

**CODAGE ET DECODAGE SEQUENTIEL DE CODES  
CONVOLUTIONNELS PERFORES**

par David Haccoun et Guy Bégin

Département de génie électrique, Ecole Polytechnique de Montréal  
C.P. 6079, Succ. "A", Montréal, Québec, Canada, H3C 3A7

**RESUME**

Dans cet article on présente le codage et le décodage séquentiel des codes convolutionnels perforés de taux de codage  $R=b/V$ . Ces codes de taux élevés sont facilement construits par élimination périodique de certains symboles codés à la sortie d'un codeur convolutionnel de taux  $R=1/V_0$ . Par rapport aux codes traditionnels de même taux et de même mémoire, ces codes offrent de grands avantages des points de vue codage et décodage sans dégradation marquée des performances d'erreur.

**ABSTRACT**

Coding and sequential decoding of punctured convolutional codes of coding rates  $R=b/V$  is presented. These high-rate codes are easily constructed by periodic elimination of certain coded symbols from the output of a low rate encoder. Compared to traditional codes having the same high rates and same memory lengths, punctured codes feature reduced coding and decoding complexities at hardly any cost in error performance.

**1. INTRODUCTION**

Un intérêt marqué se manifeste actuellement pour l'utilisation de codes convolutionnels performants à taux de codage  $R$  élevés,  $R=b/V$ ,  $1 < b < V$ , (par ex.  $R=2/3$ ,  $3/4$ , ...  $7/8$ , ...), surtout dans les applications où la largeur de bande est quelque peu limitée. En effet, pour ces codes l'expansion de largeur de bande de transmission requise n'est que  $(V/b)$  au lieu d'être égale à  $V$  comme pour les codes de taux  $R=1/V$ . Pour les codes convolutionnels de taux  $R=1/V$ , deux branches portant chacune  $V$  symboles codés émergent de chacun des noeuds de l'arbre ou du treillis d'encodage, alors que pour les codes de taux  $R=b/V$ , chaque noeud donne naissance à  $2^b$  branches. Comme les techniques traditionnelles de décodage de Viterbi ou séquentiel [1] - [4] consistent en une exploration de chemins de l'arbre ou du treillis d'encodage, l'application directe de ces techniques au décodage des codes de taux élevés conduit à une augmentation substantielle de l'effort de calcul, et devient très vite inacceptable. L'utilisation de codes dits *perforés* permet de contourner ces difficultés. On montre que de bons codes perforés de taux de codage  $R=(V-1)/V$  peuvent être facilement construits à partir des meilleurs codes de taux  $R=1/V$ . De plus leur décodage par les techniques de Viterbi ou séquentiel n'est guère plus complexe que pour les codes de taux  $1/V$ . Enfin les codes perforés peuvent être une solution très attrayante au problème du codage et décodage à taux variables.

On supposera le lecteur familier avec les notions fondamentales du codage convolutionnel et des techniques de décodage de Viterbi et séquentiel [3] - [4]. Après une brève introduction des codes convolutionnels perforés à la section 2, le décodage séquentiel des codes perforés est présenté à la section 3. Le problème de la recherche des bons codes perforés est introduit à la section 4 où quelques bons codes sont fournis. Enfin, des résultats de simulation sont donnés et commentés à la section 5.

**2. CODES CONVOLUTIONNELS PERFORES**

Les codes convolutionnels perforés sont une classe de codes convolutionnels de taux de codage élevé qui ont la particularité d'être obtenus à partir de codes convolutionnels de faible taux  $R=1/V$ , par élimination périodique (c.a.d. par *perforation*) de certains symboles à la sortie du codeur de faible taux [5]. Un code ainsi obtenu dépend du code original de faible taux, ainsi que du nombre et de la position des symboles éliminés ou *perforés*.

A titre d'illustration considérons le code convolutionnel de taux  $2/3$  et de mémoire  $M=2$  dont le treillis est montré à la Figure 1. Ce treillis comporte 4 états distincts desquels émergent 4 branches portant chacune 3 symboles codés.

