Construction de turbo-codes courts à treillis à 4 états ayant de bonnes propriétés de distance minimale

Emmanuel CADIC, Jean-Claude CARLACH

France Télécom R&D, DMR/DDH/U1
4 rue du Clos Courtel, B.P. 59, 35512 Cesson-Sévigné Cedex, France
emmanuel.cadic@rd.francetelecom.com, jeanclaude.carlach@francetelecom.com

Résumé – Cet article s'intéresse aux turbo-codes courts (longueur inférieure à 256) de rendement $\frac{1}{3}$ et $\frac{1}{2}$ dont les treillis composant ont 4 états. Ces turbo-codes sont recherchés de façon à présenter de bonnes propriétés de distance minimale ainsi que des distributions de poids les plus centrées possibles. Prendre ces différents critères de recherche a pour but de tenter de diminuer, voire de repousser, l"'error floor". Certains codes auto-duaux optimaux, dont le code de Golay [24,12,8], sont ainsi représentés sous forme de turbo-codes série ou parallèle.

Abstract – This article deals with short $\frac{1}{3}$ and $\frac{1}{2}$-rate turbo codes (length less than 256) with component trellises having 4 states. The aim is to obtain turbo-codes which have high minimum distances and good properties of the weight enumerator. By choosing these criteria, the challenge for these turbo-codes was to delay the "error-floor". We present few extremal self-dual codes, as the Golay code [24,12,8], put under the form of parallel or serial concatenated turbo-codes.

1 Introduction

L'invention des turbo-codes par Berrou et al.[1] a montré la possibilité de réaliser des systèmes de codes correcteurs d'erreurs s'approchant à quelques dixièmes de dB de la limite de Shannon à l'aide d'un decodage itératif (ou turbo) à décision douce ("soft") utilisant des algorithmes de faible complexité. Le principal objectif des turbo-codes est maintenant de repousser le phénomène d"'error floor" en augmentant leurs distances minimales et en ayant de meilleures distributions des poids. Dans cette article nous nous intéressons tout particulièrement aux turbo-codes courts, voire très courts, dont la longueur n'excède pas 256. Ces codes possèdent cependant de bonnes distances minimales sans pour autant être très complexes puisqu'ils utilisent des treillis à seulement 4 états. Dans le paragraphe 2 suivant nous expliquons comment construire les différentes sections de treillis qui seront utilisées pour élaborer les codes composants de ces turbo-codes. Dans le paragraphe 3, nous utilisons ces sections pour construire des turbo-codes série et à concaténation parallèle multiple. Finalement dans le paragraphe 4 nous récapitulons les codes déjà construits ainsi que certaines de leurs propriétés avant de présenter quelques résultats de décodage.

2 Principe

L'intérêt de la construction présentée ici est de réaliser des turbo-codes de faible complexité *via* l'utilisation de treillis à seulement 4 états. Pour élaborer la section de base de nos treillis cycliques, nous utilisons le code de Hamming étendu [8,4,4] de matrice génératrice

$$G_H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Nous considérons que les bits d'information du code composent les étiquettes de chaque section et que les deux premiers bits de redondance (resp. les deux derniers) forment l'état final (resp. initial). Une telle section de treillis est représentée par la figure 1. Chaque section ainsi construite possède 4 bits d'étiquette, 2 d'information (x_0 et x_1) et 2 de redondance (y_0 et y_1). Pour plus de possibilités de construction, nous utilisons 4 sections qui diffèrent par le positionnement des bits x_i et y_i. Ces sections sont notées H_α, H_β, H_γ et H_δ et correspondent aux bits (b_0, b_1, b_2, b_3) (Cf. figure 1) étiquetés respectivement comme suit :
(y_0, y_1, x_0, x_1), (x_0, x_1, y_0, y_1), (x_0, y_0, y_1, x_1) et (y_0, x_0, x_1, y_1).

