Codes récursifs pseudo-aléatoires et leur décodage par répliques, avec itération

Gérard Battail

Ecole Nationale Supérieure des Télécommunications
Département Communications et URA 820 du CNRS,
46, rue Barrault,
F-75634 Paris Cedex 13

RÉSUMÉ

Nous montrons que les seuls codes convolutifs capables d'imiter le codage aléatoire sont récursifs. Nous suggérons à cet effet les codes dits "pseudo-aléatoires" dont le codeur comporte un registre rebouclé à longueur maximale. Nous examinons les problèmes posés par leur décodage. Ces codes n'imitent que faiblement le codage aléatoire, en ce sens que leur distribution de poids s'écarte d'autant plus de celle du codage aléatoire que le poids est petit. Il en résulte un taux d'erreur résiduelle par bit à décroissance lente en fonction du rapport signal à bruit. S'il ne satisfait pas aux spécifications de l'utilisateur, il peut être amélioré par concaténation, notamment selon le schéma des "turbo-codes". Ceux-ci apparaissent d'ailleurs comme imitant intrinsèquement le codage aléatoire quand ils combinent un grand nombre de codes.

ABSTRACT

We show that the only possible random-like convolutional codes are recursive. We suggest for this purpose the so-called "pseudo-random" codes, with an encoder register feedback generating maximum-length sequences. We examine how to decode them. These codes are only weakly random-like since their weight distribution departs from that of random coding the more, the smaller is the weight. A residual bit error rate results which slowly decreases in terms of the signal-to-noise ratio. If it does not match the user's requirements, it can be improved by concatenation, especially according to the "turbo-codes" scheme. Turbo-codes moreover appear as intrinsically random-like when they combine a large number of component codes.

1 Introduction

Les performances exceptionnelles des "turbo-codes" ont suscité un grand intérêt mais aussi des questions, la principale étant d'en comprendre les raisons [1]. Nous avons suggéré qu'elles découlaient principalement de la récursivité de ces codes [2, 3]. En fait, les codes convolutifs récursifs ont été négligés depuis que Forney a affirmé qu'ils n'avaient pas d'intérêt particulier [4]. Cela n'est cependant vrai que pour le critère de distance minimale, dont nous avons contesté la pertinence pour les codes longs. Nous avons proposé pour critère espéré meilleur la similitude de la distribution des distances d'un code par rapport à celle qui est obtenue en moyenne par codage aléatoire (dit pour abréger critère "pseudo-aléatoire") [5] et entrepris la recherche de codes satisfaisants pour ce critère. À cet effet, nous avons introduit les codes convolutifs récursifs pseudo-aléatoires, avantageusement combinés selon le schéma des turbo-codes [2, 3]. La présente communication s'inscrit dans le prolongement direct de [2]. Nous exposons les problèmes rencontrés, notamment pour le décodage de ces codes, et comment leur combinaison selon le schéma des turbo-codes en améliore les performances. Nous montrons par ailleurs que les turbo-codes sont bons pour le critère pseudo-aléatoire quand ils combinent un grand nombre de codes composants.

2 Pourquoi un codage récursif ?

Les codes convolutifs récursifs ont la propriété d'associer des suites de poids arbitrairement grand à tous les messages de poids fini, à l'exception d'une fraction d'entre eux, qui décroît exponentiellement en fonction de la mémoire K du codeur [3, 6]. Il n'en est pas de même des codes non récursifs, de sorte qu'il est *exclu* que des codes non récursifs puissent imiter le codage aléatoire.

Au contraire, en utilisant un codeur récursif dont la matrice génératrice contient au moins une fraction rationnelle dont le dénominateur est un polynôme primitif de degré K en l'opérateur de retard D, on obtient une distribution des poids qui imite celle du codage aléatoire si K est suffisamment grand [6]. Nous désignons un tel code par "pseudo-aléatoire" puisque le codeur se comporte comme un générateur de suites de longueur maximale, souvent qualifiées ainsi. Bien que les turbo-codes, qui combinent deux codes courts avec un entrelacement, ne soient pas de ce type, non seulement ils équivalent à un code de longueur de contrainte double dont le décodage est scindé en celui des deux codes courts, donc largement facilité, mais encore ils améliorent la distribution des poids par rapport à celle de ce code unique. Nous montrons l'intérêt d'employer des codes constituants pseudo-aléatoires dans ce schéma. Nous verrons en outre que la distribution de poids obtenue avec un turbo-code combinant plusieurs codes quelconques tend vers celle du codage aléatoire quand le nombre des codes combinés augmente.