Considérons à présent un codeur convolutionnel de taux  $R=1/2$  et de mémoire  $M=2$ . Si un symbole choisi parmi 4 symboles successifs est périodiquement éliminé à la sortie de ce codeur, il en résulte alors un code de taux  $2/3$  puisque 3 symboles codés sont portés par 2 branches successives correspondant à 2 bits d'information. Il est donc clair qu'avec un choix judicieux du code de faible taux de codage et du patron de perforation, n'importe quel code de taux de codage élevé peut être, en principe, ainsi obtenu. La Figure 2 montre le treillis d'un code de taux  $1/2$  et de mémoire  $M=2$ , où chaque 4<sup>e</sup> symbole est perforé (indiqué par X sur le diagramme). Examinant ce treillis deux branches à la fois, on voit qu'il est identique à celui de la Figure 1. Un code perforé de taux de codage  $R=2/3$  a donc été obtenu à partir d'un code de taux  $R=1/2$ .

Le principe se généralise facilement. Un code perforé de taux  $R=b/V$  peut être construit à partir d'un code initial de taux  $R=1/V_0$  appelé *code origine*. Il suffit pour cela d'éliminer périodiquement un nombre  $S=(V_0b-V)$  symboles de toutes les  $b$  branches successives du code origine. Le taux de codage du code perforé ainsi obtenu est bien  $R=b/(bV_0-S)=b/V$ .



Naturellement la perforation du code origine entraîne une diminution de sa distance libre, et cette diminution sera d'autant plus importante que le nombre de symboles perforés  $S$  est élevé. Il y a donc avantage à utiliser des codes origine de taux  $R=1/2$ . Un grand nombre de bons codes perforés construits à partir de codes de taux  $1/2$  ont été ainsi proposés pour des codes de taux variant de  $2/3$  à  $16/17$  [5] - [10].

## 2.1 Matrice de perforation

Un code perforé de taux  $R=b/V$  dépend du code origine de taux  $1/V_0$ , du nombre  $b$  de branches regroupées ainsi que du nombre  $S$  et de la position des symboles perforés. L'information de position des symboles perforés peut être commodément spécifiée par une matrice binaire  $[P]$  ayant  $b$  lignes et  $V_0$  colonnes appelées *matrices de perforation*. Les éléments de  $P$  valent 0 ou 1 correspondent à l'élimination ou au maintien des symboles codés du code origine. Il est clair que  $[P]$  correspond au *patron de perforation* utilisé pour générer le code perforé désiré. Par conséquent, pour un même codeur origine plusieurs codes perforés de taux de codage différents peuvent être obtenus en modifiant le patron de perforation, c.a.d. en changeant les dimensions et les éléments de la matrice de perforation  $[P]$ . Par exemple, pour le code de la Figure 1, la matrice de perforation correspondant au code illustré à la Figure 2 est

$$[P] = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

Un codeur de codes perforés peut donc être réalisé par le codeur origine de faible taux suivi d'un perforateur commandé par la matrice de perforation (voir Figure 3).

## 3. DECODAGE PROBABILISTE DES CODES PERFORES

Les codes convolutionnels perforés se prêtent aux mêmes techniques de décodage probabiliste (décodages de Viterbi et décodage séquentiel) que les codes convolutionnels traditionnels. Cependant les difficultés (effort et complexité des calculs, mémoire requise) associées au décodage des codes de taux élevés habituels sont presque totalement éliminés en utilisant des codes perforés.

En décodage séquentiel ou de Viterbi, les opérations de décodage sont effectuées dans l'arbre ou le treillis du code origine de faible taux  $R=1/V_0$ , mais les calculs des métriques des symboles perforés  $y$  sont essentiellement inhibés. Le décodage des codes perforés de taux élevés  $R=b/V$  n'est donc guère plus complexe que celui des codes à faible taux  $R=1/V$  [6] - [10].

