



**HAL**  
open science

# Techniques de synchronisation aveugles pour les systèmes codés

Rodrigue Imad

► **To cite this version:**

Rodrigue Imad. Techniques de synchronisation aveugles pour les systèmes codés. Traitement du signal et de l'image [eess.SP]. Ecole Nationale Supérieure des Télécommunications de Bretagne - ENSTB, 2009. Français. NNT: . tel-00428895v2

**HAL Id: tel-00428895**

**<https://theses.hal.science/tel-00428895v2>**

Submitted on 2 Dec 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° d'ordre : 2009telb0108

# THÈSE

présentée à

**TÉLÉCOM BRETAGNE**

**EN HABILITATION CONJOINTE AVEC L'UNIVERSITÉ  
DE RENNES 1**

pour obtenir

le grade de : **DOCTEUR DE TÉLÉCOM BRETAGNE**

Mention : *Traitement du Signal et Télécommunications*

par

**Rodrigue Imad**

---

**Techniques de synchronisation aveugles pour les  
systèmes codés**

---

Soutenue le 25 Septembre 2009 devant la commission d'Examen :

Composition du Jury :

*Président :* Jean-François Héliard - INSA Rennes  
*Rapporteurs :* Marie-Laure Boucheret - ENSEEIHT Toulouse  
Philippe Ciblat - Télécom ParisTech  
*Examineurs :* Sébastien Houcke - Télécom Bretagne  
Ramesh Pyndiah - Télécom Bretagne  
*Invité :* Mounir Ghogho - Université de Leeds



*Say not 'I have found the truth', but rather, 'I have found a truth'.*  
Gibran Khalil Gibran, *The Prophet*



# Remerciements

Ce manuscrit est l'aboutissement de trois années de recherche effectuées au sein du département Signal et Communications de Télécom Bretagne. Je tiens ici à remercier toutes les personnes qui, par leur présence, leurs conseils et leur amitié, ont fait de ces années de thèse des moments inoubliables.

Je tiens tout d'abord à remercier Monsieur Jean-François Héliard, Professeur à l'INSA de Rennes, pour m'avoir fait l'honneur de présider mon jury de thèse.

Je remercie également Madame Marie-Laure Boucheret, Professeur à l'ENSEEIH Toulouse et Monsieur Philippe Ciblat, Maître de conférences à Télécom ParisTech, pour l'attention qu'ils ont accordée à la lecture de ce manuscrit ainsi que pour leur participation au jury de thèse.

J'adresse mes remerciements à Monsieur Mounir Ghogho, Professeur à l'Université de Leeds, pour avoir accepté de faire partie du jury de thèse ainsi qu'à l'accueil chaleureux que j'ai eu pendant mon séjour d'études à l'Université de Leeds. Les compétences scientifiques de Mounir et les discussions que nous avons eu ensemble m'ont été très précieuses.

Je souhaite également remercier Monsieur Ramesh Pyndiah, Chef du département Signal et Communications de Télécom Bretagne pour m'avoir donné la possibilité d'entreprendre mes travaux de doctorat au sein du département et pour avoir accepté de diriger ma thèse.

Toute ma gratitude et mes remerciements vont à mon encadrant Monsieur Sébastien Houcke, Maître de conférences à Télécom Bretagne. Travailler avec lui a été à la fois très enrichissant scientifiquement et facile humainement grâce à sa bonne humeur et son enthousiasme permanent. Seb, merci pour tes conseils, ta disponibilité, ta motivation, ton amitié et à tous les bons moments que nous avons passé ensemble.

Ces trois années de thèse n'auraient pas été particulièrement réussies sans l'ensemble des permanents, des doctorants et des stagiaires que j'ai eu l'occasion de rencontrer à Télécom Bretagne et qui ont contribué à créer une ambiance de travail très agréable. C'est avec beaucoup de regrets que je les quitte.

Je tiens aussi à remercier Madame Catherine Blondé pour le temps qu'elle a mis à la correction du français de ce manuscrit.

Un énorme merci à tous mes amis qui m'ont rendu la vie à côté de la thèse agréable. Ils ne peuvent pas être tous cités ici mais ils se reconnaissent.

Et pour finir, je tiens à exprimer ma très profonde gratitude et mon plus grand merci envers mes parents Josette et Imad, ma soeur Rolande et mon frère Rodolphe, pour l'amour, le soutien, et les encouragements qu'ils m'ont apportés pendant toutes ces années...

# Table des matières

Remerciements	3
Acronymes	13
Notations générales	15
Notations spécifiques	17
Introduction générale	21
<b>1 Synchronisation trame aveugle sur canal Gaussien</b>	<b>25</b>
1.1 Etat de l'art . . . . .	26
1.1.1 Méthode de synchronisation du HSS . . . . .	27
1.1.2 Méthode de synchronisation de Sun et Valenti . . . . .	29
1.2 Méthode proposée de synchronisation aveugle basée sur un critère de Maximum <i>a Posteriori</i> - Algorithme du SSS . . . . .	31
1.2.1 Comparaison avec la méthode de synchronisation de Sun et Valenti	35
1.3 Etude théorique des critères de synchronisation et équations analytiques des probabilités de fausse synchronisation . . . . .	36
1.3.1 Algorithme du HSS . . . . .	37
1.3.2 Algorithme du SSS . . . . .	38
1.3.3 Validation de l'étude théorique des algorithmes HSS et SSS . . . .	41
1.4 Propriétés de l'algorithme du SSS . . . . .	42
1.5 Conclusion . . . . .	45
<b>2 Performances et comportement des méthodes de synchronisation trame appliquées à différentes familles de codes correcteurs d'erreurs</b>	<b>47</b>
2.1 Codes LDPC . . . . .	48
2.1.1 Définition d'un code LDPC . . . . .	48
2.1.2 Synchronisation des codes LDPC . . . . .	50

2.2	Codes convolutifs . . . . .	53
2.2.1	Définition d'un code convolutif . . . . .	53
2.2.2	Synchronisation des codes convolutifs . . . . .	54
2.3	Codes BCH et codes produits . . . . .	59
2.3.1	Définitions générales . . . . .	59
2.3.2	Synchronisation trame par adaptation de la matrice de contrôle de parité - Algorithme du m-SSS . . . . .	60
2.3.3	Résultats des simulations . . . . .	63
2.4	Synchronisation trame des systèmes utilisant des modulations à grand nombre d'états . . . . .	67
2.5	Calcul du taux d'erreur trame après application de la procédure de synchronisation . . . . .	68
2.6	Conclusion . . . . .	70
<b>3</b>	<b>Estimation aveugle du résidu de porteuse</b>	<b>73</b>
3.1	Etat de l'art . . . . .	74
3.2	Estimation aveugle du résidu de porteuse en présence d'un déphasage connu par le récepteur . . . . .	75
3.2.1	Méthode proposée d'estimation aveugle du résidu de porteuse - Algorithme du C-SFM . . . . .	75
3.2.2	Méthode d'estimation du résidu de porteuse à complexité réduite - Algorithme du C-ISFM . . . . .	80
3.3	Estimation aveugle du résidu de porteuse en présence d'un déphasage inconnu par le récepteur . . . . .	82
3.3.1	Résultats des simulations . . . . .	84
3.4	Conclusion . . . . .	85
<b>4</b>	<b>Synchronisation aveugle de la phase</b>	<b>87</b>
4.1	Etat de l'art . . . . .	88
4.2	Estimation du déphasage par une expression analytique - Algorithme du SFR . . . . .	89
4.2.1	Résultats des simulations . . . . .	92
4.3	Estimation du déphasage par minimisation d'une fonction de coût - Algorithme du P-SFM . . . . .	93
4.3.1	Minimisation de la fonction de coût par la méthode de descente du Gradient . . . . .	95
4.3.2	Résultats des simulations . . . . .	97
4.4	Performances de l'algorithme du P-SFM après estimation du résidu de porteuse par le C-SFM . . . . .	99

4.5	Algorithme du P-SFM dans le cas de systèmes utilisant des modulations à grand nombre d'états . . . . .	102
4.5.1	Résultats des simulations . . . . .	104
4.6	Conclusion . . . . .	104
<b>Conclusion et perspectives</b>		<b>107</b>
<b>A Annexe du Chapitre 1</b>		<b>111</b>
A.1	Calcul de l'équation analytique de la probabilité de fausse synchronisation de la technique du HSS . . . . .	111
A.2	Calcul des paramètres $m_{t_0}$ et $\sigma_{t_0}^2$ de $\hat{\phi}(t_0)$ . . . . .	112
<b>B Annexe du Chapitre 3</b>		<b>115</b>
B.1	Calcul de l'expression analytique de la variance $\sigma_{class}^2$ . . . . .	115
<b>C Annexe du Chapitre 4</b>		<b>117</b>
C.1	Calcul de la dérivée partielle de $J_e(f_0, \tilde{\theta})$ . . . . .	117
<b>Publications</b>		<b>119</b>
<b>Bibliographie</b>		<b>121</b>
<b>Index</b>		<b>127</b>



# Liste des figures

1.1	Principe de la méthode de synchronisation trame aveugle . . . . .	28
1.2	Structure du tampon considéré par Sun et Valenti pour effectuer la synchronisation trame. . . . .	30
1.3	Validation des approximations effectuées dans le critère de synchronisation du SSS. . . . .	34
1.4	Distributions de probabilité de $\hat{\phi}$ aux positions non synchronisées et synchronisée, pour un code LDPC de taille 512 bits et à $E_b/N_0 = 0$ dB. . . . .	35
1.5	Vérification que les $\hat{\phi}(t)$ simulées suivent des distributions Gaussiennes . . . . .	39
1.6	Validation de l'étude théorique de l'approche HSS. . . . .	41
1.7	Validation de l'étude théorique de l'approche SSS. . . . .	42
1.8	$Pr[S_{t_0}(k) = 1]$ en fonction de $u_k$ pour différentes valeurs de BER. . . . .	43
1.9	Probabilité que deux éléments d'un syndrome soient indépendants . . . . .	43
1.10	Le critère de synchronisation $\hat{\phi}(t)$ tracé en fonction de la position $t$ de la fenêtre de synchronisation. . . . .	44
2.1	Performances de la méthode du SSS appliquée à trois codes LDPC différents ayant la même longueur, le même rendement et le même nombre d'éléments non nuls dans chaque ligne de leur matrice de contrôle de parité. . . . .	49
2.2	Comparaison entre les performances des deux méthodes HSS et SSS. . . . .	50
2.3	Comparaison entre la méthode de synchronisation du SSS et celle de Sun et Valenti. . . . .	51
2.4	Synchronisation trame du code LDPC utilisé dans le standard DVB-S2. . . . .	52
2.5	Les méthodes HSS et SSS appliquées aux codes convolutifs . . . . .	56
2.6	Méthodes de synchronisation du SSS et de Sun et Valenti appliquées aux codes convolutifs. . . . .	57
2.7	Comparaison entre la méthode de synchronisation de Massey et l'algorithme du SSS appliqués au code convolutif (1, 5/7). . . . .	58
2.8	Courbes de FER des codes convolutifs de rendement $R = 1/2$ et taille $n_c = 512$ bits. . . . .	59

2.9	Un Code Produit construit à partir de deux codes en blocs linéaires et systématiques. . . . .	61
2.10	Méthodes de synchronisation aveugles appliquées au code BCH étendu (32, 26). . . . .	64
2.11	Méthodes de synchronisation aveugles appliquées au code produit BCH (32, 26) <sup>2</sup> . . . . .	65
2.12	Méthode du m-SSS appliquée aux codes produits BCH (16, 11) <sup>2</sup> et (32, 26) <sup>2</sup> et comparée à la méthode de synchronisation de Massey. . . .	65
2.13	Courbes de FER du décodeur m-ABP appliqué au code produit BCH (16, 11) <sup>2</sup> . . . . .	66
2.14	Courbes de FER du décodeur m-ABP appliqué au code produit BCH (32, 26) <sup>2</sup> . . . . .	67
2.15	Synchronisation trame pour des modulations à grand nombre d'états. .	69
2.16	Courbes de FER simulé et calculé obtenu après synchronisation et décodage des codes produits. . . . .	70
3.1	$L_R(\tilde{f}, \theta_0)$ en fonction de l'erreur d'estimation du résidu de porteuse, tracée pour un déphasage connu par le récepteur. . . . .	78
3.2	Zoom sur la fonction $L_R(\tilde{f}, \theta_0)$ , tracée pour un déphasage connu par le récepteur. . . . .	79
3.3	Performances de l'algorithme du C-SFM appliqué pour l'estimation aveugle du résidu de porteuse en présence d'un déphasage connu par le récepteur. . . . .	80
3.4	Performances de l'algorithme du C-ISFM appliqué pour l'estimation aveugle du résidu de porteuse en présence d'un déphasage connu par le récepteur. . . . .	82
3.5	La fonction $L_R(\tilde{f}, 0)$ tracée en présence d'un déphasage $\theta_0 = \pi/2$ . . . .	83
3.6	La fonction $L_I(\tilde{f}, 0)$ tracée en présence d'un déphasage $\theta_0 = \pi/2$ . . . .	84
3.7	Performances de l'algorithme du C-SFM appliqué pour l'estimation aveugle du résidu de porteuse en présence d'un déphasage inconnu par le récepteur. . . . .	85
3.8	Performances de l'algorithme du C-ISFM appliqué pour l'estimation aveugle du résidu de porteuse en présence d'un déphasage inconnu par le récepteur. . . . .	86
4.1	Estimation aveugle du déphasage par l'algorithme du SFR. Le résidu de porteuse du système est supposé connu par le récepteur. . . . .	93
4.2	Les fonctions $L_R(f_0, \tilde{\theta})$ et $L_I(f_0, \tilde{\theta})$ en fonction de l'erreur d'estimation du déphasage $(\tilde{\theta} - \theta_0)$ . . . . .	94
4.3	La fonction de coût $J(f_0, \tilde{\theta})$ en fonction de l'erreur d'estimation du déphasage $(\tilde{\theta} - \theta_0)$ . . . . .	95

---

4.4	MSE de l'estimation du déphasage en présence d'un résidu de porteuse connu par le récepteur. . . . .	98
4.5	Courbes de BER obtenues après estimation du déphasage puis application du décodeur BP. Les courbes sont tracées en présence d'un résidu de porteuse connu par le récepteur. . . . .	98
4.6	La fonction de coût $J(0, \tilde{\theta})$ en présence d'un résidu de porteuse inconnu.	99
4.7	La fonction de coût $J(\hat{f}_{CSFM}, \tilde{\theta})$ après compensation par une estimée du CFO et un déphasage quelconque $\tilde{\theta}$ . . . . .	100
4.8	MSE de l'estimation de phase obtenu après estimation du résidu de porteuse. . . . .	101
4.9	Courbes de BER obtenues après estimation du résidu de porteuse et du déphasage puis application du décodeur BP. . . . .	101
4.10	La fonction de coût $J'(f_0, \tilde{\theta})$ en fonction de l'erreur d'estimation du déphasage $(\tilde{\theta} - \theta_0)$ . . . . .	103
4.11	MSE de l'estimation du déphasage dans le cas d'une modulation 16-QAM.	103



# Acronymes

16-QAM	16-state Quadrature Amplitude Modulation
ABP	Adaptive Belief Propagation
BABG	Bruit Additif Blanc Gaussien
BCH	Bose-Chaudhuri-Hocquenghem
BCJR	Bahl-Cocke-Jelinek-Raviv
BER	Bit Error Rate
BP	Belief Propagation
BPSK	Binary Phase Shift Keying
CFO	Carrier Frequency Offset
C-ISFM	CFO estimation-Initialized Syndrome Function Minimization
C-SFM	CFO estimation-Syndrome Function Minimization
DA	Data Aided
DD	Decision Directed
DVB-S2	Digital Video Broadcasting-Satellite Second Generation
EQM	Erreur Quadratique Moyenne
FER	Frame Error Rate
HDD	Hard Decision Directed
HSS	Hard Syndrome based Synchronization
LDPC	Low Density Parity Check
LLR	Log-Likelihood Ratio
m-ABP	Modified Adaptive Belief Propagation
m-SSS	Modified Soft Syndrome based Synchronization
MAP	Maximum A Posteriori
MPSK	M-array Phase Shift Keying
MSE	Mean Squared Error
MV	Maximum de Vraisemblance

NDA	Non Data Aided
P-SFM	Phase estimation-Syndrome Function Minimization
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RS	Reed-Solomon
s.e	Statistiquement équivalent
SFR	Syndrome Functions Ratio
SNR	Signal-to-Noise Ratio
SSS	Soft Syndrome based Synchronization

# Notations générales

$(*)$	Valeur conjuguée d'un nombre complexe
$\binom{n}{k}$	Nombre de combinaisons de $k$ éléments parmi $n$
$[x]$	Partie entière de $x$
$\arg$	Argument d'un nombre complexe
$\operatorname{argmin}$	Valeur de la variable pour laquelle la fonction concernée atteint son minimum
$\operatorname{argmax}$	Valeur de la variable pour laquelle la fonction concernée atteint son maximum
$\operatorname{atanh}$	Arc tangente hyperbolique
$\mathcal{B}(n, p)$	Loi Binomiale de paramètres $(n, p)$
$\cosh$	Cosinus hyperbolique
$\left. \frac{\partial h(x)}{\partial x} \right $	Dérivée partielle d'une fonction $h$ par rapport à $x$
$E[-]$	Espérance mathématique
$H^T$	Matrice transposée de $H$
$\Im$	Partie imaginaire d'un nombre complexe
$\log$	Logarithme népérien
$\mathcal{N}(m, \sigma^2)$	Loi Normale de moyenne $m$ et variance $\sigma^2$
$Pr[-]$	Probabilité
$Q(\alpha)$	Fonction $Q$
$\Re$	Partie réelle d'un nombre complexe
$\operatorname{sign}$	Signe d'une variable réelle
$\tanh$	Tangente hyperbolique



# Notations spécifiques

$b(k)$	Symbole codé et modulé
$C(n_c, n_b)$	Code correcteur d'erreurs de longueur $n_c$ et de dimension $n_b$
$\hat{d}(k)$	Décision ferme d'une estimée d'un symbole transmis
$E_b$	Energie binaire
$E_s$	Energie par symbole
$\frac{E_b}{N_0}$	Rapport entre l'énergie binaire et la densité spectrale de puissance de bruit
$\epsilon^{(m)}$	Valeur du pas de la méthode de descente du Gradient à l'itération $m$
$f_0$	Résidu de porteuse
$\tilde{f}$	Fréquence quelconque utilisée pour compenser le résidu de porteuse présent dans les échantillons reçus
$\hat{f}_{CSFM}$	Estimée du résidu de porteuse par l'algorithme du C-SFM
$\hat{f}_{class}$	Estimée du résidu de porteuse par la méthode classique
$FER_1$	FER calculé après décodage, dans le cas d'une synchronisation trame parfaite
$FER_2$	FER calculé après décodage et synchronisation trame
$\hat{\Gamma}(a(k))$	Estimée du LLR d'un bit associé à un symbole modulé par une modulation à grand nombre d'états
$H$	Matrice de contrôle de parité d'un code correcteur d'erreurs
$\hat{I}(\tilde{f}, \theta)$	Fonction de coût minimisée par le C-SFM pour l'estimation du résidu de porteuse
$\hat{J}_e(f, \tilde{\theta})$	Fonction de coût minimisée par le P-SFM pour l'estimation du déphasage
$K$	Nombre de blocs contenus dans la fenêtre de synchronisation
$\Lambda(t)$	Critère de synchronisation de Sun et Valenti calculé à une position $t$ de la fenêtre de synchronisation
$L(S_t(k))$	LLR d'un élément de syndrome calculé à la position $t$ de la fenêtre de synchronisation
$\hat{L}(S_t(k))$	Valeur proportionnelle à $L(S_t(k))$
$L_R(f, \theta)$	Fonction du LLR du syndrome calculée à partir des parties réelles des échantillons $r_{(f,\theta)}(k)$
$\hat{L}_R(f, \theta)$	Valeur estimée de $L_R(f, \theta)$
$L_I(f, \theta)$	Fonction du LLR du syndrome calculée à partir des parties imaginaires des échantillons $r_{(f,\theta)}(k)$
$\hat{L}_I(f, \theta)$	Valeur estimée de $L_I(f, \theta)$

$m$	Numéro de l'itération en cours d'un algorithme d'optimisation (Gradient ou Recuit Simulé)
$m_c$	Moyenne du LLR d'un élément de syndrome introduite dans le critère de synchronisation de Sun et Valenti
$m_t$	Moyenne de $\hat{L}(S_t(k))$
$N_0$	Densité spectrale de puissance monolatérale du bruit BABG
$n_b$	Dimension d'un code
$n_c$	Longueur d'un code
$n_r$	Nombre de relations de parité
$\nu$	Longueur de contrainte d'un code convolutif
$p_e$	Probabilité d'erreur du canal de transmission
$P_F$	Probabilité de fausse synchronisation trame
$P_{FHSS}$	Expression analytique de la probabilité de fausse synchronisation de l'algorithme HSS
$P_{FSSS}$	Expression analytique de la probabilité de fausse synchronisation de l'algorithme SSS
$P_{ind}$	Probabilité que deux éléments de syndrome soient indépendants
$p_k$	Probabilité que le $k^{i\text{ème}}$ élément du syndrome soit non nul
$P_t$	Probabilité que tous les éléments d'un syndrome soient nuls
$\psi(t)$	LLR du syndrome calculé à la position $t$
$\hat{\phi}(t)$	Variable proportionnelle au LLR du syndrome calculée à la position $t$
$\varphi(t)$	Somme des éléments du syndrome calculée à la position $t$
$q$	Nombre de bits associés à un symbole modulé
$\mathcal{Q}$	Ensemble des symboles d'une modulation donnée
$r(k)$	Echantillon reçu
$r_f(k)$	Echantillon obtenu après compensation du résidu de porteuse présent dans $r(k)$ par une fréquence $f$
$r_\theta(k)$	Echantillon obtenu après compensation du déphasage présent dans $r(k)$ par une phase $\theta$
$r_{(f,\theta)}(k)$	Echantillon obtenu après compensation respective du résidu de porteuse et du déphasage présents dans $r(k)$ par une fréquence $f$ et une phase $\theta$
$\mathbf{r}$	Séquence de $N$ échantillons reçus
$\tilde{r}(k)$	LLR d'un échantillon reçu
$R$	Rendement d'un code correcteur d'erreurs
$S_t(k)$	Valeur binaire d'un élément de syndrome calculé à la position $t$ de la fenêtre de synchronisation
$s(k)$	Symbole pilote utilisé dans les méthodes de synchronisation non aveugles
$\sigma^2$	Variance totale du bruit Gaussien
$\sigma_e^2$	Variance de chaque partie du bruit Gaussien complexe
$\sigma_{class}^2$	Expression analytique de la variance d'estimation du CFO par la méthode classique
$\sigma_t^2$	Variance de $\hat{L}(S_t(k))$
$t_0$	Instant de synchronisation trame
$t$	Position possible de l'instant de synchronisation trame
$\hat{t}_{HSS}$	Estimée de l'instant de synchronisation trame par l'algorithme du HSS

$\hat{t}_{mSSS}$	Estimée de l'instant de synchronisation trame par l'algorithme du m-SSS
$\hat{t}_{SSS}$	Estimée de l'instant de synchronisation trame par l'algorithme du SSS
$\hat{t}_{SV}$	Estimée de l'instant de synchronisation trame par la méthode de Sun et Valenti
$\theta_0$	Déphasage du canal
$\tilde{\theta}$	Phase quelconque utilisée pour compenser le déphasage présent dans les échantillons reçus
$\hat{\theta}_{HDD}$	Estimée du déphasage du canal par l'algorithme du HDD
$\hat{\theta}_P$	Estimée du déphasage du canal par le synchroniseur à puissance $P$
$\hat{\theta}_{PSFM}$	Estimée du déphasage du canal par l'algorithme du P-SFM
$\hat{\theta}_{SFR}$	Estimée du déphasage du canal par l'algorithme du SFR
$T_s$	Temps symbole
$T_0$	Température initiale nécessaire pour l'algorithme du Recuit Simulé
$u_k$	Nombre d'éléments non nuls dans la $k^{\text{ième}}$ ligne de la matrice de contrôle de parité $H$
$w(k)$	Bruit additif blanc Gaussien
$\mathbf{W}_t$	Fenêtre de synchronisation positionnée à une position $t$
$X_t(k)$	Valeur modulée de l'élément de syndrome $S_t(k)$
$Y_t(k)$	Fiabilité de l'élément de syndrome $S_t(k)$
$z(k)$	Echantillon reçu à valeur binaire



# Introduction générale

La synchronisation est une fonction critique dans les systèmes de communications numériques. Une synchronisation échouée peut avoir des effets catastrophiques sur les performances d'un système de transmission. Ainsi, le bon fonctionnement des dispositifs de synchronisation est essentiel.

Un système de transmission numérique utilisant une modulation linéaire mono-porteuse transmet des symboles à un rythme  $1/T_s$  sur une onde porteuse de fréquence  $\nu_0$ . Le récepteur doit effectuer une série de traitements afin d'être en mesure de récupérer l'information numérique transmise. Le premier de ces traitements (mis à part un premier étage de filtrage passe bande) est la démodulation du signal reçu. Le moyen le plus courant pour cela est d'utiliser un démodulateur cohérent qui consiste à multiplier le signal reçu par une onde de même fréquence et de même phase. Cette onde est générée localement. Ce type de démodulation nécessite deux opérations de synchronisation distinctes :

- **La synchronisation de la fréquence**

Au cours de la transmission, l'émetteur peut avoir un mouvement relatif par rapport au récepteur, ce qui par effet Doppler entraînera un décalage de la fréquence porteuse. Ce décalage peut aussi provenir d'une imprécision entre les oscillateurs locaux utilisés. Cette différence de fréquence est connue sous le nom de résidu de porteuse (CFO en anglais pour Carrier Frequency Offset). Elle a pour effet de "faire tourner" plus ou moins lentement la constellation des symboles émis. Evidemment, un résidu trop important entraîne une dégradation importante des performances.

- **La synchronisation de la phase**

Pour éviter un phénomène de diaphonie (interférences entre la voie en phase et la voie en quadrature), l'onde du démodulateur doit être en phase avec l'onde reçue. Le déphasage de l'onde reçue par rapport à l'onde émise provient entre autre du délais de propagation. Il sera d'autant plus important que la fréquence porteuse est grande. Au niveau des performances, il est vital de recalibrer la constellation reçue afin de minimiser l'erreur de décision à l'issue du comparateur à seuil.

Une fois la démodulation réalisée, le récepteur doit échantillonner le signal démodulé (on parle alors de signal en bande de base). Même si la chaîne considérée vérifie le critère de Nyquist, les échantillons seront obtenus sans interférence entre symboles uniquement si l'échantillonnage est réalisé aux instants optimaux. Ces instants

correspondent aux positions où le diagramme de l'oeil est le plus ouvert verticalement. Cette opération est connue sous le nom de **synchronisation rythme**. Dans tout récepteur, il est important d'avoir un dispositif qui contrôle l'instant d'échantillonnage pour éviter toute dérive du rythme d'échantillonnage. Sans ce contrôle nous pourrions nous retrouver avec des symboles en plus ou en moins si la cadence d'échantillonnage est légèrement inférieure ou supérieure à  $1/T_s$ .

Il existe un dernier type de synchronisation qui est connu sous le nom de **synchronisation trame**. Elle consiste à déterminer le début et la fin d'une trame de symboles. De plus, quand un codage canal est utilisé dans le système de transmission, il est nécessaire de découper la séquence de symboles reçus en une séquence de mots de code reçus. La mauvaise localisation des mots de code dans cette séquence entraînera des effets catastrophiques sur les performances du système.

Généralement les techniques mises en oeuvre pour réaliser les différentes synchronisations se basent sur la connaissance d'une séquence de symboles appelée séquence pilote ou séquence d'apprentissage. Cette séquence est plus ou moins longue et est répétée plus ou moins souvent en fonction des caractéristiques du canal de propagation. Ces approches sont connues sous le nom d'approches supervisées. Les symboles détectés à la réception servent à effectuer les différentes synchronisations. L'inconvénient majeur de ce type d'approches est la perte d'efficacité spectrale due à l'ajout d'une séquence connue. De plus lorsque les conditions de propagation se dégradent, il est nécessaire d'augmenter la longueur des séquences pilotes. De même, il faut aussi augmenter les séquences pilotes pour les systèmes capables de transmettre à des puissances d'émission plus faibles. En effet, les systèmes de transmission actuels utilisent de plus en plus un codage de canal sophistiqué et grâce aux gains de codage importants des nouvelles familles de codes correcteurs (Turbo codes, codes LDPC, codes produits,...), les systèmes sont capables de transmettre à de faibles SNR (Signal-to-Noise Ratio) [7, 51, 61].

Une alternative à l'approche supervisée consiste à développer des techniques dites aveugles qui ne nécessitent aucune connaissance *a priori* sur la valeur des symboles transmis. On a alors un gain immédiat en termes d'efficacité spectrale mais aussi la possibilité d'effectuer à tout instant une re-synchronisation ou de la suivre au fil des échantillons reçus : nous n'avons plus besoin d'attendre la réception des pilotes.

Nous supposons dans cette thèse que le récepteur connaît le rythme de la transmission, c'est-à-dire qu'il dispose d'une horloge lui permettant d'échantillonner les signaux au bon moment. Ainsi, le problème de la synchronisation rythme ne sera pas abordé dans ce document.

Nous considérons un canal à BABG (Bruit Additif Blanc Gaussien) et nous modélisons le  $k^{\text{ième}}$  échantillon reçu par :

$$r(k) = b(k - t_0)e^{j(2\pi k f_0 T_s + \theta_0)} + w(k), \quad (1)$$

où  $b(k)$  désigne un symbole codé, modulé et transmis,  $w(k)$  est un bruit blanc Gaussien et  $T_s$  est le temps symbole. Le paramètre  $t_0$  est un entier positif désignant le retard des échantillons transmis. Le but de la synchronisation trame est d'estimer  $t_0$ . La variable  $f_0$  désigne le résidu de porteuse du système et  $\theta_0$  est la phase introduite par le canal de propagation.

Dans cette thèse nous proposons de nouvelles techniques de synchronisation aveugles qui exploitent la structure du code correcteur d'erreurs. Nous verrons dans la suite de ce document que ces méthodes sont très performantes même pour des faibles SNR. Les objectifs de cette thèse se résument alors par l'estimation aveugle des trois paramètres suivants :

- l'instant de synchronisation trame  $t_0$ ,
- le résidu de porteuse  $f_0$ ,
- le déphasage  $\theta_0$ .

