

Nouvelle méthode de construction d'entrelaceurs pour les turbo codes

Didier LE RUYET¹, Hong SUN², Han VU THIEN¹

¹Conservatoire National des Arts et Métiers, laboratoire Signaux et Systèmes
292 rue Saint Martin, 75141 Paris Cedex 03, France

²Huazhong University of Science and Technology, Dep. Electronic and Information Engineering
430074, Wuhan, Chine

leruyet@cnam.fr, vu-thien@cnam.fr
caes@blue.hust.edu.cn

Résumé – Les turbo codes possèdent des performances proches des limites de Shannon mais cependant, à rapport signal à bruit élevé, ils présentent un effet de palier dû à leur faible distance minimale. Cet inconvénient peut être réduit en optimisant la fonction d'entrelacement. Dans cet article, nous présentons une nouvelle méthode de construction d'entrelaceurs pour les turbo codes basée sur le remplissage d'une matrice coût de poids de Hamming. L'entrelaceur est construit élément par élément en optimisant conjointement la distance minimale du turbo code et le passage de l'information extrinsèque d'un décodeur à l'autre. Nous montrons que cette approche permet d'améliorer les performances des turbo codes à rapport signal à bruit élevé.

Abstract – Turbo codes are near Shannon limit codes. Nevertheless, we can observe an error floor effect at high signal to noise ratio due to their low free distance. This drawback can be reduced by optimizing the interleaver function. In this paper, we introduce a new method for the construction of interleavers for turbo-codes based on the filling of a Hamming weight cost matrix. The interleaver construction is done step by step by optimizing both the free distance of the turbo code and the exchange of the extrinsic information between the decoders. We show how this method can improve the performances of turbo codes at high signal to noise ratio.

1 Introduction

Les codes convolutifs concaténés en parallèle ou turbo codes ont été découverts en 1993 [1]. Ces codes en bloc associent 2 codes convolutifs récurrents systématiques et une fonction d'entrelacement de N bits (figure 1).

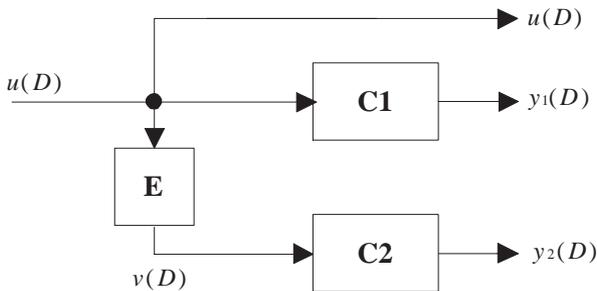


FIG. 1: codes convolutifs concaténés en parallèle

La distribution de poids de ces codes étant proche de celle d'un code aléatoire, leurs performances sont remarquables. Cependant ces codes présentent, à cause de leur faible distance minimale, un effet de palier à rapport signal à bruit élevé.

La fonction d'entrelacement influe grandement sur les performances de ces codes. Son optimisation doit permettre d'améliorer ceux-ci. Dans cet article, nous présenterons une nouvelle méthode de construction de la fonction d'en-

trelacement basée sur le remplissage d'une matrice coût de poids de Hamming. L'algorithme proposé construit l'entrelaceur élément par élément en effectuant une recherche arborescente suivant certains critères choisis au préalable.

Dans le paragraphe suivant, nous rappellerons les critères importants pour la construction d'un entrelaceur. Ensuite nous présenterons cette nouvelle méthode et montrerons comment il est possible de l'améliorer. Finalement, différents résultats seront présentés pour différentes tailles N de l'entrelaceur.

2 Critères de construction de la fonction d'entrelacement

L'optimisation d'un entrelaceur doit permettre d'améliorer la distribution de poids des turbo codes et en particulier la distance minimale de ceux-ci. Plusieurs chercheurs ont proposé des méthodes de construction de la fonction d'entrelacement basées sur optimisation de la distance minimale du code [4][6].

Cependant, à cause de la structure itérative du décodeur, l'entrelaceur doit aussi être construit de façon à garantir le meilleur passage de l'information extrinsèque d'un décodeur à l'autre. Ceci afin que les performances du décodeur itératif soient proches de celles d'un décodeur à maximum de vraisemblance a priori. Parmi les entrelaceurs assurant le meilleur passage de l'information extrinsèque

nous pouvons citer les entrelaceurs S-random [3] qui évitent les petites boucles dans l'hypertrellis et les entrelaceurs optimisant le paramètre IDS de l'entrelaceur [5] (le paramètre IDS définit le degré de corrélation entre les informations extrinsèques voisines).