L'algorithme de Viterbi a déjà été proposé pour le décodage des codes perforés, ou plus précisément, les codes perforés ont été conçus à l'origine pour réduire la complexité du décodage de Viterbi des codes de taux élevés [5]. Dans cet article, on se limite à l'application du décodage séquentiel aux codes perforés [8] - [10].

## 3.1 Décodage séquentiel des codes perforés

Sachant la séquence reçue du canal, le décodage séquentiel consiste essentiellement en la recherche et la détermination de la séquence émise la plus vraisemblable. Il utilise la structure en arbre du code, et n'explore, un chemin à la fois, que la partie de l'arbre qui paraît être la plus vraisemblable sans cependant explorer l'arbre entier. Cette exploration est effectuée le long du chemin ayant la vraisemblance (ou métrique) la plus élevée parmi tous les chemins explorés. Le chemin qui atteint le premier l'extrémité de l'arbre avec la métrique maximale est accepté comme *chemin décodé*, et pour des raisons pratiques l'arbre est limité à une longueur variant de quelques centaines à quelques milliers de branches.

Tel que mentionné plus haut, la complexité de la procédure de décodage (extension des chemins, calcul des métriques, etc) augmente très rapidement avec le taux  $R$  du code. Par conséquent le décodage séquentiel est surtout réservé aux codes de taux  $R=1/V$ . Pour les codes de taux élevés, la procédure de décodage directe n'est pas attrayante, et certaines variantes de décodage séquentiel ont été proposées pour contourner ces difficultés [11]. Cependant, l'utilisation de codes perforés rend le décodage séquentiel des codes de taux  $R=b/V$  guère plus complexe que celui des codes de taux  $R=1/V$ .

Il existe plusieurs algorithmes de décodage séquentiel et, en principe, ils peuvent tous être adaptés au décodage des codes perforés. Cependant seul l'algorithme de Zigangirov-Jelinek (Z-J), appelé *algorithme à pile* sera considéré ici.

Cet algorithme utilise une *pile* pour stocker toutes les caractéristiques des chemins explorés, et un tampon d'entrée pour stocker les séquences reçues en attente d'être décodées. La pile est une liste ordonnée où sont stockés les chemins explorés par ordre décroissant de leur métrique. Le *sommet* de la pile contient le chemin ayant la métrique maximale courante; ce chemin est donc celui qui sera prolongé. L'algorithme a donc pour objet de déterminer à chaque étape le sommet de la pile et d'en faire le prolongement jusqu'à atteindre un noeud terminal, c'est-à-dire un noeud de profondeur  $L$ , où  $L$  est la longueur de la séquence à décoder [4].

Pour le décodage des codes perforés de taux élevé, l'algorithme à pile ne requiert que des modifications minimales. Sachant le patron de perforation le décodeur avance dans l'arbre une branche à la fois, tout comme pour le décodage du code origine de taux  $R=1/V_0$ . A chaque étape, le sommet de la pile est prolongé en ses deux branches portant chacune  $V_0$  symboles, dont un certain nombre doit être éliminé. Cette élimination des symboles perforés est effectuée en inhibant simplement la valeur de leur métrique, c'est-à-dire en assignant une valeur de métrique constante (par exemple zéro) à ces symboles perforés. L'avantage de la méthode est que la détermination du chemin le plus vraisemblable courant est basée sur les accroissements de métrique de 1 branche plutôt que sur ceux de blocs de  $b$  branches. Par conséquent, les décisions de revenir en arrière et de choisir un meilleur chemin dans l'arbre peuvent être prises plus tôt,



réduisant ainsi la variabilité de l'effort de décodage. De plus, comme chaque prolongation élémentaire du sommet de la pile n'implique que deux branches, pour un code de taux  $R=b/V$  une pénétration de l'arbre implique ici  $2^b$  entrées dans la pile, plutôt que  $2^b$  entrées comme pour le décodage séquentiel habituel. Enfin, la procédure du fanion qui est normalement utilisée en décodage séquentiel pour économiser l'espace mémoire de la pile [4], peut tout aussi facilement être implantée pour les codes perforés.

