QUATORZIEME COLLOQUE GRETSI - JUAN-LES-PINS - DU 13 AU 16 SEPTEMBRE 1993

LA CAPACITÉ D'UN CANAL PEUT ÊTRE APPROCHÉE PAR DES MOYENS SIMPLES

Gérard Battail (*), Claude Berrou (**) et Alain Glavieux (**)

(*) Ecole nationale supérieure des Télécommunications,
46 rue Barrault, 75634 PARIS CEDEX 13
(**) Ecole nationale supérieure des Télécommunications de Bretagne,
B.P. 832, 29285 BREST CEDEX

RÉSUMÉ

On considère des moyens déterministes d'imiter le codage aléatoire. Les codes linéaires en blocs aléatoires et pseudo aléatoires sont d'abord introduits. Leur linéarité entraîne que le décodage en est possible par des moyens beaucoup moins complexes que la recherche exhaustive. On s'intéresse ensuite aux codes convolutifs aléatoires et pseudo aléatoires. Impliquant des suites de longueur infinie et de poids presque toujours infini, ils permettent d'approcher de très près la capacité du canal. On s'intéresse en particulier à un codeur de taux 1/2 constitué d'un registre rebouclé à longueur maximale et au décodage du code qu'il engendre.

ABSTRACT

We consider deterministic means to mimic random coding. Linear block codes, both random and pseudo random, are first introduced. Linearity makes their decoding possible by means much simpler than exhaustive search. We then consider convolutional random and pseudo random codes. Since they involve infinite length sequences, whose weight is almost always infinite, they enable to closely approach the channel capacity. We especially consider a rate 1/2 encoder made of a register with a maximum-length feedback, and decoding of the code it generates.

1. Introduction

Sous le vocable de *turbo-codes*, deux des auteurs ont récemment présenté des procédés de codage et décodage atteignant une probabilité d'erreur très petite sur le canal à bruit gaussien additif, pour un rapport signal à bruit ne dépassant la limite fixée par la capacité du canal que d'une fraction de décibel, soit un gain d'environ 2 dB par rapport aux meilleurs résultats connus [1]. En outre, ces procédés combinent des opérations simples : codage convolutif de petite longueur de contrainte, entrelacement non uniforme et décodage symbole par symbole itéré, c'est-à-dire que ces résultats surprenants sont obtenus sans complexité exagérée.

Leurs performances appellent une explication que ne paraît pas fournir la théorie classique des codes convolutifs. Les codes employés dans [1] sont systématiques et récursifs ; or, Forney a montré que l'énumérateur de poids des chemins (représentant les suites codées) qui s'écartent de l'état nul puis y reviennent est le même, que le code soit engendré par le codeur systématique récursif ou par le codeur non systématique et non récursif employant un même registre avec les mêmes connexions [2]. La distance minimale entre les suites codées (dite distance limite ou libre, en anglais *free distance*, d_{free}) est donc la même, toutes choses égales par ailleurs, que le code considéré soit récursif ou non. On a généralement déduit de ce résultat incontestable que les codes convolutifs récursifs ne sont pas particulièrement intéressants, conclusion que nous allons contester.

L'autre auteur a critiqué le critère traditionnel de distance minimale, montrant qu'il ne s'applique qu'à des codes courts c'est-à-dire, relativement au canal donné, tels que le nombre moyen d'erreurs dans un mot soit petit [3]. En effet, ce critère ne concerne que le pire des cas. Si la proportion des paires de mots à la distance minimale (celle des mots de poids minimal pour un code linéaire) est négligeable, la distance minimale n'est pas en elle-même significative. C'est la distribution *normalisée* des distances ou des poids qui est pertinente, "normalisée" signifiant que le nombre des mots d'un poids considéré est divisé par le nombre total des mots que contient le code. Prenant acte du fait que le codage aléatoire permet seul d'atteindre la capacité du canal, sans aucune restriction sur la distance minimale, il a proposé pour critère de qualité d'un code (long) une mesure de proximité entre la distribution normalisée de ses poids et celle qui est obtenue en moyenne par codage aléatoire, et montré que certains codes bons pour ce critère, mais non pour celui de distance minimale, avaient des performances intéressantes [3-5].