Nous examinons d'abord ici l'emploi d'un code convolutif pseudo-aléatoire unique où la mémoire K du codeur est prise assez grande pour que la fraction des suites codées de longueur finie soit négligeable. L'emploi d'une aussi longue mémoire paraît poser de redoutables problèmes de décodage en rendant prohibitive la complexité de l'algorithme de Viterbi. Nous montrons qu'il est possible de construire des codes de ce type facilement décodés par répliques [7, 8], avec une itération du décodage où chaque étape est précédée d'une normalisation des données provenant de l'étape précédente.

3 Codage récursif pseudo-aléatoire

3.1 Définition et propriétés

Proposons nous par exemple de construire un code systématique pseudo-aléatoire binaire de taux 1/2. Sa génératrice est la matrice ligne

$$\mathbf{G} = [1 \quad \mathrm{N}(D)/\mathrm{P}(D)] \tag{1}$$

à deux éléments : 1 (le code est systématique) et la fraction rationnelle $\mathrm{N}(D)/\mathrm{P}(D)$ (il est récursif). Pour que le code soit pseudo-aléatoire, le dénominateur $\mathrm{P}(D)$ doit être un polynôme primitif de degré K et le numérateur $\mathrm{N}(D)$ un polynôme différent, de degré au plus égal à K. La matrice $\mathbf{G}_{éq}$ du code non récursif et non systématique équivalent se déduit de $\mathbf{G}$ en la multipliant par le scalaire $\mathrm{P}(D)$, soit

$$\mathbf{G}_{éq} = [\mathrm{P}(D) \quad \mathrm{N}(D)]. \tag{2}$$

La suite $\boldsymbol{v}(D)$ émise par le codeur récursif de génératrice (1) en réponse à la suite d'information $\boldsymbol{u}(D)$ est représentée par la matrice ligne $\boldsymbol{v}(D) = [\boldsymbol{u}(D) \quad \boldsymbol{s}(D)]$, où

$$\boldsymbol{s}(D) = \boldsymbol{u}(D)\mathrm{N}(D)/\mathrm{P}(D).$$

La suite $\boldsymbol{s}(D)$ est donc représentée par une fraction rationnelle en D qui admet un développement en série infinie, sauf si $\boldsymbol{u}(D)$ est multiple de $\mathrm{P}(D)$. La proportion des suites $\boldsymbol{u}(D)$ multiples de $\mathrm{P}(D)$, donc de poids fini, est 2^{-K} et devient donc très petite pour K grand. Les suites de redondance émises sont alors presque toujours de poids croissant indéfiniment.

3.2 Décodage d'un code pseudo-aléatoire

L'emploi d'une mémoire K grande, dont on a vu l'intérêt pour doter le code de propriétés pseudo-aléatoires, entraîne l'impossibilité d'employer l'algorithme de Viterbi pour son décodage, due à la complexité du treillis qui comporte 2^K états. Si l'on ne cherche pas à rendre la distance limite (*free distance*) grande, il est possible de choisir conjointement le numérateur et le dénominateur de la fraction rationnelle de telle sorte que le décodage par répliques soit facile et puisse être itéré [7, 8]. Le processus de décodage d'un bit quelconque (d'information ou de redondance) est décrit à l'aide d'un treillis dont la complexité ne dépend que du nombre r des répliques (et non de la mémoire K du codeur comme dans l'algorithme de Viterbi). Les données sortant du décodeur sont de même forme que celles qui y entrent (des nombres réels avec le formalisme utilisé). De plus, la complète symétrie du traitement permet d'opérer une normalisation qui annule, pour l'essentiel, l'effet de la dépendance induite par le décodage sur les données sortantes, et rend donc son itération possible. Suffisamment prolongée, cette itération s'avère très efficace pour élargir le contexte en fonction duquel chaque décision est prise. On remarquera que l'itération du décodage est aussi une caractéristique importante des turbo-codes [1].