L'algorithme à pile se prête donc très facilement au décodage séquentiel des codes perforés de taux de codage élevés, et la méthode permet de conduire à une simplification et à des économies substantielles d'espace mémoire par rapport à la technique directe de décodage séquentiel. Enfin, lorsque tous les codes perforés sont issus d'un même code origine de taux  $R=1/V_0$ , un changement de taux de codage n'implique qu'un changement du patron de perforation. Par conséquent, la méthode se prête particulièrement bien pour l'implantation de systèmes à taux de codage variable.

#### 4. CODES PERFORES POUR DECODAGE SEQUENTIEL

La technique des codes perforés ayant été conçue originalement pour le décodage de Viterbi, tous les bons codes perforés de taux de codage élevés disponibles jusqu'à présent, ont une mémoire de longueur relativement faible [5] - [7]. En décodage séquentiel, les bons codes ont une mémoire beaucoup plus élevée, avec une grande distance libre et un bon profil des distances [12] - [13].

Utilisant la même approche que pour les codes perforés de faible longueur de mémoire, on peut utiliser un bon code origine connu de taux  $R=1/2$ , et de mémoire élevée, et générer les codes perforés de taux  $R=b/V$  en essayant de façon exhaustive tous les patrons de perforation. Bien que cette approche puisse fournir des codes perforés raisonnablement bons, une comparaison de ces codes avec les meilleurs codes connus de taux  $R=b/V$  et de même mémoire est limitée aux codes de taux  $R=2/3$  et  $3/4$ . En effet, les meilleurs codes connus pour le décodage séquentiel n'ont pas en général un taux de codage  $R$  supérieur à  $3/4$ .

Le Tableau 1 donne quelques codes perforés de taux  $R=2/3$  et  $3/4$  de distance libre maximale obtenus à partir des meilleurs codes connus de taux  $R=1/2$  et de mémoires  $M$  variant de 9 à 13. Dans la recherche de ces codes, les codes retenus sont ceux ayant à la fois une distance libre maximale et un bon profil des distances. En particulier, les codes du Tableau 1 délivrent le plus petit nombre  $b_{eff}$  de bits en erreur sur les chemins à la distance libre. Utilisant la même technique, les meilleurs codes perforés de taux  $2/3$  et  $3/4$  et de mémoire  $M$  atteignant 23 ont été obtenus [9] - [10]. Une analyse théorique de ces codes a montré que les performances d'erreur ne sont que très légèrement inférieures à celles des meilleurs codes connus de même taux et de même mémoire [9] - [10].

#### 5. RESULTATS DE SIMULATION

L'algorithme à pile pour le décodage de codes perforés de taux  $R=2/3$  et  $3/4$  a été simulé sur ordinateur. Chaque simulation était effectuée à des points d'opération correspondant à des rapports  $R/R_{comp}$  variant de 0.85 à 0.99. Comme on pouvait le prévoir, un très petit nombre d'événements erreurs se sont produits, et pour des codes de mémoire supérieure à  $M=15$  aucun événement erreur n'a été détecté. Par conséquent les résultats des simulations portent essentiellement sur la distribution de l'effort de calcul.

La Figure (4) montre une fonction de répartition typique obtenue avec un code de mémoire 24 et de taux de codage  $R=2/3$ , obtenu par perforation du meilleur code connu de taux  $1/2$  et mémoire 24. Le patron de perforation de ce code est donné par :

$$[P] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Sur la même Figure (4) on a tracé la fonction de répartition de l'effort de calcul, obtenue par décodage séquentiel traditionnel du code de mémoire 24 et de taux  $2/3$  de Johannesson et Paaske [13]. Les simulations ont été effectuées sur un canal binaire symétrique à  $R/R_{comp}=0.99$ . Aucune erreur de décodage n'a été observée.

