

# Synchronisation Trame Aveugle sur Canal Gaussien

Rodrigue IMAD, Sébastien HOUCKE, Catherine DOUILLARD

GET/ ENST-Bretagne - PRACOM  
Technopôle Brest Iroise - CS 83818 - 29238 Brest Cedex 3, France  
<prenom>.<nom>@enst-bretagne.fr

**Résumé** – Nous proposons dans cet article l’adaptation, au cas d’un canal Gaussien, d’une méthode de synchronisation trame aveugle initialement proposée pour un canal binaire symétrique. Cette nouvelle méthode se base sur le calcul des Logarithmes de Rapport de Vraisemblance (LRV) des éléments du syndrome. Une comparaison entre les performances des deux approches sera effectuée et les probabilités théoriques de fausse synchronisation seront données pour les deux méthodes.

**Abstract** – Being initially proposed for a binary Symetric channel, we adapt in this paper a blind frame synchronization method to a Gaussian channel. This new method is based on the computation of the Logarithmic Likelihood Ratio (LLR) of the syndrome’s elements. A comparison between the performances of these two methods will be done and the theoretical probabilities of false synchronization will be given for the two methods.

## 1 Introduction

Dans une chaîne de transmission numérique, une étape importante est le codage canal. À la réception, pour pouvoir décoder la séquence reçue, le décodeur doit être capable de trouver le début d’un mot de code. C’est ce que l’on appelle la synchronisation trame.

Traditionnellement, la synchronisation trame est réalisée par l’insertion dans la séquence envoyée, d’une séquence de bits connue du récepteur. Les codes actuels sont de plus en plus puissants et peuvent décoder des séquences possédant un nombre d’erreurs élevés. Pour garder leur performance de synchronisation, les méthodes traditionnelles doivent augmenter la longueur de leur séquence connue mais ceci va évidemment à l’encontre de l’efficacité spectrale de la transmission. L’enjeu des méthodes aveugles (fonctionnant sans séquence d’apprentissage) est donc important. Nous généralisons dans cet article une méthode de synchronisation aveugle développée pour un canal binaire symétrique [2] (données dures) afin qu’elle puisse fonctionner sur un canal gaussien (données souples). Cette méthode synchronise avant décodage et ne nécessite aucune séquence de synchronisation.

Section 2 de cet article présente les deux approches que peut prendre notre méthode de synchronisation. Nous calculons aussi pour chaque approche les probabilités théoriques de fausse synchronisation qui seront comparées aux résultats de nos simulations dans la 3<sup>e</sup> section. Une comparaison entre les per-

formances des deux approches est aussi effectuée et ceci en synchronisant des codes LDPC et quelques codes convolutifs. Finalement, la section 4 présente la conclusion et les perspectives de ce travail.

## 2 Méthode de synchronisation aveugle

Dans cet article nous considérons que l’émetteur envoie une séquence binaire constituée de mots de code. Les bits de cette séquence sont modulés par une MDP2 (Modulation de phase à deux états) et le canal de propagation considéré est un canal BABG (canal à Bruit Additif Blanc Gaussien). Nous verrons par la suite que la méthode de synchronisation est basée sur l’utilisation de la matrice de parité du code. Généralement, pour un code de rendement  $R = (n_c - n_r)/n_c$ , il est toujours possible d’obtenir la matrice de parité  $H$  de taille  $n_r \times n_c$ , où  $n_c$  représente la longueur d’un mot de code et  $n_r$  le nombre de relations de parité.

Soit  $X = [x(1), \dots, x(N)]$  la séquence reçue. Elle correspond à une séquence de  $N$  bits modulés, émise, bruitée et retardée de  $t_0$  bits correspondants au délai de propagation. Le but de la synchronisation trame est d’estimer  $t_0$ . Sans perte de généralité, nous supposons que  $t_0$  est toujours inférieur à  $n_c$ . Notons aussi  $s(r)$  le syndrome d’un mot de code reçu,  $r : s(r) = rH^T = eH^T$ , où  $e$  désigne l’erreur qui lui est associée.

A la position de synchronisation et en absence de bruit, il est clair que le syndrome d'un mot de code donné est nul. Par contre, ceci n'est généralement pas vrai lorsque nous ne sommes pas synchronisés. C'est sur cette idée que se base notre méthode de synchronisation.