Dans cet article, nous proposerons une méthode de construction de l'entrelaceur qui cherche à optimiser conjointement ces deux critères.

3 Construction de l'entrelaceur

La construction de l'entrelaceur est un problème d'optimisation. Les problèmes d'optimisation sont généralement difficiles à résoudre et utilisent souvent des méthodes heuristiques. Svirid [7] a montré que l'entrelaceur optimal du point de vue de la distance minimale doit entrelacer les séquences d'entrée $u(D)$ qui génèrent des séquences $y_1(D)$ de poids faible, avec les séquences $v(D)$ qui génèrent des séquences $y_2(D)$ de poids fort, et vice versa. Nous allons proposer une méthode de construction cherchant à respecter cette définition. Cette méthode effectue une recherche arborescente afin de construire l'entrelaceur E élément par élément.

Un entrelaceur E de taille N est défini par une matrice génératrice I de dimension $N \times N$; on a alors la relation entre la séquence entrée $u(D)$ et la sortie $v(D)$ de l'entrelaceur :

$$\mathbf{v} = \mathbf{u}I \quad \text{avec} \quad \mathbf{u} = [u_0, u_1, \dots, u_{N-1}] \\ \mathbf{v} = [v_0, v_1, \dots, v_{N-1}]$$

$$I = \{a_{ij}\}_{N \times N} \quad \text{avec} \quad a_{ij} \in \{0, 1\}$$

$$\sum_j a_{ij} = 1 \quad \text{et} \quad \sum_i a_{ij} = 1$$

On peut aussi représenter l'entrelaceur E sous la forme d'un vecteur :

$$E = [e(0), e(1), e(2), \dots, e(N \Leftrightarrow 1)]$$

$$e(i) = j \Leftrightarrow a_{ij} = 1$$

Si $a_{ij}=1$ alors le bit u_i est associé au bit v_j . Pour construire E , nous allons utiliser une matrice coût J de même dimension que I . $J = \{b_{ij}\}_{N \times N}$ b_{ij} est égal au poids de Hamming du mot code de plus faible poids généré par les séquences d'entrée $u(D)$ de poids de Hamming $w \leq w_{MAX}$ et en supposant $a_{ij} = 1$. Chaque nouvel élément $e(n)$ est choisi en respectant les deux critères définis précédemment.

Pour la structure présentée figure 1, l'algorithme de construction est le suivant :

1. initialisation :

$$b_{ij}(0) = +\infty \quad \forall i, j$$

$e(0)$ est choisi aléatoirement

2. pour $(n = 1, 2, \dots, N \Leftrightarrow 1)$:

-mise à jour de $b_{ij}(n)$ ($i \geq n$) $\forall j$:

$$b_{ij}(n) = \min \left[b_{ij}(n \Leftrightarrow 1), \min_c \left(w + \sum_{k=0}^{N-1} y_{1k} + \sum_{k=0}^{N-1} y_{2k} \right) \right]$$

avec

$$\mathcal{C} = \{u(D) = D^{l_0} + D^{l_1} + D^{l_2} + \dots + D^{l_{w-1}}\}$$

$$w = \sum_{k=0}^{N-1} u_k \leq w_{MAX} \quad \text{et} \quad w \leq n + 1$$

$$\text{avec} \quad l_0 < l_1 < \dots < l_{w-3} < n \Leftrightarrow 1, \quad (1)$$

$$l_{w-2} = n \Leftrightarrow 1, \quad l_{w-1} > n \Leftrightarrow 1 \quad (2)$$

-choix de $e(n)$:

$$\text{soient} \quad \mathcal{E}1 = \{j \mid b_{nj} \geq D_{MAX}\} \quad (3)$$

$$\mathcal{E}2 = \{j \mid |j \Leftrightarrow e(i)| > S, \\ i = n \Leftrightarrow 1, n \Leftrightarrow 2, \dots, n \Leftrightarrow S\} \quad (4)$$

$$e(n) \in \mathcal{E} = \mathcal{E}1 \cap \mathcal{E}2$$

$e(n)$ est choisie aléatoirement dans \mathcal{E} .