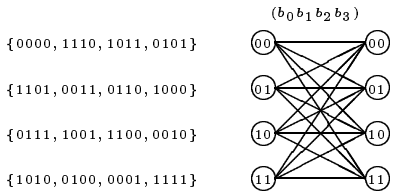

FIG. 1 – Section H

Pour permettre une plus grande modularité dans le choix des rendements construits nous introduisons 2 dernières sections H_g et H_d qui correspondent respectivement au poinçonnage des couples de bits (b_0, b_1) et (b_2, b_3) sur la section H. Ces sections de treillis sont utilisées pour la construction de turbo-codes série et parallèle. Ce choix de sections de treillis donne de bonnes propriétés au polynôme de transfert de poids (ou

"Input-Output Weight Enumerator") associé aux treillis construits à partir de ces différentes sections. Ce critère de choix est notamment plus détaillé dans [2]. La concaténation de deux sections est notée $(A|B)$ si A et B sont distinctes et on note $(A|A)$ ou A^2 sinon.

3 Représentation graphique

3.1 Construction série

La représentation graphique est très similaire à celle des codes présentés dans [3] car elle en reprend l'architecture générale en y ajoutant des liaisons verticales dans le but de diminuer le nombre de treillis à concaténer pour obtenir une bonne distance minimale. Les turbo-codes série construits sont des codes correcteurs d'erreurs en bloc systématiques de longueur $n = 2k$ comportant k bits d'information. Ces codes se présentent sous la forme d'une double concaténation série de petits codes de base $\mathcal{C}_b[n_b, k_b, d_b]$ (la figure 2 utilise uniquement le code de Hamming [8,4,4] comme code de base). Les codes de base sont dans un premier temps concaténés en série suivant un axe vertical engendrant de ce fait un treillis cyclique à $2^{\frac{k_b}{2}}$ états dans le cas binaire. Dans un second temps, les treillis ainsi obtenus sont concaténés en série selon un axe horizontal et séparés les uns des autres par des permutations (Π_i sur la figure 2) non nécessairement identiques.
Le processus d'encodage systématique est le suivant : le premier étage transforme le vecteur de bits d'information à encoder $X = (x_0, x_1, ..., x_{k-1}) = \left(x_0^{(0)}, x_1^{(0)}, ..., x_{k-1}^{(0)}\right) = X^{(0)}$ en un vecteur de bits de redondance $R^{(0)} = \left(r_0^{(0)}, r_1^{(0)}, ..., r_{k-1}^{(0)}\right)$. Les éléments de ce vecteur $R^{(0)}$ subissent alors une première permutation Π_0 pour obtenir $R^{(1)} = \left(r_{\pi_0(0)}^{(0)}, r_{\pi_0(1)}^{(0)}, ..., r_{\pi_0(k-1)}^{(0)}\right)$. On procède ainsi pour les 3 étages et les 2 permutations qui composent la structure. Le mot de code correspond alors à la concaténation du vecteur X de bits d'information avec le vecteur $R^{(2)}$ de bits de redondance. On se limite volontairement à l'utilisation de 3 étages et 2 permutations dans le but d'améliorer la qualité du décodage.

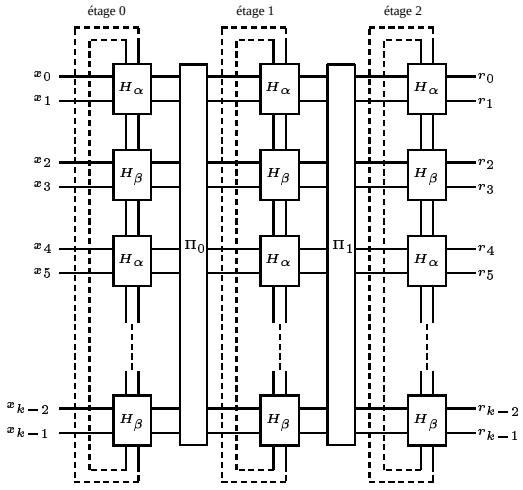

FIG. 2 – Code de paramètres $[2k, k, d_{min}]$

Les permutations Π_i sont ici des applications affines :

$$i \mapsto (a * i + b) \mod k \text{ où } (a, b) \in \mathbb{Z} \times \mathbb{Z}.$$

Nous introduisons également une construction dite "hybride" (Cf. figure 3) qui résulte d'un mélange entre la construction de la figure 1 et celle des codes présentés dans [3]. Le premier et le dernier étage de la structure restent inchangés par rapport au cas précédent, mais le second étage est remplacé par une simple concaténation de codes de Hamming étendus [8,4,4].

