

# Nouveau décodeur à complexité réduite pour codes convolutifs de rendement 1/2

J.-C. DANY<sup>1</sup>, J. ANTOINE<sup>1</sup>, L. HUSSON<sup>1</sup>, N. PAUL<sup>2</sup>, A. WAUTIER<sup>1</sup>, J. BROUET<sup>2</sup>

<sup>1</sup>Ecole Supérieure d'Electricité, Radio Dpt., Plateau de Moulon, 91192 Gif-sur-Yvette, France,

dany@supelec.fr, antoine@supelec.fr, husson@supelec.fr, wautier@supelec.fr

<sup>2</sup>Alcatel Research and Innovation, Route de Nozay, 91461 Marcoussis cedex, France,

nicolas.paul@alcatel.fr, jerome.brouet@alcatel.fr

**Résumé** – Les codes convolutifs peuvent être décodés de façon optimale à l'aide l'algorithme de Viterbi (VA). Nous proposons un décodeur à entrée souple dans lequel l'algorithme de Viterbi est employé pour identifier le vecteur d'erreur plutôt que le message d'information avec une métrique appropriée. Ce type de décodage permet d'éviter la mise en oeuvre d'un nombre important d'opérations ACS (Add Compare Select). Nous montrons que les performances atteintes sont proches de l'optimum tout en bénéficiant d'une réduction de la complexité qui est d'autant plus importante que le rapport signal à bruit (SNR) est favorable. Par exemple, pour des SNR supérieurs à 3 dB et dans le cas d'une transmission avec la modulation BPSK sur canal gaussien, au moins 88% des ACS peuvent être évités.

**Abstract** – It is well known that convolutional codes can be optimally decoded by the Viterbi Algorithm (VA). We propose a soft decoding technique where the VA is applied to identify the error vector rather than the information message. In this paper, we show that, with this type of decoding, the exhaustive computation of a vast majority of ACS (Add Compare Select) is unnecessary. Hence, performance close to optimum is achievable with a reduction of complexity. The gain in complexity of the proposed scheme is all the more important as the SNR is high. For instance, for SNR greater than 3 dB, more than 88% of computations for ACS can be avoided when transmitting BPSK symbols on a gaussian channel.

## 1. Introduction

L'utilisation de l'algorithme de Viterbi (VA) sur le message reçu dans le cas des codes convolutifs permet de corriger les erreurs au sens du maximum de vraisemblance [1][2][3]. Après avoir étudié un algorithme de décodage dur optimal, moins complexe en terme de quantité d'opérations effectuées [4], nous proposons ici une méthode de décodage souple quasi-optimale. Cette méthode comporte deux étapes. Dans un premier temps, un décodage algébrique permet de déterminer la présence des erreurs potentielles ainsi que leur position approximative ; la seconde étape consiste à corriger ces erreurs à l'aide de l'algorithme de Viterbi avec une métrique appropriée. En appliquant cet algorithme uniquement dans les zones où il y a effectivement des erreurs corrigibles, il est subséquentement possible de réduire de façon significative la complexité du décodeur.

## 2. Code convolutif de rendement 1/2

On considère dans cet article des codes convolutifs de rendement 1/2 notés  $C(2,1,M)$  où  $M$  est la longueur de contrainte. Ainsi, pour chaque bit d'information le codeur produit deux bits de code, combinaisons linéaires du bit d'information courant et des  $M-1$  bits précédents. On note  $g_1(x)$  et  $g_2(x)$  les deux polynômes générateurs du codeur,  $m(x)$  la séquence de bits d'information,  $C_1(x)$  et  $C_2(x)$  les deux

séquences produites par les blocs convolutifs et  $C(x)$  la séquence codée.

