ALGORITHME TURBO : UN NOUVEAU PROCÉDÉ DE DÉCODAGE POUR LES CODES PRODUITS

Sylvie JACQ, Ramesh PYNDIAH, Annie PICART

TELECOM BRETAGNE
Technopôle de Brest Iroise BP-832
29285 BREST

RÉSUMÉ

Cet article concerne le décodage des codes produits qui sont des codes en blocs linéaires à fort pouvoir de correction. Nous proposons un nouvel algorithme de décodage pour ce type de codes. Il s'agit d'un algorithme de décodage itératif basé sur le décodage pondéré des codes en blocs et la pondération des décisions fournies par le décodeur. Nous présentons les résultats obtenus avec différents codes produits construits à partir de codes BCH. Les performances de ces codes ont été étudiées sur le canal de Gauss ainsi que sur le canal de Rayleigh. Nous avons obtenu des gains de codage pouvant atteindre 7 dB et le rapport signal sur bruit à 10^{-5} de TEB (Taux d'Erreur Binaire) se situe environ à 2,5 dB de la limite de Shannon, quel que soit le code utilisé

ABSTRACT

This paper deals with the decoding of product codes. These linear codes have large error-correcting capability. We propose a new decoding algorithm for these codes. It's an iterative algorithm based on soft-decoding of block codes and soft-outputs of the decoder. We give the results obtained by decoding different product codes using BCH codes. The performances of these codes have been studied over the gaussian channel and also over the Rayleigh fading channel. Coding gains of up to 7 dB have been obtained and SNR (signal to noise ratio) required to achieve a BER (Bit Error Rate) of 10^{-5} is at 2,5 dB of the Shannon's limit of the different codes.

1. INTRODUCTION

Le codage correcteur d'erreurs est bien souvent indispensable dans un système de transmission numérique. De plus, de nombreux domaines, comme la transmission d'images, exigent une très bonne qualité de la transmission. Ce besoin réel implique l'utilisation de codes à haut pouvoir de correction et de décodeurs associés de plus en plus complexes. Ceci est rendu possible grâce aux énormes progrès réalisés en matière de circuits intégrés.

Il existe différentes méthodes permettant de construire des codes en blocs linéaires très performants à partir de codes plus simples. Une de ces techniques consiste à concaténer deux codes en blocs (ou plus) suivant le principe introduit par Elias en 1954 [1]. Il s'agit des codes itérés, encore appelés codes produits.

Considérons le code produit $C = C_1 \otimes C_2$ construit à partir de deux codes en blocs linéaires C_1 de paramètres (N_1, K_1, δ_1) et C_2 de paramètres (N_2, K_2, δ_2). Alors C est égal à l'ensemble des matrices dont toutes les lignes sont des mots de C_2 et dont toutes les colonnes sont des mots de C_1. On montre que les éléments de C peuvent être obtenus en prenant des matrices de $(K_1 \times K_2)$ symboles d'information puis en codant les K_1 lignes par le code C_2 et en codant les N_2 colonnes ainsi obtenues par le code C_1. Les paramètres de C sont donnés par $N = N_1.N_2$, $K = K_1.K_2$ et $\delta = \delta_1.\delta_2$.

L'algorithme que nous proposons utilise un procédé itératif basé sur un décodage pondéré à décisions pondérées des lignes et des colonnes de la matrice [2]. Ceci permet au processus itératif de fonctionner de manière optimale; une itération désigne un décodage de toutes les lignes suivi d'un décodage de toutes les colonnes. Dans cet article, nous présentons les résultats obtenus pour le décodage des codes produits de type $\mathrm{BCH}(N_1, K_1, \delta_1) \otimes \mathrm{BCH}(N_2, K_2, \delta_2)$ avec $N_1 = N_2 = 16$, 32, 64, 128, 256 et $\delta_1 = \delta_2 = 4$, 6. Le comportement de ces différents codes a été étudié sur le canal de Gauss ainsi que sur le canal de Rayleigh à l'aide de simulations de type Monte-Carlo.