## Organisation du document

Le premier chapitre de ce document traite du problème de la synchronisation trame. Après avoir brièvement présenté quelques méthodes classiques existantes, nous proposons une méthode originale d'estimation de l'instant de synchronisation trame. Cette méthode que nous désignons par la méthode du SSS (Soft Syndrome based Synchronization) est fondée sur les propriétés du code correcteur d'erreurs, en particulier sur sa matrice de contrôle de parité. En admettant quelques hypothèses, nous démontrons que la méthode du SSS est basée sur un critère MAP (Maximum a Posteriori). Dans ce même chapitre, nous développons une étude théorique sur le critère étudié et nous donnons les propriétés de la méthode du SSS. Nous démontrons que cet algorithme de synchronisation présente de meilleures performances quand des codes correcteurs d'erreurs ayant des matrices de contrôle de parité creuses sont utilisés dans le système.

Le deuxième chapitre est consacré à l'application de la méthode de synchronisation trame du SSS à des systèmes utilisant différents types de codes correcteurs d'erreurs. Nous commençons tout d'abord par les codes LDPC définis par leur matrice de contrôle de parité creuse et nous comparons les performances de notre méthode à celles de plusieurs méthodes existant dans la littérature. Les codes convolutifs sont ensuite examinés et les résultats des courbes de BER (Bit Error Rate) sont très satisfaisants. Les codes BCH et les codes produits sont traités dans la troisième partie de ce chapitre. Bien que ces codes correcteurs d'erreurs n'ont pas des matrices de contrôle de parité creuses, nous proposons une méthode originale de synchronisation de ce type de codes (algorithme du m-SSS pour modified Soft Syndrome based Synchronization). Cette méthode est basée sur l'adaptation de la matrice de contrôle de parité à la

séquence d'échantillons reçus. Les simulations précédentes sont réalisées dans le cas d'une modulation BPSK (Binary Phase Shift Keying). Une extension de notre critère de synchronisation aux cas de modulations à grand nombre d'états est effectuée.

Le troisième chapitre traite du problème d'estimation du résidu de porteuse. Nous supposons dans ce chapitre que l'instant de synchronisation trame a été correctement estimé et qu'un résidu de porteuse est présent dans le système. Dans la première partie de ce chapitre, nous considérons le cas d'un canal de transmission à déphasage connu au récepteur et nous proposons une nouvelle méthode d'estimation du résidu de porteuse. Cette technique que nous appelons technique du C-SFM (CFO estimation-Syndrome Function Minimization), est fondée sur le calcul et la minimisation d'une fonction du LLR (Log-Likelihood Ratio) du syndrome du code correcteur d'erreurs. Dans la seconde partie du chapitre, nous supposons qu'un déphasage inconnu est introduit par le canal et nous appliquons la méthode du C-SFM pour estimer le résidu de porteuse du système. Nous démontrons que cette méthode est capable d'estimer le résidu de porteuse du système indépendamment de la présence de ce déphasage inconnu. Les résultats des simulations ont montré que la méthode du C-SFM est très efficace et ses performances dépassent largement celles d'une méthode classique d'estimation du résidu de porteuse.

Dans le quatrième chapitre de ce mémoire, nous considérons le problème d'estimation du déphasage introduit par le canal de transmission. Nous supposons tout d'abord que l'instant de synchronisation trame et le résidu de porteuse sont connus par le récepteur et nous proposons deux techniques d'estimation aveugle du déphasage. Ces techniques sont basées sur le même concept étudié tout au long de cette thèse et qui se résume par le calcul des fonctions du LLR du syndrome du code correcteur d'erreurs. La première technique désignée par la méthode du SFR (Syndrome Functions Ratio) calcule une expression analytique de l'estimée du déphasage alors que la deuxième (méthode du P-SFM pour Phase estimation-Syndrome Function Minimization) est basée sur la minimisation d'une fonction de coût. Celle-ci présente de très bonnes performances et surpasse plusieurs méthodes existantes. Quand le résidu de porteuse et le déphasage sont tous les deux inconnus, estimer tout d'abord le résidu de porteuse par l'algorithme du C-SFM puis le déphasage par le P-SFM, aboutissent à de très bons résultats.

# Chapitre 1

## Synchronisation trame aveugle sur canal Gaussien

CE chapitre traite du problème de la synchronisation trame aveugle sur canal Gaussien. Nous commençons tout d'abord par un état de l'art qui présente brièvement quelques méthodes de synchronisation trame proposées dans la littérature. Ensuite, nous proposons une nouvelle technique de synchronisation trame aveugle appliquée sur un canal Gaussien et basée sur le calcul du LLR du syndrome du code correcteur d'erreurs utilisé dans le système de transmission. Cette méthode est fondée sur un critère MAP (Maximum A Posteriori) qui maximise la probabilité qu'une position d'un échantillon donné correspond au bon moment de synchronisation, sachant le signal reçu. La procédure de maximisation de cette probabilité est effectuée par le biais d'une fenêtre glissante posée sur la séquence d'échantillons reçus. Une étude théorique du critère de synchronisation introduit est ensuite effectuée. Cette étude permet de trouver les distributions de probabilité de ce critère de synchronisation et de calculer des équations analytiques des probabilités de fausse synchronisation. Finalement, nous déterminons les propriétés de la méthode de synchronisation proposée et nous démontrons que cette nouvelle technique est plus performante quand les codes correcteurs d'erreurs utilisés dans le système de communication ont des matrices de contrôle de parité creuses, c'est-à-dire des matrices contenant un grand nombre de zéros.

## 1.1 Etat de l'art

La croissance du nombre de services nécessitant des transmissions sans fils a été phénoménale lors des dix dernières années. Afin d'améliorer la qualité de la transmission, le codage du canal doit être utilisé. A la réception, pour pouvoir décoder la séquence d'échantillons reçus, 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 au récepteur. Cette séquence, appelée séquence d'apprentissage, est détectée à la réception en passant les échantillons reçus par le biais d'un corrélateur [4]. La conception et l'optimisation des séquences d'apprentissage ont été traitées dans plusieurs travaux de recherche comme dans [25, 66].

### *Méthode de synchronisation de Massey*

En 1972, Massey a décrit une règle de décision optimale pour la localisation de la position de synchronisation sur un canal BABG (Bruit Additif Blanc Gaussien). Il a démontré que l'ajout d'un terme correcteur à la sortie du corrélateur améliore les performances du système. Selon Massey, la méthode optimale d'estimation de la position de synchronisation choisit la position  $t$  qui maximise la valeur de :

$$S = \sum_{k=0}^{L-1} s(k)r(k+t) - \sum_{k=0}^{L-1} f(r(k+t)), \quad (1.1)$$

où  $L$  désigne la taille de la séquence d'apprentissage,  $s(k)$  est un symbole appartenant à cette séquence, et  $r(k)$  est un échantillon reçu.  $f$  est une fonction définie par :

$$f(x) = \frac{N_0}{2\sqrt{E_s}} \log \cosh\left(\frac{2\sqrt{E_s}x}{N_0}\right), \quad (1.2)$$

où  $N_0$  est la densité spectrale de puissance monolatérale du bruit BABG et  $E_s$  est l'énergie par symbole. Notons que le 1<sup>er</sup> terme de (1.1) n'est autre que l'inter-corrélation entre les échantillons reçus et les symboles de la séquence d'apprentissage, alors le 2<sup>ième</sup> terme est le terme correcteur. L'algorithme de synchronisation de Massey est utilisé comme référence pour plusieurs méthodes de synchronisation.

Le travail de Massey a été étendu dans [48], où les auteurs décrivent la synchronisation optimale suivant une règle de MV (Maximum de Vraisemblance) et proposent une approximation pour des SNR élevés. Afin d'améliorer la performance du système, Robertson propose dans [62] de tirer avantage de la structure du code correcteur d'erreur utilisé dans le système et de traiter la synchronisation conjointement avec le décodage. Ce travail a inspiré plusieurs auteurs comme ceux de [9] qui proposent un algorithme de synchronisation à complexité réduite, qui prend lui aussi avantage de la sortie souple du décodeur. Plutôt que de placer les symboles de la séquence d'apprentissage dans une entête séparée, les auteurs de [31] proposent de les placer parmi les données et de les coder par un codeur convolutif ou un codeur turbocode. Dans [32], une approche unifiée à la synchronisation est décrite dans laquelle les

échantillons observés sont modélisés comme étant la sortie d'une chaîne de Markov corrompue par du bruit additif blanc Gaussien.

Afin d'obtenir de bonnes performances à des faibles SNR, les méthodes traditionnelles de synchronisation sont contraintes d'augmenter la taille de la séquence d'apprentissage. Ceci réduit l'efficacité spectrale de la transmission surtout quand des codes correcteurs d'erreurs de petite taille sont utilisés. Pour cela, nous nous intéressons dans nos travaux à de nouvelles méthodes de synchronisation aveugle qui sont capables de synchroniser sans le rajout d'aucune séquence d'apprentissage. Un schéma de synchronisation aveugle a été proposé dans [53], où les auteurs présentent des algorithmes de synchronisation trame, synchronisation rythme et estimation du déphasage, et ceci suivant une procédure itérative en examinant les LLR des symboles à la sortie du décodeur. En 2005, Houcke et Sicot ont breveté une technique de synchronisation aveugle appliquée dans le cas d'un canal binaire symétrique [28]. Elle a été présentée dans [29] puis dans [45]. Cette méthode de synchronisation est présentée ci-dessous.

### 1.1.1 Méthode de synchronisation du HSS

La méthode de synchronisation aveugle que nous décrivons dans ce paragraphe est applicable pour des systèmes utilisant des codes correcteurs d'erreurs. Pour cela, il est indispensable de donner les définitions suivantes :

#### Définitions

Soit un code correcteur d'erreurs de rendement

$$R = \frac{n_b}{n_c},$$

où  $n_b$  désigne la dimension du code et  $n_c$  sa longueur. Pour ce code, il est souvent possible de trouver sa matrice de contrôle de parité  $H$  de dimension  $n_r \times n_c$ , où  $n_r = n_c - n_b$  représente le nombre d'équations de parité. L'ensemble de tous les mots de code appartenant au même système codé est défini par :

$$\mathcal{C} = \{\mathbf{c} : \mathbf{c}H^T = 0\}, \quad (1.3)$$

où  $T$  désigne la transposition et  $\mathbf{c}$  est un mot de code émis par le codeur. En utilisant la propriété d'orthogonalité de la matrice de contrôle de parité avec les mots de code, nous définissons la quantité  $\mathbf{s}$ , appelée syndrome d'erreur, par :

$$\mathbf{s} = \mathbf{y}H^T = (\mathbf{c} + \mathbf{e})H^T = \mathbf{e}H^T, \quad (1.4)$$

où  $\mathbf{y}$  est un mot de code reçu et  $\mathbf{e}$  est un mot dont les symboles non nuls représentent les erreurs. Le syndrome  $\mathbf{s}$  est nul si, et seulement si,  $\mathbf{y}$  est un mot de code. Un syndrome non nul indique la présence d'erreurs. Notons toutefois qu'un syndrome nul ne signifie pas nécessairement l'absence d'erreurs car le mot  $\mathbf{y}$  peut appartenir à l'ensemble des mots de code tout en étant différent de  $\mathbf{c}$ .

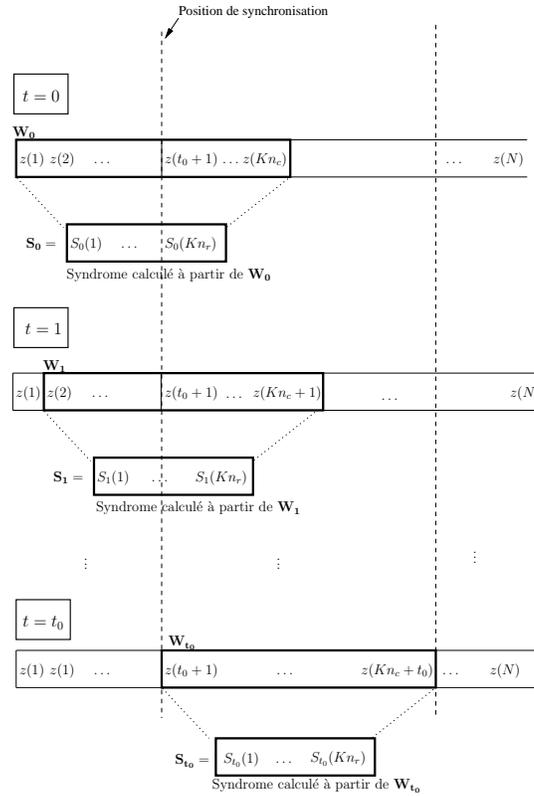


Figure 1.1 — Principe de la méthode de synchronisation trame aveugle

### Algorithme du HSS (Hard Syndrome based Synchronization)

Soit

$$\mathbf{z} = [z(1), \dots, z(N)] \quad (1.5)$$

la séquence de  $N$  échantillons reçus ayant des valeurs binaires (0 ou 1). Cette séquence est retardée de  $t_0$  bits par rapport à la séquence initialement transmise et le but de la synchronisation trame est d'estimer le retard  $t_0$ . On suppose dans la suite de ce mémoire que l'instant de synchronisation  $t_0$  est un entier appartenant à l'intervalle  $[0, n_c - 1]$ . Introduisons maintenant la séquence  $\mathbf{W}_t$  extraite de la séquence  $\mathbf{z}$ .

$$\mathbf{W}_t = [z(t), \dots, z(t + Kn_c - 1)]. \quad (1.6)$$

Appelons  $\mathbf{W}_t$  la fenêtre de synchronisation. C'est une fenêtre glissante de taille  $Kn_c$ , où l'indice  $t$  représente sa position sur la séquence  $\mathbf{z}$  et  $K$  est un entier positif différent de 0.  $\mathbf{W}_t$  peut être divisée en  $K$  blocs de longueur  $n_c$  chacun. A chaque position de la fenêtre de synchronisation et pour chacun des  $K$  blocs contenus dans cette fenêtre, on calcule le syndrome correspondant, qui est un vecteur de  $n_r$  éléments. Chaque élément

du syndrome est calculé en utilisant une équation de parité définie par une ligne de la matrice de contrôle de parité  $H$ . A la position de synchronisation, la probabilité d'avoir une équation de parité vérifiée est plus grande qu'à une position de non synchronisation. Ceci est dû au fait qu'à la position de synchronisation, le bloc testé correspond à un mot de code valide qui est corrompu uniquement par le bruit du canal, alors qu'à une position de non synchronisation et même en absence de bruit, le bloc se trouvant dans la fenêtre de synchronisation n'est pas un mot de code.

Une fois tous les éléments des syndromes calculés, on forme le vecteur :

$$\mathbf{S}_t = [S_t(1), \dots, S_t(Kn_r)]. \quad (1.7)$$

La figure 1.1 montre trois séquences différentes de  $\mathbf{W}_t$  correspondant à  $t = 0, 1$  et  $t_0$ . Ayant trouvé  $\mathbf{S}_t$ , le vecteur des syndromes, nous évaluons  $\varphi(t)$  qui désigne le nombre d'éléments non nuls dans  $\mathbf{S}_t$  :

$$\varphi(t) = \sum_{k=1}^{Kn_r} S_t(k). \quad (1.8)$$

En répétant cette procédure pour  $n_c$  valeurs successives de  $t$  et en remarquant qu'à la position de synchronisation le nombre d'éléments non nuls dans le syndrome (nombre d'erreurs) doit être minimal, l'instant de synchronisation trame est estimé par la valeur de  $t$  minimisant  $\varphi$  :

$$\hat{t}_{HSS} = \underset{t=0, \dots, n_c-1}{\operatorname{argmin}} \varphi(t). \quad (1.9)$$

La méthode présentée dans ce paragraphe et qui sera désignée dans la suite de ce document par l'algorithme du HSS (Hard Syndrome based Synchronization), est basée sur le calcul de valeurs fermes du syndrome. Malheureusement, le calcul de telles valeurs entraîne inévitablement une perte d'information. Il est donc naturel d'essayer de tirer parti de cette information et la façon la plus courante de le faire consiste à calculer les LLR des éléments du syndrome à partir du signal reçu. Une méthode de synchronisation aveugle basée sur ce concept est présentée dans le paragraphe suivant.

### 1.1.2 Méthode de synchronisation de Sun et Valenti

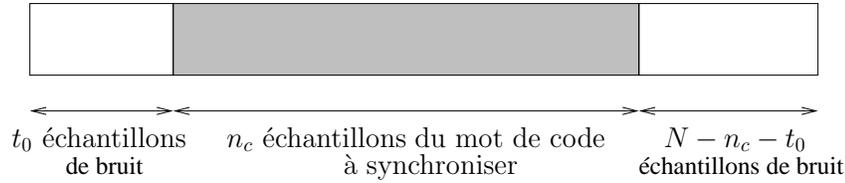
Dans la suite de ce document et sauf indication contraire, nous considérons le cas où l'émetteur transmet une séquence binaire de mots de code. Les bits de cette séquence sont modulés par une BPSK (Binary Phase Shift Keying) et le canal de propagation considéré est un canal BABG.

Soit  $b(k) = \pm 1$  le  $k^{\text{ième}}$  symbole codé et modulé à être transmis. A la réception et en supposant que le résidu de porteuse et le déphasage sont connus au récepteur, le  $k^{\text{ième}}$  échantillon reçu s'écrit sous la forme :

$$r(k) = b(k - t_0) + w(k), \quad (1.10)$$

où  $w(k)$  désigne un échantillon de bruit blanc Gaussien de variance  $\sigma^2$ . La séquence de  $N$  échantillons reçus s'écrit sous la forme :

$$\mathbf{r} = [r(1), \dots, r(N)]. \quad (1.11)$$



**Figure 1.2** — Structure du tampon considéré par Sun et Valenti pour effectuer la synchronisation trame.

Dans ce contexte de canal Gaussien, une méthode de synchronisation trame aveugle a été récemment proposée par Sun et Valenti [63, 64]. Cette méthode est appliquée à des systèmes utilisant des turbocodes comme codes correcteurs d'erreurs et est basée elle aussi sur les propriétés des matrices de contrôle de parité de ces codes. Cette technique de synchronisation aveugle est jugée optimale par ses auteurs. Cependant, ce n'est pas le cas comme nous le verrons au paragraphe 1.2.1. Dans un premier temps, nous nous contentons de décrire la méthode telle qu'elle a été proposée par ses auteurs.

Sun et Valenti supposent que le mot de code à synchroniser est entouré par des échantillons de bruit non codés et leur but est d'estimer l'instant de la synchronisation trame qui est défini comme étant la position à partir de laquelle les échantillons reçus correspondent à des symboles codés. Considérons un tampon de taille  $N$  échantillons prélevé du signal reçu. La structure de ce tampon est schématisée par la figure 1.2.

Afin d'estimer la position de synchronisation trame, le synchroniseur examine deux hypothèses : l'hypothèse  $H_0$  qui suppose un état de non synchronisation et l'hypothèse  $H_1$  indiquant une bonne position de synchronisation. L'instant de synchronisation trame est ainsi estimé suivant un estimateur MAP (Maximum *a Posteriori*) qui maximise la probabilité qu'une position  $t$  correspond au bon moment de synchronisation, sachant le signal reçu. En d'autres termes, on maximise la probabilité *a posteriori* suivante :

$$Pr[t/\mathbf{r}], t \in [0, n_c - 1], \quad (1.12)$$

où  $t$  est une position possible de l'instant de synchronisation  $t_0$ . A chaque bloc contenu dans le tampon utilisé pour la synchronisation, les auteurs calculent la somme des LLR des éléments du syndrome en la décrivant par un processus aléatoire  $e(t)$ . Ce processus suit les distributions de probabilité suivantes :

$$e(t) \sim \mathcal{N}(0, \kappa n_r m_c) \quad t \neq t_0 \quad (1.13)$$

$$\text{et } e(t) \sim \mathcal{N}(n_r m_c, \kappa n_r m_c) \quad t = t_0, \quad (1.14)$$

où  $\kappa$  est un coefficient supérieur à 2 décrivant la dépendance des éléments du syndrome. Ce coefficient est directement relié au nombre d'éléments non nuls dans les colonnes de la matrice de contrôle de parité du code correcteur d'erreurs. La variable  $m_c$  représente la moyenne du LLR d'un élément de syndrome. Selon [64], le synchroniseur trame calcule la fonction suivante :

$$\Lambda(t) = - \sum_{i=1}^t \frac{r(i)^2}{2\sigma^2} - \sum_{i=t+n_c+1}^N \frac{r(i)^2}{2\sigma^2} - \frac{(e(t) - n_r m_c)^2}{2\kappa n_r m_c}. \quad (1.15)$$

Les deux premiers termes dans (1.15) sont dus aux échantillons de bruit qui se trouvent au début et à la fin du tampon et qui entourent le mot de code à synchroniser. Le

troisième terme consiste à calculer, à chaque position  $t$ , la variable  $e(t)$  et de la comparer à la moyenne de  $e(t_0)$  pour pouvoir juger si l'on est en présence d'une position de synchronisation ou non.

Selon Sun et Valenti, l'estimée optimale de l'instant de synchronisation est obtenue par :

$$\hat{t}_{SV} = \underset{t=0, \dots, n_c-1}{\operatorname{argmax}} \{\Lambda(t)\}. \quad (1.16)$$

Remarquons que la détermination du critère de synchronisation dans (1.31) nécessite la connaissance de  $\kappa$  et  $m_c$ . Ces coefficients dépendent de la constitution de la matrice de contrôle de parité du code correcteur d'erreurs et leur calcul est indispensable avant d'appliquer la procédure de synchronisation. Cependant, bien que la valeur de  $m_c$  peut être obtenue par simulations, aucune procédure n'a été proposée pour le choix de  $\kappa$ .

## 1.2 Méthode proposée de synchronisation aveugle basée sur un critère de Maximum a Posteriori - Algorithme du SSS

Nous proposons dans ce paragraphe une nouvelle méthode de synchronisation trame aveugle qui a été inspirée par la méthode de synchronisation présentée dans le paragraphe 1.1.1 et qui est très proche de celle présentée dans le paragraphe 1.1.2. Dans notre étude, nous considérons le cas général où le bloc que nous cherchons à synchroniser est situé entre des échantillons quelconques qui sont souvent des échantillons codés, modulés et bruités provenant d'autres mots de code transmis.

Nous démontrons dans la suite que la technique de synchronisation que nous proposons est elle basée sur un critère MAP qui maximise (1.12) mais nous verrons que notre critère final de synchronisation est différent de celui proposé par Sun et Valenti. En effet, pour pouvoir confirmer qu'une position  $t$  de la fenêtre de synchronisation est la bonne position de synchronisation, nous devons vérifier que le bloc transmis à cette position est un mot de code valide **et que tous les autres blocs correspondants aux  $(n_c - 1)$  positions restantes ne le sont pas**. Comme nous l'avons déjà mentionné précédemment, la manière la plus simple pour vérifier si un bloc donné est un mot de code ou non est de calculer son syndrome à partir de la matrice de contrôle de parité du code. Pour la même raison que celle donnée dans le paragraphe 1.1.2, nous calculons le LLR du syndrome et ceci à partir des LLR des échantillons reçus.

Notons que le principe de la fenêtre glissante de taille  $Kn_c$  symboles déjà introduit dans le paragraphe 1.1.1 est aussi utilisé dans la méthode de synchronisation que nous proposons. Comme nous considérons que la position de synchronisation  $t_0$  peut prendre n'importe quelle valeur entière inférieure à  $n_c$ , nous devons faire passer la fenêtre de synchronisation sur  $n_c$  positions possibles sur la séquence d'échantillons reçus. L'étude de tous les blocs contenus dans ces  $n_c$  positions nous permet de décider si une position donnée correspond à la bonne position de synchronisation ou non. Pour des raisons de simplicité et sauf indication contraire, nous supposons dans la suite de ce document que  $K = 1$ . C'est-à-dire que la fenêtre de synchronisation utilisée contient un seul bloc de longueur  $n_c$  bits.

Soit maintenant  $\psi(t)$  le LLR du syndrome calculé à la position  $t$  de la fenêtre de synchronisation.

$$\psi(t) = \log \left( \frac{\Pr[S_t(1), \dots, S_t(n_r)] \neq 0}{\Pr[S_t(1), \dots, S_t(n_r)] = 0} \right) = \log \left( \frac{1 - P_t}{P_t} \right), \quad (1.17)$$

où  $S_t(k)$  désigne la valeur d'un élément du syndrome et  $P_t$  est la probabilité que tous les éléments du syndrome soient nuls.

Nous supposons dans la suite que les éléments du syndrome sont tous indépendants. La validité de cette hypothèse est discutée plus loin dans ce chapitre. Dans ces conditions, la probabilité  $P_t$  s'écrit sous la forme :

$$P_t = \Pr[S_t(1), \dots, S_t(n_r)] = 0 = \prod_{k=1}^{n_r} \Pr[S_t(k) = 0]. \quad (1.18)$$

Soit  $L(S_t(k)) = \log \left( \frac{\Pr[S_t(k)=1]}{\Pr[S_t(k)=0]} \right)$  le LLR du  $k^{\text{ième}}$  élément du syndrome,

$$\Pr[S_t(k) = 0] = \frac{1}{1 + \exp(L(S_t(k)))}. \quad (1.19)$$

En utilisant (1.17), (1.18), (1.19) et ayant  $\prod_{k=1}^{n_r} \left( 1 + \exp(L(S_t(k))) \right)$  largement supérieur à 1, le LLR du syndrome s'écrit sous la forme :

$$\psi(t) = \sum_{k=1}^{n_r} \log \left( 1 + \exp(L(S_t(k))) \right). \quad (1.20)$$

Maximiser (1.12) revient donc à minimiser (1.20). Ainsi, l'instant de synchronisation trame est estimé par :

$$\hat{t}_0 = \underset{t=0, \dots, n_c-1}{\operatorname{argmin}} \{ \psi(t) \}. \quad (1.21)$$

Pour des raisons de simplification, (1.20) peut être approximée par :

$$\phi(t) = \sum_{k=1}^{n_r} L(S_t(k)). \quad (1.22)$$

Remarquons aussi que (1.22) est équivalente à définir le LLR du syndrome par :

$$\phi(t) = \log \left( \frac{\Pr[S_t(1) = 1, \dots, S_t(n_r) = 1]}{\Pr[S_t(1), \dots, S_t(n_r)] = 0} \right). \quad (1.23)$$

Cette définition dans (1.23) s'interprète comme étant le cas extrême : soit toutes les équations de parité sont vérifiées soit aucune ne l'est.

Finalement, la méthode proposée de synchronisation trame aveugle estime la position de synchronisation par :

$$\hat{t}_0 = \underset{t=0, \dots, n_c-1}{\operatorname{argmin}} \{ \phi(t) \}. \quad (1.24)$$

Pour pouvoir estimer l'instant de synchronisation comme proposé ci-dessus, nous devons trouver une équation analytique du LLR des éléments du syndrome. En remarquant qu'un élément de syndrome n'est rien d'autre que la somme modulo 2 de bits et en s'inspirant par l'algorithme de décodage des codes LDPC [44], nous écrivons  $L(S_t(k))$  sous la forme :

$$L(S_t(k)) = (-1)^{u_k+1} \operatorname{atanh} \left( \prod_{j=1}^{u_k} \tanh \left( \frac{\tilde{r}(t+k_j)}{2} \right) \right), \quad (1.25)$$

où

$$\tilde{r}(i) = \frac{2}{\sigma^2} r(i) \quad (1.26)$$

est le LLR du  $i^{\text{ième}}$  échantillon reçu et  $\sigma^2$  est la variance du bruit.  $u_k$  et  $k_j$  représentent respectivement le nombre d'éléments non nuls dans la  $k^{\text{ième}}$  ligne de la matrice de contrôle de parité du code et la position du  $j^{\text{ième}}$  élément non nul dans cette  $k^{\text{ième}}$  ligne.

Selon [26], une approximation de (1.25) est donnée par :

$$\tilde{L}(S_t(k)) = (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \operatorname{sign}(\tilde{r}(t+k_j)) \right) \min_{j=1, \dots, u_k} |\tilde{r}(t+k_j)|. \quad (1.27)$$

Ayant  $\tilde{r}(i)$  proportionnel à  $r(i)$  et en supposant la variance du bruit constante durant une transmission,  $\tilde{r}(t+k_j)$  peut être remplacée par  $r(t+k_j)$  dans (1.27). De cette manière, aucune information *a priori* concernant le canal BABG n'est exigée [13].  $\tilde{L}(S_t(k))$  et  $\phi(t)$  sont alors proportionnels à :

$$\hat{L}(S_t(k)) = (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \operatorname{sign}(r(t+k_j)) \right) \min_{j=1, \dots, u_k} |r(t+k_j)| \quad (1.28)$$

$$\text{et } \hat{\phi}(t) = \sum_{k=1}^{n_r} \hat{L}(S_t(k)). \quad (1.29)$$

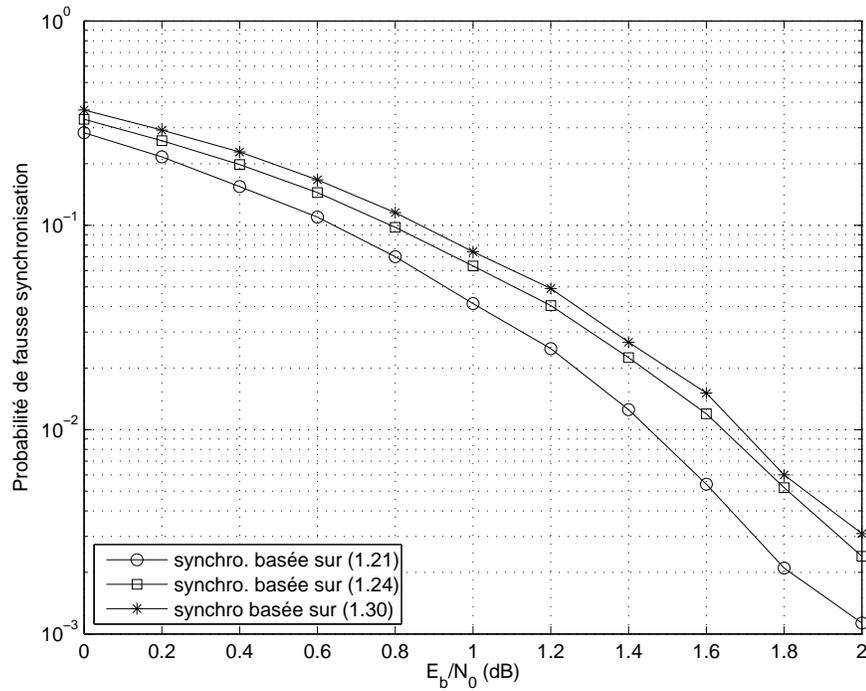
Dans la suite de ce document, la méthode de synchronisation trame aveugle que nous proposons dans ce paragraphe sera désignée par l'algorithme du SSS (Soft Syndrome based Synchronization). Cette méthode estime l'instant de synchronisation trame par :

$$\hat{t}_{SSS} = \operatorname{argmin}_{t=0, \dots, n_c-1} \{\hat{\phi}(t)\}. \quad (1.30)$$

En comparant (1.29) et (1.8), nous observons la grande ressemblance entre la méthode du HSS et celle du SSS. La première minimise la somme des éléments du syndrome alors que la deuxième revient à minimiser la somme des LLR des éléments du syndrome.

