Etude Statistique du seuil dans la détection d'entrelaceur

Guillaume SICOT[1], Sébastien HOUCKE[1]

[1]ENST Bretagne - Département Signal & Communication
Technopôle Brest-Iroise - CS 83818 - 29238 Brest Cedex 3
{guillaume.sicot} {sebastien.houcke} @enst-bretagne.fr

Résumé – Les entrelaceurs sont de plus en plus utilisés dans les systèmes de communications numériques utilisant des codes correcteurs d'erreurs. Dans un contexte non coopératif, tel que l'écoute passive, il se peut que le récepteur doit être capable d'estimer les différents paramètres de l'entrelaceur. Nous avons proposé un algorithme permettant d'estimer la taille de l'entrelaceur, le rendement de codage ainsi que le début de la trame entrelacée. Cet algorithme utilise un seuil de détection que nous proposions de fixer de manière heuristique. Dans cet article, nous montrons qu'il existe une valeur optimale de ce seuil et nous proposons une procédure itérative permettant de l'atteindre.

Abstract – Interleaving is a key component of many digital communication systems involving error correction schemes. In a non-cooperative context, such as passive listening, it is a challenging problem to estimate the interleaver parameters. Recently, we proposed an algorithm to estimate the size of the interleaver block, the code rate and to perform the blind frame synchronization of the interleaver blocks. This algorithm used an empirical threshold. In this paper we show that there exists an optimal threshold and we propose an iterative procedure that allows to converge to this optimal value.

1 Introduction

Les codes correcteurs d'erreurs possèdent généralement de meilleures propriétés de correction lorsque les erreurs sont équiréparties sur la trame plutôt que sous la forme de paquets d'erreurs. C'est pourquoi on entrelace les données avant émission. Ainsi, au niveau du récepteur, le désentrelacement permet de répartir uniformément les erreurs sur toute la trame.

Dans un contexte non-coopératif, les paramètres de l'entrelaceur et du schéma de codage doivent être estimés en aveugle afin de récupérer le message émis. Dans ce document nous allons nous intéresser à la détection des paramètres de l'entrelaceur. La technique proposée dans [1] résout ce problème dans le cas d'une transmission sans erreur. Afin d'étendre cette solution en présence d'erreurs de transmission une solution a été proposée dans [2]. Ces deux algorithmes permettent de retrouver des caractéristiques de l'entrelaceur telles que sa taille, le début du bloc de l'entrelaceur, le motif d'entrelacement utilisé mais aussi des informations sur le code utilisé telles que son rendement.

Dans un premier temps, nous présentons l'algorithme publié dans [2]. Ensuite une étude théorique des performances de cet algorithme est réalisée en particulier en ce qui concerne les limites de détection pouvant être attendues suivant le taux d'erreur présent dans le canal de transmission. Cette étude permet de mieux comprendre le fonctionnement de notre méthode et surtout donne des critères permettant de choisir objectivement la valeur d'un paramètre primordial de l'algorithme.

2 Détection en aveugle d'un entrelaceur

Nous nous intéressons au cas où le code correcteur utilisé est un code en bloc et l'entrelaceur est également un entrelaceur par bloc. L'opération de codage à l'aide d'un code en bloc est complètement définie par une matrice génératrice de rang plein G, qui transforme chaque bloc de k_c bits d'information en n_c bits codés (avec $k_c < n_c$).

Un entrelaceur peut être modélisé par une matrice de permutation P de taille $S \times S$ où S représente la taille de l'entrelaceur. En général, la taille de l'entrelaceur est un multiple de taille d'un mot de code, hypothèse que nous ferons ici. Ainsi, nous avons la relation : $S = N.n_c$, où N représente le nombre de mots de code dans le bloc de l'entrelaceur.

Par la suite, $\mathbf{X}$ désigne la séquence émise composée de M blocs entrelacés, et $\mathbf{Z}$ la séquence interceptée. $\mathbf{Z}$ est une version de $\mathbf{X}$, décalée de t_0 bits (retard de transmission) et entachée d'erreurs dues au canal de propagation. Le canal est modélisé par un canal binaire symétrique de probabilité d'erreur P_e.