2. UN NOUVEL ALGORITHME DE DÉCODAGE POUR LES CODES PRODUITS

Bien que ce nouvel algorithme soit général pour le décodage des codes produits, nous traiterons ici uniquement le cas des codes binaires. Le cas des codes M-aires est abordé dans [3].

Le décodage s'effectue ligne par ligne, puis colonne par colonne, ce qui paraît tout naturel étant donné la structure d'un code produit. Cette démarche a l'avantage de ne nécessiter qu'un seul décodeur élémentaire associé à C_1 pour le décodage des lignes et un seul décodeur élémentaire associé à C_2 pour le décodage des colonnes lors de l'implantation du circuit. C'est pourquoi nous décrivons ici uniquement le décodeur élémentaire utilisé pour le décodage des colonnes; celui employé pour le décodage des lignes est réalisé suivant le même principe. Sachant qu'un décodeur qui utilise un algorithme de décodage pondéré permet d'améliorer le gain de codage d'environ 2 dB par rapport aux performances obtenues à l'aide d'un décodeur algébrique,

nous avons opté pour l'algorithme de décodage de Chase [4]. De plus, cet algorithme est plus performant que l'algorithme GMD [5]. Comme beaucoup d'algorithmes de décodage pondérés, il génère un ensemble de mots de code qui contient , avec une forte probabilité, le mot de code se trouvant à la distance euclidienne minimale de l'observation. Il permet ainsi d'approcher les performances du décodage suivant le critère du maximum de vraisemblance *a posteriori*, pour une complexité bien moindre.

L'autre problème concerne la pondération des décisions en sortie du décodeur. Soient $E=\left[e_j\right]_{1\le j\le N_1}$ le mot binaire émis, $R=\left[r_j\right]_{1\le j\le N_1}$ l'observation (séquence reçue) et $D=\left[d_j\right]_{1\le j\le N_1}$ le mot de code fourni par le décodeur de Chase pour l'entrée R. Alors le logarithme du rapport de vraisemblance de la $j^{ème}$ décision de D, d_j, est égal à :

$$LLR_j=\ln\left(\frac{\Pr\left[e_j=+1|R\right]}{\Pr\left[e_j=-1|R\right]}\right) \qquad (2.1)$$

et il possède toute l'information contenue dans R relative à la décision d_j [6]. Ainsi, la quantité (2.1) sera la décision pondérée associée à d_j et représentative de sa fiabilité.

Pour le calcul de LLR_j, on est amené à developper $\Pr\left[e_j=a|R\right]$ où $a=\pm1$:

$$\begin{aligned}\Pr\left[e_j=a|R\right]&=\sum_{C^i\in C_1}\Pr\left[e_j=a,E=C^i|R\right]\\&=\sum_{C^i\in C_1}\Pr\left[e_j=a|E=C^i,R\right].\Pr\left[E=C^i|R\right]\\&=\sum_{C^i\in S^{a(j)}}\Pr\left[E=C^i|R\right]\end{aligned}$$

où $S^{a(j)}=\left\{C^i\in C_1 / c_j^i=a\right\}$.

De plus, à l'aide de la formule de Bayes, on a :

$$\Pr\left[E=C^i|R\right]=\frac{f(R|E=C^i).\Pr\left[E=C^i\right]}{f(R)}$$

où $f(\,.\,|E=C^i)$ est la fonction densité de probabilité (fdp) de la variable aléatoire représentant la séquence reçue conditionnellement à l'événement $\left[E=C^i\right]$ et $f(\,.\,)$ est la fdp de la variable aléatoire représentant la séquence reçue.

Dans le cas du canal gaussien, on a :

$$f(R|E=C^i)=\left(\frac{1}{\sigma\sqrt{2\pi}}\right)^{N_1}\exp\left(-\frac{1}{2\sigma^2}\left\|R-C^i\right\|^2\right)$$

où $\left\|R-C^i\right\|^2=\sum_{j=1}^{N_1}(r_j-c_j^i)^2$ et σ^2 est la variance du bruit.