Les deux séquences produites par les blocs convolutifs s'expriment par :

$$C_1(x) = m(x)g_1(x) \quad \text{et} \quad C_2(x) = m(x)g_2(x) \quad (1)$$

La séquence codée  $C(x)$  est définie par :

$$C(x) = C_1(x^2) + xC_2(x^2) = m(x^2)(g_1(x^2) + xg_2(x^2)) \quad (2)$$

c'est-à-dire :  $C(x) = m(x^2)G(x)$  avec  $G(x) = g_1(x^2) + xg_2(x^2)$

Nous avons démontré [4] que si  $C(x)$  est un mot de code, alors les puissances impaires du polynôme  $C(x)G(x)$  ont des coefficients nuls (lemme 1). L'algorithme proposé est fondé sur l'utilisation de ce lemme.

## 3. Décodage

Le décodage est réalisé en deux phases. La phase préliminaire consiste en un décodage algébrique qui permet de déterminer les zones du message reçu contenant les éventuelles erreurs détectables. Ensuite, ces erreurs sont corrigées à l'aide de l'algorithme de Viterbi. Nous décrivons comment modifier l'algorithme de façon à en réduire la complexité.

### 3.1 Décodage algébrique

Notons  $R(x)$  la séquence des échantillons reçus et  $r(x)$  le mot résultant d'une décision dure de  $R(x)$  ;  $r(x)$  est la somme du mot émis  $C(x)$  et d'une séquence d'erreur notée  $e(x)$  :

$$r(x) = C(x) + e(x) \quad \text{où} \quad e(x) = \sum_j e_j x^j \quad (3)$$

Il résulte que :

$$r(x).G(x) = m(x^2).G(x^2) + e(x).G(x) \quad (4)$$

En considérant les parties paires  $e_{\text{even}}(x)$  et impaires  $e_{\text{odd}}(x)$  de la séquence d'erreur, on peut définir :  $e(x) = e_{\text{even}}(x^2) + x.e_{\text{odd}}(x^2)$ . L'équation (4) donne alors :

$$r(x).G(x) = m(x^2).G(x^2) + E_{\text{even}}(x^2) + x.E_{\text{odd}}(x^2) \quad (5a)$$

avec :

$$\begin{cases} E_{\text{even}}(x^2) = e_{\text{even}}(x^2).g_1(x^2) + x^2.e_{\text{odd}}(x^2).g_2(x^2) \\ E_{\text{odd}}(x^2) = e_{\text{even}}(x^2).g_2(x^2) + e_{\text{odd}}(x^2).g_1(x^2) \end{cases} \quad (5b)$$

On peut démontrer que si  $x.E_{\text{odd}}(x^2)$  est nul, alors l'erreur  $e(x)$  est nulle ou indétectable (lemme 2) [4].

Des lemmes 1 et 2 on peut établir que le mot reçu est un mot de code si et seulement si  $x.E_{\text{odd}}(x^2)$  est nul. De plus, le polynôme  $E_{\text{odd}}(x) = \sum_j E_j^{\text{odd}} x^j$  permet de détecter les erreurs détectables.

### 3.2 Procédure de correction

Contrairement aux procédures usuelles de décodage [5][6][7] où l'algorithme de Viterbi est employé pour trouver la séquence d'information, la procédure que nous proposons utilise cet algorithme pour déterminer le vecteur d'erreur  $e(x)$ .

La correction du message au sens du maximum de vraisemblance correspond à la recherche de la séquence  $e'(x)$  de plus petit poids telle que :

$$\exists m'(x) / r(x).G(x) + e'(x).G(x) = m'(x^2).G(x^2) \quad (6)$$

Ceci est équivalent à identifier la séquence  $e'(x) = e'_{\text{even}}(x^2) + x.e'_{\text{odd}}(x^2)$  qui satisfait la condition :

$$E_{\text{odd}}(x^2) = e'_{\text{even}}(x^2).g_2(x^2) + e'_{\text{odd}}(x^2).g_1(x^2) \quad (7)$$

La séquence  $e'(x)$  peut être déterminée en construisant un treillis où le vecteur  $E(x)$  défini par (8) est annulé itérativement.

$$E(x) = E_{\text{odd}}(x) + e'_{\text{even}}(x).g_2(x) + e'_{\text{odd}}(x).g_1(x) \quad (8)$$

Ce treillis comporte des états  $S_k$  connectés par des branches auxquelles sont associées une métrique de branche  $\Delta$ . A chaque état est associé un polynôme  $E^{S_k}(x)$  conformément à l'équation (8). Une succession de branches conduisant à un état  $S_k$  est appelé un chemin.

