Identification des paramètres d'un code correcteur par déficience du nombre de classes

Aurélien BONVARD[1 2], Sébastien HOUCKE[1], Mélanie MARAZIN[2], Roland GAUTIER[2]

[1]Télécom Bretagne; Dépt. Signaux et Communications, CNRS,UMR 6285 Lab-STICC
Technopôle Brest-Iroise, CS 83818, 29238 Brest Cedex 3, France

[2]Université de Brest; CNRS,UMR 6285 Lab-STICC
6 avenue Victor Le Gorgeu, CS 93837, 29238 Brest cedex 3, France

aurelien.bonvard@telecom-bretagne.eu, sebastien.houcke@telecom-bretagne.eu
melanie.marazin@univ-brest.fr, roland.gautier@univ-brest.fr

Résumé – La plupart des systèmes de communication numérique actuels mettent en place des codes correcteurs d'erreurs afin de permettre l'accès à l'information à la réception, malgré les altérations dues au canal de transmission. On s'intéresse à l'identification des paramètres du code correcteur utilisé sans autre information que le flux de données reçu. Ce type de problématique se rencontre principalement dans le contexte de la guerre électronique. Nous proposons un algorithme capable de déterminer la taille des mots de code en les affectant à une classe grâce à un critère de distance. Cet article en décrit les aspects sous-jacents en adoptant un point de vue géométrique pour les cas où l'on dispose d'une information ferme et souple (LLR des bits).

Abstract – Most of current digital communication systems uses an error correcting code to allow the receiver to access the information despite the modifications due to the transmission channel. We are interested in the identification of the parameters of an error correcting code, with no more information but the received data stream. This type of problem is particularly relevant in the context of the electronic warfare. We propose an algorithm which is able to find the size of code words by sorting them in classes with a distance criterion. This paper explains the underlying aspects from a geometrical point of vue in the hard and the soft cases.

1 Introduction

Dans les télécommunications l'utilisation de codes correcteurs d'erreurs est devenue courante. Ces codes ont pour objectif d'accroître la qualité de la réception de l'information en détectant et en corrigeant les erreurs potentielles introduites par le canal. L'identification aveugle de codes correcteurs d'erreurs se révèle d'un grand intérêt dans le contexte de la guerre électronique où le type de codeur mis en place à l'émission est totalement inconnu. Dans le domaine civil, elle peut potentiellement participer à la faculté d'adaptation des systèmes radio à leur environnement (i.e. Adaptive Modulation & Coding).

Au cours des dernières années, de nombreux travaux sur la détection en aveugle de codes ont vu le jour. Des approches à partir des informations fermes, comme proposées dans [1], sont basées sur le critère du rang : les informations reçues sont ordonnées en matrice dont le nombre de colonnes varie, quand il est multiple de la taille des mots de code, le rang chute. Plus récemment, certaines méthodes s'appuient sur les informations souples avant la prise de décision afin de s'inspirer des décodeurs souples à rapport de vraisemblance. Ce type d'algorithmes est exposé dans [2] et [3] pour la reconnaissance en aveugle de codes issus d'une population finie et connue.

Dans cet article, un point de vue géométrique est adopté, comme cela a été envisagé dans [4]. Considérons un ensemble d'informations codé et transmis via un canal introduisant du bruit. Le flux de bits reçus est découpé en blocs de taille n. Les blocs sont alors regroupés pour former un ensemble de classes. Lorsque n est égal à la taille des mots de code, on constate une déficience du nombre de classe, ce qui permet d'identifier sa valeur. Ici, deux types d'informations sont considérés, ferme et souple, auxquels sont associées deux métriques : la distance de Hamming et la distance euclidienne, respectivement.

Dans la seconde section, le modèle de la transmission est défini, ainsi que la procédure de classification. Ce qui nous amène à détailler la notion de déficience en présence de bruit dans les cas ferme et souple successivement. Enfin, des résultats sont exposés pour confronter les performances des cas ferme et souple et observer l'impact de la quantité d'information reçue sur la qualité de la détection. La conclusion et les perspectives sont présentées dans la section 4.