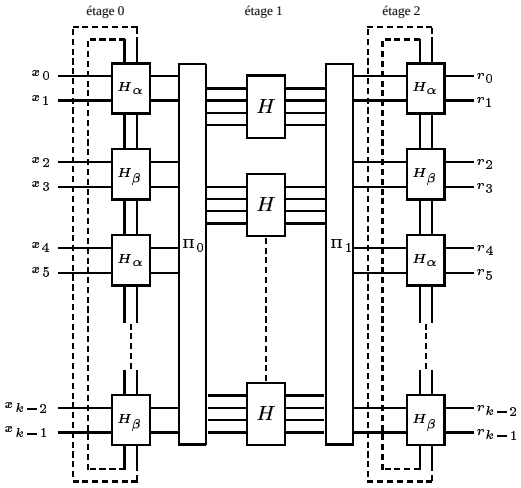

FIG. 3 – Code de paramètres $[2k, k, d_{min}]$

D'après [4], les codes présentés dans [3] atteignent la borne de Gilbert-Varshamov lorsque le nombre d'étages est suffisamment important. L'intérêt de la nouvelle strucure réside donc en partie dans sa capacité à fournir de bonnes distances minimales sans être contraint d'augmenter le nombre d'étages.

3.2 Construction parallèle

Pour illustrer le processus d'encodage, toujours en utilisant les sections de treillis de la partie 1, nous présentons le code de Golay [24,12,8] sous une forme nouvelle qui est une triple concaténation parallèle (Cf. figure 4). Chacun des treillis cycliques associé au code $\mathcal{C}$ transforme le vecteur de bits d'information $X = (x_0, \cdots, x_{k-1})$ (après une permutation différente pour chaque treillis) en un vecteur de bits de redondance.

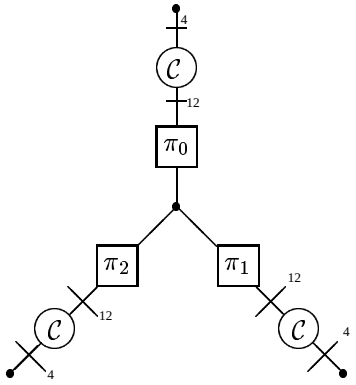

FIG. 4 – Code de Golay sous forme turbo-code parallèle

Le premier treillis (resp. le second et le troisième) fournit le vecteur de redondance $R_0 = \left(r_0^{(0)}, r_1^{(0)}, \cdots, r_{\frac{(k-1)}{3}}^{(0)}\right)$ (resp. R_1 et R_2). Le mot de code C est alors la concaténation du vecteur de bits d'information (avant permutation) et des trois vecteurs de bits de redondance.
Sur la figure 4, le treillis du code de base $\mathcal{C}$ est la concaténation de sections $(H_g|H|H_g|H_d|H|H_d)$ et correspond à un code de paramètres [16,12,2]. Les trois permutations sont $\pi_0 = (0,5,4,11,6,8,10,0,1,2,3,7), \pi_1 = D_4 \circ \pi_0$ et $\pi_2 = D_8 \circ \pi_0$ où D_s est le décalage cyclique vers la droite de s bits.