Le nombre d'états (i.e.  $2^{M-1}$ ) du treillis et le nombre de branches partant de chaque état (i.e. 2) sont identiques à ceux

du décodage effectué à l'aide d'un VA classique.

A l'instant  $n$ , pour chaque état  $S_k$ , le polynôme associé est tel que ses  $n$  premiers coefficients sont nuls (par construction). Les chemins sont caractérisés par une métrique de chemin  $m_k(n)$ , définie comme l'accumulation des métriques des branches qui le composent. Quand plusieurs chemins arrivent au même état, le survivant est celui associé à la plus petite métrique.

Etant donné que le décodage en sens du maximum de vraisemblance consiste à trouver la séquence  $e'(x)$  de poids minimum, la prise en compte des équations (7) et (8) se traduit par la définition de la métrique de branche à l'instant  $n$  donnée par l'équation (9).

$$\Delta(n) = |R_{2n+1}.e'_{\text{odd},n} + |R_{2n}.e'_{\text{even},n} \quad (9)$$

Il est possible de démontrer que la différence entre les métriques de deux chemins aboutissant au même état est identique à celle d'un VA. Il s'en suit que les performances, en terme de TEB (taux d'erreur binaire), de notre algorithme sont les mêmes que celles du décodeur classique.

A la fin de la procédure, le chemin survivant est celui qui rejoint l'état  $S_0$ , qui est l'unique état associé à un polynôme  $E^{S_0}(x)$  dont tous les éléments sont nuls. La dernière étape du décodage consiste alors à reconstituer la séquence d'information  $m'(x)$  à partir de la relation :

$$m'(x) = \frac{r(x) + e'(x)}{G(x)} \quad (10)$$

### 3.3 Exemple de décodage

Illustrons l'algorithme proposé sur le code  $C(2,1,3)$  dont la structure est représentée sur la Fig.1. Il est caractérisé par les polynômes générateurs  $g_1(x) = 1 + x + x^2$  et  $g_2(x) = 1 + x^2$ .

Le treillis correspondant à la séquence d'information  $m(x) = 1 + x^4 + x^5$  et à la séquence d'échantillons reçus ci-dessous est illustré en figure 2.

$$R(x) = (09\ 08\ 12\ -07\ -09\ 09\ 09\ -09\ 1\ 09\ -07\ 09\ -09\ 07\ 09\ 08\ -1.1\ -08\ -07\ -09)$$

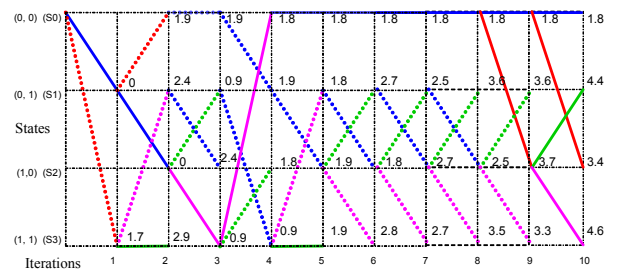


Figure 2 : treillis obtenu avec l'algorithme proposé

Pour cet exemple, considérons le chemin aboutissant à  $S_0$  à l'instant 5 :

- i) le polynôme  $E^{S_0}(x)$  est nul
- ii)  $m_k(5) \geq m_0(5)$  pour  $k \neq 0$

Nous relevons qu'après cet instant (pour  $n > 5$ ) :

- i)  $m_0(n)$  n'évolue plus ( $E^{S_0}(x) = 0$ , donc il n'y a potentiellement plus d'erreur dans la suite du message reçu)
- ii)  $m_k(n) \geq m_0(5)$  pour  $k \neq 0$  (ce chemin est le survivant)

Il serait alors possible d'arrêter la procédure de décodage à l'instant 5. Dans la suite, nous décrivons un algorithme à complexité réduite formalisant cette constatation.