2 Déficience du nombre de classes

2.1 Modèle et principes de la méthode

Nous considérons un émetteur qui utilise un code en bloc linéaire noté $C(n_c, k_c)$, avec n_c la taille des mots de code et k_c

celle des mots d'information. Un entrelaceur pseudo-aléatoire de taille n_c y est adjoint. On suppose qu'une modulation par déplacement de phase à 2 états a été mise en place à l'émission. Le signal reçu $\mathbf{y}$ contient L échantillons et il est de la forme suivante :

$$y_k = s_k + b_k, \forall k \in [\![0, L-1]\!] \tag{1}$$

où s_k est le $k^{ième}$ élément du signal émis $\mathbf{s}$ et b_k le $k^{ième}$ élément du bruit $\mathbf{b}$ introduit par le canal de transmission. Les s_k prennent leurs valeurs dans $\{\pm 1\}$.

On suppose que la synchronisation est parfaite et que la trame reçue commence au début d'un mot de code. Si on représente les mots reçus par des points dans un espace de dimension n_c, leurs positions sont altérées par le bruit. Des agglomérats se forment autour des points représentant les mots de code initialement envoyés, laissant le reste de l'espace vide.

Le principe de la méthode consiste à former des blocs $B_i^{(n)}$ de taille n comme suit :

$$B_i^{(n)} = [y_{i\cdot n}, \cdots, y_{i\cdot n+n-1}], \forall i \in [\![0\,, \lfloor \tfrac{L}{n} \rfloor - 1]\!] \tag{2}$$

où $\lfloor x \rfloor$ est la partie entière de x. Pour chaque valeur de n, le premier bloc $B_0^{(n)}$ constitue le premier représentant de la première classe : c'est un mot de référence. Les blocs suivants, $B_i^{(n)}, \forall i \neq 0$, sont comparés au mot de référence : si la distance entre les deux mots est inférieure à un seuil donné β alors ils font partie de la même classe, sinon le bloc $B_i^{(n)}, \forall i \neq 0$, devient le mot de référence d'une nouvelle classe. Une fois classé, le mot est retiré de l'ensemble des blocs créés, i.e. chaque bloc n'intègre qu'une seule et unique classe. À l'heure actuelle, le choix des représentants des classes ne dépend que de la manière dont sont parcourus les blocs. Ainsi pour chaque n, on obtient un nombre de classes m tel que :

$$m(n) = Card(C^{(n)}) \tag{3}$$

où $C^{(n)}$ est l'ensemble des mots de référence obtenus pour une valeur de n donnée, et $Card(C^{(n)})$ est son cardinal. Pour $n = n_c$, le nombre de classes, $m(n_c)$, chute de manière significative : on observe une déficience du nombre de classes. Ce phénomène est du à la redondance introduite par le code, car dans ce cas le nombre de mots de code est inférieur à 2^{n_c}. Il en va de même pour les multiples de n_c.

En l'absence de code ou dans le cas $n \neq n_c$, l'information reçue correspond à une suite de bits i.i.d.. Pour $L = 20\,000$ échantillons et $n \in [\![2, 30]\!]$, le nombre de classes détectées, $m_{sc}(n)$, est représenté sur la figure 1. Dans ce contexte, il y a potentiellement 2^n classes issues des blocs $B_i^{(n)}$, d'où la croissance exponentielle du nombre de classes pour les premières valeurs de n. Ensuite, la chute du nombre de classes est due à la limitation dans la quantité d'informations reçues : quand n augmente, le nombre de blocs diminue et la quantité de mots différents aussi.

Dans la sous section 2.2, nous considérons le cas d'un Canal Binaire Symétrique (CBS) de paramètre P_e.