Les équations (1) et (2) limitent l'ensemble \mathcal{C} des séquences d'entrée $u(D)$ à tester pour chaque n . L'équation (3) correspond à la contrainte sur la distance minimale. L'équation (4) correspond à la contrainte S-random qui impose que deux bits qui sont séparés de moins de S bits dans la séquence d'entrée $u(D)$ soient séparés d'au moins S bits dans la séquence $v(D)$. L'ensemble \mathcal{E} est l'ensemble des nouvelles positions satisfaisant les deux critères. Cette méthode permet de construire un entrelaceur avec une distance minimale D_{MAX} et de facteur S-random S . D'après [3], il est possible de construire un entrelaceur avec $S < \sqrt{N}/2$. Cependant, la prise en compte de ce critère réduisant considérablement le nombre de positions autorisées, il est conseillé de diminuer légèrement la valeur de S afin de réussir à construire un entrelaceur. Si la recherche arborescente échoue ($\mathcal{E} = \emptyset$), la procédure doit être recommencée. Pour obtenir un entrelaceur avec une distance minimale la plus grande possible, il faut répéter la procédure en incrémentant la valeur D_{MAX} jusqu'à ce qu'il ne soit plus possible de construire l'entrelaceur.

Une autre méthode peut être utilisée pour choisir $e(n)$:

$$e(n) = j' \mid b_{nj'} = \max_j(b_{nj})$$

Cependant cette solution donne de moins bons résultats que la sélection de $e(n)$ aléatoirement dans \mathcal{E} . L'algorithme proposé ci-dessus est plus proche de la définition de l'entrelaceur optimum.

Il est possible d'améliorer l'algorithme en recherchant un entrelaceur voisin en cas d'échec au cours de la construction. Pour se faire, nous remontons d'un ou plusieurs éléments dans la construction et nous essayons les solutions alternatives. Pour construire les entrelaceurs présentés dans cet article, nous avons utilisé cette méthode de recherche locale en autorisant la remontée d'un élément. Cette technique améliore sensiblement le taux de réussite de construction.

Il reste ensuite à évaluer la distribution de poids du code obtenu et à le simuler en utilisant un décodeur itératif pour vérifier ses performances.

4 Terminaison des treillis des codes convolutifs

En absence de terminaison du treillis d'un codeur convolutif, le décodage de celui ci est moins bon en fin de séquence à cause de l'indétermination de l'état final. Le rôle de la terminaison est d'éviter ceci en ajoutant en fin de séquence γ bits afin de ramener l'état interne du codeur convolutif à l'état zéro. γ est le nombre de mémoire du codeur convolutif. Pour les turbo-codes, trois solutions sont envisageables: (a) absence de terminaison pour chacun des codeurs, (b) terminaison uniquement du premier codeur, (c) terminaison des 2 codeurs.

La solution (c) implique des contraintes supplémentaires dans la construction de l'entrelaceur; nous écarterons cette solution.

(a) et (b) donnent des résultats comparables: l'indétermination de l'état final peut être facilement levée sous réserve d'utiliser un entrelaceur assurant le bon passage de l'information extrinsèque (en particulier en fin de séquence). L'augmentation de la redondance de la solution (b) par rapport à la solution (a) est compensée par une meilleure distribution de poids.

Cependant, avec la solution (b), la construction de l'entrelaceur et en particulier le remplissage de la matrice coût J est plus simple car toutes les séquences $y_1(D)$ sont de poids de Hamming finis. Nous pouvons alors utiliser les propriétés des séquences d'entrée à réponse de poids finis des codes convolutifs récursifs [8] pour déterminer $\sum_{k=0}^{N-1} y_{1k}$.

Pour les codes présentés ci-dessous la terminaison est appliquée uniquement sur le premier codeur convolutif.

5 Résultats

Pour évaluer la méthode de construction proposée ci-dessus, nous avons comparé les entrelaceurs construits avec cette méthode à différents entrelaceurs. Les résultats présentés ci-dessous ont été obtenus pour un code concaténé composé d'un entrelaceur de taille $N=105$ et deux codes récursifs systématiques (15,17). Le tableau 1 compare la distribution de poids de codes utilisant 6 entrelaceurs différents:

- $E1$: entrelaceur en bloc 5 lignes par 21 colonnes.
- $E2$: entrelaceur helical [2] 5 lignes par 21 colonnes.
- $E3$: entrelaceur random obtenu par tirage aléatoire
- $E4$: entrelaceur S-random $S=7$ [3].
- $E5$: entrelaceur construit en optimisant le paramètre IDS [5].
- $E6$: entrelaceur construit en utilisant la méthode proposée ci-dessus avec $D_{MAX}=19$ et $S=6$.

L'évaluation de la distribution de poids a été réalisée en ne prenant en compte que les séquences d'information de poids de Hamming $w \leq 6$.