Rappelons que l'étude effectuée dans ce paragraphe a été faite dans le cas d'une fenêtre de synchronisation contenant  $K = 1$  bloc. Pour une fenêtre de synchronisation contenant un nombre de blocs  $K > 1$ , il suffit de remplacer  $n_r$  par  $Kn_r$  dans (1.29) puis estimer l'instant de synchronisation comme indiqué dans (1.30).

Notons que la méthode de synchronisation proposée dans ce paragraphe a fait l'objet d'une publication dans la revue IEEE Transactions on Communications [37] et dans la conférence EUSIPCO 2007 [34].



**Figure 1.3** — Validation des approximations effectuées dans le critère de synchronisation du SSS.

## Validation des approximations effectuées dans le critère de synchronisation

Dans la méthode de synchronisation que nous venons de présenter (algorithme du SSS), nous avons effectué plusieurs approximations avant d'atteindre le critère final de synchronisation. La validité de ces approximations est vérifiée sur la figure 1.3. Sur cette figure, nous avons appliqué la méthode de synchronisation sur un code LDPC de longueur  $n_c = 512$  bits, rendement  $R = 0.5$  et ayant  $u_k = 5$  éléments non nuls dans chaque ligne de sa matrice de contrôle de parité. Les courbes tracées correspondent à la probabilité de fausse synchronisation et sont obtenues en estimant l'instant de synchronisation par les estimateurs correspondant respectivement à (1.21), (1.24) et (1.30). Sur cette figure, nous observons que les trois critères de synchronisation donnent à peu près les mêmes résultats de synchronisation. Une différence de moins de  $0.1$  dB est détectée entre les estimateurs de (1.24) et (1.30), et de moins de  $0.3$  dB entre ceux de (1.21) et (1.30), ce qui vérifie la validité de ces approximations. Cependant, l'estimateur (1.30) ne contient aucun calcul d'exponentiel et ne nécessite pas la connaissance de la variance du bruit. Pour cette raison, notre critère de synchronisation utilisé dans la suite de ce document est basé sur (1.30).

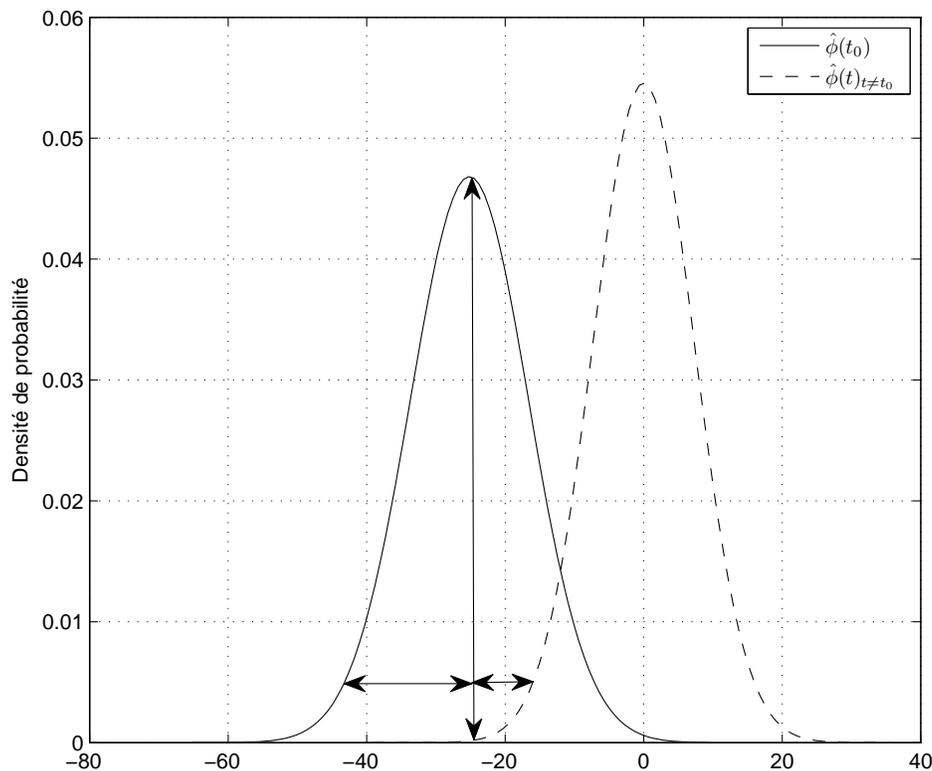


Figure 1.4 — Distributions de probabilité de  $\hat{\phi}$  aux positions non synchronisées et synchronisée, pour un code LDPC de taille 512 bits et à  $E_b/N_0 = 0$  dB.

### 1.2.1 Comparaison avec la méthode de synchronisation de Sun et Valenti

Nous effectuons dans ce paragraphe une comparaison avec la méthode de synchronisation de Sun et Valenti présentée dans le paragraphe 1.1.2 de ce chapitre. Dans [64], les auteurs affirment avoir développé une méthode optimale de synchronisation trame aveugle, basée sur un critère MAP. Cependant, nous montrons dans la suite de ce paragraphe qu'ils ont oublié un terme important dans l'expression de la probabilité *a posteriori*, ce qui donne un estimateur MAP erroné.

Rappelons que [64] traite le cas d'un mot de code situé entre des échantillons de bruit (échantillons non codés ni modulés) et le but est de déterminer le début de ce mot de code. Par contre, la méthode du SSS proposée dans le paragraphe 1.2 est applicable dans le cas général où le mot de code à synchroniser peut être situé entre des échantillons codés, modulés et éventuellement bruités. Dans ce contexte d'étude, le critère de Sun et Valenti de (1.15) que l'on cherche à maximiser devient :

$$\Lambda(t) = -\frac{(e(t) - n_r m_c)^2}{2kn_r m_c}. \quad (1.31)$$

Remarquons que la variable  $e(t)$  calculée dans le critère de Sun et Valenti est inversement proportionnelle à la variable  $\hat{\phi}(t)$  de notre critère. Ainsi, maximiser (1.31) revient à minimiser la distance :

$$\Lambda'(t) = (\hat{\phi}(t) - n_r m_{t_0})^2, \quad (1.32)$$

Il est clair que notre méthode de synchronisation et celle de Sun et Valenti sont basées sur des critères qui se ressemblent beaucoup. Par contre, la méthode proposée par Sun et Valenti présente un inconvénient majeur. En effet, afin d'estimer la position de synchronisation, les auteurs examinent seulement le cas où  $\hat{\phi}(t)$  à une position  $t$  suit la même distribution de probabilité que celle à la position synchronisée, sans vérifier que les autres  $\hat{\phi}(i)_{i \neq t}$  correspondants aux autres  $(n_c - 1)$  positions testées suivent la distribution de  $\hat{\phi}$  à une position non synchronisée. En d'autres termes, comparer les variables  $\hat{\phi}$  à la moyenne de  $\hat{\phi}(t_0)$  n'est pas suffisant pour garantir une estimation optimale de  $t_0$ .

Afin d'expliquer plus clairement notre point de vue, nous avons tracé sur la figure 1.4 les distributions de probabilité de  $\hat{\phi}$  aux positions non synchronisées et synchronisée. Nous avons testé un code LDPC de taille 512 bits, ayant 5 éléments non nuls dans chaque ligne de sa matrice de contrôle de parité et les courbes ont été tracées à un  $E_b/N_0 = 0$  dB. A partir de cette figure, prenons l'exemple suivant : supposons qu'à une position  $t_1 \neq t_0$ , nous avons  $\hat{\phi}(t_1) = -17$ , tandis qu'à la position de synchronisation  $t_2 = t_0$ ,  $\hat{\phi}(t_2) = -42$ . Afin d'estimer le bon moment de synchronisation trame et ayant la moyenne  $n_r m_{t_0} = -25$ , l'estimateur de Sun et Valenti selon (1.32) va calculer les distances :

$$\begin{aligned} \Lambda'(t_1) &= (\hat{\phi}(t_1) - n_r m_{t_0})^2 = 64 \\ \text{et } \Lambda'(t_2) &= (\hat{\phi}(t_2) - n_r m_{t_0})^2 = 289. \end{aligned}$$

Comme  $\Lambda'(t_1) < \Lambda'(t_2)$ , la méthode de Sun et Valenti va commettre une erreur de synchronisation et choisir  $t_1$  comme l'instant de synchronisation trame. Par contre, notre méthode va directement comparer  $\hat{\phi}(t_1)$  et  $\hat{\phi}(t_2)$  : ayant  $\hat{\phi}(t_2) < \hat{\phi}(t_1)$ , on estime l'instant de synchronisation trame par  $t_2$ , ce qui est équivalent à une bonne synchronisation.

L'exemple ci-dessus illustre le fait que la méthode présentée par Sun et Valenti ne donne pas la meilleure estimation de la position de synchronisation. Ceci est aussi vérifié par simulations dans le chapitre suivant.

### 1.3 Etude théorique des critères de synchronisation et équations analytiques des probabilités de fausse synchronisation

Les performances des algorithmes de synchronisation trame sont mesurées en évaluant la probabilité de fausse synchronisation en fonction du rapport entre l'énergie binaire et la densité spectrale de puissance de bruit ( $E_b/N_0$ ). Dans ce paragraphe, nous exprimons analytiquement les probabilités de fausse synchronisation. La méthode du

HSS présentée dans le paragraphe 1.1.1 est d'abord étudiée. Pour cette méthode, nous trouvons les distributions de probabilité des variables associées et nous donnons une équation analytique de la probabilité de fausse synchronisation. Une étude similaire est ensuite effectuée pour la méthode du SSS que nous venons de proposer au paragraphe 1.2.

Les distributions de probabilité impliquées dans les techniques de synchronisation étudiées peuvent servir à définir de nouveaux critères de synchronisation. En effet, ayant ces lois de probabilité, nous fixons un niveau de confiance pour lequel nous calculons un seuil de synchronisation. Ainsi, au lieu de tester toutes les positions possibles pour trouver  $\hat{\phi}$  (ou  $\varphi$ ) minimal, nous admettons que la solution de notre problème se trouve au premier  $\hat{\phi}$  (ou  $\varphi$ ) inférieur à ce seuil. En procédant de cette manière, le temps de calcul nécessaire pour l'estimation de l'instant de synchronisation pourrait être réduit.

### 1.3.1 Algorithme du HSS

Rappelons que la méthode de synchronisation du HSS présentée dans 1.1.1 estime l'instant de synchronisation comme étant la position de la fenêtre de synchronisation pour laquelle la valeur de  $\varphi(t)$  définie par (1.8) est minimale. Si à la position de synchronisation le critère  $\varphi(t_0)$  n'a pas une valeur minimale comparée aux autres positions de la fenêtre de synchronisation, on commettra alors une erreur de synchronisation. Désignons par  $P_{F_{HSS}}$  la probabilité de fausse synchronisation. Elle s'écrit sous la forme :

$$P_{F_{HSS}} = Pr[\varphi(t_0) > \min_{t=0, \dots, n_c-1, t \neq t_0} \varphi(t)] \quad (1.33)$$

Pour pouvoir calculer cette probabilité, il faut déterminer les distributions de probabilité que suivent  $\varphi(t_0)$  et  $\varphi(t)_{t=0, \dots, n_c-1, t \neq t_0}$ .

Aux positions de non-synchronisation (i.e  $t \neq t_0$ ), le bloc contenu dans la fenêtre de synchronisation contient des symboles quelconques (qui ne forment pas forcément un mot de code). Dans ce cas, une équation de parité a une probabilité de 1/2 d'être vérifiée. Ainsi, chaque élément de syndrome  $S_t(k)$  est une variable aléatoire de Bernoulli de paramètre 1/2.

A la position de synchronisation,  $S_{t_0}(k)$  suit une distribution de Bernoulli de paramètre  $p_k$  où  $p_k$  représente la probabilité que  $S_{t_0}(k)$  soit égal à 1. C'est la probabilité d'avoir un nombre impair d'erreurs parmi les  $u_k$  bits vérifiant la  $k^{\text{ième}}$  équation de parité.

$$p_k = Pr[S_{t_0}(k) = 1] = \sum_{l=0}^{\lfloor \frac{u_k-1}{2} \rfloor} \binom{u_k}{2l+1} p_e^{2l+1} (1-p_e)^{u_k-2l-1}, \quad (1.34)$$

où  $p_e$  est la probabilité d'erreur du canal.  $\binom{n}{k}$  désigne le nombre de combinaisons de  $k$  éléments parmi  $n$  et  $\lfloor x \rfloor$  est la partie entière de  $x$ .

Supposons que les éléments du syndrome sont indépendants entre eux. Nous verrons dans le paragraphe 1.4 de ce chapitre que la validité de cette supposition est liée au nombre d'éléments non nuls dans la matrice de contrôle de parité du code et à la longueur du code. Elle est d'autant plus valide que la matrice de contrôle de parité est

creuse et le code correcteur d'erreurs est long. En supposant que  $u_k$  est constant pour toutes les lignes de  $H$ , nous avons  $p_k = p \forall k$ . La variable aléatoire  $\varphi$ , qui est la somme des  $n_r$  éléments du syndrome, suit alors une distribution Binomiale :

$$\varphi(t) \sim \mathcal{B}(n_r, 1/2) \quad t \neq t_0 \quad (1.35)$$

$$\text{et } \varphi(t_0) \sim \mathcal{B}(n_r, p). \quad (1.36)$$

Ayant calculé les distributions de probabilité ci-dessus, nous démontrons que l'équation analytique de la probabilité de fausse synchronisation s'écrit :

$$P_{F_{HSS}} = 1 - \left[ \sum_{k=0}^{n_r-1} \left( \binom{n_r}{k} p^k (1-p)^{n_r-k} \sum_{j=k+1}^{n_r} \left( \frac{\binom{n_r}{j}}{2^{n_r}} \right) \right) \right]^{n_c-1}. \quad (1.37)$$

La preuve détaillée de (1.37) est donnée en Annexe A.

### 1.3.2 Algorithme du SSS

Une étude similaire à celle que nous avons faite dans le paragraphe précédent est maintenant effectuée pour la technique du SSS décrite dans le paragraphe 1.2 de ce chapitre. Nous commençons tout d'abord par trouver les distributions de probabilité de notre critère de synchronisation  $\hat{\phi}(t)$  pour  $t = t_0$  et  $t \neq t_0$ . Ensuite, nous en déduisons une équation analytique de la probabilité de fausse synchronisation.

#### Distributions de probabilité

L'équation (1.28) peut s'écrire sous la forme :

$$\hat{L}(S_t(k)) = X_t(k)Y_t(k), \quad (1.38)$$

où

$$X_t(k) = (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(r(t+k_j)) \right) = 2S_t(k) - 1 \quad (1.39)$$

est la valeur modulée de l'élément de syndrome  $S_t(k)$  et

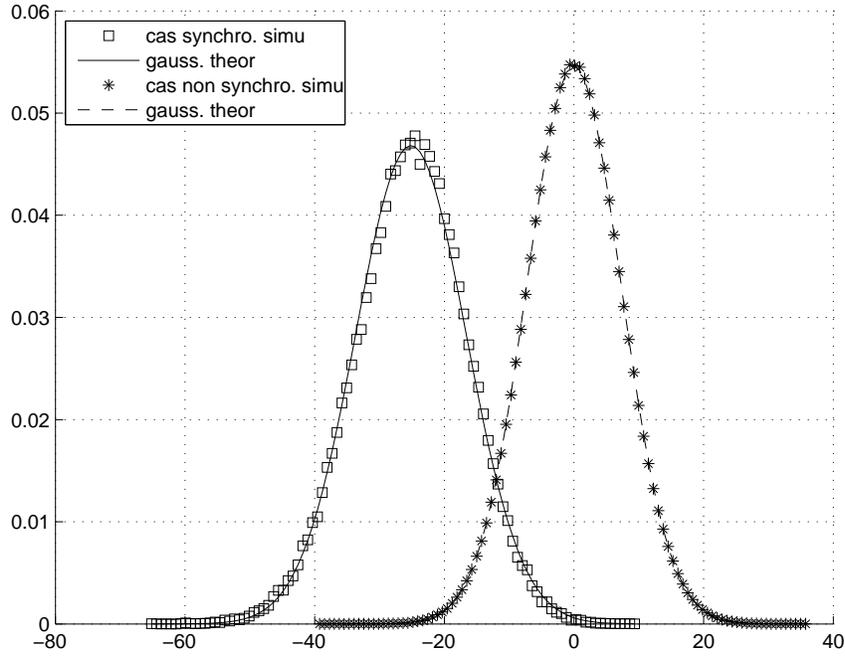
$$Y_t(k) = \min_{j=1, \dots, u_k} |r(t+k_j)| \quad (1.40)$$

est sa fiabilité. En utilisant les notations ci-dessus, (1.29) s'écrit :

$$\hat{\phi}(t) = \sum_{k=1}^{n_r} \hat{L}(S_t(k)) = \sum_{k=1}^{n_r} (2S_t(k) - 1)Y_t(k). \quad (1.41)$$

Supposons maintenant que les éléments du syndrome sont tous indépendants. En appliquant le théorème de la limite centrale,  $\hat{\phi}(t)$  suit une distribution Gaussienne :

$$\hat{\phi}(t) \sim \mathcal{N}(n_r m_t, n_r \sigma_t^2), \quad (1.42)$$



**Figure 1.5** — Vérification que les  $\hat{\phi}(t)$  simulées suivent des distributions Gaussiennes

où  $m_t$  et  $\sigma_t^2$  représentent respectivement la moyenne et la variance de  $\hat{L}(S_t(k))$ .  
 Pour  $t \neq t_0$  :

$$Pr[S_t(k) = 0] = Pr[S_t(k) = 1] = 1/2, \quad (1.43)$$

indépendamment des valeurs des échantillons reçus. On en déduit que  $S_t(k)$  et  $Y_t(k)$  sont indépendants, ce qui donne :

$$m_t = E[\hat{L}(S_t(k))] = E[X_t(k)Y_t(k)] = E[2S_t(k) - 1]E[Y_t(k)] = 0$$

et  $\sigma_t^2 = var[\hat{L}(S_t(k))] = E[Y_t(k)^2]$ .

Pour  $t = t_0$ , le signe d'un élément du syndrome dépend seulement de la présence d'une erreur. Dans ce cas,  $S_{t_0}(k)$  et  $Y_{t_0}(k)$  ne sont plus indépendants. Ainsi, le calcul de la moyenne  $m_{t_0}$  et la variance  $\sigma_{t_0}^2$  de  $\hat{L}(S_{t_0}(k))$  n'est plus si simple à effectuer. Nous proposons dans le paragraphe A.2 de l'Annexe A une démarche possible pour calculer  $m_{t_0}$  et  $\sigma_{t_0}^2$ .

En résumé :

$$\hat{\phi}(t) \sim \mathcal{N}(0, n_r \sigma_t^2) \quad t \neq t_0 \quad (1.44)$$

$$\text{et } \hat{\phi}(t_0) \sim \mathcal{N}(n_r m_{t_0}, n_r \sigma_{t_0}^2). \quad (1.45)$$

Pour valider l'hypothèse que les variables  $\hat{\phi}(t)$  suivent des Gaussiennes, nous avons tracé les fonctions de densité de probabilité des  $\hat{\phi}(t)$  obtenues par simulations et nous les avons comparées avec les courbes exactes des densités de probabilité de Gaussiennes. Le code sur lequel nous avons effectué ces tests est un code LDPC de taille  $n_c = 512$

bits, rendement  $R = 0.5$  et ayant  $u_k = 5$  éléments non nuls dans chaque ligne de  $H$ . Les résultats obtenus sont présentés sur la figure 1.5, où le  $E_b/N_0$  est égal à 0 dB. Les densités de probabilité simulées collent bien avec celles théoriques et ceci pour les deux cas, synchronisé et non synchronisé. Notre hypothèse sur les distributions de probabilité de  $\hat{\phi}(t)$  est donc vérifiée pour ce type de code.

### Expression analytique de la probabilité de fausse synchronisation

Le critère de synchronisation de la méthode du SSS étant basé sur (1.30), la probabilité de fausse synchronisation s'écrit sous la forme à :

$$P_{FSSS} = 1 - Pr \left[ \min_{t=0, \dots, n_c-1, t \neq t_0} \hat{\phi}(t) > \hat{\phi}(t_0) \right] \quad (1.46)$$

Soit  $X$  une variable aléatoire qui suit une distribution normale  $\mathcal{N}(0, 1)$ . Comme  $\hat{\phi}(t_0)$  suit la distribution  $\mathcal{N}(n_r m_{t_0}, n_r \sigma_{t_0}^2)$ , (1.46) s'écrit :

$$P_{FSSS} = 1 - Pr \left[ \min_{t=0, \dots, n_c-1, t \neq t_0} \hat{\phi}(t) > \sqrt{n_r \sigma_{t_0}^2} X + n_r m_{t_0} \right] \quad (1.47)$$

En conditionnant sur  $X$  on aura :

$$P_{FSSS} = 1 - \int_{-\infty}^{\infty} Pr \left[ \min_{t=0, \dots, n_c-1, t \neq t_0} \hat{\phi}(t) > \sqrt{n_r \sigma_{t_0}^2} X + n_r m_{t_0} \mid X = x \right] f_X(x) dx \quad (1.48)$$

où  $f_X$  est la densité de probabilité de  $X$ . Les variables  $\hat{\phi}(t)_{t=0, \dots, n_c-1, t \neq t_0}$  étant indépendantes et suivant la même distribution de probabilité, on a :

$$\begin{aligned} P_{FSSS} &= 1 - \int_{-\infty}^{\infty} \prod_{t=0, t \neq t_0}^{n_c-1} Pr \left[ \hat{\phi}(t) > \sqrt{n_r \sigma_{t_0}^2} x + n_r m_{t_0} \right] f_X(x) dx \\ &= 1 - \int_{-\infty}^{\infty} \left( Pr \left[ \hat{\phi}(t) > \sqrt{n_r \sigma_{t_0}^2} x + n_r m_{t_0} \right] \right)^{n_c-1} f_X(x) dx. \end{aligned} \quad (1.49)$$

Nous introduisons maintenant la fonction  $Q(\alpha)$  qui est souvent utilisée dans les communications numériques. Elle est appelée fonction  $Q$  et est définie par :

$$Q(\alpha) \triangleq \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{-\frac{x^2}{2}} dx. \quad (1.50)$$

Ayant  $\hat{\phi}(t) \sim \mathcal{N}(0, n_r \sigma_t^2)$ , l'expression analytique de la probabilité de fausse synchronisation devient :

$$P_{FSSS} = 1 - \int_{-\infty}^{\infty} Q^{n_c-1} \left( \sqrt{\frac{\sigma_{t_0}^2}{\sigma_t^2}} x + \sqrt{\frac{n_r}{\sigma_t^2}} m_{t_0} \right) f_X(x) dx. \quad (1.51)$$

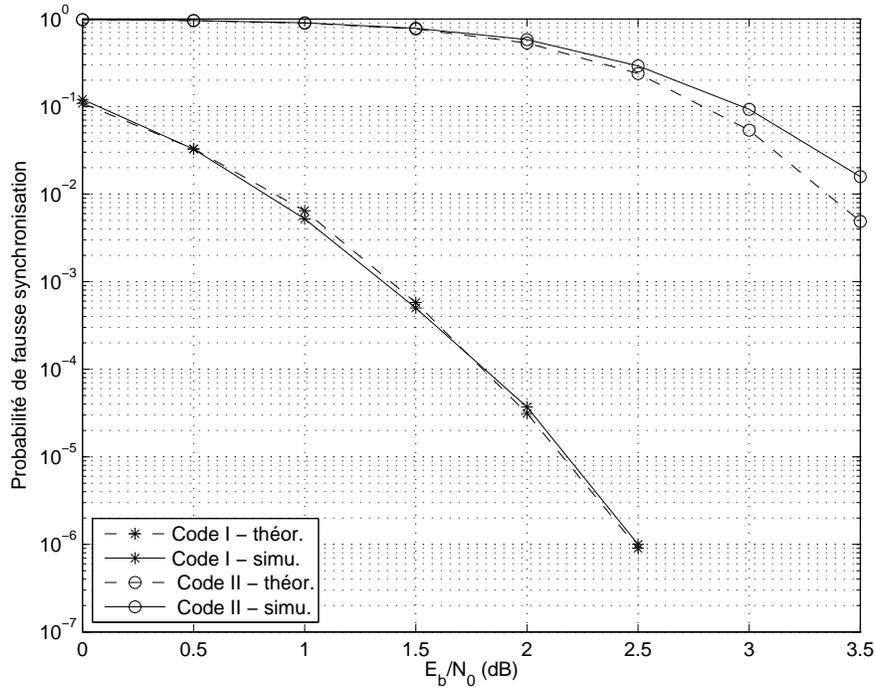


Figure 1.6 — Validation de l'étude théorique de l'approche HSS.

### 1.3.3 Validation de l'étude théorique des algorithmes HSS et SSS

Afin de valider l'étude théorique effectuée pour les deux méthodes du HSS et SSS, nous comparons dans ce paragraphe les résultats théoriques à ceux des simulations. Pour cela, nous considérons deux codes LDPC de longueur  $n_c = 511$  bits, de rendement  $R = 0.7$ , qui diffèrent seulement par la composition de leur matrice de contrôle de parité : Code I (respectivement II) a 4 ( respectivement 10) éléments non nuls dans chaque ligne de sa matrice de contrôle de parité. Bien que Code I ne soit pas un code LDPC puissant en termes de minimisation du BER, nous l'avons utilisé dans cette simulation pour illustrer un code ayant les éléments du syndrome presque indépendants et ceci pour pouvoir valider nos résultats théoriques qui ont été effectués dans le cas d'éléments du syndrome indépendants. Les courbes correspondant à l'étude théorique sont obtenues en générant des variables aléatoires suivant les mêmes distributions de probabilité de (1.35) et (1.36) pour l'algorithme du HSS et de (1.44) et (1.45) pour le SSS, puis en calculant la probabilité de fausse synchronisation.

Les résultats théoriques et simulés des deux approches HSS et SSS sont affichés respectivement dans les figures 1.6 et 1.7 où sont tracées les courbes de probabilité de fausse synchronisation en fonction du rapport  $E_b/N_0$ . Il est clair sur ces figures que les résultats théoriques et simulés sont très proches. Pour Code II, les résultats théoriques sont un peu meilleurs que ceux simulés. Ceci est dû au fait que pour ce code, chaque ligne de sa matrice de contrôle de parité contient 10 éléments non nuls, ce qui diminue le nombre d'éléments du syndrome indépendants. Par contre, dans nos calculs théoriques

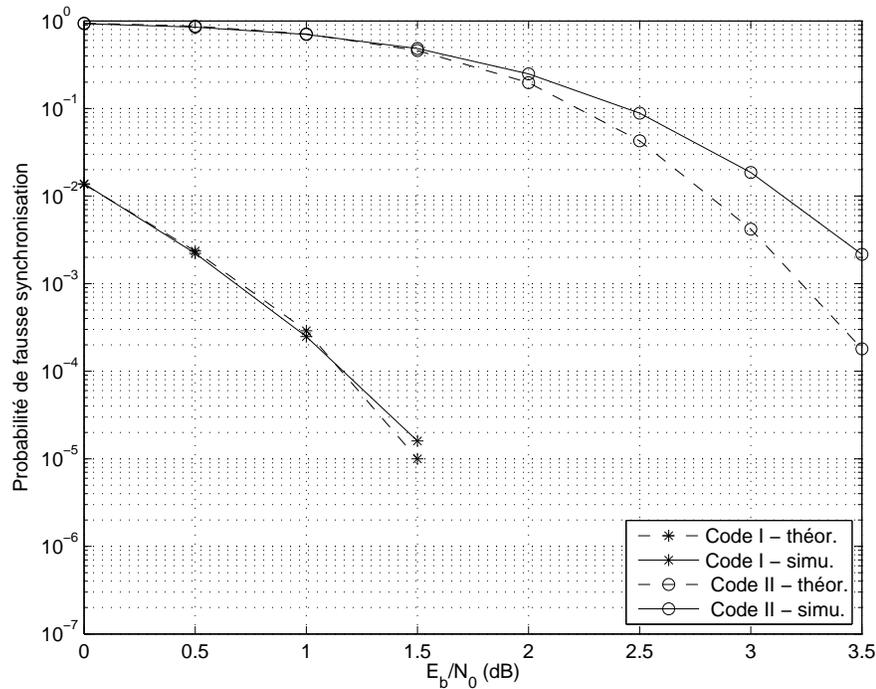


Figure 1.7 — Validation de l'étude théorique de l'approche SSS.

nous avons supposé l'indépendance de tous les éléments du syndrome, ce qui explique cette différence.

## 1.4 Propriétés de l'algorithme du SSS

Comme notre critère de synchronisation est basé sur la minimisation de (1.41), il est évident que notre méthode de synchronisation est plus performante quand la différence entre les distributions de  $\hat{\phi}(t)_{t \neq t_0}$  et  $\hat{\phi}(t_0)$  est plus grande. Cette différence est directement liée à la différence entre  $Pr[S_t(k) = 1]_{t \neq t_0}$  et  $Pr[S_{t_0}(k) = 1]$ . Les valeurs de  $Pr[S_t(k) = 1]_{t \neq t_0}$  et  $Pr[S_{t_0}(k) = 1]$  sont données respectivement dans (1.43) et (1.34).

La figure 1.8 montre l'évolution de  $Pr[S_{t_0}(k) = 1]$  en fonction de  $u_k$ , pour différentes valeurs de la probabilité d'erreur du canal,  $p_e$ . Sur cette figure, nous observons que  $Pr[S_{t_0}(k) = 1]$  tend vers  $Pr[S_t(k) = 1]_{t \neq t_0} = 1/2$  quand  $u_k$  tend vers l'infini. Nous concluons alors que notre méthode de synchronisation est plus performante pour des petites valeurs de  $u_k$ .