4 Codes construits

4.1 Propriétés

À titre indicatif, nous donnons dans un premier temps quelques polynômes d'approximation de l'énumérateur des poids (notés $W_{[n,k]}$) obtenus par la méthode [5] qui permet d'estimer ces polynômes dans le cas d'un entrelaceur moyen uniforme. Ces polynômes ne correspondent bien évidemment pas au cas de figure le plus favorable. Nous obtenons par exemple :

$$\begin{aligned} W_{[64,32]} &= 1 + 2X^8 + 72X^{10} + 1542X^{12} \\ &\quad + 22351X^{14} + 227742X^{16} + \ldots \\ W_{[128,64]} &= 1 + 13X^{16} + 430X^{18} \\ &\quad + 13162X^{20} + 326720X^{22} + \ldots \\ W_{[256,128]} &= 1 + X^{24} + 4X^{26} + 12X^{28} + 46X^{30} \\ &\quad + 496X^{32} + 16035X^{34} + \ldots \end{aligned}$$

Ces polynômes ont l'avantage de fournir une estimation relativement fiable du phénomène d'accroissement de la distance minimale en fonction de la longueur du code.
Le tableau 1 recence quelques codes construits suivant le modèle de la figure 2 alors que le tableau 2 présente des codes construits à partir de la structure "hybride" (les couples donnés pour les permutations Π_i correspondent aux valeurs de a et b de la permutation définie au 3.1). Dans ce second cas, les codes obtenus sont par ailleurs des codes auto-duaux extrémaux. Le tableau 3 propose quant à lui des codes obtenus par la multi-concaténation parallèle. Lorsque l'énumération exhaustive des mots de code n'est plus réalisable, l'estimation des distances minimales s'éffectue à l'aide d'un algorithme de calcul issu de [6].

Définition 4.1 *On note $C^\perp$ le code dual du code C défini par :*

$$C^\perp = \{y \in \mathbb{F}_2^n / \forall x \in C, x.y = 0\}$$

où "$x \cdot y$" correspond au produit scalaire euclidien $\sum_{i=1}^{n} x_i y_i$ modulo 2.

Définition 4.2 *Un code est dit auto-dual si $C = C^\perp$.*

Définition 4.3 *Un code auto-dual binaire dont tous les poids sont congrus à $0 \mod 2$ et pour lesquel au moins un mot est de poids congru à $2 \mod 4$ est dit de Type I.*

Définition 4.4 *Un code auto-dual binaire dont tous les poids sont congrus à $0 \mod 4$ est dit de Type II.*

Code de paramètres $[n,k,d]$	Permutations Π_0 et Π_1	Composition du treillis de base
[32,16,8]	(3,0) ; (3,0)	$(H_\alpha \vert H_\beta)^4$
[64,32,10]	(19,0) ; (19,0)	$(H_\alpha \vert H_\beta)^8$
[96,48,14]	(5,0) ; (11,0)	$(H_\delta \vert H_\gamma \vert H_\delta)^8$
[128,64,16]	(19,0) ; (11,0)	$(H_\alpha \vert H_\beta)^{16}$
[256,128,26 $\preceq d \preceq$ 32]	(19,0) ; (19,0)	$(H_\alpha \vert H_\beta)^{32}$

TAB. 1 – Turbo-codes série auto-duaux de Type I

Code de paramètres $[n,k,d]$	Permutations Π_0 et Π_1	Composition du treillis de base
[32,16,8]	(3,0) ; (3,0)	$(H_\alpha \vert H_\beta)^4$
[64,32,12]	(19,0) ; (19,0)	$(H_\alpha \vert H_\beta)^8$

TAB. 2 – Turbo-codes série auto-duaux de Type II

Les turbo-codes série obtenus ont de plus la particularité d'être des codes auto-duaux quelque soient les permutations affines utilisées. La démonstration de l'auto-dualité provient des deux propositions suivantes (les deux propositions sont elles mêmes simplement démontrables).