Pour l'entrelaceur $E6$, le remplissage de la matrice coût J a été fait en limitant le poids des séquences d'entrée à $w \leq w_{MAX} = 3$. En effet, l'ajout du critère S-random

TAB. 1: distribution de poids des 6 codes testés.

d	E1	E2	E3	E4	E5	E6
14			2			
15			4			
16			3	1		
17	2	1	7	2		
18	5	1	9	7	5	
19	6	6	12	21	6	6
20	5	4	21	5	10	11
21	17	5	22	17	15	14
22	19	11	35	28	23	37
23	27	16	54	24	20	34
24	29	16	56	32	37	43
25	48	100	100	39	72	60
26	131	54	142	85	104	89
27	89	134	190	96	166	130
28	280	298	277	162	251	169
29	319	187	351	237	355	217

élimine tous mots de code de faible poids obtenus avec une séquence d'entrée de poids $w \geq 4$. La distance minimale du code utilisant l'entrelaceur $E6$ est égale à 19 contre 18 pour $E5$ et 16 pour $E4$. Le turbo code ainsi obtenu devrait donc avoir de meilleures performances que les autres à rapport signal à bruit élevé. Pour l'entrelaceur $E6$, les 6 mots de code de poids égal à 19 sont tous générés par des séquences $u(D)$ de poids $w = 3$.

La figure 2 présente les bornes par réunion supérieure et les courbes de taux d'erreurs binaires (TEB) en fonction du rapport signal à bruit Eb/No pour les turbo codes utilisant les entrelaceurs $E1, E3, E4$ et $E6$. Les courbes en pointillé correspondent aux bornes par réunion supérieure (calculées à partir de la distribution de poids des codes). Les courbes en continu ont été obtenues par simulation en utilisant un décodeur itératif (canal BBAG, décodeur MAP, 8 itérations).

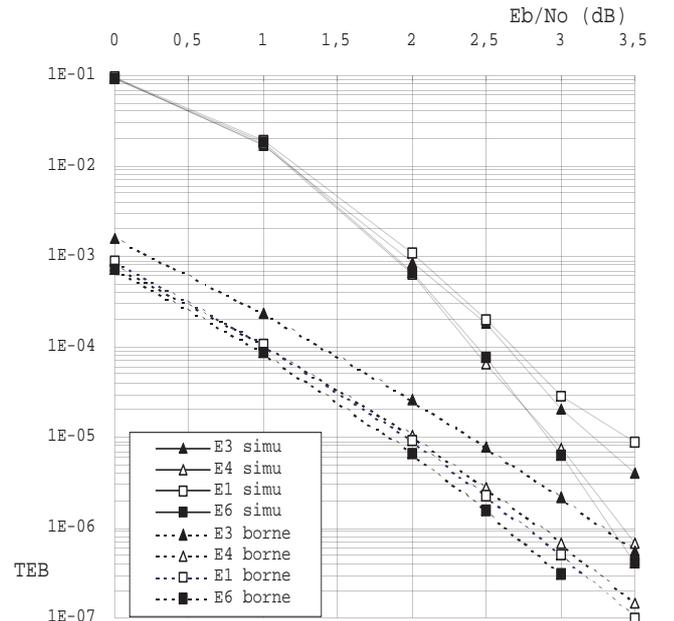


FIG. 2: Performance des différents entrelaceurs $TEB=f(Eb/No)$.

Le turbo code utilisant l'entrelaceur $E1$ présente des performances médiocres à cause du mauvais passage de l'information extrinsèque d'un décodeur à l'autre ($S=4$). Le code construit avec $E4$ présente de bonnes performances. Cependant, à rapport signal à bruit élevé, c'est le code construit avec l'entrelaceur $E6$ qui a les meilleures performances.

La figure 3 présente les courbes de taux d'erreurs trame ($P_{e\text{trame}}$) en fonction du rapport signal à bruit Eb/No .

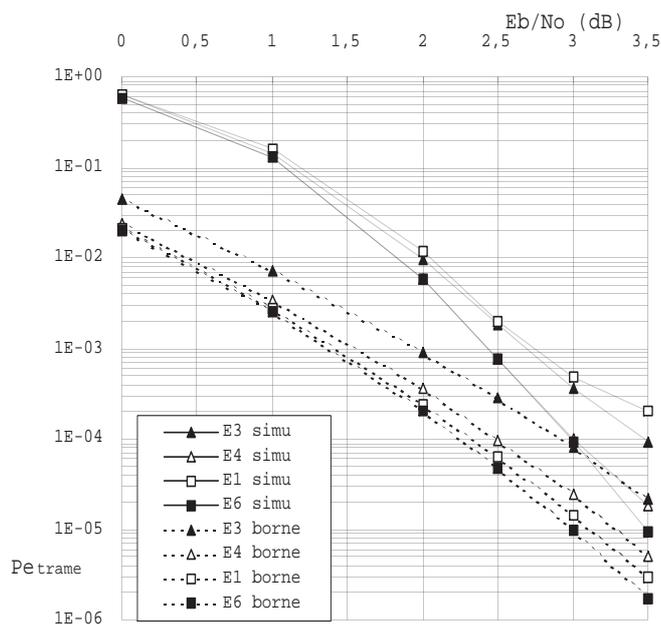


FIG. 3: Performance des différents entrelaceurs $P_{e\text{trame}}=f(Eb/No)$.