## 2.1 Synchronisation « Dure »

La méthode initialement proposée dans [2] nécessite une séquence binaire pour fonctionner. Avant de pouvoir l'appliquer, nous devons donc prendre des décisions fermes à partir de la séquence reçue. Soit  $Z = [z(1), \dots, z(N)]$  où  $z(k) = \frac{\text{sign}(x(k))+1}{2}$ . Et soit  $W_d$  une séquence extraite de la séquence  $Z$  reçue :  $W_d = [z(d), \dots, z(d + Kn_c - 1)]^T$

Appelons  $W_d$  la fenêtre de synchronisation, c'est une fenêtre glissante de taille  $Kn_c$ , où l'indice  $d$  représente sa position sur la séquence reçue.  $W_d$  peut être divisée en  $K$  blocs de taille  $n_c$  chacun.

Calculons pour chacun de ces blocs le syndrome correspondant et formons le vecteur  $S_d$  :

$$S_d = [S_d(1), \dots, S_d(Kn_r)]^T,$$

puis évaluons  $\phi(d)$ , le nombre d'éléments non nuls dans  $S_d$  :  $\phi(d) = \sum_{k=1}^{Kn_r} S_d(k)$ .

En répétant cette procédure pour  $n_c$  valeurs successives de  $d$ , l'instant de synchronisation trame est estimé comme étant la valeur de  $d$  minimisant  $\phi$  :

$$\hat{t}_0 = \underset{d=0, \dots, n_c-1}{\text{argmin}} \phi(d).$$

## Probabilité théorique de fausse synchronisation

Soit  $P_F$  la probabilité de fausse synchronisation :

$$P_F = Pr[\phi(t_0) > \min_{d=0, \dots, n_c-1, d \neq t_0} \phi(d)]$$

Pour pouvoir calculer cette probabilité, il faut déterminer les lois que suivent  $\phi(t_0)$  et  $\phi(d)$ .

Chaque élément du syndrome suit une loi de Bernoulli de paramètre  $1/2$  dans le cas non synchronisé, et  $p$  dans le cas synchronisé.

$$\begin{aligned} p &= Pr(S_{t_0}(k) = 1) \\ &= \sum_{l=0}^{\lfloor \frac{u_k-1}{2} \rfloor} C_{u_k}^{2l+1} p_e^{2l+1} (1-p_e)^{u_k-2l-1} \end{aligned} \quad (1)$$

où  $u_k$  est le nombre d'éléments non nuls dans la  $k^{ième}$  ligne de  $H$ ,  $p_e$  la probabilité d'erreur du canal et  $C_n^k$  le nombre de combinaisons de  $k$  éléments parmi  $n$ .

En supposant que les éléments du syndrome sont indépendants et  $u_k$  constant pour toutes les lignes de  $H$ ,  $\phi$  suit une loi Binomiale :

$$\begin{aligned} \phi(d) &\propto \mathcal{B}(Kn_r, 1/2) & d \neq t_0 \\ \text{et } \phi(t_0) &\propto \mathcal{B}(Kn_r, p) \end{aligned}$$

La probabilité de fausse synchronisation est alors

$$P_F = 1 - \left[ \sum_{k=0}^{Kn_r-1} \left( C_{Kn_r}^k p^k (1-p)^{Kn_r-k} \cdot \sum_{j=k+1}^{Kn_r} \left( \frac{C_{Kn_r}^j}{2^{Kn_r}} \right) \right) \right]^{n_c-1}$$

Intéressons nous maintenant à l'adaptation de cette méthode au cas d'un canal BABG.

## 2.2 Synchronisation « Souple »

L'utilisation d'une décision ferme sur la séquence reçue entraîne inévitablement une perte d'information. Il est donc naturel d'essayer de tirer parti de cette information. La façon la plus courante pour le faire consiste à calculer le LRV des bits reçus. Dans notre contexte, nous sommes intéressés par le LRV de  $S_d(k)$ . Pour simplifier les notations, supposons que la fenêtre de synchronisation possède une taille de  $n_c$  bits.