La normalisation effectuée est basée sur la remarque que, pour un bruit gaussien additif indépendant du message codé, la valeur relative *a priori* d'une variable aléatoire binaire B, grandeur définie par

$$a = \ln \frac{\Pr(B=0)}{\Pr(B=1)}, \tag{3}$$

disponible en sortie du démodulateur si B est l'hypothèse faite en réception quant au bit émis, est une variable aléatoire de variance égale au double de sa moyenne. Le décodage consiste à calculer la valeur relative *a posteriori*, de même définition (3) à ceci près que les probabilités y sont calculées en tenant compte des contraintes dues au codage. On lui restitue les propriétés statistiques d'une variable affectée de bruit gaussien indépendant du signal, ce qui rend possible l'itération du décodage, en restaurant cette relation entre moyenne et variance en multipliant les valeurs relatives *a posteriori* par un coefficient approprié. Moyenne et variance sont évaluées à partir des données disponibles, donc à partir de moyennes temporelles.

En interprétant le décodage par répliques comme un procédé de diversité calculée, où chaque bit est reconstitué en fonction d'un certain ensemble d'autres bits reçus, que l'on dira son contexte, on voit que l'itération a pour effet d'élargir progressivement le contexte en fonction duquel la décision sur un bit est prise, tendant asymptotiquement vers le contexte infini qui est la propriété fondamentale du codage convolutif pseudo-aléatoire. Des taux d'erreurs par bit petits sont obtenus ainsi près de la capacité du canal.

Cependant, si petite que soit la proportion des suites de faible poids, elle est du même ordre que l'inverse du nombre des états dans le treillis associé au décodage par l'algorithme de Viterbi. D'ailleurs, il est bien connu que ce treillis est commun au codeur récursif et au codeur non-récursif équivalents. Les différences de probabilité des chemins dans les deux cas ne paraissent pas facilement exploitables par un décodeur, et on peut même se demander si c'est possible. Nous sommes donc dans le cas d'un code qui imite faiblement le codage aléatoire, c'est-à-dire bien dans la partie centrale de la distribution mais non dans les "queues" de celle-ci [9].

Le décodage n'est simple qu'avec un nombre de répliques r petit. Pour de faibles valeurs du rapport signal à bruit, les erreurs sont fréquentes aux bas rapports signal à bruit si les poids les plus faibles sont petits. Or, le poids minimal est la somme des poids du numérateur et du dénominateur de la fraction rationnelle. Pour améliorer la probabilité d'erreur résiduelle en conservant un code pseudo-aléatoire unique, il faudrait donc augmenter le nombre r des répliques, donc les poids des polynômes $\mathrm{P}(D)$ et $\mathrm{N}(D)$, aux dépens de la simplicité du décodage.

4 Combinaison de codes récursifs pseudo-aléatoires selon le schéma des turbo-codes

La distribution des poids ne diffère cependant de celle du codage aléatoire que par l'existence d'un *petit* nombre de suites de *faible* poids. Pour un taux inférieur à la capacité, les suites erronées sont donc presque sûrement voisines de la suite émise, puisqu'elles s'en déduisent par addition d'une suite représentée par un multiple du polynôme P(D), choisi de faible poids pour faciliter le décodage. Le taux d'erreur par bit y reste petit. Les suites erronées appartiennent ainsi presque certainement à un sous-ensemble très minoritaire de suites du code, qui contient la suite émise. Le décodage reste donc efficace si on l'interprète comme destiné à *diminuer l'incertitude* quant à la suite émise, même s'il est erroné au sens habituel. De plus, il est facile dans ce cas de réduire le taux d'erreur par bit en choisissant les suites d'information émises dans un autre sous-ensemble de toutes les suites possibles. On est ainsi conduit très naturellement à un schéma de *concaténation avec entrelacement*, celui même des turbo-codes.

On remarquera que l'entrelacement, supposé aléatoire, a pour résultat le *produit de convolution* des distributions de poids des suites de redondance composantes. En outre le poinçonnage, qui sert à maintenir un taux d'émission constant, ramène la distribution obtenue au même support que la distribution initiale tout en en conservant la forme. Comme nous l'avons vu, la distribution de poids d'un code composant récursif pseudo-aléatoire est voisine de celle qui est obtenue en moyenne par codage aléatoire, à part l'excès de suites de faible poids qui entraîne une dégradation du taux d'erreur par bit par rapport au codage aléatoire. Le produit de convolution des distributions de poids opéré par la concaténation avec entrelacement conserve cette forme dans la partie centrale mais en diminue les "queues", notamment vers les poids faibles ce qui améliore notablement le taux d'erreur. La figure 1 représente le schéma du codeur d'un turbo-code combinant $N+1$ codes composants de taux 1/2, dont le taux est ramené à cette même valeur par poinçonnage périodique. Le choix du nombre N permet d'ajuster la probabilité d'erreur résiduelle par bit aux spécifications de l'utilisateur. En outre, quelle que soit la distribution de poids initiale des codes composants (même si ceux-ci ne sont pas pseudo-aléatoires), celle qui est obtenue ainsi converge pour N grand vers la distribution moyenne du codage aléatoire, par effet de tendance vers la loi de Gauss.