2. Codes linéaires aléatoires et pseudo aléatoires

Peut-on mettre en œuvre concrètement le codage aléatoire, ou plutôt un codage *pseudo aléatoire*, déterministe mais possédant les propriétés attendues du codage aléatoire ? Le codage aléatoire a été jusqu'à présent considéré surtout comme un moyen de démonstration théorique, sans que son emploi effectif ait été vraiment envisagé. La complexité du codage aléatoire, en ce qui concerne notamment le décodage, est évidente si l'on s'en tient au schéma naïf qui consiste à tirer au hasard chacun des mots du code indépendamment des autres, chaque symbole étant lui-même tiré indépendamment des autres avec une probabilité $1/q$ pour un alphabet de taille q. On constitue ainsi une liste de mots dont le décodage ne peut être qu'exhaustif, donc d'une complexité prohibitive pour les ordres de grandeur où le codage est utile. Nous allons maintenant

introduire des codes *linéaires* aléatoires et pseudo aléatoires dont le décodage est beaucoup plus simple pourvu que l'on accepte une légère dégradation des performances par rapport à l'exacte optimalité.

Soit un code linéaire en blocs (n,k) ayant une matrice génératrice sous la forme systématique

$$\mathbf{G} = [\mathbf{I}_k \mid \mathbf{P}], \tag{1}$$

où $\mathbf{I}_k$ est la matrice unité d'ordre k et $\mathbf{P}$ est une matrice à k lignes et $n-k$ colonnes. Si tous les éléments de la matrice $\mathbf{P}$ sont tirés au hasard indépendamment les uns des autres, on définit ainsi un code aléatoire. Écrivons en effet un vecteur de ce code sous la forme $v = [u \mid s]$, où v est un vecteur à n composantes, u et s, respectivement à k et $n-k$ composantes, sont le vecteur d'information et le vecteur de redondance. Les vecteurs u et s sont aléatoires et indépendants puisque :

u est le vecteur d'information, aléatoire par hypothèse ;

s est aléatoire, car il résulte d'une combinaison linéaire des lignes de $\mathbf{P}$ qui sont aléatoires ;

u et s sont indépendants, car issus de deux tirages indépendants, l'un immédiat (celui de u) et l'autre "fossile", ce terme imagé pour signifier qu'il s'agit d'un tirage antérieur fait une fois pour toutes.

Or, il existe des algorithmes de décodage qui n'exploitent que la linéarité du code et dont la complexité est très inférieure à celle du décodage exhaustif, par exemple [6-9]. Certains de ces algorithmes sont optimaux dans leur principe mais leur mise en œuvre peut être considérablement simplifiée au prix d'une légère dégradation. D'autres sont fondés sur une approximation initiale, au bénéfice de la simplicité. Ils peuvent donc être employés pour décoder les codes linéaires aléatoires que nous venons d'introduire.

De plus, il n'y a que des avantages à remplacer le tirage aléatoire de la matrice $\mathbf{P}$ par une construction déterministe pseudo aléatoire. En effet, le codage en est évidemment simplifié et, surtout, les propriétés obtenues sont voisines des propriétés *moyennes* du codage aléatoire, alors qu'un véritable tirage aléatoire ne fournirait une suite typique, représentative de la moyenne des suites, qu'avec une probabilité d'autant plus grande qu'elle est longue. La matrice $\mathbf{P}$ peut être construite en en prenant pour éléments successifs, lus par exemple ligne par ligne, ceux d'une suite de longueur maximale $l = 2^m - 1$ engendrée par un registre de longueur m. Les propriétés souhaitées de la distribution des poids sont obtenues pourvu que $l \geq k(n-k)$.

Nous allons maintenant examiner comment appliquer des idées similaires aux codes convolutifs.

3. Codes convolutifs aléatoires et pseudo aléatoires

Impliquant des suites de longueur infinie, les codes convolutifs ne sauraient être considérés comme courts, même sur des canaux très peu bruyants. Selon [3], le critère de distance minimale ne leur est donc pas pertinent. L'argument qui justifie l'équivalence du code non systématique et non récursif et du code systématique et récursif, construits à partir d'un même registre avec les mêmes connexions, n'est basé que sur l'identité de la fonction qui énumère les poids. Le poids minimal des suites codées, en particulier, est le même. Pour un critère fondé sur la distribution normalisée des distances, cette équivalence n'est plus pertinente si les poids finis et donc le poids minimal donnent une contribution nulle ou négligeable à la distribution normalisée (qui concerne ici, rappelons-le, des suites infinies). Tel est précisément le cas avec les codes convolutifs aléatoires et pseudo aléatoires.