Chaque élément  $S_d(k)$  du syndrome est la somme (modulo 2) de  $u_k$  bits :

$$S_d(k) = W_d h_k^T = \sum_{i=1}^{u_k} z(d+k_i) \quad (2)$$

où  $k_i$  est la position du  $i^{ième}$  élément non nul de la  $k^{ième}$  ligne de  $H$ . Nous obtenons facilement le LRV de chaque élément de cette somme :

$$L(z(d+k_i)) = \frac{2}{\sigma^2} x(d+k_i)$$

où  $\sigma^2$  représente la variance du bruit du canal. Une approximation du LRV d'un élément du syndrome est alors donnée par [1] :

$$L(S_d(k)) = (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(L(z(d+k_j))) \right) \beta \quad (3)$$

avec  $\beta = \min_{j=1, \dots, u_k} |L(z(d+k_j))|$ .

Le critère que nous considérons maintenant est :

$$\phi_L(d) = \sum_{k=1}^{Kn_r} L(S_d(k))$$

et la position de synchronisation est estimée par :

$$\hat{t}_0 = \underset{d=0, \dots, n_c-1}{\text{argmin}} \phi_L(d)$$

En remarquant que  $L(z(k))$  et  $x(k)$  sont proportionnels et en supposant que la variance du bruit est constante pour une transmission, nous pouvons remplacer  $L(z(d+k_j))$  par  $x(d+k_j)$  dans (3) et nous n'avons donc pas besoin de connaître la variance du bruit.  $\phi_L(d)$  est alors estimée par :

$$\hat{\phi}_L(d) = \sum_{k=1}^{Kn_r} \hat{L}(S_d(k)) = \sum_{k=1}^{Kn_r} (2S_d(k) - 1) Y_d(k)$$

où  $Y_d(k) = \min_{j=1, \dots, u_k} |x(d+k_j)|$ .

### Probabilité théorique de fausse synchronisation

Cette fois, nous ne sommes plus en présence de variables discrètes, mais continues. En appliquant le théorème de la limite centrale,  $\hat{\phi}_L(d)$  suit une loi Gaussienne :  $\hat{\phi}_L(d) \propto \mathcal{N}(Kn_r m_d, Kn_r \sigma_d^2)$ , où  $m_d$  et  $\sigma_d^2$  représentent respectivement la moyenne et la variance de  $\hat{L}(S_d(k))$ .

Pour  $d \neq t_0$ ,  $S_d(k)$  et  $Y_d(k)$  sont indépendants et  $E[S_d(k)] = 1/2$ . Donc,  $m_d = 0$  et  $\sigma_d^2 = E[Y_d(k)^2]$  que nous estimons par simulations.

Par contre, quand  $d = t_0$ ,  $S_{t_0}(k)$  et  $Y_{t_0}(k)$  ne sont plus indépendants.  $m_{t_0}$  se met alors sous la forme :

$$m_{t_0} = Pr(S_{t_0}(k) = 0)E[\hat{L}(S_{t_0}(k))|_{S_{t_0}(k)=0}] + Pr(S_{t_0}(k) = 1)E[\hat{L}(S_{t_0}(k))|_{S_{t_0}(k)=1}]$$

où  $Pr(S_{t_0}(k) = 1) = p$ , calculée dans (1).

Calculons maintenant

$$E[\hat{L}(S_{t_0}(k))|_{S_{t_0}(k)=1}] = E[Y_{t_0}(k)|_{S_{t_0}(k)=1}].$$

Rappelons que nous sommes en présence d'une modulation MDP2. Donc, un bit reçu  $x(i)$  se met sous la forme suivante :  $x(i) = b(i) + w(i)$ , où  $b(i) = \pm 1$  correspond à un bit modulé et  $w(i)$  le bruit.

$S_{t_0}(k) = 1$  signifie qu'il y a un nombre impair de bits erronés parmi les  $u_k$  bits  $z(t_0 + k_j)$  intervenant dans (2). Et comme  $z(k) = \frac{\text{sign}(b(k) + w(k)) + 1}{2}$ , cela veut dire que parmi les  $u_k$  éléments, il y a un nombre impair d'éléments  $w(k)$  qui, additionnés à  $b(k)$ , changent le signe de  $b(k)$ .

Remarquons qu'une erreur se produit quand :

$$\begin{aligned} & b(k) = +1 \quad \text{et} \quad w(k) < -1 \\ \text{ou} \quad & b(k) = -1 \quad \text{et} \quad w(k) > 1. \end{aligned}$$

Et comme

$$Pr(b(k) = 1) = Pr(b(k) = -1) = 1/2$$

et  $w(k)$  un bruit gaussien centré, nous pouvons dire

$$E[Y_{t_0}(k)|_{S_{t_0}(k)=1}] = E\left[\min_{j=1, \dots, u_k} |1 + w(t_0 + k_j)|_{B_1}\right],$$

où  $B_1$  est l'événement : {Parmi  $u_k$  échantillons de bruit, un nombre impair de ces éléments a sa valeur inférieure à  $-1$ }.