Proposition 4.1 *Soient $\mathcal{C}_b$ un code auto-dual de paramètres $[n_b, k_b, d_b]$ et $\mathcal{S}_b$ la section de treillis construite en prenant les k_b bits d'information en étiquettes, les $\frac{k_b}{2}$ premiers bits de redondance comme état entrant et les $\frac{k_b}{2}$ derniers bits de redondance comme état sortant. Alors, lorsqu'il existe, le code associé à la concaténation de ces sections sous forme d'un treillis cyclique est un code auto-dual.*

Proposition 4.2 *Soit $\mathcal{T}$ un treillis cyclique associé à un code $\mathcal{C}$ auto-dual. Alors les codes associés à la concaténation série de ces treillis via des permutations quelconques sont des codes eux aussi auto-duaux.*

4.2 Exemple de décodage

On note P4 un code de paramètres [400,200] construit à partir de la concaténation parallèle de deux treillis cycliques composés uniquement de la section de treillis H_γ. La figure 5 présente la courbe de taux d'erreur binaire de ce code en fonction du rapport signal à bruit pour un décodage effectué sur un canal gaussien à bruit blanc additif utilisant une modulation de phase à 2 états (MDP-2). Un tel décodage est réalisé en 15 itérations. Les courbes des codes CT (3,1), qui résulte de la concaténation de 3 arbres, et TC (21,37), qui est un turbo code utilisant des treillis à 16 états, sont donnés dans [7]. Les courbes pour ces deux derniers codes correspondent à un décodage en 18 itérations.
La figure 6 présente les courbes de taux d'erreur binaire en fonction du rapport signal à bruit pour deux codes de rendement $\frac{1}{3}$: un code [300,100,18] et un code [600,200] notés respectivement P3 et P6. Là aussi le canal est gaussien à bruit blanc additif et la modulation est une MDP-2.
La figure 5 présente un phénomène intéressant qui se manifeste à partir d'environ $2,5\ dB$. En effet pour ce rapport signal à bruit on constate que les courbes associées aux codes CT (3,1) et TC (21,37) deviennent légèrement convexes, ce qui traduit l'apparition de l"'error floor". En revanche la courbe du code P4 accentue son aspect concave et continue donc de "plonger".

Rendement $\frac{k}{n}$	Code de paramètres $[n, k, d]$	Nombre de treillis concaténés
Rendement $\frac{1}{3}$	[60,20,8]	4
	[120,40,16]	4
	[240,80,24]	4
Rendement $\frac{1}{2}$	**[24,12,8]**	3
	[32,16,8]	4
	[128,64,12]	4
	[256,128,18]	4

(en caractères gras les codes auto-duaux extrémaux)

TAB. 3 – Turbo-codes parallèle de rendement $\frac{1}{3}$ et $\frac{1}{2}$

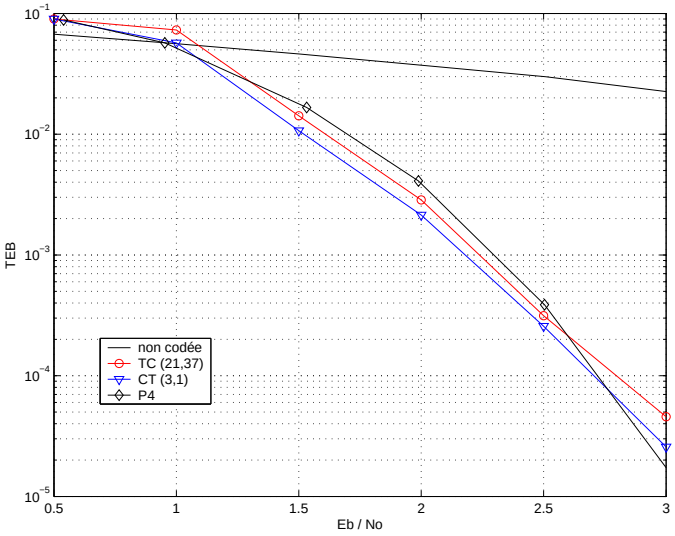

FIG. 5 – Comparaison de décodage de codes [400,200]