D'une manière très semblable à celle qui a été brièvement exposée pour les codes en blocs, on peut définir des codes convolutifs aléatoires et pseudo aléatoires. Pour simplifier, nous allons maintenant considérer uniquement des codes de taux 1/2, mais la généralisation à un taux quelconque est immédiate. D'ailleurs, le poinçonnage permet de l'augmenter à volonté à partir d'un code de base de taux 1/2. Nous nous restreindrons aussi à des codes binaires. Soit le code convolutif de matrice génératrice

$$\mathbf{G} = [1 \;\; \mathrm{R}(D)], \tag{2}$$

où $\mathrm{R}(D)$ est la série formelle associée à une suite aléatoire binaire semi-infinie, l'indéterminée D représentant l'opérateur de retard unitaire. On a donc

$$\mathrm{R}(D) = r_0 + r_1 D + r_2 D^2 + \cdots \tag{3}$$

où $r_0, r_1, \cdots$ sont les symboles de la suite, résultant de tirages aléatoires binaires successifs, indépendants et équiprobables, effectués antérieurement une fois pour toutes. L'analogie de forme des matrices génératrices (1) et (2) est évidente.

Alors, à toute suite d'information représentée par le scalaire $u(D)$ est associée la suite codée, représentée par le vecteur à deux composantes :

$$v(D) = u(D)\mathbf{G} = [u(D) \;\; u(D)\mathrm{R}(D)].$$

Les composantes du vecteur codé sont deux suites aléatoires semi-infinies et indépendantes, pour les mêmes raisons qu'avec les codes en blocs. En effet, $u(D)$ est intrinsèquement aléatoire, tandis que $u(D)\mathrm{R}(D)$ est une combinaison linéaire de suites déduites de $\mathrm{R}(D)$ par décalage, ayant les mêmes propriétés aléatoires que $\mathrm{R}(D)$ elle-même. La distribution des distances du code ainsi défini est donc la même que celle du codage aléatoire, mais il s'agit ici, à la différence des codes en blocs envisagés ci-dessus, de suites de longueur et poids infinis. De ce fait, ces codes sont de bons candidats pour approcher de très près la capacité du canal.

Pour la mise en œuvre effective d'un codage pseudo aléatoire de ce type, le codage récursif, et lui seul, permet d'obtenir des suites codées de poids croissant indéfiniment en réponse à une suite d'information de poids fini. Les codes convolutifs récursifs sont caractérisés par une matrice génératrice dont au moins un élément est une fraction rationnelle en l'indéterminée D. Une telle fraction rationnelle est développable en une série formelle périodique, donc de poids infini. Ainsi, le codage fait correspondre une suite de redondance de poids infini à toutes les suites d'information de poids fini (messages), à l'exception de celles dont la représentation polynomiale en D est un multiple du dénominateur. La proportion des messages représentés par un multiple du dénominateur est égale à q^{-K} pour un alphabet à q éléments, K désignant la longueur de contrainte du codeur ; elle peut donc être rendue négligeable en choisissant K assez grand.

À elles seules, ces remarques paraissent suffire à expliquer les performances des turbo-codes, en y interprétant la concaténation avec entrelacement non uniforme comme un moyen simple de construire un code récursif de grande longueur de contrainte mais où le décodage est scindé en celui des deux codes constituants, donc simplifié.