Nous avons également construit des entrelaceurs de taille 196, 256, 320 et 424. Pour ces différents entrelaceurs, les performances obtenues pour $P_e = 10^{-4}$ sont présentées figure 4. La construction de ces entrelaceurs a nécessité d'une à quelques heures de calcul sur un Pentium 2, 400 MHz. Les résultats obtenus sont à 0,8dB de la borne inférieure relative au codage aléatoire.

6 Conclusion

Une nouvelle méthode de construction d'entrelaceurs pour les turbo codes basée sur le remplissage d'une matrice coût a été présentée. Par rapport aux autres méthodes proposées, nous avons cherché à optimiser conjointement la distance minimale du turbo code et le passage de l'information extrinsèque d'un décodeur à l'autre. Les simulations effectuées ont montré que cette approche permet d'améliorer les performances des turbo codes à rapport signal à bruit élevé. Cette méthode de construction peut également s'appliquer pour des codes avec perforation ou des codes avec plusieurs entrelaceurs.

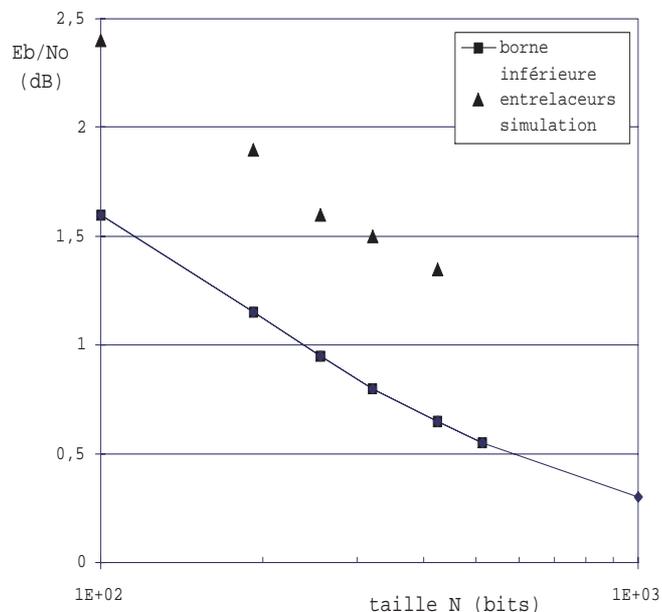


FIG. 4: Performance des entrelaceurs de taille 105, 196, 256, 320 et 424 bits $Eb/No=f(N)$ pour $P_e = 10^{-4}$.

Références

- [1] C. Berrou, A. Glavieux, P. Thitimajshima. "Near Shannon limit error correcting coding and decoding: Turbo-codes". Proc. of Int. Conf. on Comm. (ICC93), pp. 1064–1070, 1993.
- [2] A. S. Barbulescu, S. S. Pietrobon. "Terminating the trellis of turbo-codes in the same state". Electronics Letters, Vol. 31, No. 1, pp. 22–23, Jan. 1995.
- [3] S. Dolinar, D. Divsalar. "Weight distributions for turbo codes using random and nonrandom permutations". TDA Progress Report 42-122, Jet Propulsion Lab., Pasadena, CA, Aug. 1995.
- [4] A. K. Khandani. "Design of turbo code interleaver using Hungarian method". Electronics Letters, Vol. 34, No. 1, pp. 63–65, Jan. 1998.
- [5] J. Hokfelt, O. Edfors, T. Maseng. "Interleaver design for turbo codes based on the performance of iterative decoding". Proc. of ICC'99, Vancouver, Canada, June. 1999.
- [6] J. Hokfelt, T. Maseng. "Methodical interleaver design for turbo codes". Proc. of Int. Symp. on Turbo Codes and Related Topics, Brest, France, Sept. 1997.
- [7] Y. V. Svirid. "Weight distributions and bounds for turbo-codes". European Trans. on Telecommunications, Vol. 6, No. 5, pp. 543–555, Sept. 1995.
- [8] D. Le Ruyet, H. Sun, H. Vu Thien. "Properties of finite response input sequences of recursive convolutional codes". Proc. of AAEC, Hawaii, USA, Nov. 1999.