Cependant, la condition sur la différence entre  $Pr[S_t(k) = 1]_{t \neq t_0}$  et  $Pr[S_{t_0}(k) = 1]$  n'est pas suffisante pour garantir de bonnes performances de synchronisation [29]. Une autre condition est l'indépendance des éléments  $S_t(k)$ ,  $k = 1 \dots n_r$ . Dire que deux éléments d'un syndrome sont indépendants est équivalent à dire que deux lignes de

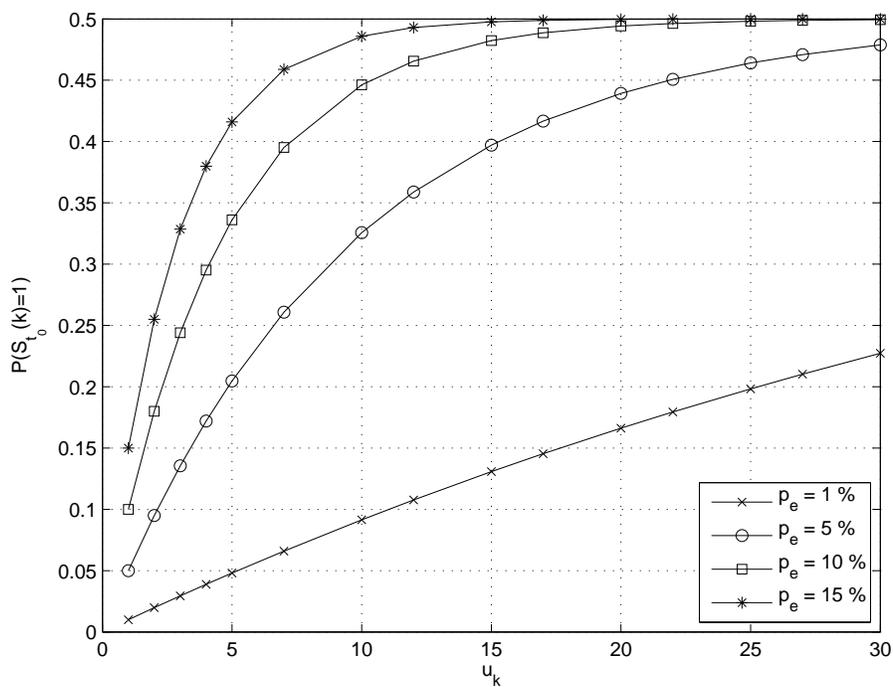


Figure 1.8 —  $Pr[S_{t_0}(k) = 1]$  en fonction de  $u_k$  pour différentes valeurs de BER.

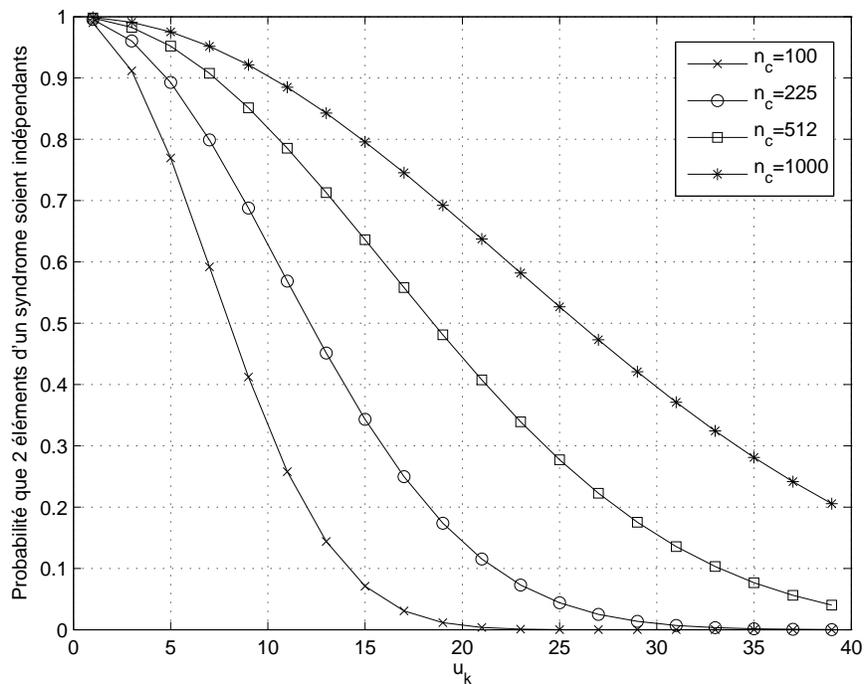
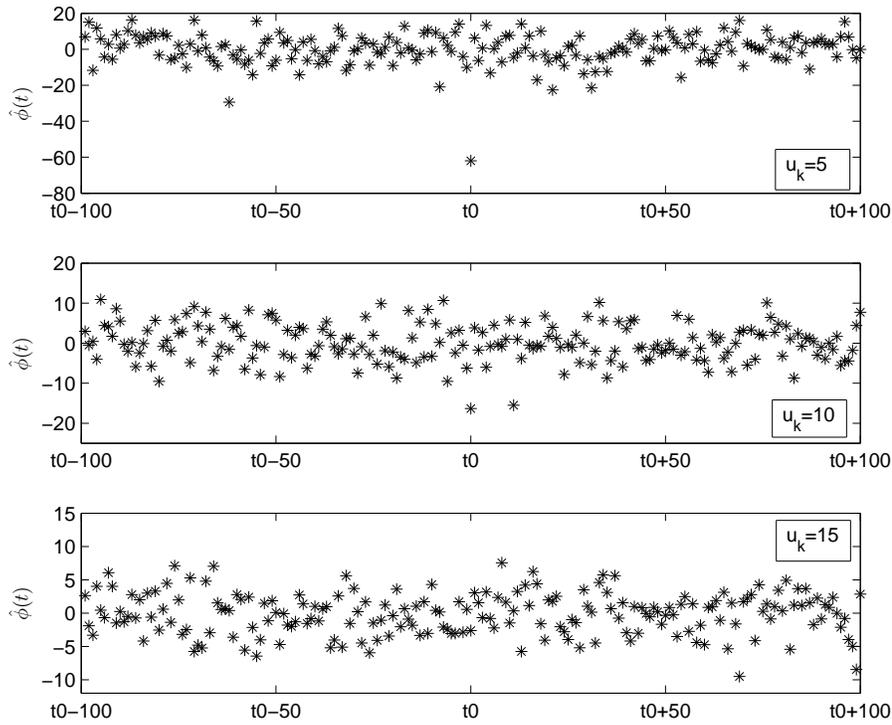


Figure 1.9 — Probabilité que deux éléments d'un syndrome soient indépendants



**Figure 1.10** — Le critère de synchronisation  $\hat{\phi}(t)$  tracé en fonction de la position  $t$  de la fenêtre de synchronisation.

la matrice de contrôle de parité  $H$  n'ont aucun élément non nul aux mêmes positions (i.e colonnes). Appelons  $P_{ind}$  cette probabilité. En supposant que  $u_k$  est constant pour toutes les lignes de  $H$ , nous avons :

$$P_{ind} = \frac{\binom{n_c - u_k}{u_k}}{\binom{n_c}{u_k}}. \quad (1.52)$$

La figure 1.9 représente la probabilité que deux éléments d'un syndrome soient indépendants. Cette probabilité a été tracée en fonction de  $u_k$ . Il est clair que la probabilité d'indépendance de deux éléments d'un syndrome est d'autant plus grande que le nombre d'éléments non nuls dans chaque ligne de  $H$  est plus petit et la longueur d'un mot de code plus grande.

Afin de valider l'étude effectuée dans ce paragraphe, nous traçons sur la figure 1.10 des valeurs du critère  $\hat{\phi}(t)$  que nous cherchons à minimiser pour estimer l'instant de synchronisation. Ces valeurs sont tracées en fonction de la position  $t$  de la fenêtre de synchronisation à un rapport  $E_b/N_0 = 3 \text{ dB}$  et ceci pour des différentes valeurs de  $u_k$ . Le code correcteur d'erreurs sur lequel notre méthode de synchronisation trame est appliquée est un code LDPC de longueur  $n_c = 512$  bits et rendement  $R = 0.5$ . Pour  $u_k = 5$  par exemple, il est clair que  $\hat{\phi}(t)$  a une seule valeur minimale et ceci pour  $t = t_0$ . En outre, la différence entre  $\hat{\phi}(t_0)$  et  $\hat{\phi}(t)_{t \neq t_0}$  étant assez grande, ceci garantit

une bonne estimation de l'instant de synchronisation. Cependant, plus  $u_k$  augmente, plus il est difficile de faire la discrimination entre  $\hat{\phi}(t_0)$  et les autres  $\hat{\phi}(t)_{t \neq t_0}$ . Pour  $u_k = 15$ , la figure 1.10 montre que le minimum de  $\hat{\phi}(t)$  est atteint pour une valeur de  $t \neq t_0$ . La synchronisation trame suivant l'algorithme du SSS est alors impossible dans ce cas.

En conclusion, les codes correcteurs d'erreurs ayant une matrice de contrôle de parité creuse sont de bons candidats pour la méthode de synchronisation trame du SSS. Ceci est le cas des codes LDPC [20, 51].

## 1.5 Conclusion

Nous avons traité dans ce chapitre du problème de la synchronisation trame. Après avoir présenté quelques méthodes existant dans la littérature, nous avons proposé une nouvelle méthode de synchronisation trame aveugle (algorithme du SSS), fondée sur un critère MAP et appliquée à des systèmes utilisant des codes correcteurs d'erreurs. Nous avons aussi effectué une étude théorique concernant la méthode en question en trouvant les distributions de probabilité suivies par le critère de synchronisation et donnant une équation analytique de la probabilité de fausse synchronisation. Nous avons déterminé également les propriétés de la méthode du SSS et démontré que cette méthode de synchronisation trame est plus performante quand des codes ayant des matrices de contrôle de parité creuses sont utilisés dans le système.

Dans le chapitre suivant, nous analysons les performances de la méthode du SSS en l'appliquant à différents types de codes correcteurs d'erreurs.



## Chapitre 2

# Performances et comportement des méthodes de synchronisation trame appliquées à différentes familles de codes correcteurs d'erreurs

DANS ce chapitre, nous appliquons les méthodes de synchronisation trame décrites dans le chapitre précédent à plusieurs types de codes correcteurs d'erreurs. Dans le paragraphe 2.1, nous considérons le cas des codes LDPC définis par une matrice de contrôle de parité creuse. Pour ce type de codes correcteurs d'erreurs, nous comparons les performances de l'algorithme du SSS proposé à celles de plusieurs méthodes existantes et nous montrons que notre méthode de synchronisation trame aveugle présente les meilleures performances en termes de probabilité de fausse synchronisation. Le paragraphe 2.2 est consacré aux codes convolutifs qui peuvent être aussi définis par des matrices de contrôle de parité creuses. Appliquée à ce type de codes correcteurs d'erreurs, la méthode de synchronisation proposée se montre très performante. Ensuite, nous considérons au paragraphe 2.3 le cas de systèmes utilisant des codes BCH et des codes produits. Ces codes sont caractérisés par des matrices de contrôle de parité non creuses. Pour cela, nous proposons de faire évoluer notre technique de synchronisation trame d'origine afin qu'elle fonctionne aussi pour des codes ayant des matrices de contrôle de parité non creuses. Cette méthode, que nous appelons m-SSS (modified Soft Syndrome based Synchronization), est basée sur l'adaptation de la matrice de contrôle de parité aux échantillons reçus. Les résultats des simulations sont très encourageants. Ceci nous a permis de publier cette méthode originale de synchronisation trame dans la conférence ICC 2009. Enfin, dans le paragraphe 2.4, nous présentons une généralisation de notre méthode de synchronisation trame proposée pour qu'elle soit applicable à des systèmes utilisant des modulations à grand nombre d'états. Une expression analytique du taux d'erreur trame calculé après synchronisation et décodage est donnée au paragraphe 2.5. Finalement, le paragraphe 2.6 conclut le chapitre.

Tous les résultats des simulations affichés dans ce chapitre ont été obtenus par simulations de Monte Carlo. Jusqu'à 1000000 de réalisations ont été effectuées, où à chaque réalisation, le bruit, les bits d'information et le retard du canal ont été choisis

aléatoirement.

Notons aussi que dans toutes les simulations effectuées dans ce document, le rapport entre l'énergie binaire et la densité spectrale de puissance de bruit  $E_b/N_0$  a été défini en tenant compte du rendement du code correcteur d'erreurs. Si une séquence d'apprentissage est utilisée à la transmission (cas de la méthode de Massey par exemple), le nombre de bits non utiles insérés est aussi pris en considération dans la définition du  $E_b/N_0$ .

## 2.1 Codes LDPC

### 2.1.1 Définition d'un code LDPC

Les codes LDPC (Low Density Parity Check codes) ont été introduits au début des années 60 par Gallager [20]. Ils ont été oubliés pendant quasiment 30 ans et ce n'est qu'après l'invention des turbocodes par Berrou et al. [7], que ces codes furent redécouverts dans les années 90 par MacKay et Neal [51]. Les codes LDPC constituent une nouvelle famille de codes en blocs linéaires caractérisés par leur matrice de contrôle de parité creuse. Un code LDPC peut également être représenté par son graphe factoriel [44]. Un graphe factoriel est un graphe bipartite contenant deux types de noeuds : les noeuds de données représentant le mot de code et les noeuds de parité représentant les fonctions de vérification de parités.

On distingue deux types de codes LDPC :

- *Les codes LDPC réguliers* : Un code LDPC est dit régulier si sa matrice de contrôle de parité  $H$  contient un nombre constant de 1 dans chaque ligne et un nombre constant de 1 dans chaque colonne. Ces nombres sont notés respectivement  $d_c$  et  $d_v$ . Le rendement d'un code LDPC régulier vérifie la relation suivante :

$$R \geq 1 - \frac{d_v}{d_c}. \quad (2.1)$$

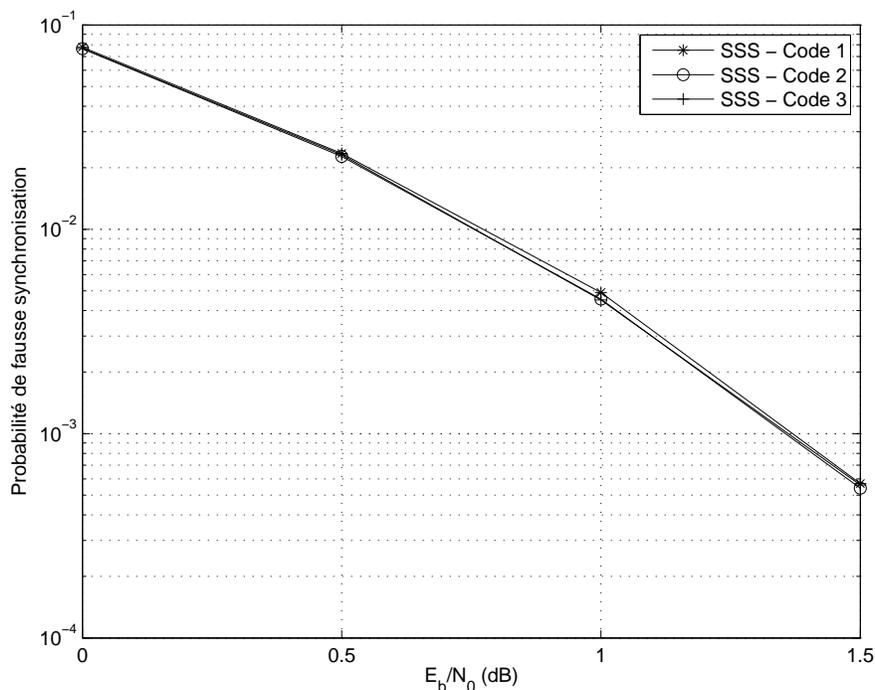
- *Les codes LDPC irréguliers* : C'est une classe plus générale des codes LDPC qui a été introduite par Luby et al. [47]. Les codes LDPC irréguliers sont caractérisés par une distribution non uniforme du nombre de branches connectées aux différents types de noeuds sur le graphe factoriel. L'irrégularité d'un code LDPC est spécifiée par deux polynômes :

$$\lambda(x) = \sum_i \lambda_i x^{i-1} \quad \text{et} \quad \rho(x) = \sum_j \rho_j x^{j-1}, \quad (2.2)$$

où  $\lambda_i$  (respectivement  $\rho_j$ ) est la proportion des branches connectées aux noeuds de données (respectivement noeuds de parité) de degré  $i$  (respectivement degré  $j$ ). Ces deux polynômes sont liés au rendement  $R$  du code par :

$$R \geq 1 - \frac{\sum_j \rho_j / j}{\sum_i \lambda_i / i}. \quad (2.3)$$

Les égalités dans (2.1) et (2.3) sont atteintes si la matrice de contrôle de parité du code LDPC est de rang plein.



**Figure 2.1** — Performances de la méthode du SSS appliquée à trois codes LDPC différents ayant la même longueur, le même rendement et le même nombre d'éléments non nuls dans chaque ligne de leur matrice de contrôle de parité.

Le décodage des codes LDPC s'effectue par un algorithme itératif dit algorithme à propagation de croyance (ou BP pour Belief Propagation). Comme son nom l'indique, cet algorithme est basé sur la propagation de messages tout au long du graphe factoriel associé au code. Le principe de propagation de croyance est l'application de la règle de Bayes sur chaque bit du mot de code, d'une façon itérative, afin d'estimer les probabilités *a posteriori* de chaque bit [44].

Durant les dernières années, de nombreuses études ont été menées pour optimiser la construction et le décodage des codes LDPC. Ceci a permis à ces codes d'atteindre des probabilités d'erreur très faibles pour des SNR assez bas. Les codes LDPC sont alors introduits dans plusieurs applications pratiques.

Les critères de synchronisation étudiés dans cette thèse sont directement liés au nombre d'éléments non nuls dans chaque ligne de la matrice de contrôle de parité du code correcteur d'erreurs et ne dépendent pas de leurs performances de décodage. Pour illustrer cette proposition, nous appliquons sur la figure 2.1 la méthode du SSS à trois codes LDPC différents, ayant la même taille ( $n_c = 512$  bits), le même rendement ( $R = 0.5$ ) et le même nombre d'éléments non nuls dans chaque ligne de la matrice de contrôle de parité  $H$  ( $u_k = 4$ ). Cependant, comme les matrices  $H$  de ces codes sont générées aléatoirement, chaque code LDPC parmi ces trois a une matrice de contrôle de parité différente. Comme nous l'observons sur la figure 2.1, les performances de la méthode du SSS appliquée aux trois codes LDPC sont les mêmes, ce qui n'est pas

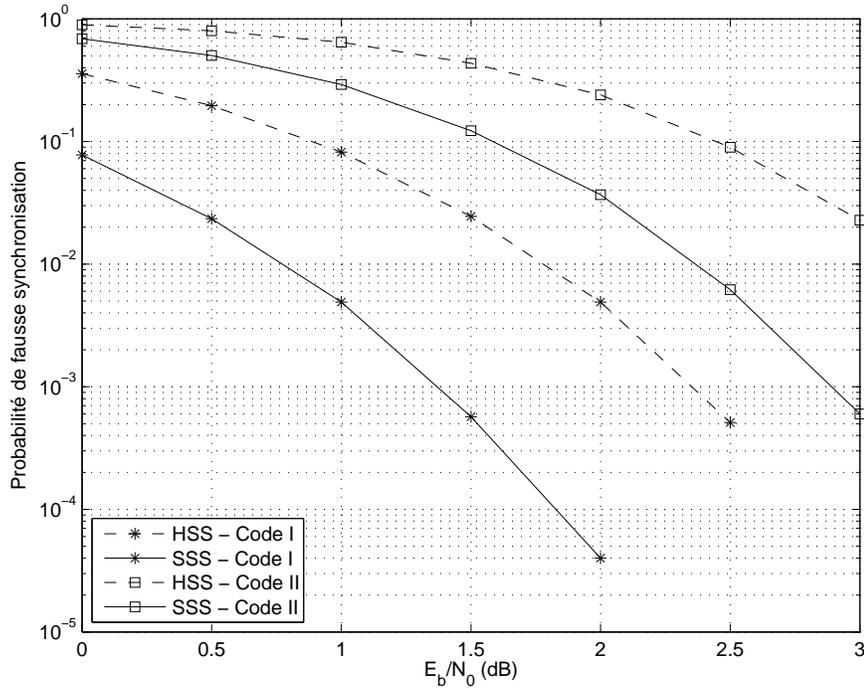


Figure 2.2 — Comparaison entre les performances des deux méthodes HSS et SSS.

forcément le cas pour les performances de décodage de ces codes.

Ainsi, dans la suite de ce document, et sauf indication contraire, les codes LDPC sur lesquels nous appliquons nos techniques de synchronisation ne sont pas forcément des codes LDPC optimisés. On les définit par la longueur  $n_c$ , le rendement  $R$  et l'entier  $u_k$  désignant le nombre de 1 dans chaque ligne de  $H$ .

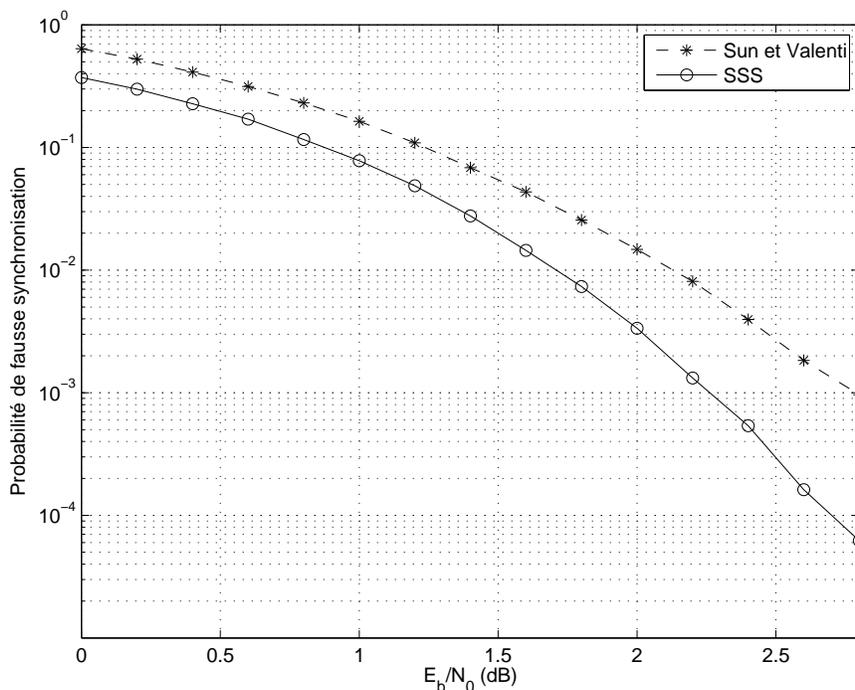
## 2.1.2 Synchronisation des codes LDPC

### Comparaison entre les méthodes HSS et SSS

Nous effectuons dans ce paragraphe une comparaison entre les performances des deux approches de synchronisation présentées dans les paragraphes 1.1.1 (algorithme du HSS) et 1.2 (algorithme du SSS).

La figure 2.2 montre cette comparaison quand deux codes LDPC de longueur  $n_c = 512$  bits et rendement  $R = 0.5$  sont utilisés. Ces codes, notés Code I et Code II, ont respectivement 4 et 6 éléments non nuls dans chaque ligne de leur matrice de contrôle de parité. Sur cette figure, il est clair que la méthode du SSS, basée sur un critère MAP, est plus performante que le HSS. Pour  $u_k = 4$  et une probabilité de fausse synchronisation égale à  $10^{-2}$ , une différence d'environ 1 dB existe entre les deux méthodes.

En comparant maintenant les courbes tracées pour les codes I et II des deux cas HSS et SSS, nous remarquons que les performances de ces méthodes de synchronisation se dégradent lorsque le nombre d'éléments non nuls dans la matrice de contrôle de parité



**Figure 2.3** — Comparaison entre la méthode de synchronisation du SSS et celle de Sun et Valenti.

$H$  augmente. Ceci valide les résultats de l'étude effectuée au paragraphe 1.4 où nous avons montré que ces techniques de synchronisation aveugle sont plus performantes quand elles sont appliquées à des codes correcteurs d'erreurs ayant des matrices de contrôle de parité creuses.

### Comparaison entre la méthode de synchronisation du SSS et celle de Sun et Valenti

En 2004, Sun et Valenti ont proposé une méthode de synchronisation aveugle appliquée aux codes LDPC [64]. Cette méthode est décrite au paragraphe 1.1.2. Effectuons maintenant une comparaison entre les performances de la méthode proposée par Sun et Valenti et celle que nous proposons au paragraphe 1.2 (algorithme du SSS). Le code testé est le code LDPC de taille  $n_c = 512$  bits, rendement  $R=0.5$  et ayant  $u_k = 5$  éléments non nuls dans chaque ligne de sa matrice de contrôle de parité. A partir de la figure 2.3, nous observons que la méthode de synchronisation trame du SSS est plus performante que celle de Sun et Valenti. Une différence d'environ  $0.5$  dB est atteinte pour une probabilité de fausse synchronisation égale à  $10^{-3}$ . Cette différence peut être vitale pour des applications nécessitant de bonnes performances de décodage à des faibles SNR.

Les codes LDPC utilisés dans les simulations précédentes ne sont pas forcément des codes à grand pouvoir correcteur d'erreurs. En effet, pour que la méthode de

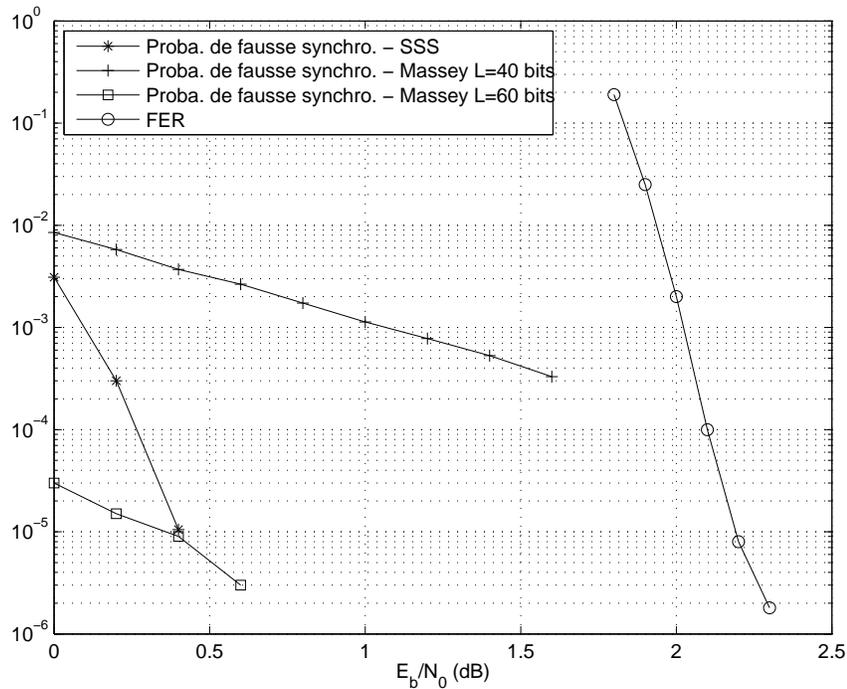


Figure 2.4 — Synchronisation trame du code LDPC utilisé dans le standard DVB-S2.

synchronisation aveugle ait un sens, il faut être capable de synchroniser à un SNR plus faible ou égal au point de fonctionnement du code. Dans le paragraphe suivant, nous testons l'efficacité de notre méthode de synchronisation trame après l'avoir appliquée à un code LDPC optimisé.

### Synchronisation d'un code LDPC optimisé

Traçons sur la figure 2.4 les performances de notre méthode de synchronisation appliquée au le code LDPC utilisé dans le codage canal du standard DVB-S2 (Digital Video Broadcasting-Satellite, Second Generation), utilisé pour la transmission du contenu multimédia par satellite. Le code LDPC en question est un code de rendement  $R = 2/3$  et de taille  $n_c = 16200$  bits. Sur cette même figure, nous traçons aussi la courbe de FER (Frame Error Rate) du code LDPC. Cette courbe fournie par la société TurboConcept [1] est obtenue après 50 itérations du décodeur itératif dans le cas d'une synchronisation parfaite. Le grand écart entre ces deux courbes (environ 2 dB) et l'allure de la courbe de la probabilité de fausse synchronisation, montrent que la synchronisation aveugle n'entraînera pas de perte significative de performances en termes de FER.

Cependant, calculer le LLR du syndrome à partir des 10800 équations de parité pour appliquer ainsi l'algorithme du SSS, s'avère d'une complexité importante et nécessite un temps de calcul assez long. De plus, comme le code LDPC utilisé dans ce cas a une grande taille, le rajout d'une séquence d'apprentissage pour effectuer

la synchronisation n'aura pas beaucoup d'impact sur l'efficacité spectrale de la transmission. En effet, nous avons appliqué la méthode de synchronisation de Massey sur un système utilisant ce même code LDPC et les résultats sont affichés sur la figure 2.4. Il est clair sur cette figure que pour une séquence d'apprentissage de taille  $L = 60$  bits, de très bonnes performances de synchronisation sont atteintes. Toutefois, comme la longueur du code est de 16200 bits, les 60 bits rajoutés ne constituent que 0.37% de la longueur totale du code. Ainsi, la perte en efficacité spectrale due à l'insertion de ces bits pilotes est négligeable. C'est pourquoi dans le cas de codes longs, on préférera certainement utiliser les méthodes classiques de synchronisation, telle que la méthode de synchronisation non aveugle de Massey.

Appliquée aux codes LDPC, la méthode de synchronisation proposée a montré son efficacité comparée à d'autres méthodes de synchronisation. Ces mêmes comparaisons sont effectuées au paragraphe suivant mais cette fois-ci pour des systèmes utilisant les codes convolutifs comme codes correcteurs d'erreurs.

## 2.2 Codes convolutifs

### 2.2.1 Définition d'un code convolutif

Inventés par Elias en 1954 [18], les codes convolutifs forment une famille de codes correcteurs d'erreurs très utilisés en pratique vu la simplicité de leur codage/décodage et leurs bonnes performances. Le principe des codes convolutifs se base sur le codage de  $n_b$  bits d'information en  $n_c$  bits codés. Contrairement aux codes en blocs, ces  $n_c$  bits ne dépendent pas seulement du bloc des  $n_b$  bits d'information à l'entrée du codeur, mais aussi des  $l$  blocs précédents. Le codeur est réalisé alors à l'aide d'un registre à décalage en mémorisant toujours  $l$  blocs d'information.

Un code convolutif est défini par son rendement  $R = n_b/n_c$  et ses polynômes générateurs qui fournissent la logique combinatoire reliant les différentes mémoires du registre. On définit aussi la longueur de contrainte du code donnée par  $\nu = l + 1$ . Dans la littérature, on parle souvent de trois principales méthodes de représentation d'un code convolutif. Le diagramme d'états, le diagramme en treillis et le diagramme en arbre. Un code convolutif peut aussi être défini par une matrice de contrôle de parité spécifique. Ceci est effectué en passant par les représentations algébriques de ces codes comme indiqué dans [41].