Ainsi, en tenant compte de l'équiprobabilité des séquences émises, on a :

$$LLR_j=\ln\left(\frac{\sum_{C^i\in S^{+1(j)}}\exp\left(-\frac{1}{2\sigma^2}\left\|R-C^i\right\|^2\right)}{\sum_{C^i\in S^{-1(j)}}\exp\left(-\frac{1}{2\sigma^2}\left\|R-C^i\right\|^2\right)}\right)$$

Cependant, ce calcul n'est pas envisageable dans le cas où C_1 possède un grand nombre de mots de code.

Si on note $C^{(j,+1)}$ le mot de code à distance euclidienne minimale de la séquence reçue R et tel que $c_j^{(j,+1)}=+1$, et $C^{(j,-1)}$le mot de code à distance euclidienne minimale de R et tel que $c_j^{(j,-1)}=-1$, alors on peut approximer LLR_j par :

$$LLR_j\approx\frac{1}{2\sigma^2}\left(\left\|R-C^{(j,-1)}\right\|^2-\left\|R-C^{(j,+1)}\right\|^2\right) \qquad (2.2)$$

Cette approximation de LLR_j a le même signe que d_j :

$$\begin{aligned}&\frac{1}{2\sigma^2}\left(\left\|R-C^{(j,-1)}\right\|^2-\left\|R-C^{(j,+1)}\right\|^2\right)\\&\qquad=\left[\frac{1}{2\sigma^2}\left(\left\|R-C^{(j,-d_j)}\right\|^2-\left\|R-D\right\|^2\right)\right].d_j\end{aligned}$$

Le mot $C^{(j,-d_j)}$ est recherché dans l'ensemble de mots de code généré par l'algorithme de Chase.

Le développement de l'expression (2.2) donne

$$LLR_j\approx\frac{2}{\sigma^2}\left(r_j+\sum_{\substack{l=1,l\neq j\\c_l^{(j,-1)}\neq c_l^{(j,+1)}}}^{N_1}r_l.c_l^{(j,+1)}\right)$$

En posant

$$w_j=\sum_{\substack{l=1,l\neq j\\c_l^{(j,-1)}\neq c_l^{(j,+1)}}}^{N_1}r_l.c_l^{(j,+1)} \qquad (2.3)$$

et en normalisant par le facteur $\frac{\sigma^2}{2}$, nous avons une approximation de LLR_j normalisé, r_j', associée à d_j :

$$r_j'=r_j+w_j \qquad (2.4)$$

Lorsque l'algorithme de Chase permet de déterminer le mot $C^{(j,-d_j)}$, le calcul de r_j' se fait en utilisant l'expression (2.3). Dans le cas contraire, on utilise la relation

$$r_j'=\beta.d_j \qquad (2.5)$$

où β est une constante [2].

Le schéma bloc du décodeur pour le $i^{ème}$ décodage est le suivant

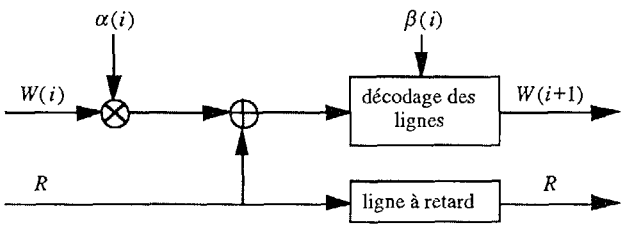

***figure 2.1** : diagramme du décodeur élémentaire utilisé pour le traitement des lignes à la $i^{ème}$ itération*

où $W(i)=\left[w_j(i)\right]_{1\le j\le N_1}$ a été défini précédemment (voir l'expression (2.3)), $\beta(i)$ est la constante utilisée dans (2.5) et $\alpha(i)$ est un facteur de pondération tenant compte du fait que la variance de la v.a. représentant R est inférieure à la variance de la v.a. représentant $W(i)$ lorsque le TEB est élevé.

3. PERFORMANCE DES TURBO CODES EN BLOC SUR CANAL GAUSSIEN

Nous présentons maintenant les résultats obtenus sur le canal de Gauss avec une MDP-4 lorsque l'information est codée par un code $BCH(N_1,K_1,\delta_1)\otimes BCH(N_2,K_2,\delta_2)$, où $N_1 = N_2 = 16$, 32, 64, 128, 256 et $\delta_1 = \delta_2 = 4$, 6.