2.2 Cas ferme

Une décision ferme est prise sur y_k afin de pouvoir considérer le cas d'un CBS de paramètre P_e :

- si $y_k > 0$ alors $\hat{y}_k = 1$
- si $y_k < 0$ alors $\hat{y}_k = 0$

Faisons l'hypothèse que la classification est parfaite, c'est-à-dire que chaque bloc issu du même mot de code est rangé dans la même classe, pour $l \in \mathbb{N}^*$on distingue deux cas :

- $n \neq l \cdot n_c$: dans ce cas, le nombre de classes obtenues correspond au cas sans codage.
- $n = l \cdot n_c$: dans ce cas, le nombre de classes issues des blocs $B_i^{(n)}$ est de $2^{l \cdot k_c} < 2^n$.

Dans le cas ferme, l'objectif est de rassembler tous les mots qui sont au maximum à une distance de Hamming donnée du mot de référence. Prenons l'exemple d'un code $LDPC(20, 10)$ ($n_c = 20$ et $k_c = 10$). Pour $L = 20\,000$ et $n \in [\![2, 30]\!]$, le cas sans bruit est représenté sur la figure 1 en considérant que la valeur du seuil à adopter sur la distance de Hamming pour créer les différentes classes est $\beta_F = 0$. Le nombre de classes $m_{sc}(n)$ en l'absence de code est superposé au nombre de classes $m(n)$ en présence du code, et nous vérifions bien qu'il y a une déficience du nombre de classes pour $n = n_c = 20$. Par contre, on remarque d'autres déficiences pour des valeurs de n différentes de n_c. En effet, même sous l'hypothèse d'une classification parfaite, il s'avère que la redondance introduite par le code peut avoir un impact sur le nombre de classes. Dans la suite, afin de pouvoir détecter la déficience prépondérante, nous proposons d'observer le nombre de classes normalisé $\varphi(n) = m(n)/m_{sc}(n)$.

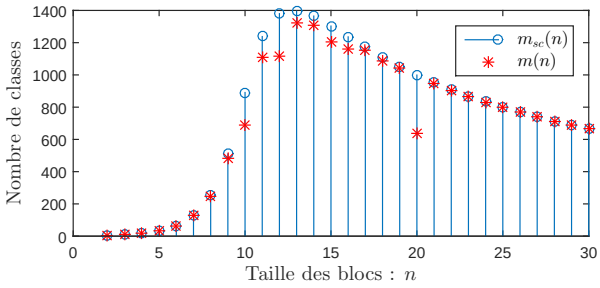

FIGURE 1 – Comparaison des résultats avec et sans code en l'absence de bruit pour un code $LDPC(20, 10)$

Dans le cas d'une transmission bruitée, la qualité de la classification est primordiale pour obtenir une bonne détection de n_c. C'est dans ce cas que la métrique utilisée pour comparer deux blocs joue un rôle important. Dans le cas présent, nous avons estimé une valeur optimale du seuil β_F de manière empirique. Pour cela, nous reconsidérons le code $LDPC(20, 10)$ précédent pour $L = 20\,000$. La figure 2 présente le nombre de classes normalisé $\varphi(n)$ dans le cas sans bruit pour des valeurs autour de $n = 20$ et pour $\beta_F \in \{1, 3, 5, 7\}$. La plus grande déficience apparaît pour $\beta_F = 3$: dans la suite nous considérerons donc cette valeur du seuil dans le cas ferme pour

le $LDPC(20, 10)$.

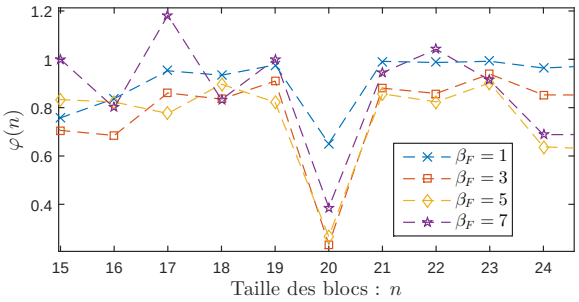

FIGURE 2 – Déficiences pour différentes valeurs de β_F pour un code $LDPC(20, 10)$

L'opération de normalisation permet d'observer l'impact du code sur le nombre de classe malgré la limitation due à la valeur de L. La figure 3 présente l'évolution de $\varphi(n)$ en présence de bruit caractérisé par $P_e = 0.01$. Nous pouvons vérifier que la plus grande déficience apparaît pour $n = n_c = 20$.