Plusieurs algorithmes de décodage existent pour décoder les codes convolutifs, dont les plus répandus sont l'algorithme de Viterbi et l'algorithme MAP (Maximum A Posteriori). Ce dernier est connu sous le nom BCJR (pour les noms de ces inventeurs Bahl, Cocke, Jelinek et Raviv) [3]. L'algorithme de Viterbi consiste à rechercher la séquence des blocs d'information la plus vraisemblable, tandis que l'algorithme MAP détermine à chaque instant le symbole d'information le plus vraisemblable. Ainsi, un décodeur se basant sur l'algorithme de Viterbi garantit une probabilité d'erreurs minimale sur les séquences décodées, alors qu'avec le MAP, on minimise la probabilité d'erreurs sur les symboles d'information.



Or en absence de bruit,  $Y_t(k) = 1$ . Ainsi, à la position de synchronisation nous avons :

$$\hat{\phi}(t_0) = -n_r. \quad (2.9)$$

Dans le cas général où la matrice de contrôle de parité n'a pas une structure particulière et selon (1.43) :

$$Pr[S_t(k) = 0] = Pr[S_t(k) = 1] = 1/2, \quad t \neq t_0 \quad (2.10)$$

D'où :

$$E[\hat{\phi}(t)] = 0, \quad t \neq t_0 \quad (2.11)$$

où  $E[.]$  désigne l'espérance mathématique. Cette différence entre  $\hat{\phi}(t_0)$  (Eq. (2.9)) et  $\hat{\phi}(t)_{t \neq t_0}$  (Eq. (2.11)) garantit de bonnes performances pour la méthode du SSS. Cependant, pour les codes convolutifs (ou les codes en blocs cycliques) et à cause de la structure particulière de leur matrice de contrôle de parité  $H$ , (2.11) n'est plus vérifiée pour toutes les valeurs de  $t \neq t_0$ . En effet :

Supposons maintenant que le code utilisé est le code convolutif (1,5/7) et que nous sommes à la position  $t = t_0 + 2n$ , où  $n$  est un entier naturel quelconque. En s'appuyant sur (2.8) et (2.6), calculons la valeur du critère  $\hat{\phi}(t)$  pour cette position de non synchronisation :

$$\begin{aligned} \hat{\phi}(t) &= \sum_{k=1}^{n_r} (2S_t(k) - 1) \\ &= \sum_{k=1}^{n_r} (2S_{t_0}(k+n) - 1) \\ &= \sum_{i=n+1}^{n_r} (2S_{t_0}(i) - 1) \\ &= -(n_r - n) \end{aligned} \quad (2.12)$$

En comparant maintenant (2.9) et (2.12), nous remarquons que la différence entre  $\hat{\phi}(t_0)$  et  $\hat{\phi}(t)_{t \neq t_0}$  n'est plus si grande et elle est d'autant plus petite que nous nous approchons de la bonne position de synchronisation (que  $n$  diminue). Ceci entraîne une confusion sur la bonne position de synchronisation et par suite, une dégradation dans les performances de la méthode de synchronisation trame du SSS.

La solution la plus simple au problème mis en évidence dans ce paragraphe est de rompre cette structure spécifique des matrices de contrôle de parité de ce type de codes. Ceci est effectué en utilisant un entrelaceur pseudo-aléatoire de taille  $n_c$  bits à l'émission qui va permuter les bits dans chaque mot de code, et par conséquent, les colonnes de la matrice  $H$ .

### Méthodes du HSS et SSS appliquées aux codes convolutifs

Appliquons maintenant les méthodes HSS et SSS présentées respectivement aux paragraphes 1.1.1 et 1.2 du chapitre 1, à des systèmes utilisant des codes convolutifs.

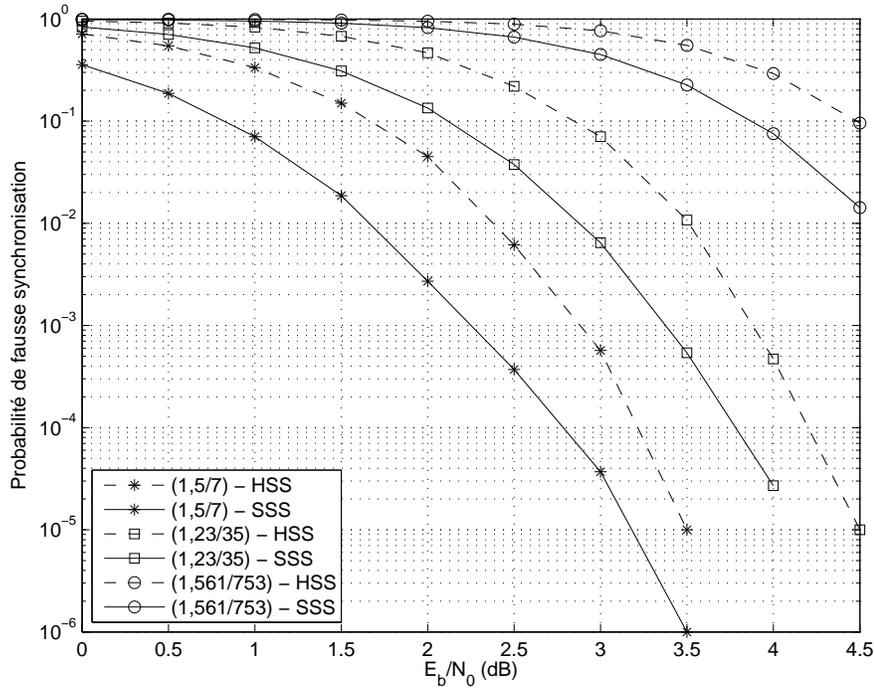


Figure 2.5 — Les méthodes HSS et SSS appliquées aux codes convolutifs

Les codes convolutifs que nous avons testés sont des codes systématiques et récurrents, de polynômes générateurs  $(1, 5/7)$ ,  $(1, 23/35)$  et  $(1, 561/753)$ . Ces codes ont la même taille ( $n_c = 512$  bits), le même rendement ( $R = 0.5$ ) et des longueurs de contrainte égales respectivement à 3, 5 et 9.

Il est clair sur la figure 2.5 que, quel que soit le code convolutif utilisé, la méthode du SSS présente toujours de meilleures performances que la technique du HSS. Pour le code convolutif  $(1, 5/7)$  et à une probabilité de fausse synchronisation égale à  $2 \cdot 10^{-2}$ , on note une différence d'environ  $0.7$  dB entre les deux méthodes.

Notons aussi que le nombre d'éléments non nuls dans la matrice de contrôle de parité augmente avec la longueur de contrainte du code. Il est donc prévisible que les performances de notre méthode de synchronisation trame appliquée au code convolutif  $(1, 5/7)$  soient meilleures que celles des codes  $(1, 23/35)$  et  $(1, 561/753)$ . Ceci est également confirmé par les courbes tracées sur la figure 2.5.

Ces résultats ont été présentés à la conférence EUSIPCO (European Signal Processing Conference) en Septembre 2007 [34].

### Méthodes de synchronisation du SSS et de Sun et Valenti appliquées aux codes convolutifs

Dans ce paragraphe, nous appliquons la méthode de Sun et Valenti expliquée au paragraphe 1.1.2 aux codes convolutifs et nous comparons ses performances à celles de l'algorithme du SSS. Les courbes de la probabilité de fausse synchronisation en fonction du  $E_b/N_0$  sont tracées sur la figure 2.6 pour les codes convolutifs  $(1, 5/7)$  et  $(1, 23/35)$ .

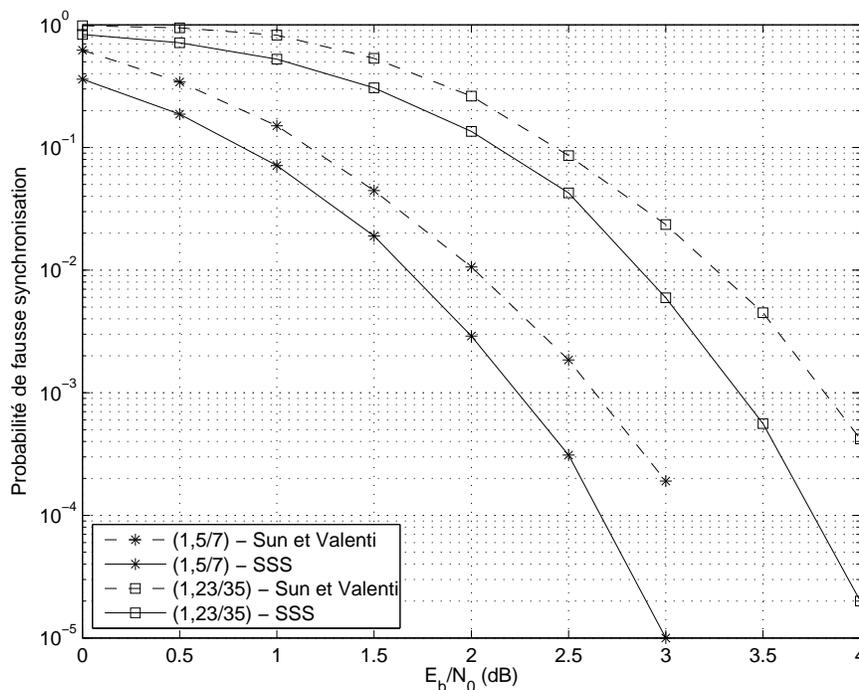


Figure 2.6 — Méthodes de synchronisation du SSS et de Sun et Valenti appliquées aux codes convolutifs.

Le même comportement que celui obtenu avec les codes LDPC est constaté ici : la méthode de synchronisation trame aveugle du SSS présente de meilleures performances que celle de Sun et Valenti. Pour une probabilité de fausse synchronisation égale à  $10^{-3}$ , une différence d'environ  $0.4 \text{ dB}$  existe entre les deux méthodes.

### Méthodes de synchronisation du SSS comparée à la technique de synchronisation non aveugle de Massey

La méthode la plus classique de synchronisation trame est celle décrite par Massey en 1972 [52]. Cette méthode localise une séquence d'apprentissage périodiquement introduite dans les données binaires. Selon Massey, la meilleure règle de synchronisation trame est de sélectionner la position de la fenêtre de synchronisation qui maximise la somme d'une corrélation et d'un terme correcteur. La méthode de synchronisation de Massey est décrite au paragraphe 1.1 du chapitre 1. La figure 2.7 montre une comparaison entre la méthode de synchronisation du SSS et la méthode classique de Massey. Ces techniques de synchronisation ont été appliquées au code convolutif  $(1, 5/7)$  de taille  $n_c = 512$  bits et de rendement  $R = 0.5$ . Pour la méthode de synchronisation de Massey, trois différentes tailles d'une séquence d'apprentissage ont été testées,  $L = 15, 25$  et  $40$  bits. A partir de la figure 2.7, nous observons que la méthode du SSS présente de meilleures performances que celle introduite par Massey. Même avec une séquence d'apprentissage de taille  $L = 40$  bits, la méthode que nous proposons est plus performante pour des  $E_b/N_0 > 3.5 \text{ dB}$ , zone dans laquelle les codes convolutifs sont généralement

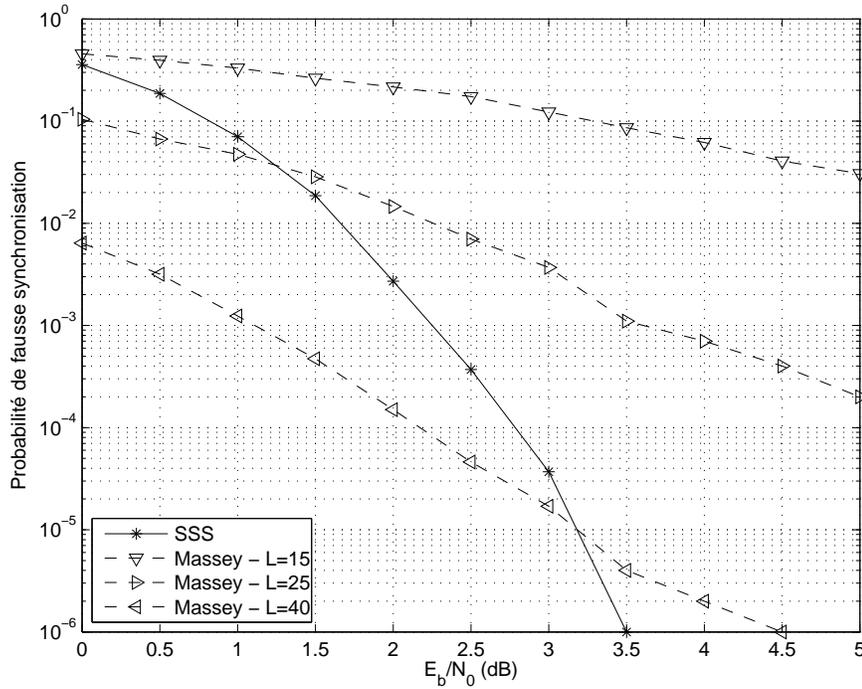


Figure 2.7 — Comparaison entre la méthode de synchronisation de Massey et l'algorithme du SSS appliqués au code convolutif (1, 5/7).

utilisés. Notons que  $L = 40$  bits est équivalent à 8% de la taille du code. Ainsi, en appliquant la méthode proposée de synchronisation aveugle, un gain d'environ 8% en termes d'efficacité spectrale est atteint.

### Performances du décodeur MAP après application de la méthode de synchronisation du SSS

Afin d'évaluer l'efficacité de la méthode de synchronisation du SSS, nous traçons sur la figure 2.8 les courbes de FER obtenues après avoir décodé les codes (1, 5/7) et (1, 23/35) utilisés précédemment par le décodeur MAP. Deux cas sont considérés : dans le premier, nous supposons une synchronisation parfaite tandis que dans le deuxième cas, la position de synchronisation est estimée en utilisant la méthode du SSS. Sur cette figure, les courbes de FER du code (1, 5/7) sont à peu près les mêmes. Pour le code (1, 23/35), une petite dégradation est observée entre le cas de la synchronisation parfaite et après application de la méthode de synchronisation proposée. Mais à partir de  $E_b/N_0 > 3.5$  dB, les deux courbes de FER se confondent et aucune dégradation dans les performances n'est détectée. Sachant que les codes convolutifs fonctionnent à de tels  $E_b/N_0$ , nous concluons que la méthode de synchronisation trame du SSS est bien adaptée à ce type de codes.

Les systèmes que nous avons synchronisés jusqu'à maintenant utilisent des codes ayant des matrices de contrôle de parité plus ou moins creuses. Cependant, il existe

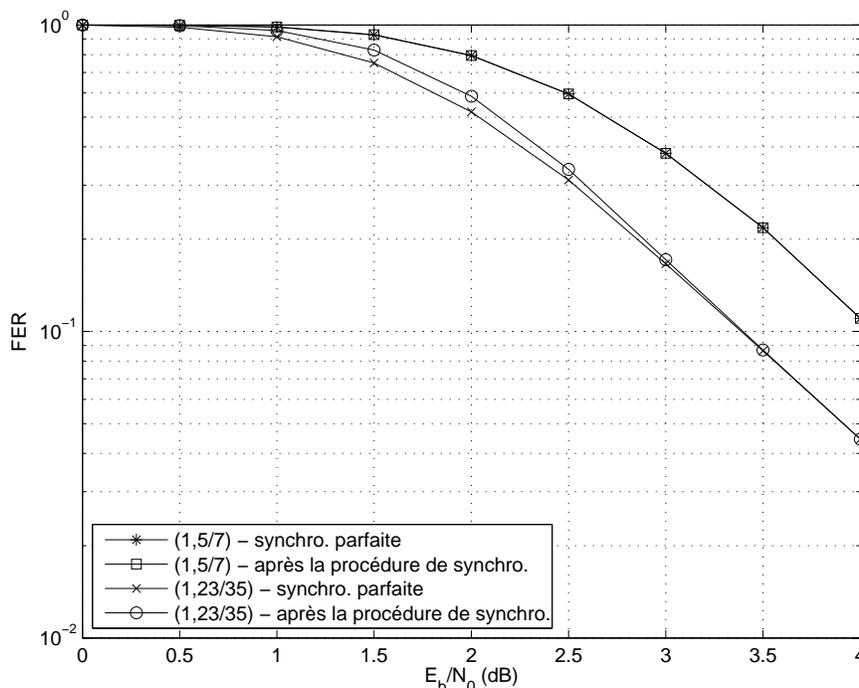


Figure 2.8 — Courbes de FER des codes convolutifs de rendement  $R = 1/2$  et taille  $n_c = 512$  bits.

de nombreux codes correcteurs d'erreurs utilisés dans la pratique et qui n'ont pas de matrices de contrôle de parité creuses, tels que les codes BCH et les codes produits. Le problème de la synchronisation de ces codes est considéré dans le paragraphe suivant.

## 2.3 Codes BCH et codes produits

### 2.3.1 Définitions générales

Les techniques de synchronisation trame aveugle que nous proposons dans ce paragraphe sont spécialement conçues pour les codes en blocs ayant des matrices de contrôle de parité non creuses, comme les codes BCH et les codes produits. Pour mieux comprendre la structure de ces codes et les techniques utilisées pour leur codage/décodage, nous donnons les définitions ci-dessous :

*Codes cycliques* : Les codes cycliques représentent la classe la plus importante des codes en blocs linéaires. Un code en bloc linéaire est cyclique si toute permutation circulaire à droite de  $j$  symboles d'un mot de code redonne un mot de code.

*Codes étendus* : A partir d'un code en bloc  $C(n_c, n_b)$ , on peut construire un code  $C(n_c + 1, n_b)$  en ajoutant à la fin de chaque mot de code un symbole égal à 1 (respectivement à 0) si le mot de code comprend un nombre impair (respectivement pair) de

1. Ce nouveau code est appelé code étendu et sa distance minimale est augmentée de 1 par rapport au code initial. La matrice de contrôle de parité d'un code étendu s'écrit sous la forme :

$$H_e = \begin{pmatrix} & & & 0 \\ & H & & \vdots \\ & & & 0 \\ 1 & \dots & 1 & 1 \end{pmatrix} \quad (2.13)$$

où  $H$  est la matrice de contrôle de parité du code  $C(n_c, n_b)$ .

*Codes BCH* : Les codes BCH qui tiennent leur nom des initiales de leurs inventeurs Bose, Chaudhuri et Hocquenghem, sont des codes cycliques capables de corriger  $t$  erreurs dans un bloc de  $n_c$  symboles. En d'autres termes, ce sont des codes construits de manière à ce que leur distance minimale soit supérieure ou égale à  $2t + 1$ .

Deux types de décodage peuvent être utilisés pour décoder les codes BCH. Un décodage ferme (décodage à maximum de vraisemblance *a posteriori* ou bien décodage à partir du syndrome) et un décodage pondéré (décodage à maximum de vraisemblance *a posteriori* ou bien décodage par l'algorithme de Chase [12]).

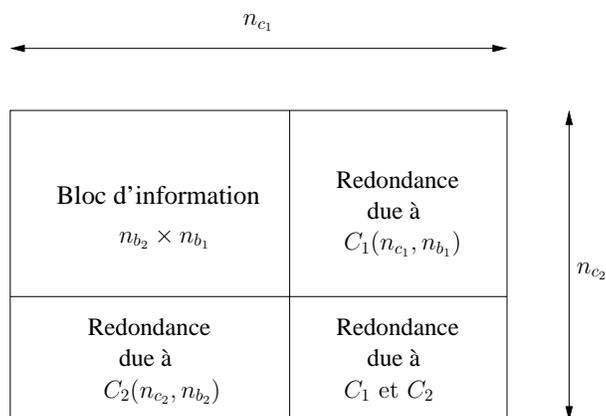
*Codes Produits* : L'invention des codes produits est due à Elias en 1954. Un code produit est un code multidimensionnel construit à partir de codes élémentaires [18]. Soient  $C_1(n_{c_1}, n_{b_1})$  et  $C_2(n_{c_2}, n_{b_2})$  deux codes en blocs linéaires et systématiques de longueurs respectives  $n_{c_1}$  et  $n_{c_2}$  et dimensions respectives  $n_{b_1}$  et  $n_{b_2}$ . La dimension d'un code représente la taille du bloc de données contenu dans ce code. A partir des codes élémentaires  $C_1(n_{c_1}, n_{b_1})$  et  $C_2(n_{c_2}, n_{b_2})$ , nous construisons le code produit  $C(n_{c_1}n_{c_2}, n_{b_1}n_{b_2})$  comme schématisé par la figure 2.9 et décrit ci-dessous :

$n_{b_2}$  blocs de données de dimension  $n_{b_1}$  bits chacun sont codés par le codeur  $C_1$ .  $n_{b_2}$  mots de code de longueur  $n_{c_1}$  bits chacun sont alors obtenus et placés dans un vecteur bidimensionnel. Chaque colonne de ce vecteur est ensuite codée par le codeur  $C_2$  pour produire un mot de code de longueur  $n_{c_2}$  bits. Le résultat final est un code produit de longueur  $n_{c_1} \times n_{c_2}$  et dimension  $n_{b_1} \times n_{b_2}$ .

Les codes produits sont habituellement décodés par l'algorithme de Chase-Pyndiah [61]. C'est un algorithme de décodage fondé sur le critère turbo où les décodeurs lignes et colonnes du code produit échangent entre eux des informations extrinsèques sur les bits.

### 2.3.2 Synchronisation trame par adaptation de la matrice de contrôle de parité - Algorithme du m-SSS

Les codes BCH et les codes produits ont des matrices de contrôle de parité non creuses. Il est donc prévisible que la méthode de synchronisation trame proposée au paragraphe 1.2 (méthode du SSS) ait des difficultés pour synchroniser ces types de codes. Nous proposons dans ce paragraphe d'effectuer quelques modifications à notre critère de synchronisation pour avoir une nouvelle technique de synchronisation trame



**Figure 2.9** — Un Code Produit construit à partir de deux codes en blocs linéaires et systématiques.

aveugle spécialement conçue pour ces types de codes. L'idée principale de cette technique que nous appelons m-SSS (modified Soft Syndrome based Synchronisation) est inspirée de l'algorithme de décodage du m-ABP (Modified Adaptive Belief Propagation) décrit dans la suite.

### Algorithme de décodage du m-ABP

L'algorithme BP est considéré comme étant l'algorithme de référence pour le décodage des codes en blocs linéaires de type LDPC. Cependant, tout comme pour notre algorithme de synchronisation, l'algorithme BP n'est pas approprié si la matrice de contrôle de parité du code en bloc n'est pas creuse, cas des codes BCH, des codes produits et des codes RS. Un algorithme BP adapté, connu sous le nom ABP (Adaptive Belief Propagation), a été récemment proposé pour le décodage des codes RS [39, 40]. Le principe consiste à adapter la matrice de contrôle de parité au cours du processus itératif de l'algorithme BP en fonction de l'évolution de la fiabilité des bits d'information. Cette adaptation a pour but de diminuer le nombre de 1 des colonnes de la matrice associées aux bits d'information les moins fiables. L'algorithme du ABP étant complexe à implémenter, une version modifiée (m-ABP) a été proposée par Jégo et Gross dans [38]. L'algorithme m-ABP appliqué aux codes produits BCH est présenté ci-dessous.

Tout d'abord, les symboles appartenant à chaque ligne (colonne) du code produit reçu sont réarrangés suivant leur fiabilité. En d'autres termes, nous classons ces symboles par leurs valeurs absolues. Ensuite, les  $n_r$  colonnes de la matrice de contrôle de parité  $H$  correspondant aux  $n_r$  symboles les moins fiables sont réduits pour obtenir une matrice identité carrée. Ceci est effectué par la méthode d'élimination de Gauss. L'objectif de cette procédure est de réduire le nombre de 1 dans la partie de la matrice de contrôle de parité qui est associée aux bits les moins fiables. L'étape suivante du décodeur m-ABP est l'algorithme itératif classique du BP : chaque ligne et colonne du code produit est décodée par un décodeur BP où chaque itération est appelée "itération locale". Les décodeurs lignes et colonnes échangent entre eux des informations extrinsèques par un processus itératif. Nous désignons par "itération globale" chaque itération de ce pro-

cessus itératif. Finalement, les décisions fermes sont effectuées à la dernière itération globale.

Une fois appliqué aux produits BCH, l'algorithme du m-ABP est plus performant que le ABP classique et présente les mêmes performances en termes de BER que le Chase-Pyndiah [38]. Dans le paragraphe suivant, nous décrivons comment utiliser une partie de l'algorithme de décodage présenté ci-dessus dans la méthode de synchronisation trame aveugle que nous proposons pour les codes BCH et les codes produits.

### Synchronisation des codes BCH

La méthode de synchronisation que nous proposons dans ce paragraphe est basée sur le même concept que celui de la technique de synchronisation trame aveugle proposée au paragraphe 1.2 du chapitre précédent, mais avec quelques petites modifications. Les codes BCH étant des codes courts, nous devons augmenter la taille de la fenêtre de synchronisation afin d'obtenir des performances acceptables pour notre méthode de synchronisation. Dans le paragraphe 1.2 et pour des raisons de simplicité, nous avons supposé que la fenêtre de synchronisation contient un seul bloc de longueur  $n_c$  bits. Dans le cas des codes BCH, nous utilisons une fenêtre glissante de taille  $Kn_c$  où  $K$  est un entier supérieur ou égal à 1.

A chaque position de la fenêtre glissante et pour chaque bloc de taille  $n_c$  contenu dans cette fenêtre, nous arrangeons ses symboles suivant leur fiabilité puis nous adaptons la matrice de contrôle de parité correspondante, comme décrit au paragraphe précédent. Une fois cette opération effectuée, nous calculons le même critère de synchronisation que celui de (1.29), mais cette fois-ci pour une taille de la fenêtre de synchronisation égale à  $Kn_c$ . La position de synchronisation est alors estimée par :

$$\hat{t}_{mSSS} = \underset{t=0, \dots, n_c-1}{\operatorname{argmin}} \{ \hat{\phi}(t) \} = \underset{t=0, \dots, n_c-1}{\operatorname{argmin}} \left\{ \sum_{k=1}^{Kn_c} \hat{L}(S_t(k)) \right\}. \quad (2.14)$$

### Synchronisation des codes produits

Dans le cas des codes produits, nous profitons de leur structure multidimensionnelle et proposons une méthode de synchronisation spécialement conçue pour ce type de codes. Soit un code produit  $C(n_{c_1}n_{c_2}, n_{b_1}n_{b_2})$  bidimensionnel construit à partir des deux codes élémentaires  $C_1(n_{c_1}, n_{b_1})$  et  $C_2(n_{c_2}, n_{b_2})$ , qui sont des codes en blocs linéaires et systématiques. Le code produit étant de longueur  $n_c = n_{c_1}n_{c_2}$ , nous considérons une fenêtre glissante de taille  $Kn_{c_1}n_{c_2}$ . Pour chaque bloc de taille  $n_{c_1}n_{c_2}$  contenu dans la fenêtre glissante, nous divisons la procédure proposée de synchronisation trame en deux parties.

Tout d'abord, nous arrangeons les symboles dans chaque ligne du code produit suivant leur fiabilité. Chaque ligne permutée entraîne une adaptation de la matrice de contrôle de parité du code  $C_1$ , comme expliqué au paragraphe précédent. Ensuite, pour chaque ligne du code produit reçu, nous calculons les LLR des éléments du syndrome suivant l'équation (1.28), et ceci en utilisant la matrice de contrôle de parité adaptée du code  $C_1$ . Soit  $\hat{\phi}_1(t)$  la somme de ces LLR.

Chaque colonne du code produit étant un mot de code de  $C_2$ , la deuxième partie de notre méthode de synchronisation proposée est identique à la précédente mais cette fois-ci appliquée aux colonnes du code reçu. Le résultat est  $\hat{\phi}_2(t)$ , somme des LLR des éléments du syndrome des colonnes du code, obtenue en utilisant la matrice de contrôle de parité adaptée du code  $C_2$ .

Finalement, nous calculons un nouveau critère de la méthode proposée :

$$\hat{\phi}_p(t) = \hat{\phi}_1(t) + \hat{\phi}_2(t) \quad (2.15)$$

et la position de synchronisation trame est estimée par :

$$\hat{t}_{mSSS} = \underset{t=0, \dots, n_c-1}{\operatorname{argmin}} \{ \hat{\phi}_p(t) \}. \quad (2.16)$$

### 2.3.3 Résultats des simulations

Dans le cas des codes convolutifs étudiés au paragraphe 2.2, nous avons démontré que la structure spécifique de leur matrice de contrôle de parité entraîne une dégradation des performances de la méthode de synchronisation du SSS. Ce même problème est rencontré dans le cas des codes cycliques. Ainsi, tout comme pour les codes convolutifs, nous proposons pour les codes en blocs cycliques d'utiliser un entrelaceur pseudo-aléatoire à l'émission pour permuter les symboles codés. Cet entrelaceur doit être connu au récepteur pour que la propriété cyclique du code soit récupérée avant le décodage. De plus, la taille de la fenêtre de synchronisation doit être un multiple de la longueur de l'entrelaceur.

Nous présentons dans ce paragraphe les résultats des simulations effectuées pour analyser les performances de la méthode du m-SSS proposée au paragraphe précédent pour la synchronisation des codes BCH et des codes produits. Les performances de cette méthode sont comparées à celles de la technique du SSS proposée au paragraphe 1.2 du chapitre 1.

La figure 2.10 montre une comparaison entre les deux méthodes de synchronisation une fois appliquées au code BCH étendu (32,26). Pour une fenêtre de synchronisation de longueur  $K = 1$  bloc, il est clair que les deux méthodes de synchronisation n'arrivent pas à synchroniser les codes BCH. Ceci est dû à la petite longueur du code (32 bits) et au faible nombre d'équations de parité (6 équations). Une solution à ce problème est d'augmenter la taille de la fenêtre de synchronisation. Ainsi, pour  $K = 5$  blocs et en appliquant la méthode du m-SSS, une probabilité de fausse synchronisation égale à  $4,5 \cdot 10^{-4}$  est atteinte à un  $E_b/N_0$  égal à 5 dB. Pour cette même probabilité, une différence de 2 dB est observée entre la version originale de la méthode de synchronisation (algorithme du SSS) et celle que nous proposons pour la synchronisation des codes BCH (algorithme du m-SSS).