En normalisant les composantes du vecteur $W(i)$, nous avons utilisé, pour tous les codes étudiés, des valeurs $\alpha(i)$ et $\beta(i)$ identiques :

$$\alpha(1..8) = [0.0, 0.3, 0.35, 0.5, 0.7, 0.9, 1.2, 1.5]$$
$$\beta(1..8) = [0.2, 0.4, 0.6, 0.8, 1.0, 1.0, 1.0, 1.0].$$

Nous avons indiqué dans le tableau 3.1 le rendement des différents codes ainsi que la borne supérieure du gain asymptotique $(G_a)_{max} = 10.\log(R\delta)$ avec $R = \frac{K_1.K_2}{N_1.N_2}$ et $\delta = \delta_1.\delta_2$

code produit	R	$(G_a)_{max}$
BCH(16,11,4)⊗BCH(16,11,4)	0.473	8.787 dB
BCH(16,7,6)⊗BCH(16,7,6)	0.191	8.383 dB
BCH(32,26,4)⊗BCH(32,26,4)	0.660	10.238 dB
BCH(32,21,6)⊗BCH(32,21,6)	0.431	11.904 dB
BCH(64,57,4)⊗BCH(64,57,4)	0.793	11.035 dB
BCH(64,51,6)⊗BCH(64,51,6)	0.635	13.591 dB
BCH(128,120,4)⊗BCH(128,120,4)	0.879	11.481 dB
BCH(128,113,6)⊗BCH(128,113,6)	0.779	14.480 dB
BCH(256,247,4)⊗BCH(256,247,4)	0.931	11.730 dB

tableau 3.1 - Rendement et valeur de $(G_a)_{max}$, calculés pour chacun des codes étudiés

Afin d'évaluer les performances de l'algorithme de décodage itératif du paragraphe 2, nous avons calculé la différence ΔG entre $(G_a)_{max}$,et G_4, le gain de codage obtenu à la 4ème itération pour un TEB de 10^{-5}. Ces résultats figurent dans le tableau 3.2.

C_1 (= C_2)	G_4 à 10^{-5}	ΔG à 10^{-5}	SNR pour TEB = 10^{-5}	ΔL_S à 10^{-5}
BCH(16,11,4)	6.25 dB	2.54 dB	3.35 dB	3.26 dB
BCH(16,7,6)	5.90 dB	2.48 dB	3.70 dB	4.70 dB
BCH(32,26,4)	6.55 dB	3.69 dB	3.05 dB	2.50 dB
BCH(32,21,6)	7.05 dB	4.85 dB	2.50 dB	2.73 dB
BCH(64,57,4)	6.15 dB	4.89 dB	3.45 dB	2.44 dB
BCH(64,51,6)	6.75 dB	6.84 dB	2.80 dB	2.34 dB
BCH(128,120,4)	5.45 dB	6.03 dB	4.10 dB	2.78 dB
BCH(128,113,6)	6.20 dB	8.28 dB	3.35 dB	2.39 dB
BCH(256,247,4)	4.80 dB	6.93 dB	4.80 dB	3.29 dB

tableau 3.2 *- Comparaison des performances sur le canal de Gauss en présence de MDP-4*

D'autre part, le rapport signal sur bruit nécessaire pour obtenir un TEB de 10^{-5} à la 4ème itération est également donné dans le tableau, ainsi que l'écart entre ce rapport signal sur bruit et la limite théorique de Shannon L_S du code : $L_S = 10.\log\left(\frac{2^{2R}-1}{2R}\right)$.

La quantité ΔG augmente avec R et δ. En effet, bien que la quantité $(G_a)_{max}$ augmente également avec R et δ, le gain asymptotique des codes n'est atteint que pour des TEB très faibles ($< 10^{-7}$), lorsque R et δ sont grands; les pentes des courbes sur la figure 3.1 le confirment.