Il est facile en outre d'imiter le codage aléatoire avec un code récursif, c'est-à-dire de satisfaire à la lettre au critère proposé dans [3]. Il est connu depuis longtemps que certains rebouclages d'un registre à décalage, notamment à longueur maximale, permettent d'engendrer des suites qui imitent les suites aléatoires et sont de ce fait dites *pseudo aléatoires*. Il s'agit de suites périodiques dont la période est $l = 2^K - 1$ pour un registre de longueur K et peut donc être rendue très grande en choissant K suffisamment grand. Le codeur binaire le plus simple fonctionnant selon ce principe est constitué d'un registre avec rebouclage à longueur maximale où le symbole réinjecté à l'entrée du registre est additionné modulo 2 au symbole d'information. Celui-ci est émis (le codage est systématique), ainsi que le symbole entrant dans le registre (par exemple), ce qui définit bien un code de taux 1/2. La matrice génératrice de ce code, sous forme systématique, est donc

$$\mathbf{G} = [1 \;\; 1/P(D)], \tag{4}$$

où $P(D)$, qui décrit les connexions du registre, est alors un polynôme primitif pour le corps d'extension à 2^K éléments. Par rapport à la matrice génératrice des codes récursifs habituellement employés, par exemple dans les turbo-codes, on a remplacé le polynôme en numérateur par 1. Il en résulte une faible distance limite d_{free}, par exemple 4 si $P(D)$ est de poids 3, mais cela n'a pas d'inconvénient selon notre critère. Un exemple simple de codeur de cette forme est donné sur la figure 1 pour $K = 3$. La figure 2 représente le diagramme de transition associé à ce codeur, mettant en évidence un chemin fermé de poids non nul mais correspondant à une suite d'information nulle.

On peut d'ailleurs facilement améliorer la distance limite en combinant des bits contenus dans le registre dans le sens de l'avant, ce qui a pour effet de substituer au numérateur 1 un polynôme $N(D)$ de degré au plus égal à K et de poids supérieur à 1, choisi afin d'augmenter au maximum la distance limite. La matrice génératrice du code devient alors

$$\mathbf{G} = [1 \;\; N(D)/P(D)]. \tag{5}$$

Les performances ainsi obtenues sont meilleures que celles du code défini par (4) si K est petit et si le rapport signal à bruit est grand, mais ces codes ne sont utiles que si K est grand et au voisinage de la capacité, donc avec un mauvais rapport signal à bruit. Tout ce qui va suivre s'applique indifféremment aux codes définis par les génératrices (4) ou (5).

Le codage défini par les matrices (4) ou (5) associe à chaque suite d'information une suite de redondance qui est la plupart du temps de poids infini, sauf pour une fraction 2^{-K} du nombre total des suites d'information. On peut rendre cette fraction négligeable en choisissant K suffisamment grand, condition qui doit aussi être remplie pour imiter effectivement le codage aléatoire.

On peut même empêcher l'émission de suites de poids fini en insérant dans le message un symbole supplémentaire 0 toutes les fois que le contenu du registre du codeur risque de devenir nul (ce qui aurait lieu si un symbole 1 était appliqué à l'entrée du codeur). Alors, seules sont émises des suites de poids infini. La contrepartie de cet avantage est une augmentation du taux d'émission, petite si la longueur de contrainte K est grande. Par ailleurs, ce procédé ne peut être employé que si la probabilité d'une erreur de décodage est effectivement négligeable, pour permettre à coup sûr d'effacer à la réception les symboles supplémentaires insérés dans le message à