2.1 Sans erreur de transmission

Dans cette partie, nous supposons que le canal de transmission n'introduit aucune erreur de transmission (*i. e.* $P_e = 0$). Burel *et al.* proposent [1] de construire une matrice $H(n_a, d)$ à partir de la séquence $\mathbf{Z}$. Pour ce faire, on découpe la séquence $\mathbf{Z}$ à laquelle on a préalablement enlevé les d premiers bits, en blocs K contenant n_a bits. Ensuite ces K blocs de taille n_a forment les lignes de la matrice $H(n_a, d)$. Ainsi la matrice $H(n_a, d)$ est une matrice de dimension $K \times n_a$. Burel *et al.* proposent d'étudier le coefficient $\rho(n_a, d)$ défini par (1) pour différentes valeurs de n_a et de d.

$$\rho(n_a, d) = \frac{\text{rank}(H(n_a, d))}{n_a} \tag{1}$$

Il apparaît que $\forall\, d$, $\rho(n_a, d)$ est égal à 1, sauf lorsque n_a est un multiple de la taille de l'entrelaceur, c'est-à-dire lorsque $n_a = kS$, avec k un entier non nul. En effet, dans ce cas particulier, des corrélations entre colonnes apparaissent dues à la redondance introduite par le code correcteur d'erreur.

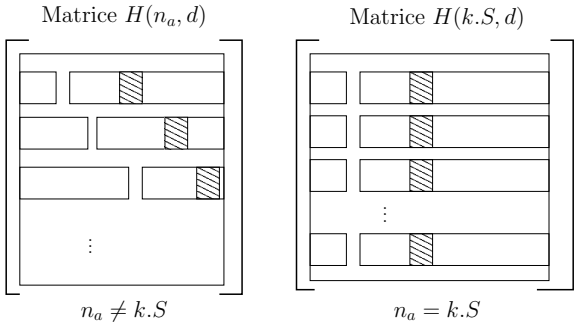

FIG. 1 – La matrice $H(n_a, d)$

La figure 1 présente les deux cas de figure qui peuvent se présenter au niveau de la matrice $H(n_a, d)$. L'élément grisé dans la figure 1 représente un bit de redondance : ce bit est donc une combinaison linaire d'autres bits appartenant au même bloc. Lorsque $n_a = kS$, cette même relation linéaire est aussi vérifiée par la ligne suivante de la matrice $H(kS, d)$ et donc par toute la colonne. La colonne hachurée est une combinaison linéaire d'autres colonnes et la matrice $H(kS, d)$ n'est pas de rang plein. Dès que $n_a \neq kS$ cette propriété n'est plus vérifiée. La déficience du rang de $H(n_a, d)$ permet ainsi d'estimer la taille de l'entrelaceur. Une fois S estimée, l'argument minimum de $\rho(kS, d)$ en fonction de d permet de trouver le début de la trame de l'entrelaceur et de déduire le rendement du code utilisé.

2.2 Avec erreurs de transmission

Dans le contexte d'une écoute passive, la séquence $\mathbf{Z}$ est vraisemblablement entachée d'erreurs, ainsi la matrice $H(n_a, d)$ devient généralement de rang plein même pour $n_a = kS$. Par conséquent la méthode précédemment citée devient peu performante. Pour pallier à ce problème, nous avons proposé dans [2] de construire la matrice $H(n_a, d)$ de la même manière que dans la méthode précédente. La matrice $H(n_a, d)$ est ensuite trianguralisée à l'aide de l'algorithme du pivot de Gauss modifié pour fonctionner dans l'espace binaire $\mathbb{F}_2$. Il est possible de définir l'opération réalisée par cet algorithme par la relation linéaire suivante :

$$A_1 H(n_a, d) A_2 = L(n_a, d) \tag{2}$$

où A_1 représente les permutations sur les lignes de $H(n_a, d)$, et A_2 représentant les opérations sur les colonnes. Enfin $L(n_a, d)$ représente la matrice triangulaire inférieure résultant de cet algorithme.Soit B_i la variable qui représente le nombre de 1 dans la partie inférieure de la colonne i de la matrice $L(n_a, d)$ (cf. figure 2).

Définissons par $\phi_{n_a}(i)$ la variable aléatoire suivante : $\phi_{n_a}(i) = \frac{B_i}{m_B}$ où m_B est un facteur de normalisation égal à

$$m_B = \frac{1}{2}\left(\lfloor \frac{MS}{n_a} \rfloor - n_a\right).$$

Il correspond à la moyenne de B_i si les bits de la colonne i sont indépendants et équiprobables (ce qui est le cas si $n_a \neq kS$).