Le même raisonnement peut être fait dans le cas où  $S_{t_0}(k) = 0$  pour aboutir finalement à :

$$m_{t_0} = -(1-p)E\left[\min_{j=1, \dots, u_k} |1 + w(t_0 + k_j)|_{B_2}\right] + pE\left[\min_{j=1, \dots, u_k} |1 + w(t_0 + k_j)|_{B_1}\right],$$

où  $B_2$  est l'événement : {Parmi  $u_k$  échantillons de bruit, zéro ou un nombre pair de ces éléments a sa valeur inférieure à  $-1$ }.

Pour la variance, on obtient facilement :

$$\sigma_{t_0}^2 = E\left[\left(\min_{j=1, \dots, u_k} |x(t_0 + k_j)|\right)^2\right] - m_{t_0}^2$$

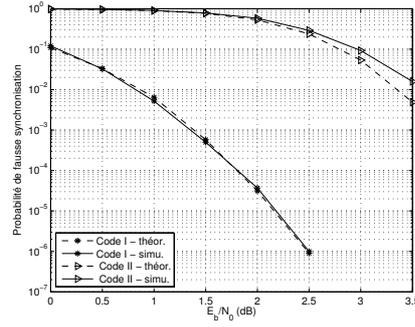


FIG. 1 – Synchronisation Dure de codes LDPC

Ce qui nous donne en supposant l'indépendance des éléments du syndrome

$$\hat{\phi}_L(t_0) \propto \mathcal{N}(Kn_r m_{t_0}, Kn_r \sigma_{t_0}^2).$$

La probabilité de fausse synchronisation est alors :

$$P_F = 1 - \left[ 1 - \frac{1}{2} \operatorname{erfc} \left( \frac{-Kn_r m_{t_0}}{\sqrt{2Kn_r(\sigma_{t_0}^2 + \sigma_d^2)}} \right) \right]^{n_c - 1}.$$

### 3 Simulations

Afin d'estimer les performances de nos algorithmes de synchronisation, nous avons estimé la probabilité de fausse synchronisation. L'évaluation de cette probabilité est réalisée par simulation de Monte Carlo : jusqu'à 1000000 réalisations ont été effectuées, où à chaque réalisation, le bruit, les bits d'information et le retard du canal ont été choisis aléatoirement. Considérons deux codes LDPC de taille 511 bits, de rendement 0.7, qui diffèrent seulement par la composition de leur matrice de parité : Code I (respectivement II) a 4 (respectivement 10) éléments non nuls sur chaque ligne de sa matrice de parité. En prenant une fenêtre de synchronisation de taille 1 bloc (511 bits), nous avons appliqué les deux méthodes de synchronisation sur ces codes. La figure 1 représente la probabilité de fausse synchronisation en fonction du Rapport Signal sur Bruit ( $E_b/N_0$ ), pour la méthode Dure. Il est clair sur cette figure que les résultats théoriques et simulés sont très proches. Pour code II, les résultats théoriques sont un peu meilleurs que ceux simulés et ceci est dû au fait que pour ce code, chaque ligne de sa matrice de parité contient 10 éléments non nuls, ce qui diminue le nombre d'éléments de syndrome indépendants. Par contre, dans nos calculs théoriques nous avons supposés l'indépendance de tous les éléments du syndrome, ce qui explique cette différence. En comparant les courbes tracées pour les codes I et II, nous remarquons que les performances de notre méthode de synchronisation se dégradent lorsque le