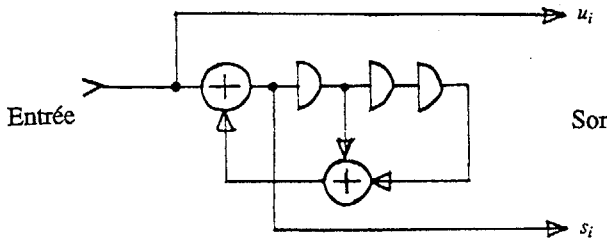

Fig. 1. Exemple de codeur récursif pseudo aléatoire

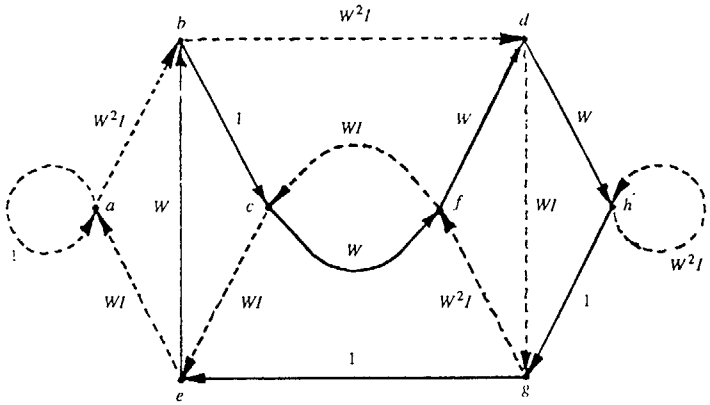

Fig. 2. Diagramme de transitions du codeur de la figure 1
Les états sont notés a, b, ... h, a représentant l'état 0. Les branches sont marquées avec les indéterminées W et I. L'exposant de W est le poids de la branche tandis que celui de I est le bit d'information correspondant. Tous les bits d'information sont 0 le long du chemin fermé dessiné en trait plein, mais le poids (la somme des exposants de W pour toutes ses branches) n'en est pas nul.

l'émission. Si cette condition n'était pas respectée, les erreurs de décodage entraîneraient des décalages dans la suite restituée au destinataire.

4. Décodage des codes convolutifs aléatoires et pseudo aléatoires

La complexité du décodage optimal, réalisé par exemple par l'algorithme de Viterbi, conduit à chercher des algorithmes non exactement optimaux.

On peut décoder le code qui vient d'être décrit avec une variante itérée du décodage par répliques [6]. L'ensemble des répliques linéairement indépendantes à partir duquel est écrite la règle de décision est infini, ce qui oblige à le tronquer. De ce fait, le décodage n'est pas exactement optimal, mais l'itération élargit indéfiniment le contexte en fonction duquel les décisions sont prises. Son emploi est cependant rendu délicat par la dépendance que les décisions antérieures induisent entre les symboles, qui oblige à introduire des facteurs de pondération correctifs.

Un procédé plus simple, non exactement optimal lui aussi mais dépourvu de ces inconvénients, peut être déduit du décodage par résolution d'un système d'équations implicites [9] convenablement généralisé aux codes convolutifs. Ainsi, avec le code convolutif de matrice génératrice (2), le système à résoudre s'écrit

$$A_i = a_i + z_i \prod_{k=1}^{i} [t(A_{i-k})]^{r_k} + \cdots + r_j z_{i+j} \prod_{k=0, k\neq j}^{i+j} [t(A_{i+j-k})]^{r_k} + \cdots, \tag{6}$$

avec $i = 0, 1, 2, ...$, où a_i et z_i sont respectivement les valeurs relatives *a priori* des i-ièmes symboles d'information et de redondance, A_i est la valeur relative *a posteriori* du i-ième symbole d'information et $t(x) = [\exp(x)-1]/[\exp(x)+1] = \tanh(x/2)$. Rappelons que nous définissons la valeur relative a d'une variable aléatoire binaire B par le rapport de vraisemblance logarithmique

$$a = \ln \frac{\Pr(B=0)}{\Pr(B=1)}.$$

La locution "*a priori*" signifie que l'estimation des probabilités ne concerne que les symboles reçus pris séparément et "*a posteriori*" qu'au contraire elle tient compte des contraintes du code. Une règle de décodage symbole par symbole, telle que la résolution de (6), est un moyen d'exprimer les valeurs relatives *a posteriori* en fonction des valeurs relatives *a priori* [10].

Les coefficients $\{r_i\}$ ont été définis en (3) pour une séquence aléatoire vraie R(D) ou bien, dans le cas d'un code pseudo aléatoire, sont les coefficients du développement en série formelle du second élément de la matrice génératrice (4) ou (5). Bien entendu, il est nécessaire en pratique de tronquer ce système à un nombre fini s d'équations, les indices j dans chacune d'elles étant eux-mêmes limités à $s-1$. D'autre part, la fonction $t(x)$ devient très voisine de ± 1 dès que $|x|$ atteint une valeur moyennement grande, fait que l'on peut exploiter pour beaucoup simplifier les calculs.

La résolution approximative de (6) peut être effectuée par itération, en prenant pour valeurs initiales des $\{A_i\}$ les valeurs relatives *a priori* $\{a_i\}$ des symboles d'information. La complexité de la résolution de (6) croît assez lentement en fonction de ce nombre pour qu'on puisse le choisir grand. Chaque décision dépend alors d'un large contexte, condition nécessaire pour obtenir des performances proches de la limite ultime qui le suppose infini.