Si $n_a \neq kS$, $\forall\, i \in \{1, .., n_a\}$, B_i suit une loi binomiale de paramètres $(K - n_a, \frac{1}{2})$ et il est facile de montrer que :

$$\forall i \in \{1, .., n_a\},\ \lim_{M\to\infty} \phi_{n_a}(i) \xrightarrow{\mathcal{P}} 1 \tag{3}$$

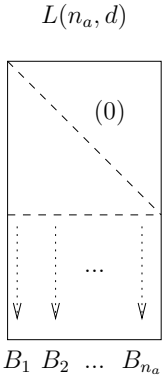

FIG. 2 – La matrice $L(n_a, d)$

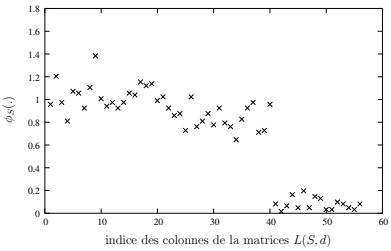

FIG. 3 – Exemples des valeurs prises par ϕ pour $n_a = kS$

où $\xrightarrow{\mathcal{P}}$ signifie la convergence en probabilité.

Si $n_a = kS$, alors nous devrions trouver des colonnes dépendantes dans $H(n_a, d)$, c'est-à-dire nous devrions avoir des colonnes de zéros dans la matrice $L(n_a, d)$. Soit $\mathcal{I}$ l'ensemble de ces colonnes. En supposant que la partie triangulaire supérieure de $H(n_a, d)$ ne contienne aucune erreur alors pour $i \in \mathcal{I}$, B_i suit une loi Binomiale de paramètres $(K - n_a, P)$ et nous avons :

$$\forall i \in \mathcal{I} \lim_{M\to\infty} \phi_{kS}(i) \xrightarrow{\mathcal{P}} = 2P$$

avec :

$$P = 1 - \sum_{l=0}^{\lfloor \frac{p_i}{2} \rfloor} \binom{p_i}{2l} P_e^{2l} (1 - P_e)^{p_i - 2l} \tag{4}$$

où p_i représente le nombre minimal de combinaisons linéaires nécessaires pour obtenir la colonne i de $L(kS, d)$ et P est la probabilité qu'un élément de la colonne i est égale à 1. De plus, si $n_a \neq kS$, $\phi_{n_a}(i)$ possède le même comportement que $\phi_{kS}(i)$ pour $i \notin \mathcal{I}$.

La figure 3 nous montre que l'écart entre les deux comportements de $\phi_S(.)$ est significatif même pour des valeurs de M finies. Cette figure a été réalisée avec une code de Hamming $(7, 4)$, un entrelaceur de taille 56, un taux d'erreur binaire du canal de propagation $p_e = 0.08$ et une durée d'observation de 10000 bits. Pour ce code particulier, p_i est égal à 4 $\forall i$.

Précisons que les erreurs se produisant dans la partie triangulaire supérieure de $H(kS, d)$ ont l'effet suivant : nous risquons d'ajouter une colonne que nous ne devrions pas ajouter et ainsi nous risquons de "perdre" une corrélation. Dans [2], nous avons proposé une procédure itérative permettant de contrer ce phénomène. En effet le Pivot de Gauss utilise exclusivement la partie triangulaire supérieure de $H(n_a, d)$ pour obtenir $L(n_a, d)$,donc si nous relançons l'algorithme sur une autre matrice $H(n_a, d)$ obtenue par une permutation de ses lignes, nous

détecterons éventuellement d'autres corrélations et les performances de l'algorithme sont ainsi améliorées.

Pour estimer la taille de l'entrelaceur, nous procédons par une recherche exhaustive sur le paramètre n_a. Pour chaque valeur de n_a, nous calculons $\phi_{n_a}(i)$. L'estimateur $\hat{S}$ de la taille de l'entrelaceur est alors :

$$\hat{S} = \text{Argmax}_{n_a} \left(\text{Card}(\{i = 1, \ldots, n_a | \phi_{n_a}(i) < \beta\})\right)$$

où β est un seuil fixé. Le choix de ce seuil conditionne les performances de notre algorithme. Dans cette article nous proposons de trouver la valeur optimale de ce seuil.