## 4. Décodage à complexité réduite

Décrivons les conditions nécessaires et suffisantes permettant d'éviter tout ou une partie des itérations de la procédure.

### 4.1 Vecteur $E^{S_0}$ nul dès la première itération

Le cas où le vecteur  $E^{S_0}$  est nul dès la première itération correspond à un vecteur  $E_{odd}(x)$  nul. Autrement dit, le message est reçu sans erreur détectable. Le parcours du treillis est inutile.

### 4.2 Vecteur $E^{S_0}$ nul à partir d'une itération

Nous avons établi [4] que si à un instant  $n$ , les propriétés (11) et (12) sont satisfaites, il est inutile de poursuivre la procédure itérative : le chemin survivant est celui qui atteint l'état  $S_0$  à l'instant  $n$  et qui reste à cet état après ce nœud (Lemme 3).

$$E^{S_0}(x) = (0 \ 0 \ \dots \ 0) \quad (11)$$

$$m_k(n) \geq m_0(n) \text{ pour } k \neq 0 \quad (12)$$

### 4.3 Vecteur $E^{S_0}$ comportant une suite de zéros suivie d'un 1

Supposons maintenant que à l'instant  $n$ , la condition (12) soit satisfaite et que le vecteur  $E^{S_0}(x)$  soit sous la forme suivante (représentation vectorielle) :

$$E^{S_0}(x) = (\underbrace{0 \dots 0}_n \dots \underbrace{0 \dots 0}_N 1 \times \dots \times) \quad (13)$$

Cette configuration ne peut survenir que si il y a au moins  $N$  zéros consécutifs dans le polynôme  $E_{odd}(x)$ .

Dans le cas d'un décodage dur, nous avons démontré [4] qu'il était inutile d'effectuer les itérations entre les étapes  $n$  et  $n+N-K$  (la valeur du recul  $K$  étant uniquement liée à la longueur de contrainte du code par  $K=M-1$ ).

Dans le cas du décodage souple, les métriques dépendent des échantillons reçus. Conséquemment, les valeurs de  $K$  sont différentes pour chaque message reçu. Il est alors possible de déterminer sa valeur à l'aide d'un parcours inversé du treillis comme décrit ci-après.

Pour chaque état  $S_k$ , un parcours du treillis est effectué de l'instant  $n+N$  vers l'instant  $n$  jusqu'à ce que la condition (12) soit satisfaite. A cet instant, noté  $n+N-K^{S_k}$ , le chemin survivant qui atteint l'état  $S_k$  à l'instant  $n+N$  provient de l'état  $S_0$  à l'instant  $n$ . La valeur du recul  $K$  à considérer est le maximum des  $K^{S_k}$ . Cette détermination, induisant  $2^{M-1}$  parcours de treillis supplémentaires, limite la réduction de complexité offerte par la procédure proposée (courbe  $K$  variable Fig. 5).

Toutefois, il est possible d'éviter les parcours inversés de treillis en choisissant une valeur a priori pour le recul  $K$ . Les

performances en terme de TEB, sont d'autant plus proches de l'optimum que  $K$  est grand (Fig. 4) ; à l'inverse, la complexité décroît avec  $K$  (Fig. 5).

## 4.4 Algorithmes proposés

Sur la base des principes décrits dans les paragraphes 4.1, 4.2, 4.3, nous proposons un algorithme de décodage souple (PCT : Parcours Conditionnel du Treillis) dont les performances sont proches des performances optimales.