Par exemple, avec $P(D) = 1 + D^2 + D^5$ la partie de redondance de la matrice (4), tronquée à ses 6 premières lignes et colonnes et écrite en binaire, s'écrit :

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ & 1 & 0 & 1 & 0 & 1 \\ & & 1 & 0 & 1 & 0 \\ & & & 1 & 0 & 1 \\ & & & & 1 & 0 \\ & & & & & 1 \end{bmatrix}$$

Le système (6) correspondant s'écrit, pour $i = 0$:

$$A_0 = a_0 + z_0 + z_2 t(A_2) + z_4 t(A_2) t(A_4) + z_5 t(A_1) t(A_3) t(A_5)$$

$$A_1 = a_1 + z_1 + z_3 t(A_3) + z_5 t(A_0) t(A_3) t(A_5)$$

$$A_2 = a_2 + z_2 t(A_0) + z_4 t(A_0) t(A_4)$$

$$A_3 = a_3 + z_3 t(A_1) + z_5 t(A_0) t(A_1) t(A_5)$$

$$A_4 = a_4 + z_4 t(A_0) t(A_2)$$

$$A_5 = a_5 + z_5 t(A_0) t(A_1) t(A_3)$$

Ceci n'est qu'un exemple et il faudrait un nombre beaucoup plus grand de termes et d'équations pour obtenir des résultats ayant un intérêt pratique.

Le procédé de décodage qui vient d'être sommairement décrit ne fait aucun usage de la forte structure algébrique du codeur et pourrait également être employé avec une suite aléatoire vraie, résultant d'un tirage préalable. Nous n'avons pas réussi jusqu'ici à exploiter cette structure de façon efficace, problème que nous considérons comme ouvert.

5. Conclusion

Les codes convolutifs récursifs à grande longueur de contrainte, notamment ceux qui utilisent un générateur de suites pseudo aléatoires, atteignent des performances voisines de la limite ultime fixée par la capacité du canal. Leur décodage est possible par des moyens simples. Le principal facteur qui limite les performances devient le retard total admissible pour les opérations de codage et décodage, plutôt que les propriétés intrinsèques du code.

Références

[1] C. Berrou, A. Glavieux et P. Thitimajshima, Near Shannon Limit Error-Correcting Coding and Decoding : Turbo-Codes, ICC'93, Genève, 23-26 mai 1993, pp. 1064-1070

[2] G.D. Forney Jr, Convolutional Codes I: Algebraic Structure, IEEE Trans., vol. **IT-16**, n° 6, nov. 1970, pp. 720-738

[3] G. Battail, Construction explicite de bons codes longs, Annales Télécommunic., vol. **44**, n° 7-8, juil.-août 1989, pp. 392-404

[4] G. Battail, Codage déterministe imitant le codage aléatoire, 13-ième colloque GRETSI, Juan-les-Pins, 16-20 septembre 1991, pp. 397-400

[5] G. Battail, H. Magalhães de Oliveira et W. Zhang, Codage déterministe imitant le codage aléatoire pour le canal à bruit gaussien additif, Annales Télécommunic., vol. **47**, n° 9-10, sept.-oct. 1992, pp. 433-447

[6] G. Battail, M. Decouvelaere et P. Godlewski, Replication Decoding, IEEE Trans. vol. **IT-25**, n° 3, mai 1979, pp. 332-345

[7] G. Battail, Décodage pondéré optimal des codes linéaires en blocs I.- Emploi simplifié du diagramme du treillis, Annales Télécommunic., vol. **38**, n° 11-12, nov.-déc. 1983, pp. 443-459

[8] G. Battail et J. Fang, Décodage pondéré optimal des codes linéaires en blocs II. - Analyse et résultats de simulation, Annales Télécommunic., vol. **41**, n° 11-12, nov.déc. 1986, pp. 580-604

[9] G. Battail, Décodage par résolution d'un système d'équations implicites analogiques, Annales Télécommunic., vol. **45**, n° 7-8, juil.-août 1990, pp. 393-409

[10] G. Battail, Le décodage pondéré en tant que procédé de réévaluation d'une distribution de probabilité, Annales Télécommunic., vol. **42**, n° 9-10, sep.-oct. 1987, pp. 499-509