Un examen des courbes de la Figure (4) montre que les fonctions de répartition sont quasi identiques pour le code perforé et le code normal. Cependant le nombre d'entrées dans la pile est nettement moindre pour le code perforé que pour le code normal. Mais un calcul ne correspondant pas exactement aux mêmes opérations dans les deux cas de décodage, une comparaison plus judicieuse des efforts de calcul peut être basée sur les temps de décodage respectifs des deux algorithmes. On observe alors des temps de décodage nettement plus faibles avec l'algorithme pour codes perforés, qu'avec celui des codes habituels. Pour les codes de taux  $R=2/3$  cette réduction des temps de décodage est de l'ordre de 40%, et augmente avec le taux de codage, démontrant encore l'avantage de la méthode.

#### 6. CONCLUSION

Les codes convolutionnels perforés à taux de codage élevés offrent de grands avantages par rapport aux codes traditionnels de même taux. Les codeurs convolutionnels pour codes perforés de taux  $R=(b/V)$  ne sont guère plus complexes que ceux pour codes de taux  $R=1/V$ . Les techniques de décodage de Viterbi et séquentiel peuvent s'appliquer au décodage des codes perforés de taux élevés sans accroissement de complexité par rapport au décodage des codes de taux  $1/V$ . Par conséquent, dans toutes les applications où l'utilisation du codage convolutionnel à taux de codage élevé est indiquée, l'utilisation des codes perforés offre des avantages marqués et cette solution est d'autant plus attrayante que ce taux est élevé.

#### REFERENCES

- [1] A.J. VITERBI, "Convolutional Codes and their Performance in Communications Systems", IEEE Trans. on Comm. Tech., Vol. COM-19, Oct. 1971.



[2] R.M. FANO, "A Heuristic Discussion of Probabilistic Decoding", IEEE Trans. Inf. Theory, Vol. IT-9, April 1963.

[3] V.K. BHARGAVA, D. HACCOUN, R. MATYAS, P. NUSPL, "Digital Communications by Satellite", J. Wiley, New York, 1981.

[4] D. HACCOUN, "Variabilité de calculs et débordements de décodeurs séquentiels à pile", Traitement du signal, Vol. 3, No.3, 1986, pp. 127-143.

[5] J.B. CAIN, G. C. CLARK and J.M. GEIST, "Punctured Convolutional Codes of Rate  $(n-1)/n$  and Simplified Maximum Likelihood Decoding", IEEE Trans. on Inf. Theory, Vol. IT-25, Jan. 1979.

[6] Y. YASUDA, Y. HIRATA, K. NAKAMURA, S. OTANI, "Development of Variable-Rate Viterbi Decoder and its Performance Characteristics", 6th International conference on digital Satellite Communication, Phoenix, Sept. 1983.

[7] Y. YASUDA, K. KASHIKI, Y. HIRATA, "High-Rate Punctured Convolutional Codes for Soft Decision Viterbi Decoding", IEEE Trans. on com., Vol. COM-32, March 1984, pp. 315-319.

[8] G. BEGIN and D. HACCOUN, "Decoding of Punctured Convolutional Codes by the Stack Algorithm", Book of Abstracts, 1986 IEEE Intern. Symposium on Information Theory, Ann Arbor, Michigan, Oct. 1986.

[9] W.W. WU, D. HACCOUN, Y. HIRATA, R. PEILE, "Coding for Satellite Communications", to appear in the IEEE Journal of Selected Areas in Communications", 1987.

[10] D. HACCOUN, G. BEGIN, "Probabilistic Decoding of Punctured convolutional Codes", to be published in IEEE Trans. on Communications.

[11] D. HACCOUN, C. NAIYUN, "Variants of the Stack Algorithm for the Decoding of High-Rate Convolutional Codes", Proc. Intern. Satellite Comm. Conf., Ottawa, June 1983, pp 21.4.1-21.4.4

[12] R. JOHANNESSON, "Some Long Rate One-Half Binary Convolutional Codes with an Optimal Distance Profile" IEEE Trans. on Inf. Theory, IT-22, pp. 629-631, Sept. 1976.

