemment positives. Par exemple, on a observé que plus de 96% des branches sont reçues sans erreur avec un code de taux  $3/4$  à  $E_b/N_0=8,25$  dB (correspondant à  $R/R_{comp}=0,82$ ), dans un canal à quantification ferme. Lorsqu'une branche est reçue sans erreur, l'algorithme rejette toutes les extensions de nœud prolongé à l'exception de celle correspondant à la branche « zéro erreur ». Cette branche devant être obligatoirement prolongée à l'étape suivante, son insertion dans la pile devient donc inutile. Exploitant cette observation l'algorithme avec seuil a été modifié pour n'insérer dans la pile que les branches de métriques négatives [16, 17]. Des résultats préliminaires avec des codes de taux  $2/3$  et  $3/4$  sur des canaux à quantification ferme montrent que l'espace mémoire requis par le nouvel algorithme appelé Algorithme à Pile Simplifié n'est plus que 5% environ de celui requis par l'algorithme avec seuils de rejets.

## 6. Conclusions

Dans cet article nous avons montré que par un choix judicieux d'un seuil de rejet qui élimine les chemins inutiles qu'un décodeur séquentiel aurait dû examiner, l'algorithme à pile peut s'avérer très attrayant pour

le décodage des codes convolutionnels à taux élevés. Des économies substantielles d'espace mémoire du décodeur peuvent être réalisées sans porter atteinte à la performance d'erreur ou à l'effort de calcul du décodeur. De plus, la mise en œuvre de la technique est très simple. Ces avantages sont d'autant plus grands que le taux de codage est élevé, rendant l'algorithme de décodage séquentiel à pile particulièrement attrayant dans les canaux limités en largeur de bande.

Bien que cette technique soit applicable avec tout code convolutionnel de taux élevé, en pratique son utilisation est tributaire de la disponibilité de codes bien appropriés au décodage séquentiel : grande longueur de contrainte, distance libre élevée, bon profil des distances, etc. La pénurie de bons codes convolutionnels à taux élevés peut être contournée par l'utilisation de codes dits perforés, avec décodage dans un arbre à deux branches par nœud [4, 18-19]. Cependant, même dans ces cas, la technique des seuils de rejet peut encore être avantageusement utilisée.

*Manuscrit reçu le 1<sup>er</sup> décembre 1986.*

## BIBLIOGRAPHIE

- [1] R. M. FANO, A Heuristic Discussion of Probabilistic Decoding, *IEEE Trans. on Inform. Theory*, IT-9, avril 1963, p. 64-73.
- [2] A. J. VITERBI, Convolutional Codes and Their Performance in Communication Systems, *IEEE Trans. on Com. Tech.*, COM-19, octobre 1971.
- [3] V. BHARGAVA, D. HACCOUN, R. MATYAS et P. NUSPL, *Digital Communications by Satellite*, John Wiley, New York, octobre 1981.
- [4] W. W. WU, D. HACCOUN, R. PEILE et Y. HIRATA, Coding for Satellite Communication, *IEEE Journal on Selected Areas in Communications*, SAC-5, n° 4, mai 1987, p. 724-748.
- [5] F. JELINEK, A Sequential Decoding Algorithm Using a Stack, *IBM Journal of Research and Development*, 13, novembre 1969.
- [7] D. HACCOUN, A Markov Chain Analysis of the Sequential Decoding Metric, *IEEE Trans. on Inform. Theory*, IT-26, n° 1, janvier 1980, p. 109-113.
- [6] J. L. MASSEY, Variable-Length Codes and the Fano Metric, *IEEE Trans. in Inform. Theory*, IT-18, Janv. 1972, p. 196-198.
- [8] I. M. JACOBS et E. R. BERLEKAMP, A Lower Bound to the Distribution of Computation for Sequential Decoding, *IEEE Trans. on Inform. Theory*, IT-13, avril 1967, p. 167-174.
- [9] D. HACCOUN et M. FERGUSON, Generalized Stack Algorithms for the Decoding of Convolutional Codes, *IEEE Trans. on Inform. Theory*, IT-21, novembre 1975, p. 638-651.
- [10] R. P. CHEVILLAT et D. J. COSTELLO, A Multiple Stack Algorithm for Erasure-free Decoding of Convolutional Codes, *IEEE Trans. on Comm.*, COM-25, décembre 1977, p. 1460-1470.
- [11] D. HACCOUN, Variabilité de calculs et débordements de décodeurs séquentiels à pile, *Traitement du signal*, 3, n° 3, décembre 1986, p. 127-143.
- [12] J. M. WOZENCRAFT et I. M. JACOBS, *Principles of Communications Engineering*, J. Wiley, New York, 1965.

- [13] D. HACCOUN, A Branching Process Analysis of the Average Number of Computations of the Stack Algorithm, *IEEE Trans. on Inform. Theory*, IT-30, n° 3, mai 1984, p. 497-508.
- [14] E. PAASKE, Short Binary Convolutional Codes with Maximal Free Distance for Rates 2/3 and 3/4, *IEEE Trans. on Inform. Theory*, IT-20, sept. 1974, p. 683-686.
- [15] R. JOHANNESSON et E. PAASKE, Further Results on Binary Convolutional Codes with an Optimum Distance Profile, *IEEE Trans. on Inform. Theory*, IT-24, mars 1978, p. 264-268.
- [16] Y. FEQUIERE, Simplification du décodage séquentiel des codes convolutionnels de taux élevés, *Mémoire M. Sc.* A., École Polytechnique de Montréal, déc. 1986.
- [17] D. HACCOUN et CHEN NAIYUN, Variants of the Stack Algorithm for the Decoding of High Rate Codes by Sequential Decoding, *1st Canadian Domestic and International Satellite Communications Conference*, Ottawa, juin 1983, p. 21.4.1-21.4.5.
- [18] D. HACCOUN et G. BEGIN, Codage et décodage séquentiel de codes convolutionnels perforés, 11<sup>e</sup> colloque *GRETSI*, Nice, juin 1987, p. 221-224.
- [19] G. BEGIN et D. HACCOUN, High Rate Punctured Convolutional Codes : Structure Properties and Construction Technique, *IEEE Trans. on Com.*, juin 1987, soumis.