5 Conclusion

Nous avons vu dans cet article que des codes possédant de bonnes distances minimales, pour des rendement $\frac{1}{2}$ et $\frac{1}{3}$, peuvent être mis sous la forme de turbo-codes série ou à concaténation parallèle multiple. Il est de plus très probable que les turbo-codes courts de bonne distance minimale ne nécessitent pas l'augmentation de la complexité des treillis des codes composants. Nous envisageons actuellement de construire des codes sur l'anneau $\mathbb{Z}_4$ à l'aide de la méthode présentée dans cet article en remplaçant le code binaire Hamming étendu [8,4,4] par l'Octacode [8,4,6] sur $\mathbb{Z}_4$ (ce code est équivalent au Nordström-Robinson sur $\mathbb{Z}_2$ [8]). Ce nouveau composant permet d'augmenter la distance minimale du code et peut être naturellement associé avec une modulation de phase à 4 états. Un autre aspect intéressant de ce code est sa complexité car il fournit un treillis à 16 états et est par conséquent comparable à un turbo code duo-binaire possédant 16 états. Par ailleurs, des simulations concernant le décodage itératif des codes présentés dans cet article sont en cours de réalisation.

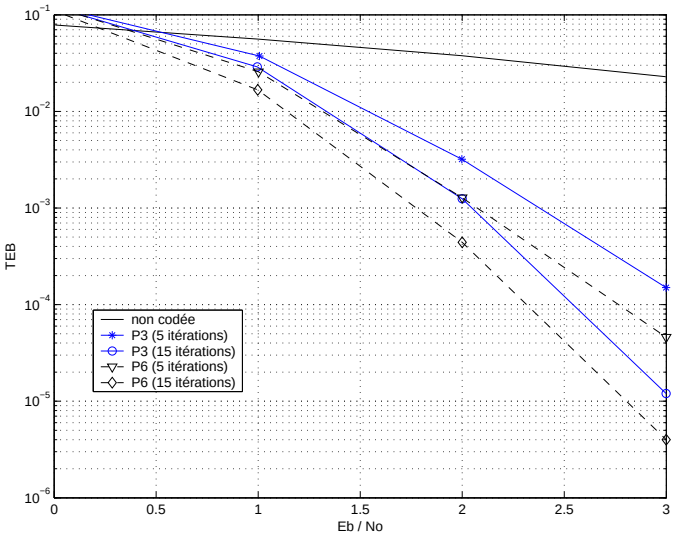

FIG. 6 – Décodage de codes de rendement $\frac{1}{3}$

Références

[1] C. Berrou, A. Glavieux et P. Thitimajshima, "Near Shannon limit Error Correcting Coding and Decoding : Turbo Codes", *Proceedings of ICC'93*, Vol. 2/3, pp.1064-1070, May 1993, Geneva, Switzerland.

[2] E. Cadic et J.C. Carlach, "Building Short Turbo Codes With High Minimum Distance Using 4-State Tail-Biting Trellises", *ISIT2003 Proceedings*, Yokohama (Japan), July 2003.

[3] J.C. Carlach and A. Otmani and C. Vervoux, "A New Scheme for Building Good Self-Dual Codes", *ISIT2000 Proceedings*, pp. 476, Sorrento (Italy), June 2000.

[4] G. Olocco et J.P. Tillich, "A familly of self-dual codes which behave in many respects like random linear codes of rate $\frac{1}{2}$", *ISIT2001 Proceedings*, Washington DC (USA), June 2001.

[5] S. Benedetto and D. Divsalar and G. Montorsi and F. Pollara, "Serial concatenation of interleaved codes : performance analysis, design and iterative decoding", *IEEE Transactions on Information Theory*, IT 44, no.3, pp.909-926, May 1998.

[6] A. Canteaut et F. Chabaud, "A New Algorithm for Finding Minimum-Weight Words in a Linear Code : Application to McEliece's Cryptosystem and to Narrow-Sence BCH Codes of Length 511", *IEEE Transactions on Information Theory*, vol. 44, no. 1, Janvier 1998.

[7] Li Ping and K.Y. Wu, "Concatenated Tree Codes", 2^{nd} International Symposium on Turbo Codes & Related Topics, Brest, France, 2000.

[8] A.R. Hammons, P.V. Kumar, A.R. Calderbank, N.J.A. Sloane and P. Solé, "The $\mathbb{Z}_4$-Linearity of Kerdock, Preparata, Goethals, and Related Codes", *IEEE Transactions on Information Theory*, vol. 40, 301-319, March 1994.