[13] R. JOHANNESSON, E. PAASKE, "Further Results on Binary Convolutional Codes with an Optimum Distance Profile", IEEE Trans. on Inf. Theory, IT-24, pp. 264-268, March 1978.

[14] K. J. LARSEN, "Short Convolutional Codes with Maximal Free Distance for Rates  $1/2$ ,  $1/3$  and  $1/4$ ", IEEE Trans. on Inf. Theory, Vol. IT-19, pp. 371-372, May 1973.

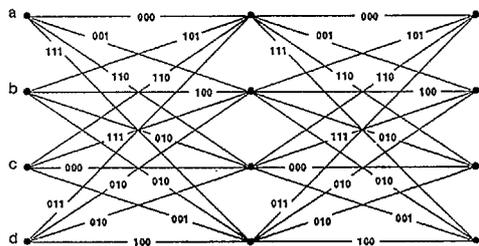


Figure 1: Treillis d'un code R=2/3 M=2

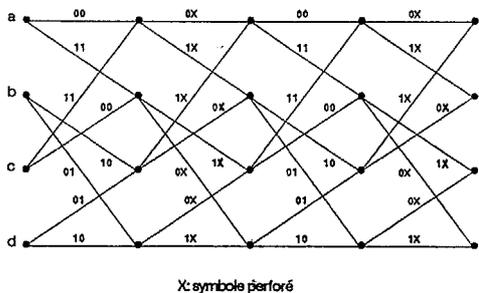


Figure 2: Treillis d'un code perforé équivalent

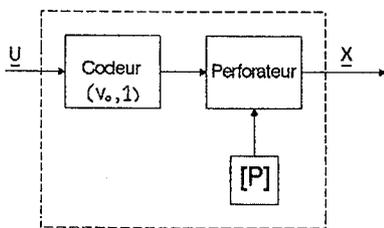


Figure 3: Codeur perforé, vu de l'extérieur.

Tableau 1: Codes perforés de taux R=2/3 et 3/4

CODE ORIGINE R=1/2 <sup>(1)</sup>		CODE 2/3 PERFORE			CODE 3/4 PERFORE					
M	G <sub>1</sub>	G <sub>2</sub>	d <sub>f</sub>	b <sub>df</sub>	[P]	d <sub>f</sub>	b <sub>df</sub>	[P]	d <sub>f</sub>	b <sub>df</sub>
9	1167	1545	12		11 10	7	3	110 101	6	14
10	2335	3661	14	94	10 11	8	8	100 111	6	3
11	4335	5723	15	76	11 10	9	51	100 111	7	34
12	10533	17661	16	152	11 10	9	25	101 110	8	47
13 <sup>(2)</sup>	21675	27123	16	22	11 10	10	32	100 111	8	4

Notes : (1) Meilleurs codes de Johannesson et Paaske [13]  
(2) Code de Larsen [14]

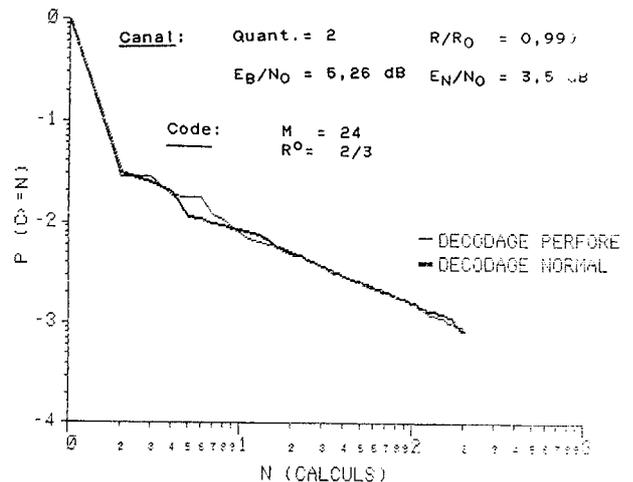


Figure 4: Cumulative P(C >= N) du code perforé R=2/3, M=24