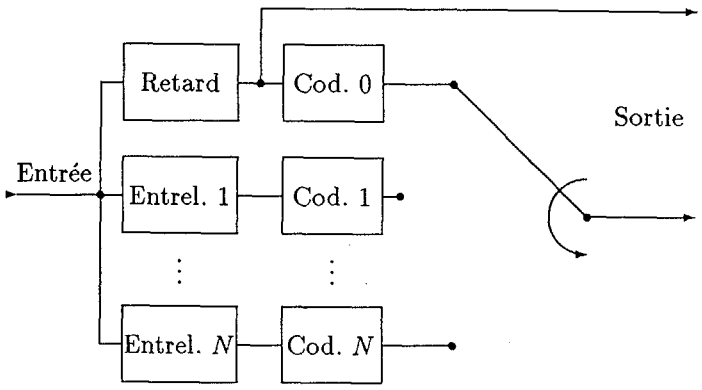

Figure 1: Codeur d'un turbo-code pseudo-aléatoire de taux 1/2. Les boîtes marquées "Cod. 0, Cod. 1, ..., Cod. N" engendrent les bits de redondance correspondant à la suite d'entrée selon un code pseudo-aléatoire de taux 1/2. Les boîtes marquées "Entrel. 1, ..., Entrel. N" représentent des entrelaceurs, dont le retard est compensé dans la boîte marquée "Retard". Le poinçonnage périodique maintient le taux 1/2.

5 Conclusion

La recherche de codes convolutifs dont la distribution de poids imite celle du codage aléatoire nous a conduit à proposer des codes pseudo-aléatoires, récursifs comme ceux qu'ont employés Berrou *et al.* [1], et où le registre du codeur est en outre rebouclé à longueur maximale.

La difficulté du décodage nous a conduit à préférer des codes de faible poids minimal, dont le décodage risque d'être affecté de l'addition d'une suite de faible poids. La combinaison de plusieurs de ces codes selon le schéma des turbo-codes apparaît alors comme comme un moyen simple de diminuer le taux d'erreur par bit pour l'ajuster aux spécifications de l'utilisateur. En outre, impliquant un produit de convolution des distributions de poids, la combinaison d'un grand nombre de codes par ce schéma a pour résultat une distribution de poids qui tend vers celle du codage aléatoire, quelle que soit celle des codes composants.

Références

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

[2] Gérard Battail, Claude Berrou et Alain Glavieux, "La capacité d'un canal peut être approchée par des moyens simples," *14-ième Colloque Gretsi*, pp. 623–626, Juan-les-Pins, 13-16 Sept. 1993.

[3] G. Battail, C. Berrou et A. Glavieux, "Pseudo-random recursive convolutional coding for near-capacity performance," *Proc. GLOBECOM'93, Communication Theory Mini-Conference*, Vol. 4, pp. 23–27, Houston, U.S.A., 29 nov.-2 déc. 1993.

[4] G.D. Forney Jr, "Convolutional Codes I: Algebraic Structure," *IEEE Trans. Inf. Th.*, Vol. IT-16, No. 6, pp. 720–738, nov. 1970.

[5] G. Battail, "Construction explicite de bons codes longs," *Annales Télécommunic.*, Vol. 44, No. 7–8, pp. 392–404, juil.-août 1989.

[6] G. Battail, "Codage convolutif récursif pseudo-aléatoire," proposé aux *Annales Télécommunic.*

[7] G. Battail et M. Decouvelaere, "Décodage par répliques," *Annales Télécommunic.*, Vol. 31, No. 11-12, pp. 387-404, nov.-déc. 1976.

[8] G. Battail, M. Decouvelaere et P. Godlewski, "Replication Decoding," *IEEE Trans. Inf. Th.*, Vol. IT-25, No. 3, pp. 332-345, mai 1979.

[9] G. Battail, "On random-like codes," *4-th Canadian Workshop on Information Theory*, Lac Delage, Québec, Canada, 29–31 mai 1995.