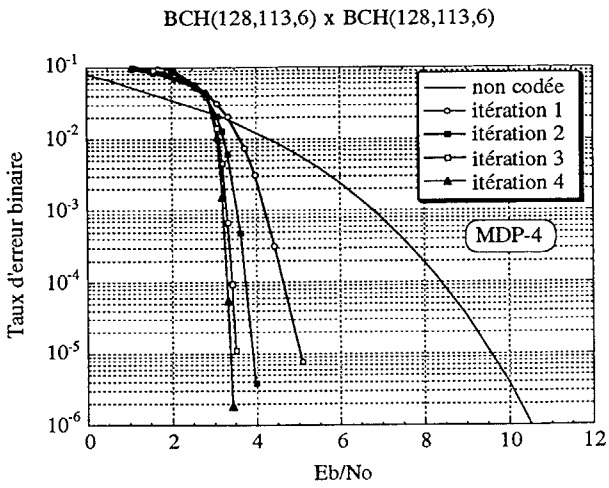

figure 3.1 *: canal de Gauss décodage du BCH(128,113,6)⊗BCH(128,113,6)*

4. PERFORMANCE DES TURBO CODES EN BLOC SUR CANAL DE RAYLEIGH

Comme pour le cas du canal de Gauss, les composantes du vecteur $W(i)$ ont été normalisées et les valeurs de $\alpha(i)$ et $\beta(i)$ utilisées sont les mêmes.

Le tableau 4.1 donne la comparaison des performances sur le canal de Gauss et sur le canal de Rayleigh.

C_1 (= C_2)	SNR pour TEB = 10^{-5} GAUSS	SNR pour TEB = 10^{-5} RAYLEIGH	Δ SNR
BCH(16,11,4)	3.35 dB	7.30 dB	3.95 dB
BCH(16,7,6)	3.70 dB	5.80 dB	2.10 dB
BCH(32,26,4)	3.05 dB	7.90 dB	4.85 dB
BCH(32,21,6)	2.50 dB	5.40 dB	2.90 dB
BCH(64,57,4)	3.45 dB	9.40 dB	5.95 dB
BCH(64,51,6)	2.80 dB	7.30 dB	4.50 dB
BCH(128,120,4)	4.10 dB	11.85 dB	7.75 dB
BCH(128,113,6)	3.35 dB	9.50 dB	6.15 dB
BCH(256,247,4)	4.80 dB	14.35 dB	9.55 dB

tableau 4.1 *- Comparaison des performances sur le canal gaussien et sur le canal de Rayleigh en présence de MDP-4*

La dégradation des performances, comparées à celles obtenues sur canal de Gauss, est faible. Cette dégradation est d'autant plus faible que le rendement du code est petit.

De plus, on observe là encore une très forte pente pour la courbe de taux d'erreur. Pour un pouvoir de correction donné, cette pente augmente avec la longueur du code. De même, pour une longueur de code donnée, la pente des courbes augmente avec le pouvoir de correction (voir figure 4.1).

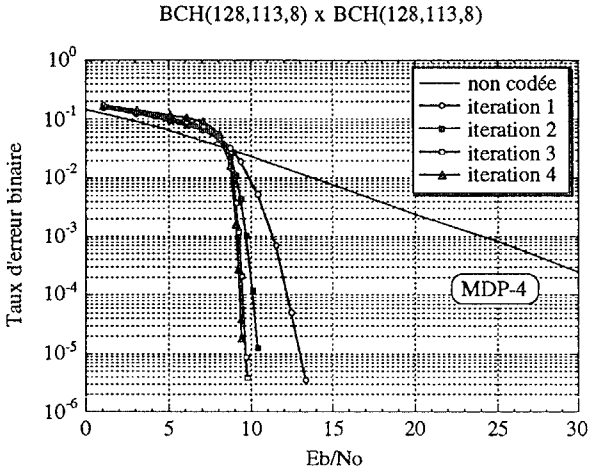

figure 4.1 *: canal de Rayleigh*
décodage du BCH(128,113,6)⊗BCH(128,113,6)
à l'aide de l'algorithme turbo.

Ces résultats ont été obtenus sans supposer la connaissance des coefficients α_{ij}, ce qui représente une simplification non négligeable. Si on dispose des α_{ij}, le taux d'erreurs est alors divisé par 2 environ.

La figure 4.2 présente une estimation de la distribution de $w_j|e_j=+1$, qui désigne w_j conditionnellement à l'événement $[e_j=+1]$. On observe que $w_j|e_j=+1$ converge vers une loi de Gauss au fil des itérations, sur canal de Rayleigh.