L'algorithme PCT permet de réduire la complexité d'autant plus efficacement que le SNR est grand. Une réduction supplémentaire de la complexité peut être obtenue en utilisant le traitement algébrique préliminaire pour déceler les messages qui ont une faible probabilité d'être décodés avec succès. Ces trames peuvent être rejetées avant décodage, ce qui conduit ainsi à une réduction de complexité qui est d'autant plus importante que le SNR est faible.

La figure 3 montre le taux d'erreur paquet (TEP) après décodage en fonction du nombre de 1 (noté  $N_1$ ) du vecteur  $E_{odd}(x)$ .

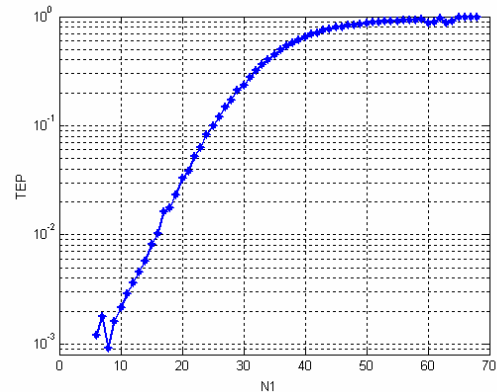


Figure 3 : TEP en fonction de  $N_1$  canal BBAG, 50000 trames de 100 bits d'information, BPSK

Lorsque  $N_1$  dépasse un seuil  $N_{1min}$ , le TEP après décodage est proche de 1. Nous proposons un deuxième algorithme (MND : Message Non Décodé) :

Si  $N_1 < N_{1min}$ , alors on applique l'algorithme PCT.

Si  $N_1 \geq N_{1min}$ , alors la trame est rejetée sans décodage.

## 5. Simulations

Nous avons évalué les algorithmes proposés par simulation dans le cas d'une transmission utilisant la modulation BPSK sur un canal à bruit blanc additif gaussien (BBAG) avec la méthode de Monte Carlo portant sur 50000 trames de 100 bits d'information.

Les figures 4 et 5 montrent que l'on peut obtenir avec l'algorithme PCT des performances en terme de TEB proches de l'optimal avec une complexité réduite lorsque la valeur de  $K$  est fixée a priori. Par exemple pour  $K=3$  et un SNR de 3 dB, la dégradation est inférieure à 0,2 dB tandis que le nombre d'ACS (Add Compare Select) effectués représente seulement 12 % des opérations du VA.

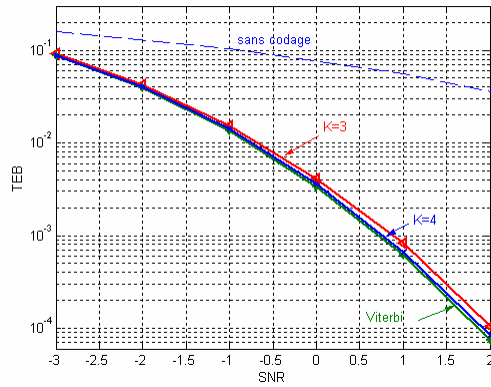


Figure 4 : TEB en fonction du SNR pour différentes valeurs du paramètre  $K$  – Algorithme PCT

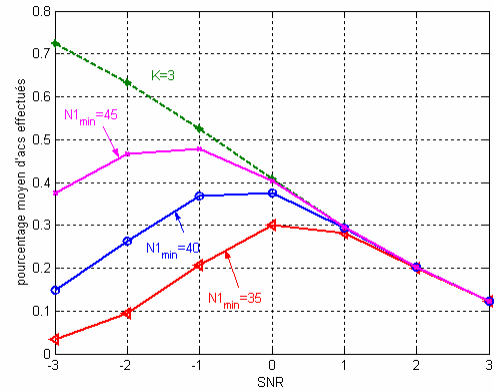


Figure 7 : pourcentage moyen d'ACS effectués (par rapport au VA) en fonction du SNR pour différentes valeurs de  $N1_{min}$  – Algorithme MND