Dans la suite de ce chapitre, la seule différence qui existe entre les appellations SSS et m-SSS est l'étape de l'adaptation de la matrice de contrôle de parité à la fiabilité des symboles. Nous considérons maintenant le code produit BCH (32, 26)<sup>2</sup>. Cette notation signifie que le code produit  $C$  est construit à partir de deux codes BCH étendus (32, 26) identiques. La figure 2.11 montre une comparaison entre la méthode du SSS et celle

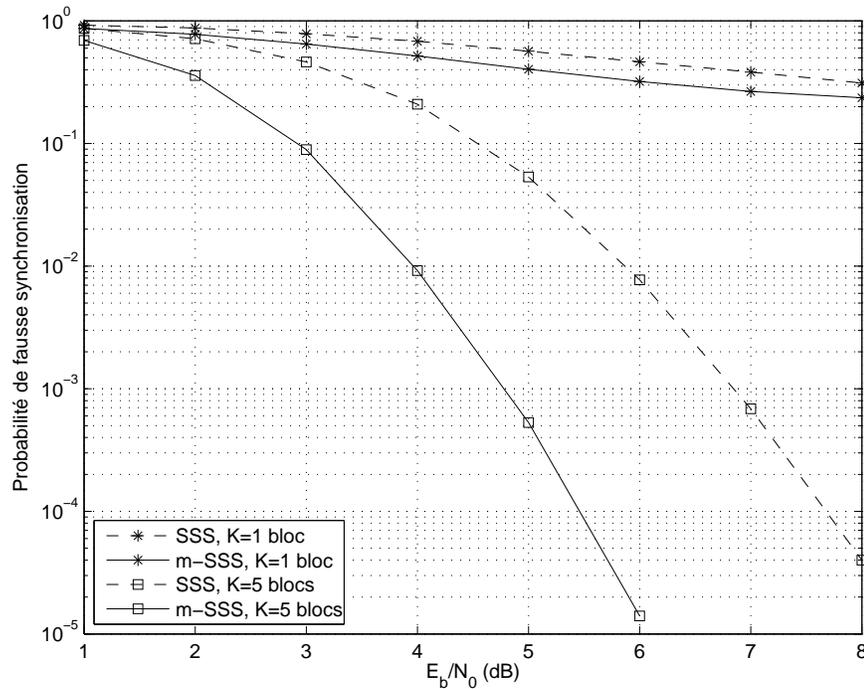


Figure 2.10 — Méthodes de synchronisation aveugles appliquées au code BCH étendu (32, 26).

du m-SSS. Pour la même taille de la fenêtre de synchronisation, il est clair que la technique du m-SSS est la plus performante. Un gain d'environ 1,3 dB est atteint pour une probabilité de fausse synchronisation égale à  $10^{-3}$ .

Dans la figure 2.12, nous évaluons les performances de la méthode du m-SSS appliquée aux codes produits BCH  $(16, 11)^2$  et  $(32, 26)^2$ . Bien que le code  $(16, 11)^2$  possède 8 éléments non nuls dans chaque ligne de sa matrice de contrôle de parité  $H$  et que le code  $(32, 26)^2$  en possède 16, la synchronisation du dernier présente de meilleures performances pour un  $E_b/N_0$  supérieur à 1,5 dB. Ceci est dû au fait que le code  $(32, 26)^2$  est plus long que le code  $(16, 11)^2$  (1024 bits à la place de 256) et a un nombre plus grand d'équations de parité (6 à la place de 5). Sur cette même figure, nous comparons aussi nos résultats à ceux obtenus par la méthode de synchronisation non aveugle de Massey. Pour une séquence d'apprentissage de longueur  $L = 20$  bits (environ 8% de la longueur du code  $(16, 11)^2$  et 2% de celle du code  $(32, 26)^2$ ), les pentes des courbes correspondant à la méthode proposée décroissent beaucoup plus rapidement que celles des courbes de Massey. Pour le code produit  $(32, 26)^2$  et à une probabilité de fausse synchronisation égale à  $10^{-3}$ , il existe une différence d'environ 2 dB entre l'algorithme du m-SSS ( $K = 1$  bloc) et la méthode de Massey ( $L = 20$  bits).

Afin d'évaluer l'efficacité de l'algorithme proposé, nous avons tracé les courbes de FER obtenues après avoir décodé les codes produits en utilisant le décodeur m-ABP et appliqué la méthode de la synchronisation du m-SSS. Trois itérations locales et six itérations globales ont été effectuées pour l'algorithme du m-ABP. La figure 2.13

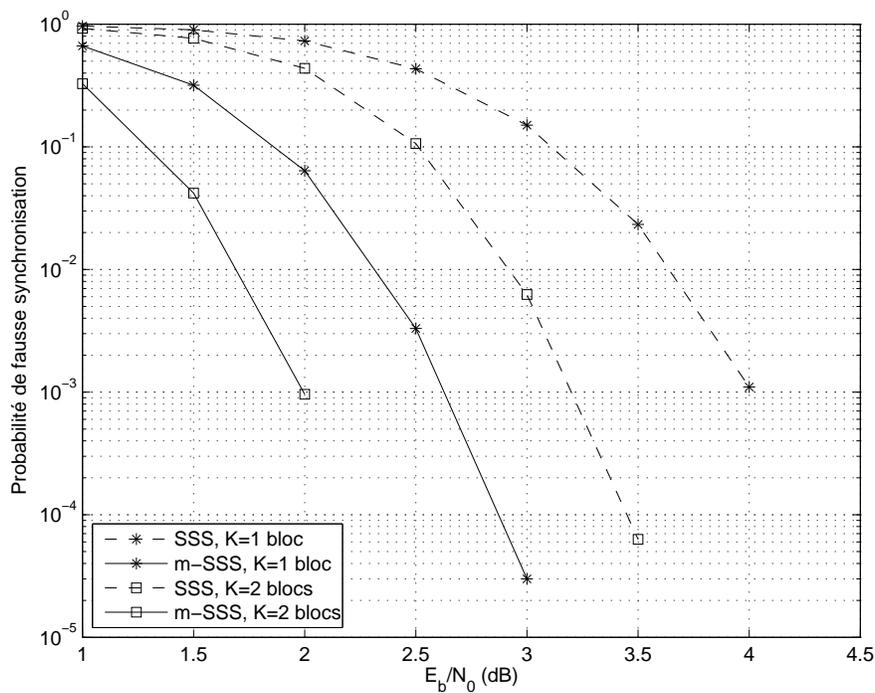


Figure 2.11 — Méthodes de synchronisation aveugles appliquées au code produit BCH  $(32, 26)^2$ .

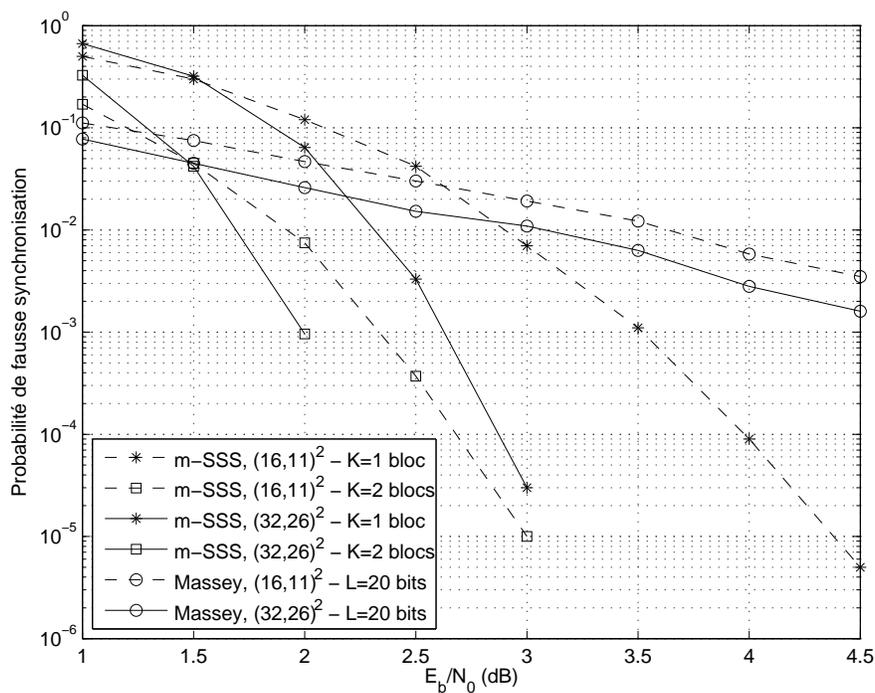


Figure 2.12 — Méthode du m-SSS appliquée aux codes produits BCH  $(16, 11)^2$  et  $(32, 26)^2$  et comparée à la méthode de synchronisation de Massey.

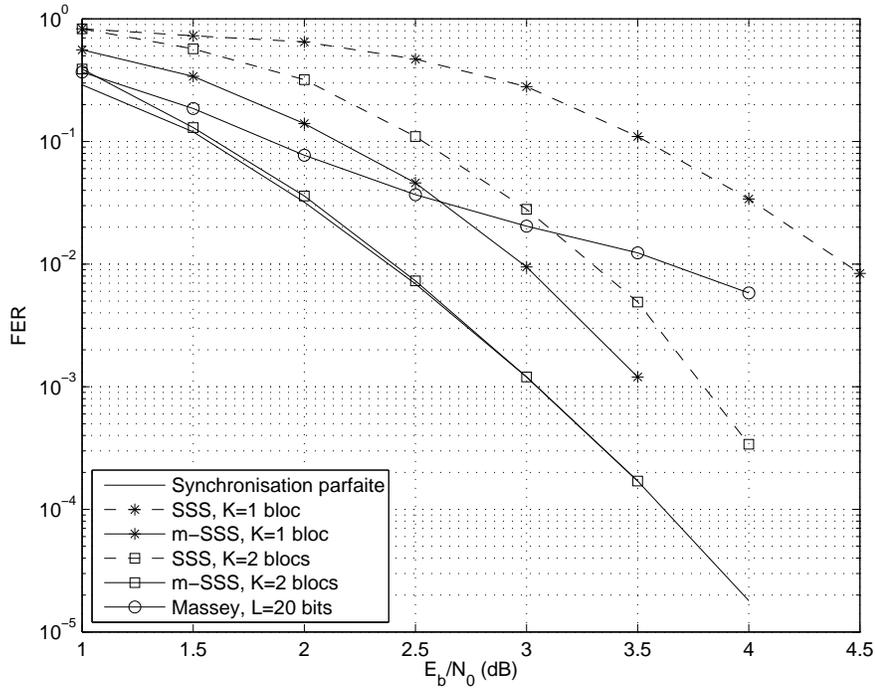


Figure 2.13 — Courbes de FER du décodeur m-ABP appliqué au code produit BCH  $(16, 11)^2$ .

montre les courbes de FER obtenues après le décodage du code produit BCH  $(16, 11)^2$ . Il est clair que la méthode du SSS telle qu'elle a été proposée au paragraphe 1.2 n'est pas convenable pour la synchronisation des systèmes utilisant des codes produits. Pour la courbe de FER obtenue après synchronisation par la méthode de Massey, on voit qu'elle n'arrive pas à converger vers la courbe de synchronisation parfaite. Par contre, avec la méthode du m-SSS, une dégradation de  $0.6 \text{ dB}$  seulement est observée entre la courbe de FER correspondant à la synchronisation parfaite et celle obtenue après application du m-SSS ( $K = 1$  bloc). Et lorsque  $K = 2$  blocs, les courbes de FER obtenues après synchronisation par le m-SSS et dans le cas de la synchronisation parfaite, sont à peu près les mêmes.

Dans le cas du code produit BCH  $(32, 26)^2$ , la figure 2.14 montre que même pour  $K = 1$  bloc, la courbe de FER tracée après avoir appliqué l'algorithme du m-SSS est la même que celle obtenue dans le cas d'une synchronisation parfaite.

Notons aussi que même si la méthode du m-SSS est coûteuse en termes de coût de calcul, le fait de l'appliquer économise la première étape du décodeur m-ABP.

Reste à signaler que la méthode du m-SSS proposée dans ce paragraphe a été présentée dans la conférence ICC 2009 : IEEE International Conference on Communications [36].

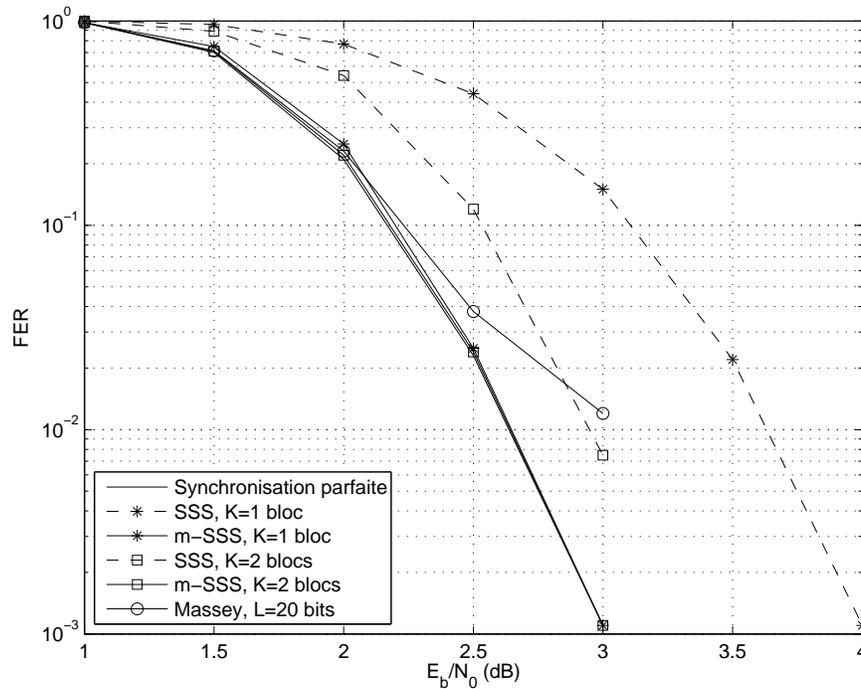


Figure 2.14 — Courbes de FER du décodeur m-ABP appliqué au code produit BCH  $(32, 26)^2$ .

## 2.4 Synchronisation trame des systèmes utilisant des modulations à grand nombre d'états

L'étude que nous avons présentée jusqu'à maintenant et les techniques de synchronisation trame proposées ont été introduites dans le cas d'une modulation BPSK. En absence d'un déphasage et d'un résidu de porteuse, ces mêmes algorithmes peuvent être utilisés avec une modulation QPSK (Quadrature Phase Shift Keying) et des résultats identiques à ceux d'une modulation BPSK sont obtenus. En effet, après transmission des symboles QPSK dans le canal Gaussien, il suffit à la réception de prendre les parties réelle et imaginaire de chaque échantillon puis de les classer de nouveau comme étant une série d'échantillons réels. On revient alors au cas BPSK et la même procédure de synchronisation est effectuée donnant ainsi les mêmes performances.

Dans ce paragraphe, nous effectuons une généralisation de notre méthode de synchronisation pour qu'elle soit adaptée à des modulations à grand nombre d'états. Soit une modulation à  $2^q$  états, où  $q$  bits sont associés à chaque symbole. Notons  $\mathcal{Q}$  l'ensemble des symboles de cette modulation. A chaque  $q$ -uplet  $(a(1), a(2), \dots, a(q))$ , le modulateur fait correspondre un symbole  $\gamma$  appartenant à  $\mathcal{Q}$ .

Pour pouvoir adapter nos méthodes de synchronisation trame présentées dans ce document à n'importe quel type de modulation, nous devons estimer tout d'abord les LLR des bits codés et ensuite appliquer les critères de synchronisation. Inspirés par une approximation proposée dans [60] et utilisée dans [10], nous nous servons de chaque échantillon reçu  $r(k)$  pour estimer le LLR des  $q$  bits qui y sont associés. Cette estima-

tion est donnée par :

$$\hat{\Gamma}(a((k-1)q+i)) = \min_{\substack{\gamma \in \mathcal{Q} \\ a(i)=0}} \frac{|r(k) - \gamma|^2}{\sigma^2} - \min_{\substack{\gamma \in \mathcal{Q} \\ a(i)=1}} \frac{|r(k) - \gamma|^2}{\sigma^2}, \quad i = 1, \dots, q \quad (2.17)$$

Ayant estimé les LLR de tous les bits d'un mot de code, nous pouvons maintenant appliquer toutes les méthodes de synchronisation proposées dans ce document, en remplaçant dans les critères calculés, la vraie valeur d'un échantillon reçu par l'estimée de son LLR que nous venons de calculer dans (2.17). Ainsi, notre méthode de synchronisation trame d'origine proposée au paragraphe 1.2 sera dans ce cas basée sur les critères :

$$\hat{L}(S_t(k)) = (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\hat{\Gamma}(a(t+k_j))) \right) \min_{j=1, \dots, u_k} |\hat{\Gamma}(a(t+k_j))| \quad (2.18)$$

$$\text{et } \hat{\phi}(t) = \sum_{k=1}^{Kn_r} \hat{L}(S_t(k)), \quad (2.19)$$

et l'instant de synchronisation trame est estimé par :

$$\hat{t}_{SSS} = \underset{t=0, \dots, n_c-1}{\text{argmin}} \{ \hat{\phi}(t) \}. \quad (2.20)$$

Nous rappelons que les variables  $u_k$  et  $k_j$  utilisées dans (2.18) représentent respectivement le nombre d'éléments non nuls dans la  $k^{\text{ième}}$  ligne de la matrice de contrôle de parité du code et la position du  $j^{\text{ième}}$  élément non nul dans cette  $k^{\text{ième}}$  ligne. Dans (2.19),  $K$  désigne le nombre de blocs de longueur  $n_c$  bits contenu dans la fenêtre de synchronisation.

Appliquons maintenant la méthode de synchronisation proposée dans ce paragraphe à un système utilisant une modulation 16-QAM (16-state Quadrature Amplitude Modulation). La figure 2.15 montre les résultats des simulations une fois cette méthode appliquée au code LDPC de longueur  $n_c = 512$  bits, rendement  $R = 0.5$  et ayant  $u_k = 4$  éléments non nuls dans chaque ligne de sa matrice de contrôle de parité. Pour une fenêtre de synchronisation de taille  $K = 1$  mot de code et comparée au cas d'une modulation BPSK/QPSK, une dégradation d'environ 2.5 dB est observée pour une probabilité de fausse synchronisation égale à  $10^{-3}$ . Cette différence est réduite à 0.5 dB si une fenêtre de taille  $K = 2$  mots de codes est utilisée pour la synchronisation.

## 2.5 Calcul du taux d'erreur trame après application de la procédure de synchronisation

Les courbes du taux d'erreur trame (ou FER) présentées dans les figures précédentes ont été obtenues après synchronisation et décodage par les décodeurs appropriés à chaque code. Cependant, il est possible de trouver une expression analytique du FER après synchronisation et décodage, en fonction de la probabilité de fausse

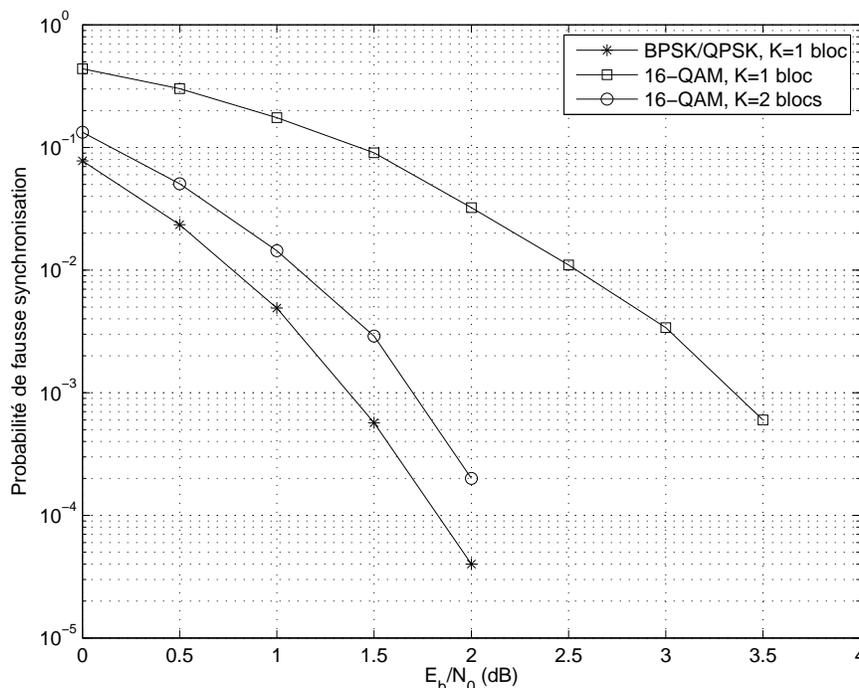


Figure 2.15 — Synchronisation trame pour des modulations à grand nombre d'états.

synchronisation et du FER obtenu dans le cas d'une synchronisation parfaite. Cette expression que nous calculons dans ce paragraphe est valide quels que soient le code correcteur d'erreurs utilisé et la méthode de synchronisation appliquée.

Soient les événements suivants :

$A$  : Une trame décodée est erronée.

$B$  : Une bonne synchronisation a lieu.

En appliquant le théorème de Bayes, la probabilité qu'une trame décodée soit erronée est égale à :

$$Pr[A] = Pr[A/B]Pr[B] + Pr[A/\bar{B}]Pr[\bar{B}], \quad (2.21)$$

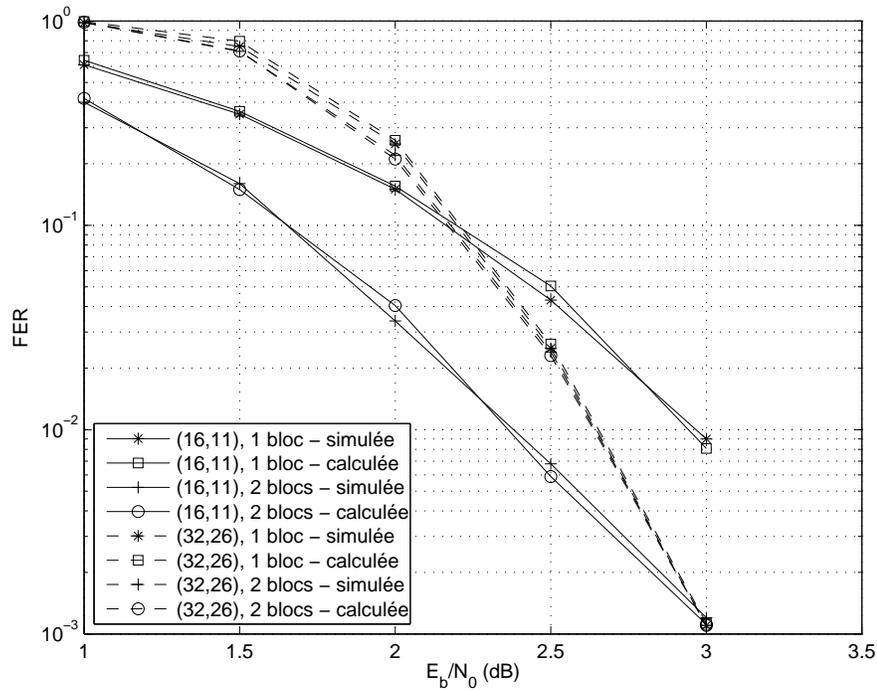
où  $\bar{B}$  désigne l'événement complémentaire de  $B$ . En admettant qu'une trame décodée est toujours erronée quand une fausse synchronisation trame a lieu, nous avons :

$$Pr[A/\bar{B}] = 1. \quad (2.22)$$

Ainsi, (2.21) devient égale à :

$$Pr[A] = Pr[A/B]Pr[B] + Pr[\bar{B}]. \quad (2.23)$$

Remarquons maintenant que  $Pr[A]$  et  $Pr[A/B]$  ne sont rien autre que les FER à la sortie du décodeur, obtenus respectivement après synchronisation (noté  $FER_2$ ) et dans le cas d'une synchronisation parfaite ( $FER_1$ ). Quant à  $Pr[\bar{B}]$ , c'est la probabilité



**Figure 2.16** — Courbes de FER simulé et calculé obtenu après synchronisation et décodage des codes produits.

de fausse synchronisation,  $P_F$ . Ainsi, l'expression analytique du FER calculée après synchronisation et décodage est donnée par :

$$FER_2 = (1 - P_F)FER_1 + P_F. \quad (2.24)$$

Afin de vérifier la validité de l'expression ci-dessus, nous traçons sur la figure 2.16 les FER calculés et simulés des codes produits BCH  $(16, 11)^2$  et  $(32, 26)^2$ , obtenus après avoir appliqué la méthode de synchronisation du m-SSS. Nous observons sur cette figure que les courbes simulées et calculées sont quasiment les mêmes, ce qui valide (2.24).

Ainsi, afin d'étudier l'efficacité d'une méthode quelconque de synchronisation trame en termes de taux d'erreur trame, il suffit de calculer (2.24) sans être obligé d'appliquer le décodeur après chaque procédure de synchronisation. Dorénavant, nous privilégierons ce mode de calcul afin de réduire significativement le temps de simulation.

## 2.6 Conclusion

Dans ce chapitre, nous avons appliqué la méthode de synchronisation trame que nous avons proposée dans le chapitre précédent (méthode du SSS), à plusieurs types de codes correcteurs d'erreurs. Pour des systèmes utilisant des codes LDPC, nous avons comparé les performances de la méthode du SSS à plusieurs méthodes de synchronisation proposées dans la littérature. Notre technique de synchronisation basée sur le

critère MAP présente les meilleures performances. Les résultats des simulations ont aussi montré que la probabilité de fausse synchronisation augmente avec le nombre d'éléments non nuls se trouvant dans chaque ligne de la matrice de contrôle de parité du code correcteur d'erreurs.

Pour pouvoir appliquer la méthode de synchronisation du SSS aux codes convolutifs, nous avons proposé d'utiliser un entrelaceur pseudo-aléatoire à l'émission et ceci pour rompre la structure circulaire de la matrice de contrôle de parité de ce type de codes. Encore une fois, la méthode du SSS surpasse les autres méthodes de synchronisation. Les courbes des taux d'erreur trame obtenues après synchronisation et décodage ont montré que la méthode de synchronisation proposée est bien adaptée aux codes convolutifs.

Les codes BCH et les codes produits étant des codes à matrices de contrôle de parité non creuses, nous avons proposé dans ce chapitre de modifier notre critère de synchronisation d'origine et d'ajouter une étape d'adaptation de la matrice de contrôle de parité en donnant un poids plus faible aux symboles les moins fiables. Pour les codes produits, nous profitons de la forme par laquelle ces codes sont construits, pour effectuer la synchronisation. Les courbes de FER obtenues après application de la méthode de synchronisation du m-SSS sont quasiment les mêmes que celles de la synchronisation parfaite.

Tous les résultats précédents ont été obtenus dans le cas d'une modulation BPSK. Nous avons alors étendu notre méthode de synchronisation trame aveugle pour qu'elle soit applicable quel que soit le type de la modulation utilisée et encore une fois, notre méthode de synchronisation présente des performances très satisfaisantes.



# Chapitre 3

## Estimation aveugle du résidu de porteuse

DANS ce chapitre, le problème d'estimation du résidu de porteuse (ou CFO pour Carrier Frequency Offset) est considéré. Nous supposons que les instants de synchronisation trame et symbole sont parfaitement estimés et que le type de modulation utilisée à la transmission est une BPSK. Ainsi, un échantillon reçu s'écrit sous la forme :

$$r(k) = b(k)e^{j(2\pi kf_0 T_s + \theta_0)} + w(k), \quad (3.1)$$

où  $b(k) = \pm 1$  est le  $k^{\text{ième}}$  symbole transmis,  $\theta_0$  est le déphasage introduit par le canal de propagation et  $w(k)$  est un nombre complexe désignant un bruit blanc Gaussien. Le résidu de porteuse  $f_0$  à estimer est supposé dans la gamme de quelques pourcents de la rapidité de modulation  $1/T_s$ . Dans la suite de ce document, nous supposons que  $f_0$  est uniformément distribué entre  $-0.1/T_s$  et  $0.1/T_s$ .

Ce chapitre est divisé en deux parties. Dans la première, nous supposons que le déphasage  $\theta_0$  est connu par le récepteur et nous cherchons à estimer le résidu de porteuse  $f_0$ . Dans la seconde partie du chapitre, le même problème d'estimation du résidu de porteuse est considéré mais cette fois-ci en présence d'un déphasage  $\theta_0$  inconnu. Nous montrons dans cette partie que nous sommes toujours capables d'estimer le CFO et cela indépendamment de la présence du déphasage.

L'organisation de ce chapitre est comme suit. Dans la première section, un état de l'art est présenté dans lequel nous parlons brièvement de quelques méthodes d'estimation du résidu de porteuse étudiées dans la littérature. Ensuite, nous considérons le cas d'un déphasage connu par le récepteur et nous proposons une méthode d'estimation du CFO basée sur le calcul et la minimisation d'une fonction du LLR du syndrome obtenue à partir de la matrice de contrôle de parité du code correcteur d'erreurs. En raison de la présence de nombreux minima locaux, le minimum global de la fonction de coût est obtenu en appliquant l'algorithme itératif du Recuit Simulé. Les résultats des simulations montrent que la méthode proposée présente de très bonnes performances et surpasse une méthode classique d'estimation du CFO.

Quand le déphasage introduit par le canal de transmission est inconnu, nous calculons une nouvelle fonction de coût à minimiser et nous montrons que même en présence de ce déphasage, la méthode proposée d'estimation du CFO est très performante.

Reste à signaler que la méthode d'estimation aveugle du résidu de porteuse proposée dans ce chapitre a fait l'objet d'une soumission d'un article dans la revue IEEE Transactions on Wireless Communications [35].

### 3.1 Etat de l'art

Dans la littérature, plusieurs algorithmes ont été introduits pour estimer le résidu de porteuse d'un système. Une méthode d'estimation de la fréquence à partir d'une séquence d'échantillons corrompus par du bruit blanc Gaussien a été proposée par Tretter dans [65]. Cette méthode est fondée sur une régression linéaire sur la phase instantanée du signal reçu. Dans [42], Kay présente un estimateur de fréquence pour le cas d'un signal composé d'une seule sinusoïde complexe. Cet estimateur est inspiré de celui de Tretter mais il est un peu plus robuste au bruit. Fitz propose dans [19] un algorithme d'estimation de fréquence applicable à des problèmes de communications nécessitant une estimation rapide et récursive de la fréquence porteuse.

Dans le cas de signaux modulés, plusieurs algorithmes d'estimation de porteuse ont été proposés. Dans [22], les auteurs présentent un détecteur de fréquence selon le maximum de vraisemblance. Dérivés de ce principe de maximum de vraisemblance, plusieurs algorithmes d'estimation de fréquence ont été proposés dans [15] pour les cas DA (Data aided), DD (Decision Directed) et NDA (Non Data Aided). Une nouvelle famille d'estimateurs NLLS (NonLinear Least Squares) a été introduite dans [68] pour l'estimation jointe du déphasage, du résidu de porteuse et du Doppler pour des transmissions utilisant des modulations MPSK (M-ary Phase Shift Keying). Les performances de l'estimateur NLLS pour l'estimation aveugle du résidu de porteuse à des faibles rapports signal-sur-bruit ont été étudiées dans [14].