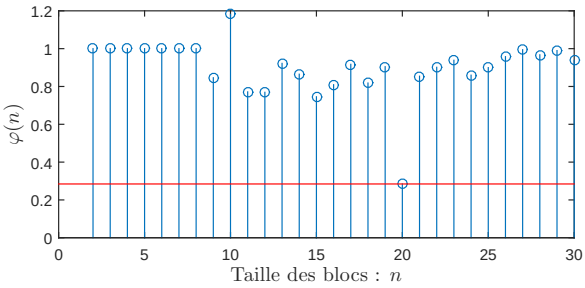

FIGURE 3 – Résultats normalisés dans le cas ferme pour un code $LDPC(20, 10)$ et $P_e = 0.01$

2.3 Cas souple

Dans cette sous-section, nous généralisons notre procédure au cas souple. Les altérations survenues lors de la transmission sont modélisées par le biais d'un canal à Bruit Blanc Additif Gaussien (BBAG) de variance σ^2. De manière analogue au cas ferme, il faut déterminer un seuil pour définir le rayon des classes, ainsi que la métrique. Nous avons choisi la distance euclidienne pour répondre à ce besoin.

Considérons deux blocs distincts de taille n, $B_i^{(n)}$ et $B_j^{(n)}$, $\forall i, j \in [\![0\,, \lfloor \frac{L}{n} \rfloor - 1]\!]$, issus de la même séquence de symboles $[s_{i\cdot n}, \cdots, s_{i\cdot n+n-1}] = [s_{j\cdot n}, \cdots, s_{j\cdot n+n-1}]$. Par définition :

$$\begin{aligned} d_E^2\left(B_i^{(n)}, B_j^{(n)}\right) &= \sum_{k=0}^{n-1}((s_{i\cdot n+k} + b_{i,k}) - (s_{j\cdot n+k} + b_{j,k}))^2 \\ &= \sum_{k=0}^{n-1}(b_{i,k} - b_{j,k})^2 \end{aligned} \quad (4)$$

où les $b_{i,k}$ et les $b_{j,k}$ sont les $k^{ièmes}$ éléments du bruit affectant respectivement les blocs $B_i^{(n)}$ et $B_j^{(n)}$, avec $k \in [\![0, n-1]\!]$, et $d_E\left(B_i^{(n)}, B_j^{(n)}\right)$ leur distance euclidienne.

On note $\mathbf{b}_i$ et $\mathbf{b}_j$ le bruit ayant affecté les blocs $B_i^{(n)}$ et $B_j^{(n)}$, respectivement. Les $\mathbf{b}_i = \{b_{i,k}\}$ et $\mathbf{b}_j = \{b_{j,k}\}$ suivent une loi normale de moyenne $\mu = 0$ et de variance σ^2. La différence $X_k = b_{i,k} - b_{j,k}$ est par conséquent la somme de deux variables aléatoires indépendantes, et suit elle-même une loi normale de moyenne $\mu_{tot} = 0$ et de variance $\sigma_{tot}^2 = 2\cdot\sigma^2$. Par conséquent, la variable X, telle que :

$$\begin{aligned} X &= \sum_{k=0}^{n-1}\left(\frac{X_k - \mu_{tot}}{\sigma_{tot}}\right)^2 = \sum_{k=0}^{n-1}\left(\frac{X_k}{\sqrt{2}\cdot\sigma}\right)^2 \\ &= \frac{1}{2\cdot\sigma^2}\cdot\sum_{k=0}^{n-1}X_k^2 = \frac{1}{2\cdot\sigma^2}\cdot d_E^2\left(B_i^{(n)}, B_j^{(n)}\right) \end{aligned} \quad (5)$$