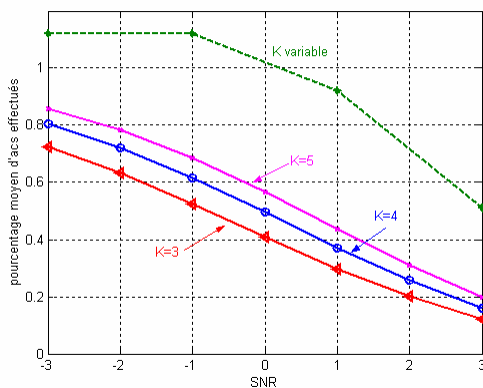


Figure 5 : pourcentage moyen d'ACS effectués (par rapport au VA) en fonction du SNR pour différentes valeurs de  $K$  – Algorithme PCT

Les figures 6 et 7 illustrent la réduction supplémentaire de complexité obtenue grâce à l'algorithme MND pour des SNR faibles. En choisissant  $K=3$  et  $N1_{min}=40$ , un gain de complexité est obtenu pour les SNR inférieurs à 1 dB (par exemple : pour un SNR de  $-3$  dB, la réduction de complexité qui est de 28 % avec le PCT passe à 85 % avec le MND). De plus, quel que soit le SNR le nombre d'ACS est limité à 40 % de ceux du VA, pour une dégradation inférieure à 0,2 dB en terme de TEP.

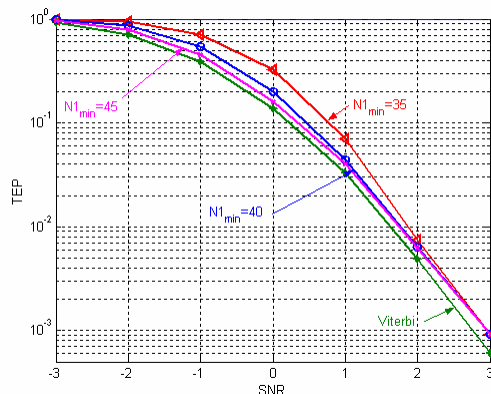


Figure 6 : TEP en fonction du SNR pour différentes valeurs du paramètre  $N1_{min}$  – Algorithme MND

## 6. Conclusion

Dans cet article, nous proposons et étudions deux nouveaux algorithmes à complexité réduite pour le décodage des codes convolutifs de rendement  $\frac{1}{2}$ . Ils permettent d'obtenir une réduction significative de la complexité tout en garantissant des performances proches de l'optimum. Le décodeur PCT tire parti du petit nombre d'erreurs lorsque le SNR est élevé. L'algorithme MND évite de surcroît le décodage inutile des messages contenant un nombre d'erreurs trop important, ce qui est le cas pour les SNR faibles. Des études sont en cours, d'une part pour appliquer cette approche sur des codes de rendement quelconque, d'autre part pour évaluer les performances d'un décodeur à sorties souples en vue d'une utilisation dans le cadre des turbos codes.

## Références

- [1] G. D. Forney, JR, *The Viterbi Algorithm*.
- [2] G. D. Forney, JR, *Convolutional codes I: Algebraic Structure*, IEEE Trans. on Information Theory, vol. IT16, No.6 November 1970.
- [3] A.J. Viterbi, *Convolutional codes and their Performance in Communication Systems*, IEEE Trans. on Communications Technology, vol. COM19, No. 5 October 1971.
- [4] J C. Dany *et al.*, *Complexity Algorithm for Optimal Hard Decoding of Convolutional Code*, EPMCC03, April 2003.
- [5] S. Lin and D.J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, 1983.
- [6] W.W. Peterson and E.J. Weldon, Jr., *Error-Correcting Codes*, 2<sup>nd</sup> edition, MIT Press: Cambridge, Mass., 1972.
- [7] J.C. Dany, *Théorie de l'information et codage*, SUPELEC, Gif sur Yvette, France, 2002.