Dans le cas NDA, une méthode classique d'estimation du CFO est souvent utilisée. Pour une modulation BPSK, cette méthode, caractérisée par sa complexité réduite et ses bonnes performances, estime le résidu de porteuse par [15] :

$$\hat{f}_{class} = \frac{1}{4\pi DT_s} \arg \left\{ \sum_{k=D}^{N-1} (r(k)r^*(k-D))^2 \right\}. \quad (3.2)$$

Notons que  $N$  désigne le nombre d'échantillons utilisés pour l'estimation du CFO et  $D$  est un coefficient à régler. Dans la suite de ce document, nous fixons  $N = n_c$  et  $D = 1$ . Une expression approximative de la variance de l'erreur d'estimation de l'estimateur ci-dessus est calculée dans [5] pour des faibles valeurs de  $f_0 T_s$ . Elle est donnée par :

$$\sigma_{class}^2 \approx \frac{1}{4\pi^2 T_s} \left[ \frac{1}{N} (8\sigma_e^2 + 16\sigma_e^2 + 8\sigma_e^8) + \frac{1}{N^2} (2\sigma_e^2 + 2\sigma_e^4) \right], \quad (3.3)$$

où  $\sigma_e^2 = \frac{\sigma^2}{2}$  est la variance de chaque partie (réelle ou imaginaire) du bruit additif blanc Gaussien.

Les algorithmes présentés ci-dessus donnent de bonnes performances pour des SNR élevés. Cependant, comme les codes actuels fonctionnent pour des faibles SNR, nous

nous sommes intéressés à des algorithmes plus performants. Nous présentons dans ce chapitre des nouveaux algorithmes d'estimation aveugle du résidu de porteuse basés sur le calcul et la minimisation des fonctions du LLR du syndrome obtenues à partir des parties réelles et imaginaires des échantillons reçus. Les résultats des simulations montrent que ces algorithmes sont très efficaces mêmes pour des niveaux de bruit très élevés.

## 3.2 Estimation aveugle du résidu de porteuse en présence d'un déphasage connu par le récepteur

### 3.2.1 Méthode proposée d'estimation aveugle du résidu de porteuse - Algorithme du C-SFM

Nous considérons dans ce paragraphe le problème d'estimation du résidu de porteuse en présence d'un déphasage connu par le récepteur. Après avoir compensé ce déphasage connu présent dans les échantillons reçus, nous obtenons :

$$r_{\theta_0}(k) = r(k)e^{-j\theta_0}. \quad (3.4)$$

La méthode d'estimation aveugle du résidu de porteuse que nous proposons dans ce paragraphe est appliquée à des systèmes utilisant des codes correcteurs d'erreurs. Cette méthode est décrite par la procédure adaptative suivante :

Le résidu de porteuse présent dans les échantillons  $r_{\theta_0}(k)$  est compensé par une fréquence quelconque  $\tilde{f}$ . Cette procédure de compensation se traduit par :

$$r_{(\tilde{f}, \theta_0)}(k) = r_{\theta_0}(k)e^{-j2\pi k \tilde{f} T_s}. \quad (3.5)$$

Ensuite, nous évaluons une fonction de coût  $L_R(\tilde{f}, \theta_0)$ , qui est une fonction du LLR du syndrome du code correcteur d'erreurs utilisé dans le système. Cette fonction est calculée à partir des échantillons  $r_{(\tilde{f}, \theta_0)}(k)$ . Nous verrons dans la suite que cette fonction de coût est minimale pour  $\tilde{f} = f_0$ . Ainsi, trouver l'argument minimal de cette fonction en  $\tilde{f}$  permet d'estimer le résidu de porteuse du système.

Dans le paragraphe 1.2, nous avons proposé une méthode de synchronisation trame basée sur le calcul et la minimisation du LLR du syndrome. Inspirés par ce même principe, nous définissons la fonction de coût  $L_R(\tilde{f}, \theta_0)$  en question comme étant la moyenne du LLR du syndrome calculée à partir des parties réelles des échantillons

$r_{(\tilde{f}, \theta_0)}(k)$ . Cette fonction est définie par :

$$\begin{aligned} L_R(\tilde{f}, \theta_0) &= E \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Re(r_{(\tilde{f}, \theta_0)}(k_j))) \right) \min_{j=1, \dots, u_k} |\Re(r_{(\tilde{f}, \theta_0)}(k_j))| \right) \right] \\ &= E \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Re(r_{\theta_0}(k_j) e^{-j(2\pi k_j \tilde{f} T_s)})) \right) \right. \right. \\ &\quad \left. \left. \cdot \min_{j=1, \dots, u_k} |\Re(r_{\theta_0}(k_j) e^{-j(2\pi k_j \tilde{f} T_s)})| \right) \right], \end{aligned} \quad (3.6)$$

où  $E[\cdot]$  désigne l'opérateur de l'espérance mathématique et  $\Re(\cdot)$  est la partie réelle d'un nombre complexe. Rappelons que  $u_k$  désigne le nombre d'éléments non nuls dans la  $k^{\text{ième}}$  ligne de la matrice de contrôle de parité  $H$ . Dans la suite de ce chapitre et pour des raisons de simplicité, nous supposons que  $u_k$  est pair et constant pour toutes les lignes de  $H$ . Notons que la raison pour laquelle nous avons introduit une espérance dans la définition de  $L_R(\tilde{f}, \theta_0)$  est pour avoir une fonction déterministe et être ainsi capables de l'étudier.

Calculons maintenant la valeur minimale de la fonction  $L_R(\tilde{f}, \theta_0)$ . Pour cela, remarquons qu'un échantillon reçu est statistiquement équivalent (s.e) à :

$$r(k) \stackrel{\text{s.e}}{=} (b(k) + w(k)) e^{j(2\pi k f_0 T_s + \theta_0)}. \quad (3.7)$$

Ainsi, nous avons :

$$r_{(\tilde{f}, \theta_0)}(k) \stackrel{\text{s.e}}{=} (b(k) + w(k)) e^{j(2\pi k (f_0 - \tilde{f}) T_s)}. \quad (3.8)$$

L'équation (3.6) est égale à :

$$L_R(\tilde{f}, \theta_0) = \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} E \left[ \left( \prod_{j=1}^{u_k} \text{sign}(\Re(r_{(\tilde{f}, \theta_0)}(k_j))) \right) \min_{j=1, \dots, u_k} |\Re(r_{(\tilde{f}, \theta_0)}(k_j))| \right] \right). \quad (3.9)$$

Pour un système de transmission utilisant une modulation de type BPSK et en supposant que  $u_k$  est constant et pair, (3.9) devient égale à :

$$\begin{aligned} L_R(\tilde{f}, \theta_0) &= - \sum_{k=1}^{n_r} \left( E \left[ \left( \prod_{j=1}^{u_k} \text{sign} \left( (b(k_j) + w_1(k_j)) \cos(2\pi k_j (f_0 - \tilde{f}) T_s) \right. \right. \right. \right. \\ &\quad \left. \left. \left. - w_2(k_j) \sin(2\pi k_j (f_0 - \tilde{f}) T_s) \right) \right) \min_{j=1, \dots, u_k} \left| (b(k_j) + w_1(k_j)) \cos(2\pi k_j (f_0 - \tilde{f}) T_s) \right. \right. \\ &\quad \left. \left. - w_2(k_j) \sin(2\pi k_j (f_0 - \tilde{f}) T_s) \right| \right] \right) \end{aligned} \quad (3.10)$$

où  $w_1(k)$  et  $w_2(k)$  représentent respectivement les composantes réelles et imaginaires du bruit  $w(k)$ .

Remarquons maintenant que :

$$\begin{aligned}
 E & \left[ \left( \prod_{j=1}^{u_k} \text{sign} \left( (b(k_j) + w_1(k_j)) \cos(2\pi k_j(f_0 - \tilde{f})T_s) - w_2(k_j) \sin(2\pi k_j(f_0 - \tilde{f})T_s) \right) \right) \right. \\
 & \cdot \left. \min_{j=1, \dots, u_k} \left| (b(k_j) + w_1(k_j)) \cos(2\pi k_j(f_0 - \tilde{f})T_s) - w_2(k_j) \sin(2\pi k_j(f_0 - \tilde{f})T_s) \right| \right] \\
 & \leq E \left[ \min_{j=1, \dots, u_k} \left| (b(k_j) + w_1(k_j)) \cos(2\pi k_j(f_0 - \tilde{f})T_s) - w_2(k_j) \sin(2\pi k_j(f_0 - \tilde{f})T_s) \right| \right] \\
 & \leq E \left[ \min_{j=1, \dots, u_k} \left( |b(k_j) + w_1(k_j)| + |w_2(k_j) \sin(2\pi k_j(f_0 - \tilde{f})T_s)| \right) \right] \quad (3.11)
 \end{aligned}$$

En substituant (3.11) dans (3.10) on obtient :

$$L_R(\tilde{f}, \theta_0) \geq -n_r E \left[ \min_{j=1, \dots, u_k} \left( |b(k_j) + w_1(k_j)| + |w_2(k_j) \sin(2\pi k_j(f_0 - \tilde{f})T_s)| \right) \right]. \quad (3.12)$$

Une condition nécessaire pour que  $L_R(\tilde{f}, \theta_0)$  atteigne sa borne minimale est :  $\tilde{f} = f_0$ . Cette condition devient suffisante dès lors que  $w_1(k)$  ne vient pas changer le signe de  $b(k)$ .

Rappelons que nous sommes toujours dans le cas d'une modulation BPSK. En considérant maintenant un canal de transmission sans bruit, (3.12) devient :

$$L_R(\tilde{f}, \theta_0) \geq -n_r. \quad (3.13)$$

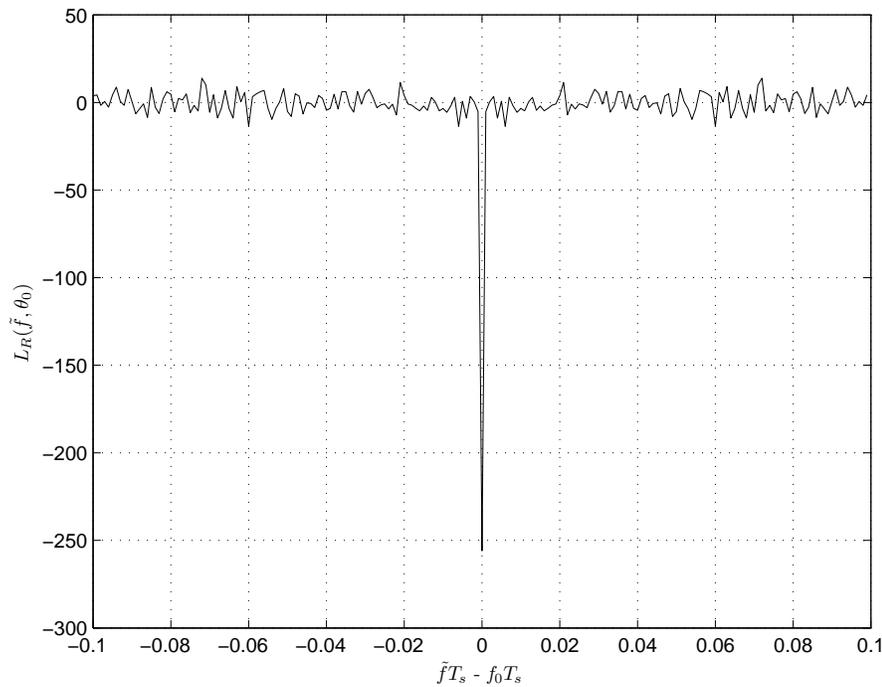
Ainsi, en absence de bruit, la fonction  $L_R(\tilde{f}, \theta_0)$  est minorée par  $-n_r$ . Elle atteint cette valeur minimale pour  $\tilde{f} = f_0$ . Ces résultats peuvent être vérifiés par simulations. Pour cela, nous traçons sur la figure 3.1 les variations de  $L_R(\tilde{f}, \theta_0)$  en fonction de  $(\tilde{f}T_s - f_0T_s)$  dans le cas d'un canal sans bruit. La fonction  $L_R(\tilde{f}, \theta_0)$  a été calculée pour un système utilisant un code LDPC de taille  $n_c = 512$  bits, rendement  $R = 0.5$  et  $u_k = 4$ . En observant cette figure, il est clair que la fonction  $L_R(\tilde{f}, \theta_0)$  admet un minimum pour  $\tilde{f} = f_0$  (i.e  $\tilde{f}T_s - f_0T_s = 0$ ) et la valeur de ce minimum est égale à  $-256$  (pour le code LDPC utilisé dans cette simulation,  $n_r = 256$ ). Ce qui valide l'étude que nous venons d'effectuer.

La fonction  $L_R(\tilde{f}, \theta_0)$  étant minimale pour une fréquence  $\tilde{f} = f_0$ , notre but est de trouver la fréquence  $\tilde{f}$  qui minimise  $L_R$ . Cette fréquence sera une estimée du résidu de porteuse du système. L'équation (3.6) peut être estimée par :

$$\begin{aligned}
 \hat{L}_R(\tilde{f}, \theta_0) & = \frac{1}{K} \sum_{i=0}^{K-1} \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Re(r_{\theta_0}(k_j + in_c) e^{-j(2\pi(k_j + in_c)\tilde{f}T_s)})) \right) \right) \right. \\
 & \cdot \left. \min_{j=1, \dots, u_k} \left| \Re(r_{\theta_0}(k_j + in_c) e^{-j(2\pi(k_j + in_c)\tilde{f}T_s)}) \right| \right] \quad (3.14)
 \end{aligned}$$

où  $K$  est le nombre de mots de code utilisés pour calculer l'espérance mathématique de  $L_R$ . Dans le reste de ce document et pour des raisons de simplicité, nous supposons que  $K = 1$ . Finalement, le résidu de porteuse du système est estimé par :

$$\hat{f}_{CSFM} = \underset{\tilde{f}}{\text{argmin}} \hat{L}_R(\tilde{f}, \theta_0). \quad (3.15)$$



**Figure 3.1** —  $L_R(\tilde{f}, \theta_0)$  en fonction de l'erreur d'estimation du résidu de porteuse, tracée pour un déphasage connu par le récepteur.

Dans la suite de ce document, l'algorithme d'estimation que nous proposons et qui est basé sur la minimisation d'une fonction du syndrome, sera désigné par l'algorithme du C-SFM (CFO estimation-Syndrome Function Minimization).

Faisons maintenant un zoom sur une partie de la fonction  $L_R(\tilde{f}, \theta_0)$  et traçons la courbe résultante sur la figure 3.2. Comme nous l'observons, pour  $\tilde{f}T_s - f_0T_s \neq 0$ , la courbe de  $L_R(\tilde{f}, \theta_0)$  contient plusieurs minima locaux. Pour cela, il est clair que nous ne pouvons pas utiliser un algorithme de type Gradient pour résoudre le problème d'optimisation. Dans ce document, nous proposons d'utiliser l'algorithme du Recuit Simulé [43, 11] pour minimiser la fonction du LLR du syndrome et estimer ainsi le résidu de porteuse du système.

### Algorithme du Recuit Simulé

Dans sa forme originale, l'algorithme de Recuit Simulé est basé sur l'analogie entre la simulation de la recuisson des solides et la solution des grands problèmes d'optimisation combinatoire. Recuit est le processus de chauffage d'un solide et de son refroidissement lent de manière à éliminer les tensions et les imperfections du cristal. Au cours de ce processus, l'énergie libre du solide est réduite au minimum. Le chauffage initial est nécessaire pour éviter d'être piégé dans un minimum local. Chaque fonction peut être considérée comme l'énergie libre d'un système et par conséquent, l'étude et l'imitation de ce processus devrait permettre de résoudre notre problème d'optimisation.

Soit  $h$  la fonction que nous cherchons à minimiser. L'algorithme itératif du Recuit

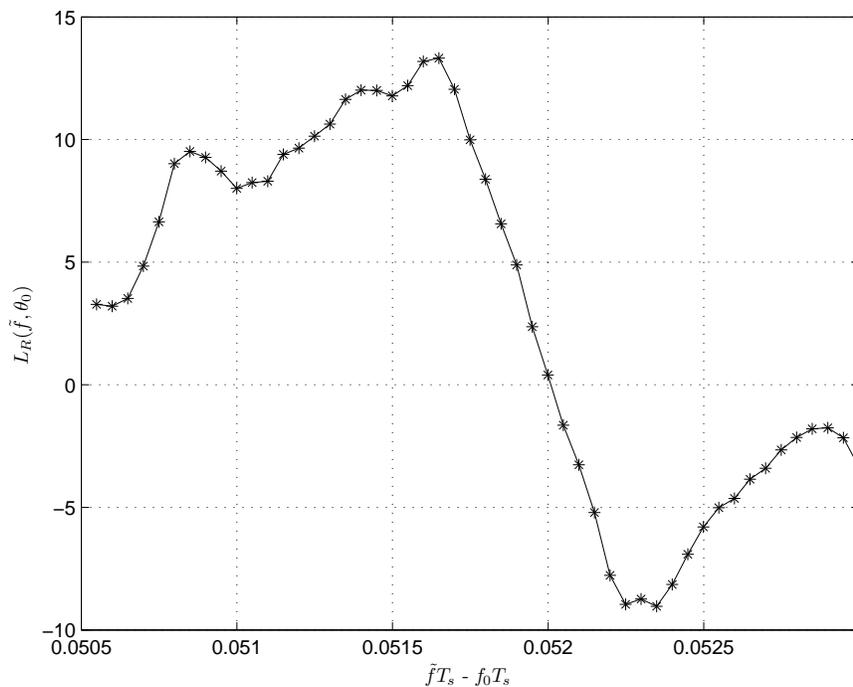


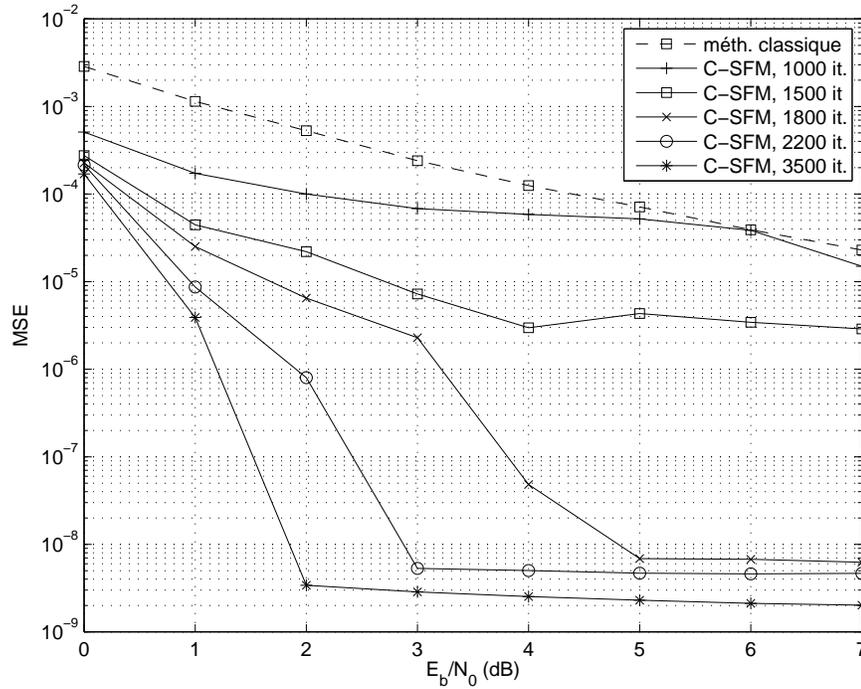
Figure 3.2 — Zoom sur la fonction  $L_R(\tilde{f}, \theta_0)$ , tracée pour un déphasage connu par le récepteur.

Simulé se résume comme suit :

- A l'itération 0 :
  - \* Initialiser  $x^{(0)}$ ,  $T_0$  et  $a$ , où  $x^{(m)}$  est la solution du problème de minimisation après  $m$  itérations,  $T_0$  est la température initiale et  $a$  est le coefficient de baisse de température.
- A l'itération  $m$  :
  - \* Générer une variable  $z$  suivant une loi uniforme.
  - \* Si  $(h(z) - h(x^{(m-1)})) \leq 0$ , accepter  $x^{(m)} = z$
  - \* Sinon
    - ▷ Générer une variable  $u$  suivant une loi uniforme entre 0 et 1,
    - ▷ Accepter  $x^{(m)} = z$  si  $(\exp(-(\frac{h(z)-h(x^{(m-1)})}{T_0 a^m})) \geq u)$ .
- Sortir quand le nombre maximal d'itérations est atteint ou bien qu'un critère d'arrêt est vérifié.

### Résultats des simulations

Afin d'analyser les performances de l'algorithme du C-SFM proposé pour l'estimation aveugle du résidu de porteuse, nous avons tracé les courbes de MSE (Mean Squared Error) obtenues par des simulations de Monte Carlo, où, pour chaque réalisation, un résidu de porteuse uniformément réparti entre  $-0.1/T_s$  et  $0.1/T_s$  a été tiré aléatoirement. Rappelons que le déphasage introduit par le canal de propagation



**Figure 3.3** — Performances de l'algorithme du C-SFM appliqué pour l'estimation aveugle du résidu de porteuse en présence d'un déphasage connu par le récepteur.

est supposé connu par le récepteur. L'algorithme du C-SFM a été comparé à la méthode classique d'estimation du résidu de porteuse décrite par (3.2). La figure 3.3 montre les courbes de MSE de l'algorithme du C-SFM et de la méthode classique appliqués à un code LDPC de taille  $n_c = 512$  bits, rendement  $R = 0.5$  et ayant  $u_k = 4$  éléments non nuls dans chaque ligne de sa matrice de contrôle de parité. Pour l'algorithme du Recuit Simulé utilisé dans le C-SFM, les paramètres suivants ont été utilisés :

- ▷ Température initiale  $T_0$  : 500 °C,
- ▷ Coefficient de baisse de température  $a$  : 0.9,
- ▷ fréquence d'entrée  $\tilde{f}$  initialisée à : 0.01
- ▷ Taille de l'intervalle de recherche de la variable uniforme  $z$  :  $0.2/T_s$ .

A partir de la figure 3.3, il est clair que si nous augmentons le nombre d'itérations de l'algorithme du Recuit Simulé, les performances de l'algorithme du C-SFM pour l'estimation du résidu de porteuse sont améliorées. Ces performances dépassent clairement celle de la méthode classique. Pour 3500 itérations du Recuit Simulé, une MSE d'environ  $3 \cdot 10^{-9}$  est atteinte pour un  $E_b/N_0$  égal à 2 dB seulement.

### 3.2.2 Méthode d'estimation du résidu de porteuse à complexité réduite - Algorithme du C-ISFM

Le principal inconvénient de l'algorithme du C-SFM proposé dans le paragraphe précédent pour l'estimation du CFO, est la partie itérative du Recuit Simulé pour la-

quelle de nombreux paramètres doivent être initialisés. Afin de réduire la complexité et le temps de calcul de cet algorithme, notre objectif est maintenant de réduire le nombre d'itérations du Recuit Simulé utilisé pour la minimisation de la fonction  $L_R(\tilde{f}, \theta_0)$  exprimée dans (3.6). Ainsi, au lieu d'initialiser aléatoirement la fréquence d'entrée  $\tilde{f}$  à la première itération du Recuit Simulé, nous proposons de lancer d'abord l'algorithme classique d'estimation du résidu de porteuse décrit par (3.2). La fréquence  $\hat{f}_{class}$  estimée par cet algorithme sert de fréquence d'initialisation pour le Recuit Simulé. L'algorithme d'estimation du CFO ainsi obtenu est désigné par le C-ISFM (CFO estimation-Initialized Syndrome Function Minimization).

Remarquons aussi que le nombre d'itérations du Recuit Simulé augmente avec la taille de l'intervalle de recherche de  $f_0$ . Ainsi, pour l'algorithme du C-ISFM, nous proposons de réduire cet intervalle de  $[-0.1/T_s, 0.1/T_s]$  à l'intervalle  $[\hat{f}_{class} - 3\sqrt{\sigma_{class}^2}, \hat{f}_{class} + 3\sqrt{\sigma_{class}^2}]$ , où  $\sigma_{class}^2$  est la variance théorique de l'estimation du résidu de porteuse de la méthode classique.

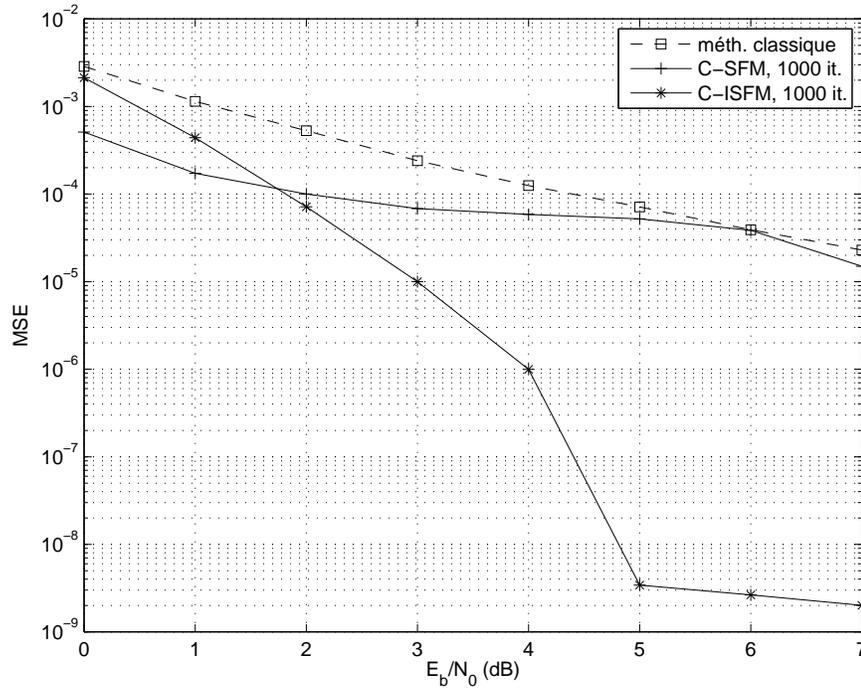
Dans [5], les auteurs calculent une expression approximative de  $\sigma_{class}^2$  en supposant le résidu de porteuse nul. Leur expression, donnée dans (3.3), est aussi valable pour des petites valeurs de  $f_0 T_s$ . Dans ce document, nous calculons une nouvelle expression de  $\sigma_{class}^2$ . Cette expression est obtenue sans recourir à l'hypothèse d'un résidu de porteuse nul. Pour des grandes valeurs de  $N$  nous démontrons que :

$$\sigma_{class}^2 \approx \frac{1}{\pi^2 T_s^2 N} (2\sigma_e^4 + 4\sigma_e^6 + 2\sigma_e^8). \quad (3.16)$$

Le calcul détaillé de (3.16) est donné en Annexe B.1.

## Résultats des simulations

Appliquons maintenant la méthode du C-ISFM à un système de communications utilisant le même code LDPC déjà utilisé dans ce chapitre. Les courbes correspondantes de MSE en fonction du  $E_b/N_0$  sont affichées sur la figure 3.4. Comme nous le voyons, initialiser la fréquence d'entrée à  $\hat{f}_{class}$  et réduire l'intervalle de recherche comme proposé ci-dessus (algorithme du C-ISFM), donnent de meilleurs résultats pour un nombre fixe d'itérations. Pour seulement 1000 itérations du Recuit Simulé, nous atteignons maintenant une MSE de  $3.10^{-9}$  pour un  $E_b/N_0$  égal à 5 dB, alors qu'en appliquant l'algorithme du C-SFM d'origine et pour le même nombre d'itérations (1000 itérations), une MSE d'environ  $5.10^{-5}$  est obtenue pour un  $E_b/N_0$  égal à 5 dB.



**Figure 3.4** — Performances de l'algorithme du C-ISFM appliqué pour l'estimation aveugle du résidu de porteuse en présence d'un déphasage connu par le récepteur.

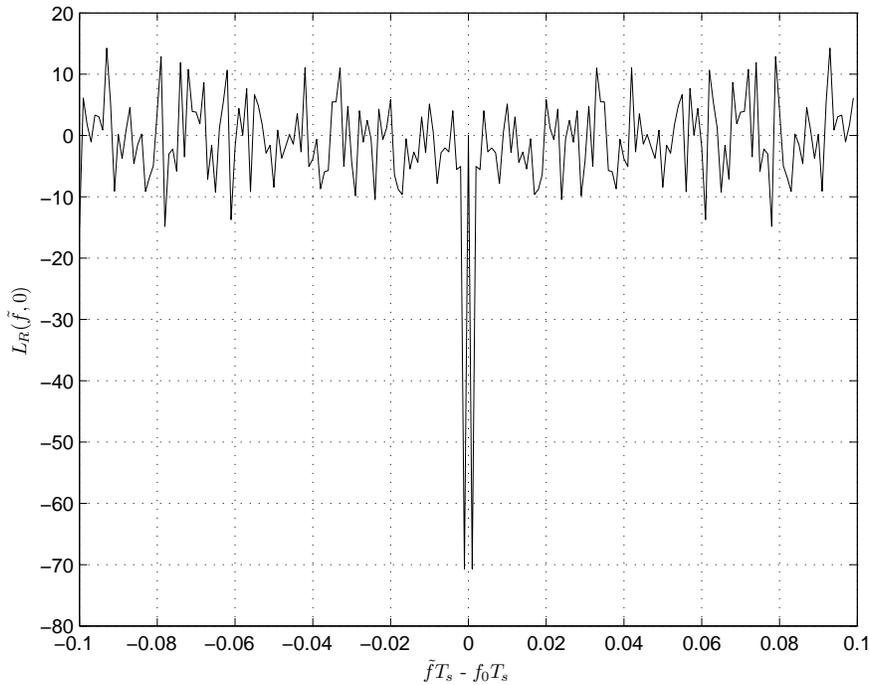
### 3.3 Estimation aveugle du résidu de porteuse en présence d'un déphasage inconnu par le récepteur

Jusqu'à présent, nous avons considéré dans ce document que le déphasage introduit par le canal de transmission est connu par le récepteur, ce qui n'est pas toujours le cas. Nous supposons maintenant que  $\theta_0$  est inconnu et notre objectif est toujours d'estimer le résidu de porteuse du système indépendamment de ce déphasage.