suit une loi du χ^2 à n degrés de liberté. Grâce à la fonction de répartition de cette loi et la table de distribution correspondante (*cf* [5]), il est possible de déterminer β_S, le seuil pour délimiter les différentes classes dans le cas souple. Pour cela, on fixe P, la probabilité que X dépasse une valeur donnée α :

$$\begin{aligned} P &= Pr\left(X \geqslant \alpha\right) = Pr\left(\frac{d_E^2\left(B_i^{(n)}, B_j^{(n)}\right)}{2\cdot\sigma^2} \geqslant \alpha\right) \\ &= Pr\left(d_E\left(B_i^{(n)}, B_j^{(n)}\right) \geqslant \sqrt{2\cdot\alpha}\cdot\sigma\right) \\ &= Pr\left(d_E\left(B_i^{(n)}, B_j^{(n)}\right) \geqslant \beta_S\right) \\ \Rightarrow\ & \beta_S = \sqrt{2\cdot\alpha}\cdot\sigma \end{aligned} \quad (6)$$

Reconsidérons l'exemple du code $LDPC(20, 10)$ dans le cas d'un BBAG de variance $\sigma^2 = 0.1$. Le nombre de classes normalisé $\varphi(n)$, pour $L = 20\,000$ et $n \in [\![2, 30]\!]$ est représenté sur la figure 4. Nous vérifions bien que la plus grande déficience du nombre de classes est visible pour $n = n_c = 20$. Dans le cas souple, les valeurs successives de $m_{sc}(n)$ sont obtenues en appliquant le même niveau de bruit que celui contenu dans le signal reçu à une suite de bits i.i.d.. En effet, dans le cas souple, l'espace des mots forme un ensemble continu, contrairement au cas ferme où l'espace des mots est un ensemble discret. Dans le cas souple, le bruit déplace les mots hors des points d'accumulation existant dans le cas ferme, ce qui crée des régions de forte densité. Par conséquent, le bruit a une influence directe sur la formation de classes, même dans le cas d'une suite de symboles générés aléatoirement.

3 Détection et performances

Dans cette section, les résultats exposés concernent un code $LDPC(20, 10)$ et les probabilités de détection ont été calculées sur la base de 500 itérations de Monte Carlo.

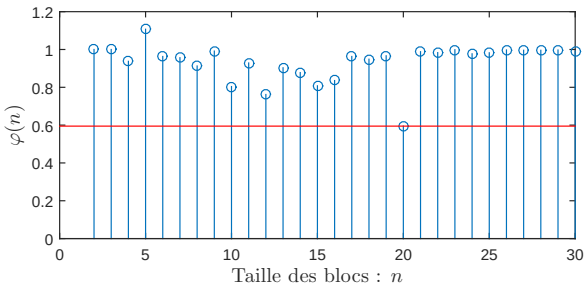

FIGURE 4 – Résultats normalisés pour un code $LDPC(20, 10)$ dans le cas souple pour un RSB de 23 db

3.1 Performances dans les cas ferme et souple

Le pouvoir de détection de notre algorithme dépend très fortement de la quantité de bruit dans l'information reçue. Pour $L = 20\,000$, la figure 5 fait état de la probabilité de détection en fonction de la quantité de bruit présent dans le signal reçu. Il s'avère que notre algorithme est plus efficace dans le cas souple. En effet, en considérant que $\beta_F = 3$ dans le cas ferme, nous constatons que la probabilité de détection est plus élevée lors du traitement des informations souples pour des valeurs du RSB comprises entre 0 et 15 dB.