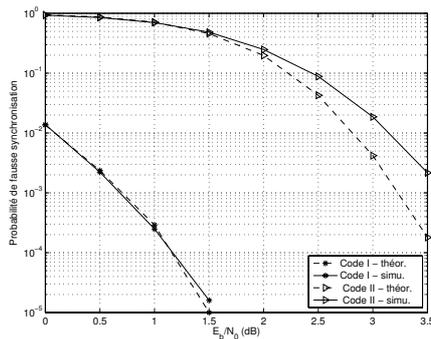


FIG. 2 – Synchronisation Souple de codes LDPC

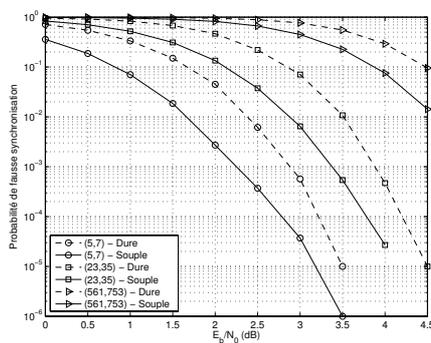


FIG. 3 – Synchronisations Dure et Souple de quelques codes convolutifs

nombre d'éléments non nuls dans  $H$  augmente.

La figure 2 montre les performances théoriques et simulées de la synchronisation souple, pour les deux codes LDPC déjà cités. Le même comportement que celui de la synchronisation dure est obtenu ici.

En comparant maintenant les figures 1 et 2, il est clair que la synchronisation souple présente de meilleures performances : Prenons par exemple code I, pour une probabilité de fausse synchronisation égale à  $10^{-3}$ , un écart de  $0.62 \text{ dB}$  se trouve entre les deux méthodes.

Traisons maintenant le cas des codes convolutifs. Pour ce type de codes, les éléments non nuls de la matrice de parité sont répartis en forme d'« escalier ». Ceci cause une dégradation des performances de la méthode de synchronisation. Une solution à ce problème est d'entrelacer les bits dans chaque mot de code en utilisant un entrelaceur pseudo-aléatoire de taille  $n_c$  bits.

Pour les simulations effectuées, nous avons synchronisé des codes convolutifs de polynômes générateurs (5, 7), (23, 35) et (561, 753), ayant tous une taille de 512 bits et des longueurs de contrainte égales respectivement à 3, 5 et 9. La figure 3 montre une comparaison entre les deux méthodes de synchro-

nisation pour les trois types de codes convolutifs déjà cités. Le même comportement que celui obtenu avec les codes LDPC est obtenu ici : La synchronisation souple est toujours meilleure.

Prenons le code (5,7), nous observons un écart de  $0.7 \text{ dB}$  entre les deux méthodes, pour une probabilité de fausse synchronisation égale à  $2 \cdot 10^{-2}$ . Remarquons que plus la longueur de contrainte d'un code augmente, plus la matrice de parité correspondante est "pleine" et plus les performances de la méthode de synchronisation se dégradent, conformément aux observations de [2]. Notons que si ces trois codes convolutifs ont été décodés par un décodeur type MAP (Maximum A Posteriori), un taux d'erreur trame d'environ  $10^{-2}$  correspondrait à des  $E_b/N_0$  compris entre 5 et 6  $\text{dB}$ . Cependant la figure 3 montre qu'à partir de 4  $\text{dB}$ , la synchronisation des codes (5, 7) et (23, 35) n'a donné aucune erreur sur les 1000000 réalisations testées.

## 4 Conclusion

Nous avons introduit dans cet article une nouvelle approche de synchronisation trame aveugle basée sur la somme des LRV des éléments d'un syndrome. Comparée avec celle de la synchronisation dure, cette approche souple donne de meilleurs résultats. Nous avons aussi estimé les probabilités théoriques de fausse synchronisation et les valeurs trouvées sont quasiment les mêmes que celles obtenues par la procédure de synchronisation. Suite à ce travail et à la vue de ces résultats très encourageants, notre but sera d'appliquer cette méthode de synchronisation aveugle à d'autres types de codes, en particulier les turbocodes.

## Références

- [1] J. Hagenauer. Iterative decoding of binary block and convolutional codes. *IEEE trans. on Information Theory*, vol. 42, no. 2, pp. 429-445, March 1996.
- [2] S. Houcke and G. Sicot. Blind Frame Synchronization for Block Code. In *Proceedings of EUSIPCO, European Sig. Proc. Conf*, Florence, Italy, September 2006.
- [3] S. Houcke, G. Sicot and M. Debbah. Blind Detection for block interleaved division multiple access. In *Proceedings of GLOBECOM*, San Francisco, USA, 2006. .
- [4] R. Johannesson, K. Sh. Zigangirov. Fundamentals of Convolutional Coding, IEEE series on Digital and Mobile Communication, IEEE Press, 1999.