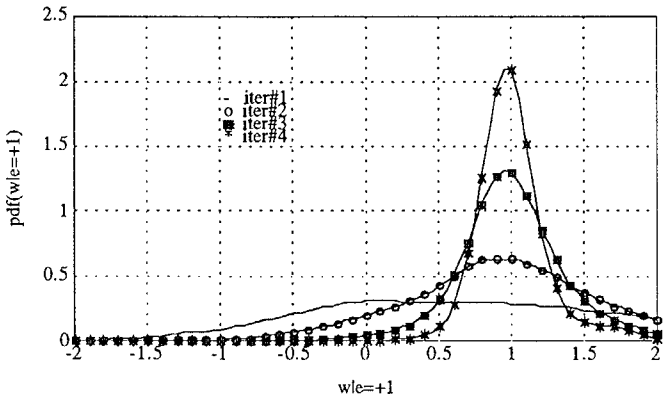

figure 4.2 *: distribution de* $w_j|e_j=+1$
(après normalisation, pour le BCH(64,57,4)⊗BCH(64,57,4))

5. CONCLUSION

Dans cet article, nous avons présenté les performances des codes produits contruits à partir de codes BCH binaires utilisant un nouvel algorithme de décodage itératif basé sur un décodage pondéré à décisions pondérées : turbo codes en blocs.

Les résultats obtenus par simulations sur canal de Gauss mettent en évidence des gains de codage pouvant atteindre 7 dB, à un TEB de 10^{-5}. Cette performance, jamais été atteinte auparavant avec les codes produits, se distingue également par une très forte pente des courbes de TEB.

Sur canal de Rayleigh, nous observons un phénomène similaire. Les résultats présentés sont obtenus sans la connaissance *a priori* des bruits multiplicatifs de Rayleigh. On évite ainsi leur estimation, ce qui représente une simplification non négligeable. On observe une légère dégradation des performances, en comparaison de celles obtenues sur le canal de Gauss. Cette dégradation est très faible (≈ 2 dB) lorsque le rendement du code prend des petites valeurs, c'est-à-dire losque l'effet de diversité se fait plus ressentir.

Les codes produits avec décodage itératif présentent le plus grand intérêt pour des applications en communications numériques et en stockage des données. Les codes produits étudiés, avec décodage itératif, sont à 2,5 dB de la limite de Shannon, pour un taux d'erreur binaire de référence de 10^{-5}. Actuellement, ces résultats sont comparables à ceux des turbo-codes convolutifs introduits en 1993 par Berrou [7], dont les performances sont parmi les meilleures connues à ce jour.

REMERCIEMENTS

Cette étude a été financée par le Centre National d'Études des Télécommunications (CNET) et le Centre Commun d'Études de Télédiffusion et Télécommunications (CCETT).

RÉFÉRENCES

[1] P. Elias
"Error-free coding"
IRE Trans. on Inf. Theory, vol. IT-4, pp. 29-37, September 1954.

[2] R. Pyndiah, A. Glavieux, A. Picart, S. Jacq
"Near optimum decoding of product codes"
IEEE Globecom'94, Vol 1/3, San Fransisco, pp. 339-343, 1994.

[3] O. Aitsab, R. Pyndiah
"Performances des turbos-codes en blocs Q-aires 'Reed-Solomon'"
sera publié au Gretsi'95

[4] D. Chase
"A class of algorithms for decoding block codes with channel measurement information"
IEEE Transactions on Information Theory, vol. IT-18, n°1, pp. 170-182, january 1972.

[5] J.E.M. NILSSON
"Difference between two soft-decision decoding algorithms"
Electronics letters, vol. 30, n°20, pp. 1665-1666, 29th September 1994.

[6] G.D. Forney
"Concatenated codes"
M.I.T. Press, Cambridge, Massachusetts, 1966.

[7] C. Berrou, A. Glavieux, P. Thitimajshima
"Near Shannon limit error-correcting coding and decoding : Turbo-codes"
Proc. IEEE International Conference on Communications (ICC'93) pp. 1064-1070, Geneva 1993.