3 Détermination du seuil optimal

Le seuil est considéré optimal dans le sens où il minimise la probabilité de mauvaise détection P_{md} d'une colonne théoriquement dépendante (une colonne appartenant à l'ensemble $\mathcal{I}$). Connaissant la loi de probabilité de ϕ, nous sommes en mesure de déterminer le seuil optimal.

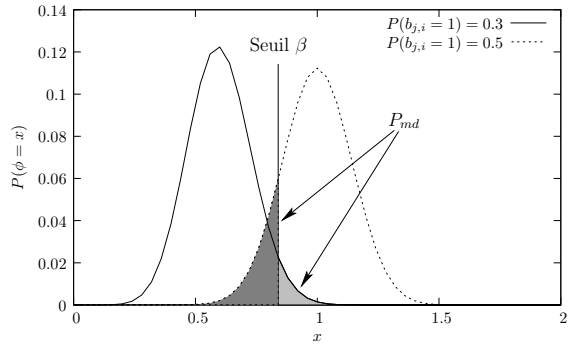

FIG. 4 – Définition de P_{md}

La figure 4 nous présente un exemple de densités de probabilités[1] de $\phi_S(i)$, celle de gauche correspond à une colonne appartenant à l'ensemble $\mathcal{I}$ et celle de droite correspond à une colonne indépendante. De plus nous avons représenté un seuil possible ainsi que la probabilité de mauvaise décision, P_{md}, associée. P_{md} correspond à la somme de 2 probabilités : la probabilité d'avoir $\phi_S(i) > \beta$ alors que la colonne i est une colonne dépendante, et la probabilité d'avoir $\phi_S(i) < \beta$ alors que la colonne i est une colonne dépendante.

Par conséquent le problème est de déterminer le seuil qui minimise P_{md}, ce seuil dépendant *a priori* de m_B, P_e et de p_j. Ainsi il est possible de définir le seuil β^* comme ci-dessous :

$$\beta^* = \arg\min_{\beta} P_{md}(m_B, P, \beta) \tag{5}$$

En utilisant les équations (3) et (4), on obtient l'expression suivante :

$$P_{md}(m_B, P, \beta) = \sum_{i=0}^{\lfloor m_B.\beta \rfloor} \binom{i}{2m_B} (0.5)^{2m_B} + \sum_{i=\lfloor m_B.\beta \rfloor + 1}^{2m_B} \binom{i}{2m_B} P^i (1-P)^{2m_B - i} \tag{6}$$

La résolution de ce problème d'optimisation permet d'obtenir la valeur théorique du seuil optimal β^*.

[1]Ces densités de probabilités sont normalement discrètes, néanmoins par un souci de clarté nous avons préféré les représenter sous forme continue

3.1 Propriétés du seuil optimal β^*

Cette section illustre l'influence des paramètres m_B et P sur β et en particulier sur le seuil optimal β^*.

Dans un premier temps, nous étudions l'influence de β sur P_{md} et cela pour différentes valeurs de P_e. La figure 5 présente P_{md} en fonction de β où l'on a fixé $m_B = 50$ et P est calculé pour $p_i = 6$ et pour différents P_e (cf. eq.(4)).

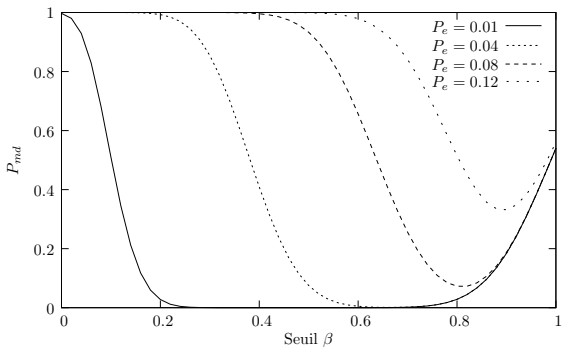

FIG. 5 – $P_{md}(m_B, P, \beta)$ en fonction de β avec $m_B = 50$, et P calculé avec $p_i = 6$

Comme le montre la figure 5, lorsque le canal est peu sévère (P_e faible), la valeur du seuil a peu d'influence sur les performances de bonne détection : en effet pour $P_e = 0.01$ si le seuil $\beta \in [0.2, 0.8]$, la valeur de P_{md} est proche de zéro. Si l'on connaît P_e et p_i, il est tout à fait possible de déterminer le seuil optimal à partir de ces courbes.