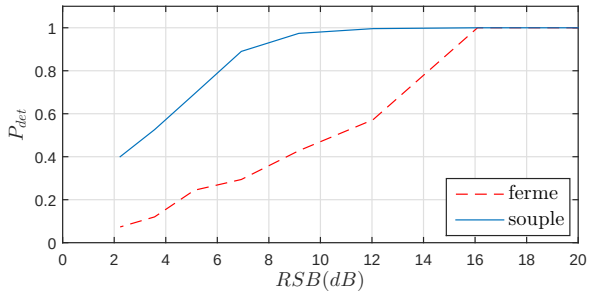

FIGURE 5 – Probabilité de détection dans les cas ferme et souple en fonction du rapport signal sur bruit

3.2 Impact de L

Les résultats proposés dans la sous-section précédente ne rendent pas compte de l'incidence de la valeur de L sur la probabilité de détection de notre algorithme. Plus L est grand, plus $P_{dét}$ augmente. La figure 6 permet d'observer l'évolution de la probabilité de détection en fonction de différentes valeurs de la quantité d'information reçue, L, dans les cas ferme et souple pour un RSB égal à 30 dB. Il s'avère que la limite de détection est moins élevée dans le cas ferme : la prise de décision décrite au début de la sous-section 2.2 provoque une perte d'information qui diminue la capacité de détection pour une valeur de L donnée. En effet, lorsque $L \in [\![10\,000, 15\,000]\!]$, la probabilité de détection est plus grande dans le cas souple.

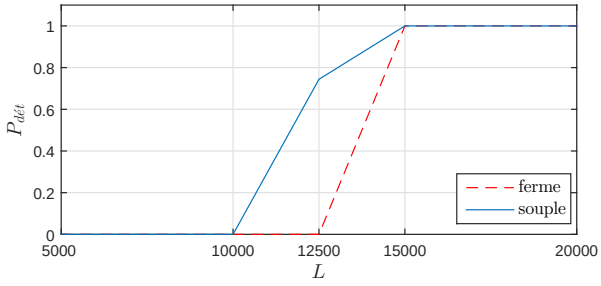

FIGURE 6 – Probabilité de détection dans les cas ferme et souple en fonction de la quantité d'information reçue

4 Conclusion

L'interprétation géométrique du problème d'identification aveugle permet de mettre en avant le caractère lacunaire de l'espace des mots de code comparé à celui des mots d'information. Le cas souple rend les agglomérats de mots reçus plus denses, ce qui permet une discrimination par classes plus efficace. L'algorithme présenté ordonne l'information reçue par classes et permet de déterminer la taille des mots de code quand une chute du nombre de classes est décelée. Nous avons montré l'importance du nombre d'échantillons reçus sur la détection. Il serait intéressant d'établir un seuil sur le nombre minimal de bits interceptés pour garantir la détection à un niveau de bruit donné. De même, notre algorithme pourrait être amélioré par la définition d'un critère d'arrêt : dès que $\varphi(n)$ est inférieur à un seuil donné, on arrête la recherche sur n. Enfin, un estimateur de la variance du bruit, σ^2, pourrait être mis en place : en effet, $m(n_c)$ dépend aussi de β (β_F dans le cas ferme et/ou β_S dans le cas souple), et étudier la fonction $\beta \mapsto m(n_c, \beta)$ permettrait d'estimer σ^2.

Références

[1] Y. Zrelli, R. Gautier, M. Marazin, E. Rannou and E. Radoi. *Focus on theoretical properties of blind convolutional codes identi ?cation methods based on rank criterion*. MTA Review, vol. XXII, no 4, pp. 213-234, Dec. 2012.

[2] R. Moosavi et E. G. Larsson. *Fast Blind Recognition of Channel Codes*. IEEE Transactions on Communications, vol. 62, NO. 5, pp. 1393-1304, 2014.

[3] T. Xia et H.-C. Wu. *Blind Identification of Nonbinary LDPC Codes Using Averge LLR of Syndrome a Posteriori Probability*. IEEE Communications Letters, vol. 17, NO. 7, pp. 1301-1304, 2013.

[4] C. E. Shannon. *Probability of Error for Optimal Codes in a Gaussian Channel*. The Bell System Technical Journal, vol. 38, NO. 3, pp. 611-656, 1959.

[5] G. Saporta. *Probabilités, Analyse de données et Statistique*. 2ème édition, Éditions Technip, p.93 et p.565-568, 2006.