Maintenant, nous illustrons figure 6 la dépendance du seuil optimal β^* par rapport à la variable p_i. On remarque que plus p_i est grand plus le seuil optimal β^* est grand. En d'autres termes plus p_i est grand et plus difficile sera la détection.

Enfin, nous illustrons l'influence de m_B sur β^*. m_B n'a que peu d'influence sur β^* et cela peut facilement être expliqué. En effet si m_B augmente, alors la variance de la densité de probabilité des B_i diminue. Cela n'influence donc pas la position du seuil. m_B possède donc un effet important sur la valeur de P_{md} comme le montre la figure 7. Sur cette courbe sont représentées les valeurs de P_{md} lorsque que l'on choisit le seuil optimal β^*. Cette figure montre en particulier que P_{md} baisse significativement avec m_B, par exemple pour $P_e = 0.1$, $P_{md} = 0.35$ pour $m_B = 25$, et on trouve que $P_{md} = 0.06$ lorsque $m_B = 100$. Nous proposons maintenant une procédure permettant d'estimer le taux d'erreur binaire du canal (*i.e.* P_e) et ainsi d'obtenir une estimation du seuil optimal β^*.

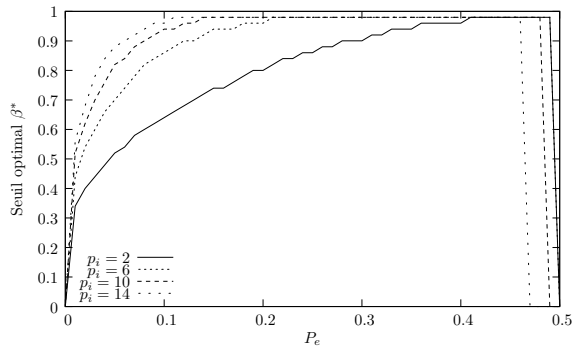

FIG. 6 – Le seuil optimal β^* en fonction de p_i

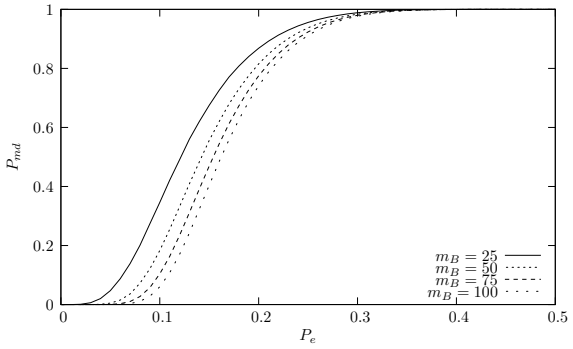

FIG. 7 – P_{md} en fonction de P_e lorsque $\beta = \beta^*$

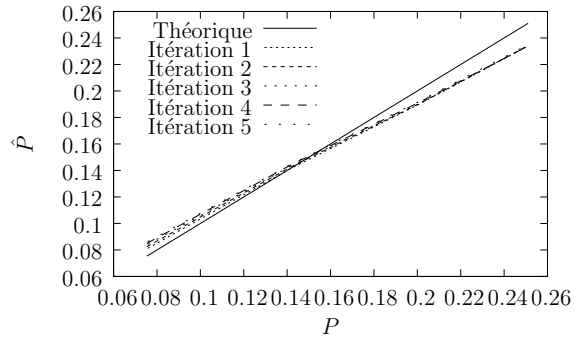

FIG. 8 – Estimation of P

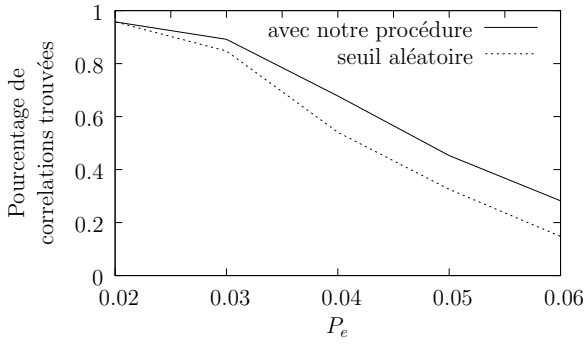

FIG. 9 – Gain de performance avec notre procédure

3.2 Choix du paramètre β

Nous avons vu que le paramètre dépend de p_i, P_e et m_B. m_B est connu du récepteur, cependant p_i, P_e sont inconnus. Donc nous sommes *a priori* incapables de fixer le seuil au début de l'algorithme à sa valeur optimale. Nous proposons cependant une procédure itérative compatible avec l'algorithme itératif initial permettant d'adapter la valeur du seuil.

Après une itération de l'algorithme initial, nous sommes éventuellement en mesure de trouver une valeur de n_a pour laquelle on détecte des colonnes dépendantes. Cette valeur correspond à la taille de l'entrelaceur. Soit $\mathcal{K}$ l'ensemble de ces colonnes détectées. Grâce à la matrice A_2, on obtient les p_i pour ces colonnes (p_i représente le nombre de colonnes impliquées pour obtenir la colonne i de $L(S, d)$). Ensuite en utilisant la valeur des $\phi_S(i)$ pour $i \in \mathcal{K}$, il est possible d'estimer P :

$$\hat{P} = \frac{1}{\text{Card}(\mathcal{K})} \sum_{i \in \mathcal{K}} \frac{\phi_{\hat{S}}(i)}{2}.$$

L'équation (4) permet alors d'estimer P_e. Enfin la figure 6 nous permet d'estimer la valeur du seuil optimal.

3.3 Validation du modèle et performances de notre nouvel algorithme.

Tout d'abord, nous validons expérimentalement la loi de probabilité suivie par $\phi(.)$ (*i.e.* équation (4)). Nous avons utilisé un entrelaceur pseudo aléatoire de taille 56 et un code correcteur de Hamming (7,4) (*i.e.* $p_i = 4$). Le nombre de bits interceptés est fixé à 10000 bits ($m_B = 122$ pour $n_a = 56$). 3000 tirages de Monte-Carlo ont été réalisés où pour chaque essai la séquence émise, les erreurs et l'entrelaceur ont été tirés aléatoirement. La figure 8 représente la moyenne de $\hat{P}$ ainsi que sa valeur théorique asymptotique. Nous constatons que les valeurs obtenues par simulation sont proches des valeurs théoriques.

Enfin, nous illustrons l'amélioration apportée par notre procédure d'adaptation du seuil par rapport à notre méthode initiale. Le contexte de simulation est le même que précédemment. Le seuil initial est tiré aléatoirement entre 0.3 et 0.7. Afin de quantifier le gain apporté par notre procédure, on suppose que la taille de l'entrelaceur a été correctement estimée (*i.e.* au moins une corrélation a été détectée). La figure 9 présente le pourcentage de corrélations trouvées après 20 itérations de notre algorithme de détection. Le nombre de corrélations trouvées avec notre procédure de seuil adaptatif est plus grand : en effet pour un TEB de 4% et après 20 itérations, nous trouvons environ 14% de corrélations de plus que notre algorithme initial. Nous obtenons les mêmes performances que notre algorithme initial mais seulement après 13 itérations soit un gain de 7 itérations. Avec notre nouvel algorithme, nous serons donc en mesure de mieux reconstruire le motif de l'entrelaceur (*i.e.* plus de correlations disponibles), de mieux trouver le début de la trame entrelacée.

4 Conclusion

Les performances de notre premier estimateur aveugle des paramètres d'un entrelaceur dépendent de notre capacité à choisir un seuil de manière appropriée. L'étude théorique menée dans cet article nous permet de définir un seuil optimal qui minimise la probabilité de mauvaise détection d'une corrélation due au code correcteur d'erreurs. Nous avons montré qu'il existe un seuil optimal mais que malheureusement ce seuil dépend de paramètres inconnus du récepteur (par exemple le TEB). Cependant, nous proposons une procédure permettant d'estimer ces paramètres et ainsi d'adapter le seuil pour qu'il converge vers sa valeur optimale. Ainsi nous pouvons espérer atteindre les meilleures performances attendues avec cet algorithme. De plus cette procédure nous permet d'estimer le taux d'erreur du canal de propagation.

Références

[1] G. Burel and R. Gautier, "Blind estimation of encoder and interleaver characteristics in a non cooperative context," *IASTED - CIIT*, November 2003.

[2] G. Sicot and S. Houcke, "Blind detection of interleaver parameters," in *Proc. of ICASSP*, Philadelphia (USA), 2005.