La fonction de coût  $L_R(\tilde{f}, \theta_0)$  de (3.6) ne peut pas être calculée puisque le déphasage  $\theta_0$  est inconnu. Nous calculons alors la fonction  $L_R(\tilde{f}, 0)$  obtenue sans compenser le déphasage se trouvant dans les échantillons reçus.

$$L_R(\tilde{f}, 0) = E \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Re(r(k_j) e^{-j(2\pi k_j \tilde{f} T_s)})) \right) \right) \cdot \min_{j=1, \dots, u_k} |\Re(r(k_j) e^{-j(2\pi k_j \tilde{f} T_s)})| \right]. \quad (3.17)$$

Etudions maintenant les variations de la fonction  $L_R(\tilde{f}, 0)$ . Pour cela, prenons le cas extrême en supposant que le déphasage  $\theta_0$  du canal est égal à  $\pi/2$  et traçons pour ce cas la fonction  $L_R(\tilde{f}, 0)$  de (3.17) sur la figure 3.5. On observe sur cette figure que le minimum de  $L_R(\tilde{f}, 0)$  n'est plus atteint pour  $\tilde{f} = f_0$ . Ainsi, en présence d'un déphasage inconnu, appliquer la méthode d'estimation proposée dans le paragraphe 3.2



**Figure 3.5** — La fonction  $L_R(\tilde{f}, 0)$  tracée en présence d'un déphasage  $\theta_0 = \pi/2$ .

et minimiser la fonction  $L_R(\tilde{f}, 0)$  qui est une fonction du LLR du syndrome calculée à partir des parties réelles des échantillons reçus, n'aboutit plus à une estimée du résidu de porteuse.

Introduisons maintenant la fonction

$$L_I(\tilde{f}, 0) = E \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Im(r(k_j) e^{-j(2\pi k_j \tilde{f} T_s)})) \right) \cdot \min_{j=1, \dots, u_k} |\Im(r(k_j) e^{-j(2\pi k_j \tilde{f} T_s)})| \right) \right], \quad (3.18)$$

qui est une fonction du LLR du syndrome calculée à partir des parties imaginaires ( $\Im(\cdot)$ ) des échantillons reçus. Reprenons le cas où  $\theta_0 = \pi/2$  et traçons la fonction  $L_I(\tilde{f}, 0)$  sur la figure 3.6.  $L_I(\tilde{f}, 0)$  étant minimale pour  $\tilde{f} = f_0$ , minimiser, dans ce cas, cette fonction en termes de  $\tilde{f}$  donne une bonne estimation du résidu de porteuse  $f_0$ .

Malheureusement, le déphasage  $\theta_0$  du canal est inconnu et par suite, nous sommes incapables de choisir quelle fonction il faut minimiser ( $L_R(\tilde{f}, 0)$  ou  $L_I(\tilde{f}, 0)$ ) pour estimer le résidu de porteuse. Pour remédier à ce problème, nous combinons ces deux fonctions ensemble et nous introduisons une nouvelle fonction de coût définie par :

$$I(\tilde{f}, 0) = L_R(\tilde{f}, 0) + L_I(\tilde{f}, 0). \quad (3.19)$$

Ainsi, quelle que soit la valeur du déphasage  $\theta_0$ , minimiser maintenant la fonction  $I(\tilde{f}, 0)$  en fonction de  $\tilde{f}$  en appliquant le critère du C-SFM, aboutit à une estimée du

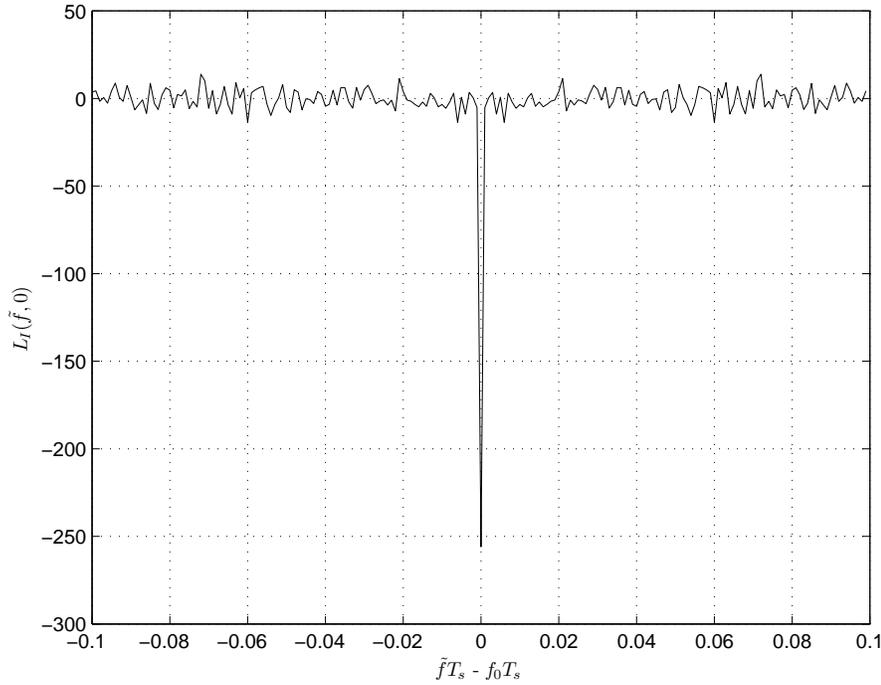


Figure 3.6 — La fonction  $L_I(\tilde{f}, 0)$  tracée en présence d'un déphasage  $\theta_0 = \pi/2$ .

résidu de porteuse du système et ceci indépendamment de la valeur de ce déphasage inconnu. Notons aussi que le fait de considérer les deux fonctions  $L_R(\tilde{f}, 0)$  et  $L_I(\tilde{f}, 0)$  conjointement réduit l'effet du bruit du canal de transmission.

Remarquons finalement que (3.18) est estimée par :

$$\hat{L}_I(\tilde{f}, 0) = \frac{1}{K} \sum_{i=0}^{K-1} \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Im(r(k_j + in_c) e^{-j2\pi(k_j + in_c)\tilde{f}T_s})) \right) \right) \cdot \min_{j=1, \dots, u_k} |\Im(r(k_j + in_c) e^{-j2\pi(k_j + in_c)\tilde{f}T_s})| \right]. \quad (3.20)$$

En utilisant (3.14) et (3.20), la fonction de coût de (3.19) s'écrit :

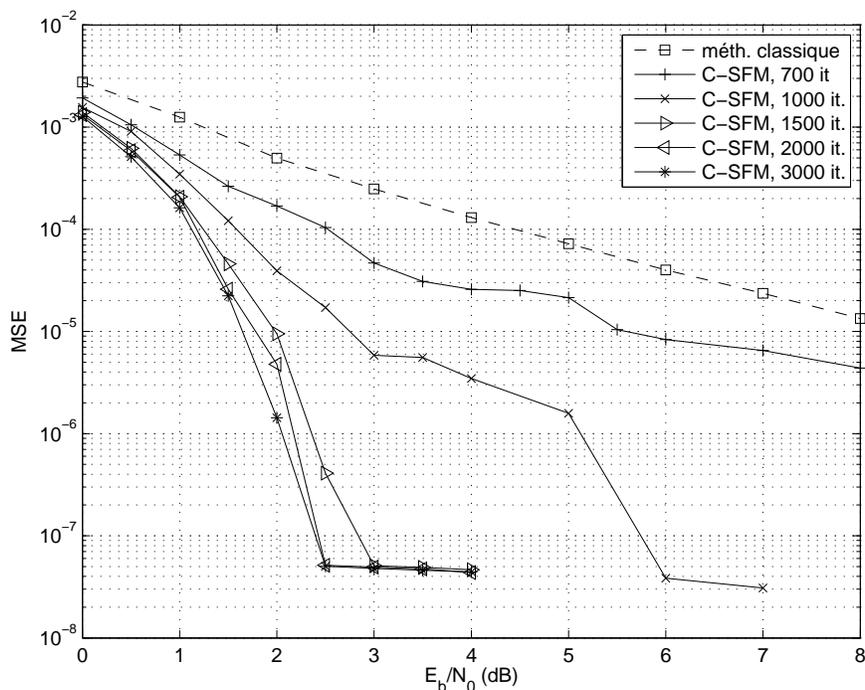
$$\hat{I}(\tilde{f}, 0) = \hat{L}_R(\tilde{f}, 0) + \hat{L}_I(\tilde{f}, 0) \quad (3.21)$$

et le résidu de porteuse du système est estimé par :

$$\hat{f}_{CSFM} = \underset{\tilde{f}}{\operatorname{argmin}} \hat{I}(\tilde{f}, 0). \quad (3.22)$$

### 3.3.1 Résultats des simulations

Evaluons maintenant les performances de l'algorithme du C-SFM dans le cas d'un canal de transmission à déphasage inconnu  $\theta_0$ . Pour les simulations que nous avons effectuées, un résidu de porteuse a été aléatoirement généré entre  $-0.1/T_s$  et  $0.1/T_s$ , en



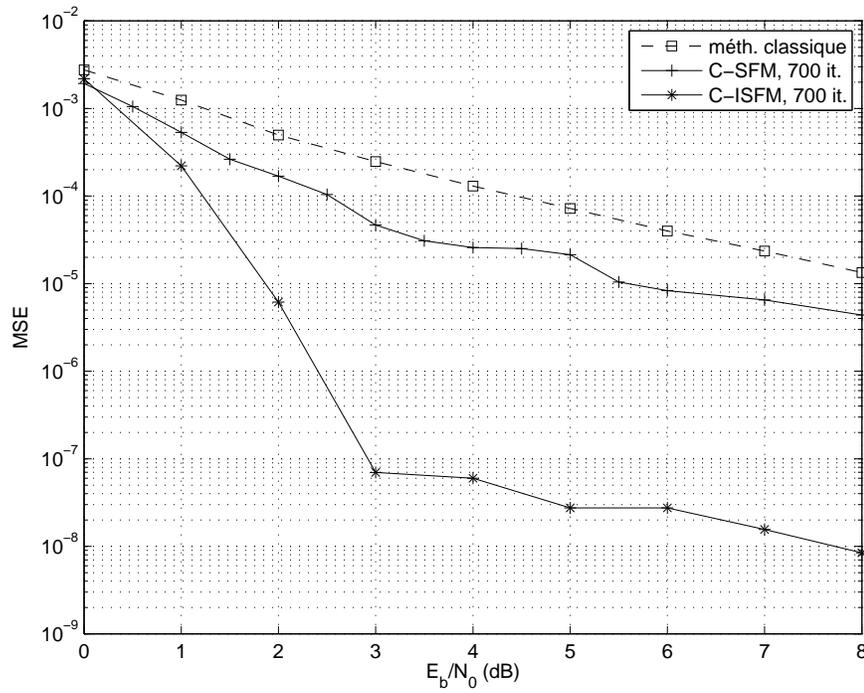
**Figure 3.7** — Performances de l'algorithme du C-SFM appliqué pour l'estimation aveugle du résidu de porteuse en présence d'un déphasage inconnu par le récepteur.

présence d'un déphasage  $\theta_0$  tiré suivant une loi uniforme dans l'intervalle  $[-\pi/2, \pi/2]$ . La figure 3.7 montre les courbes de MSE en fonction du  $E_b/N_0$  obtenues après avoir appliqué l'algorithme du C-SFM. Comme nous l'observons, notre méthode d'estimation aveugle du résidu de porteuse surpasse largement la méthode classique utilisée dans la littérature. Une MSE d'environ  $5 \cdot 10^{-8}$  est atteinte pour un  $E_b/N_0$  égal seulement à 2.5 dB et ceci pour 1500 itérations du Recuit Simulé.

Appliquons maintenant l'algorithme du C-ISFM proposé au paragraphe 3.2.2 de ce chapitre. Les courbes de MSE correspondantes sont tracées sur la figure 3.8. Pour seulement 700 itérations du Recuit Simulé, on atteint maintenant une MSE de  $7 \cdot 10^{-8}$  pour un  $E_b/N_0$  égal à 3 dB. La différence entre les performances de l'algorithme du C-ISFM et de la méthode classique est bien visible.

### 3.4 Conclusion

Nous avons considéré dans ce chapitre le problème d'estimation du résidu de porteuse. Nous avons proposé des algorithmes originaux d'estimation aveugle basés sur le calcul et la minimisation de fonctions du LLR du syndrome obtenues à partir des parties réelles et imaginaires des échantillons reçus. Due à la présence de plusieurs minima locaux dans ces fonctions, la procédure de minimisation est effectuée en utilisant l'algorithme itératif du Recuit Simulé. Les résultats des simulations montrent que



**Figure 3.8** — Performances de l'algorithme du C-ISFM appliqué pour l'estimation aveugle du résidu de porteuse en présence d'un déphasage inconnu par le récepteur.

nos techniques d'estimation aveugle du résidu de porteuse pour un système codé sont très performantes. De plus, la méthode d'estimation proposée est capable d'estimer le résidu de porteuse indépendamment de la valeur du déphasage.

Le problème d'estimation du déphasage introduit par le canal de propagation est considéré dans le chapitre suivant.

# Chapitre 4

## Synchronisation aveugle de la phase

NOUS considérons dans ce chapitre le problème d'estimation du déphasage introduit par le canal de transmission. Bien que les systèmes de communication actuels soient très efficaces dans le cas d'une détection cohérente, une dégradation non négligeable dans leurs performances est observée en présence d'un déphasage inconnu. Ainsi, l'estimation de ce déphasage avec une erreur d'estimation minimale est exigée pour une transmission efficace.

Nous supposons dans la suite de ce document que l'instant de synchronisation trame et l'instant de synchronisation symbole sont parfaitement estimés au récepteur. Dans ces conditions, un échantillon reçu s'écrit sous la forme :

$$r(k) = b(k)e^{j(2\pi k f_0 T_s + \theta_0)} + w(k), \quad (4.1)$$

où  $\theta_0$  est le déphasage inconnu que nous cherchons à estimer.

Nous admettons dans la première partie de ce chapitre que le résidu de porteuse du système est parfaitement estimé au récepteur. Ainsi, en compensant son effet des échantillons reçus nous obtenons :

$$r_{f_0}(k) = r(k)e^{-j2\pi k f_0 T_s}. \quad (4.2)$$

Nous proposons dans ce chapitre deux techniques aveugles d'estimation du déphasage. Ces techniques sont basées sur le même concept étudié et développé tout au long de cette thèse et qui est fondé sur le calcul de fonctions de LLR de syndrome obtenues à partir de la matrice de contrôle de parité du code correcteur d'erreurs. Ces deux techniques d'estimation ont été présentées respectivement dans les conférences SPAWC 2008 et SPAWC 2009. Les résultats des simulations montrent que la deuxième technique proposée (algorithme du P-SFM), qui est basée sur la minimisation d'une fonction de coût, présente de très bonnes performances en termes de MSE et surpasse toutes les méthodes testées. Cependant, ces méthodes d'estimation du déphasage ne sont pas robustes en présence d'un résidu de porteuse inconnu. Une solution à ce problème est d'estimer tout d'abord le résidu de porteuse par le C-SFM et ensuite le déphasage par le P-SFM. De très bons résultats en termes de BER sont ainsi obtenus.

## 4.1 Etat de l'art

Le problème d'estimation du déphasage introduit par le canal de propagation a toujours été un centre d'intérêt des chercheurs du domaine de traitement de signal pour les communications numériques. Dans la littérature, deux approches de synchronisation sont souvent utilisées : l'algorithme conventionnel du NDA (Non Data Aided) [67] et le HDD (Hard Decision Directed) [55]. Ces approches supposent que seulement le type de modulation utilisée à la transmission est connu par le récepteur, ce qui est généralement le cas. Basé sur un critère de MV, un algorithme d'estimation du déphasage est présenté dans [16] où seulement le cas des constellations de type MPSK a été considéré. Les résultats de [16] ont été étendus par Moeneclaey et de Jonghe dans [57] pour couvrir d'autres types de constellations comme par exemple les QAM. Les auteurs de cet article ont démontré que le synchroniseur à puissance  $P$  est optimal (dans le sens du MV) pour de faibles SNR. Le synchroniseur à puissance  $P$  estime la phase du canal par :

$$\hat{\theta}_P = \frac{1}{P} \arg \left[ E [b(k)^{*P}] \sum_{k=1}^N r_{f_0}(k)^P \right]. \quad (4.3)$$

Pour le cas BPSK, la variable  $P$  dans l'équation ci-dessus est égale à 2, on retrouve alors le synchroniseur proposé dans [54]. Pour toute modulation de type MPSK,  $P = M$  [21, 16]. Quant aux modulations de type QAM, l'estimateur de phase optimal est obtenu pour  $P = 4$  [57]. Sans perte de généralité, on suppose que :

$$E [b(k)^2] = 1, \quad (4.4)$$

ce qui correspond à une constellation à énergie moyenne égale à l'unité. Des algorithmes dérivés du MV et du synchroniseur à puissance  $P$  ont été présentés dans [23].

Utilisé généralement comme une référence pour plusieurs algorithmes de récupération de phase, l'algorithme HDD estime le déphasage par :

$$\hat{\theta}_{HDD} = \arg \left( \sum_{k=1}^N r_{f_0}(k) \hat{d}(k)^* \right), \quad (4.5)$$

où les symboles  $\{\hat{d}(k)\}_{k=1, \dots, N}$  désignent des décisions fermes d'estimées de  $N$  symboles transmis. Ces décisions sont obtenues à partir des échantillons  $\{r_{f_0}(k)\}_{k=1, \dots, N}$ . L'opérateur (\*) représente la valeur conjuguée d'un nombre complexe.

Quand les méthodes conventionnelles présentées ci-dessus sont appliquées pour l'estimation de la phase, l'estimée résultante présente une ambiguïté de phase due aux symétries rotationnelles de la constellation utilisée. La suppression de cette ambiguïté peut être effectuée par un algorithme DA (Data Aided) qui exploite une séquence pilote insérée dans le flux de données transmises et connue au récepteur [8]. Par conséquent, une perte d'efficacité spectrale est attendue. La procédure de suppression de l'ambiguïté de phase peut être remplacée par un codage différentiel. Ceci entraîne malheureusement une dégradation dans le BER résultant et exige des changements au niveau du décodeur dans le cas d'un décodage itératif [27].

Les algorithmes conventionnels d'estimation de la phase présentent de bonnes performances pour des SNR élevés. Cependant, si des codes correcteurs d'erreurs sont utilisés par le système de transmission, exploiter les propriétés de ces codes pourrait aboutir à de meilleures performances. Plusieurs algorithmes ont proposé de combiner la procédure d'estimation de la phase avec le décodage itératif, comme par exemple dans [46, 56, 59, 70]. Dans [69], en plus de l'estimation conjointe de la phase et du décodage, les auteurs traitent le problème de suppression de l'ambiguïté de phase. Une méthode d'estimation du déphasage appliquée à des transmissions par satellite a été proposée dans [17]. Cette méthode exploite la redondance apportée par un code ayant une seule équation de parité et qui a été spécialement introduit dans le système de transmission pour effectuer la procédure de synchronisation.

Nous proposons dans ce chapitre de nouvelles techniques d'estimation du déphasage introduit par le canal de propagation. Ces techniques, basées sur les propriétés du code correcteur d'erreurs utilisé dans la chaîne de transmission, sont présentées dans la suite.

## 4.2 Estimation du déphasage par une expression analytique - Algorithme du SFR

Les techniques d'estimation du déphasage que nous proposons dans ce chapitre sont basées sur le même critère : calcul des fonctions du LLR du syndrome du bloc reçu, à partir de la matrice de contrôle de parité du code correcteur d'erreurs. Dans ce chapitre et sauf indication contraire, nous supposons que le résidu de porteuse du système est connu par le récepteur.

Réécrivons les fonctions de coût introduites dans (3.6) et (3.18) du chapitre 3 en compensant le résidu de porteuse présent dans les échantillons reçus par sa valeur connue ( $f_0$ ) et sans compenser le déphasage (le déphasage est supposé inconnu et notre but est de l'estimer). Ces fonctions désignant des fonctions du LLR du syndrome calculées à partir des parties réelles et imaginaires des échantillons reçus sont respectivement données par :

$$L_R(f_0, 0) = E \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Re(r_{f_0}(k_j))) \right) \min_{j=1, \dots, u_k} |\Re(r_{f_0}(k_j))| \right) \right] \quad (4.6)$$

et

$$L_I(f_0, 0) = E \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Im(r_{f_0}(k_j))) \right) \min_{j=1, \dots, u_k} |\Im(r_{f_0}(k_j))| \right) \right] \quad (4.7)$$

Rappelons que  $u_k$  désigne le nombre d'éléments non nuls dans la  $k^{\text{ième}}$  ligne de  $H$  et  $k_j$  est la  $j^{\text{ième}}$  position de cette  $k^{\text{ième}}$  ligne.

Les équations (4.6) et (4.7) sont estimées par :

$$\hat{L}_R(f_0, 0) = \frac{1}{K} \sum_{i=0}^{K-1} \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Re(r_{f_0}(k_j + in_c))) \right) \right) \cdot \min_{j=1, \dots, u_k} |\Re(r_{f_0}(k_j + in_c))| \right] \quad (4.8)$$

et

$$\hat{L}_I(f_0, 0) = \frac{1}{K} \sum_{i=0}^{K-1} \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Im(r_{f_0}(k_j + in_c))) \right) \right) \cdot \min_{j=1, \dots, u_k} |\Im(r_{f_0}(k_j + in_c))| \right], \quad (4.9)$$

où  $K$  désigne le nombre de mots de codes utilisés pour estimer les espérances mathématiques. Dans la suite, nous supposons que le déphasage  $\theta_0$  est constant sur ces  $K$  mots de code.

Développons maintenant les expressions des fonctions du LLR du syndrome présentées dans (4.8) et (4.9). Elles sont égales à :

$$\hat{L}_R(f_0, 0) = \frac{1}{K} \sum_{i=0}^{K-1} \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(b(k_j + in_c) \cos \theta_0 + w_1(k_j + in_c)) \right) \right) \cdot \min_{j=1, \dots, u_k} |b(k_j + in_c) \cos \theta_0 + w_1(k_j + in_c)| \right] \quad (4.10)$$

et

$$\hat{L}_I(f_0, 0) = \frac{1}{K} \sum_{i=0}^{K-1} \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(b(k_j + in_c) \sin \theta_0 + w_2(k_j + in_c)) \right) \right) \cdot \min_{j=1, \dots, u_k} |b(k_j + in_c) \sin \theta_0 + w_2(k_j + in_c)| \right]. \quad (4.11)$$

Les variables  $w_1(k)$  et  $w_2(k)$  désignent respectivement les composantes réelles et imaginaires du bruit  $w_{f_0}(k)$  défini par :

$$w_{f_0}(k) = w(k) e^{-j2\pi k f_0 T_s}. \quad (4.12)$$

Soit maintenant la fonction  $L_b$  qui correspond à la valeur du critère sans bruit et sans déphasage. Elle est définie par :

$$L_b = \frac{1}{K} \sum_{i=0}^{K-1} \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(b(k_j + in_c)) \right) \min_{j=1, \dots, u_k} |b(k_j + in_c)| \right) \right]. \quad (4.13)$$

Pour le cas d'une modulation BPSK, nous avons :

$$L_b = -n_r \quad \forall u_k. \quad (4.14)$$

Afin de comprendre l'intérêt des deux fonctions introduites dans (4.6) et (4.7), étudions les deux cas ci-dessous pour un canal supposé sans bruit.

**1<sup>er</sup> cas :  $u_k$  est pair :**

Dans le cas d'un canal sans bruit, (4.10) et (4.11) s'écrivent sous la forme :

$$\hat{L}_R(f_0, 0) = L_b |\cos \theta_0| = -n_r |\cos \theta_0| \quad (4.15)$$

$$\hat{L}_I(f_0, 0) = L_b |\sin \theta_0| = -n_r |\sin \theta_0| \quad (4.16)$$

A partir des deux équations ci-dessus, nous déduisons que :

$$|\tan \theta_0| = \frac{\hat{L}_I(f_0, 0)}{\hat{L}_R(f_0, 0)}. \quad (4.17)$$

**2<sup>ième</sup> cas :  $u_k$  est impair :**

Quand  $u_k$  est impair et en absence de bruit, nous avons :

$$\hat{L}_R(f_0, 0) = \begin{cases} -n_r |\cos \theta_0| & \text{si } \theta_0 \in [0, \frac{\pi}{2}] \cup [\frac{3\pi}{2}, 2\pi] \\ n_r |\cos \theta_0| & \text{si } \theta_0 \in [\frac{\pi}{2}, \frac{3\pi}{2}] \end{cases} \quad (4.18)$$

$$\hat{L}_I(f_0, 0) = \begin{cases} -n_r |\sin \theta_0| & \text{si } \theta_0 \in [0, \pi] \\ n_r |\sin \theta_0| & \text{si } \theta_0 \in [\pi, 2\pi] \end{cases} \quad (4.19)$$

Ainsi, quelle que soit la valeur de  $\theta_0$  nous avons :

$$|\tan \theta_0| = \pm \frac{\hat{L}_I(f_0, 0)}{\hat{L}_R(f_0, 0)}. \quad (4.20)$$

Les équations (4.17) et (4.20) nous donnent une expression analytique de  $\theta_0$  lorsque le bruit est nul. En présence de bruit, nous pouvons toujours utiliser cette expression pour avoir une estimée du déphasage. Cette estimée est donnée par :

$$\hat{\theta}_{SFR} = \text{atan} \left( \pm \frac{\hat{L}_I(f_0, 0)}{\hat{L}_R(f_0, 0)} \right) \quad \forall u_k. \quad (4.21)$$

Lorsque la variance du bruit  $\sigma^2$  tend vers 0,  $\hat{\theta}_{SFR}$  tend vers  $\theta_0$ . Une dégradation des performances d'estimation est prévisible lorsque  $\sigma^2$  augmente.

Remarquons maintenant que (4.21) donne une estimée de  $\theta_0$  avec une ambiguïté de  $\frac{\pi}{2}$ . Cependant, nous pouvons faire évoluer ce critère pour aboutir à une estimation du déphasage avec une ambiguïté  $\pi$ . En effet, il suffit de calculer  $\hat{L}_R(f_0, \delta\theta)$  et  $\hat{L}_I(f_0, \delta\theta)$ , où  $\delta\theta$  est une phase quelconque. L'étude de l'évolution des fonctions  $\hat{L}_R$  et  $\hat{L}_I$  en fonction de  $\delta\theta$  détermine les deux quadrants possibles d'appartenance de  $\theta_0$  et donne ainsi une estimée du déphasage avec une ambiguïté de  $\pi$ . Nous avons :

**Pour  $u_k$  pair :**

Selon (4.15) et (4.16), nous avons :

- Si  $\theta_0 \in [0, \frac{\pi}{2}] \cup [\pi, \frac{3\pi}{2}]$ ,  $\hat{L}_R$  est une fonction croissante et  $\hat{L}_I$  est décroissante.
- Si  $\theta_0 \in [\frac{\pi}{2}, \pi] \cup [\frac{3\pi}{2}, 2\pi]$ ,  $\hat{L}_R$  est une fonction décroissante et  $\hat{L}_I$  est croissante.

Pour  $u_k$  impair :

Les équations (4.18) et (4.19) montrent qu'il est difficile d'étudier les variations des fonctions  $\hat{L}_R$  et  $\hat{L}_I$  pour le cas d'un  $u_k$  impair. Cependant, en remarquant que :

$$|\hat{L}_R(f_0, 0)| = n_r |\cos \theta_0| \quad (4.22)$$

et

$$|\hat{L}_I(f_0, 0)| = n_r |\sin \theta_0| \quad (4.23)$$

nous obtenons :

- Si  $\theta_0 \in [0, \frac{\pi}{2}] \cup [\pi, \frac{3\pi}{2}]$ ,  $|\hat{L}_R|$  est une fonction décroissante et  $|\hat{L}_I|$  est croissante.
- Si  $\theta_0 \in [\frac{\pi}{2}, \pi] \cup [\frac{3\pi}{2}, 2\pi]$ ,  $|\hat{L}_R|$  est une fonction croissante et  $|\hat{L}_I|$  est décroissante.

Ainsi, appliquer (4.21) puis étudier l'évolution des fonctions  $\hat{L}_R$  et  $\hat{L}_I$  (où  $|\hat{L}_R|$  et  $|\hat{L}_I|$ ) nous permet d'avoir une estimée du déphasage avec une ambiguïté de  $\pi$ .

Dans la suite de ce chapitre, l'algorithme d'estimation du déphasage proposé dans ce paragraphe sera désigné par l'algorithme du SFR (Syndrome Functions Ratio).

### 4.2.1 Résultats des simulations

Les performances de l'algorithme du SFR sont évaluées en traçant les courbes de MSE en fonction du  $E_b/N_0$ . Des simulations de Monte Carlo ont été effectuées où, pour chaque réalisation, un déphasage a été tiré aléatoirement et uniformément entre  $-\pi/2$  et  $\pi/2$ . Les performances de la méthode du SFR sont comparées à celles de l'algorithme du HDD et du synchroniseur à Puissance  $P$  qui estiment le déphasage par respectivement (4.5) et (4.3). Rappelons que le type de la modulation utilisée est une BPSK. Dans ce cas, (4.3) devient égale à :

$$\hat{\theta}_P = \frac{1}{2} \arg \left[ \sum_{k=1}^N r(k)^2 \right]. \quad (4.24)$$

Dans la suite de ce chapitre, nous supposons que  $N = n_c$ . Appliquons maintenant ces méthodes d'estimation à deux codes LDPC de taille  $n_c = 512$  bits et de rendement  $R = 0.5$ . Ces deux codes diffèrent seulement par le nombre d'éléments non nuls se trouvant dans chaque ligne de leur matrice de contrôle de parité. Code I a  $u_k = 4$  (pair) tandis que Code II a  $u_k = 5$  (impair). Notons que les performances de l'algorithme du HDD et du synchroniseur à puissance  $P$  sont indépendantes des propriétés du code correcteur d'erreurs utilisé (et par suite de la valeur de  $u_k$ ), ce qui n'est pas le cas pour le SFR. Les courbes de MSE obtenues après avoir appliqué ces méthodes d'estimation sont tracées sur la figure 4.1. Sur cette figure, nous observons que notre méthode d'estimation est plus performante que l'algorithme du HDD. En revanche, elle n'arrive pas à surpasser le synchroniseur à puissance  $P$  qui se montre plus performant.

Remarquons aussi que les performances de la méthode du SFR se dégradent quand le nombre d'éléments non nuls dans chaque ligne de la matrice de contrôle de parité augmente. Rappelons qu'au paragraphe 1.4, nous avons étudié l'influence de la densité de la matrice de contrôle de parité sur les performances de la méthode de synchronisation trame du SSS. Cette même analyse reste valide pour les méthodes d'estimation

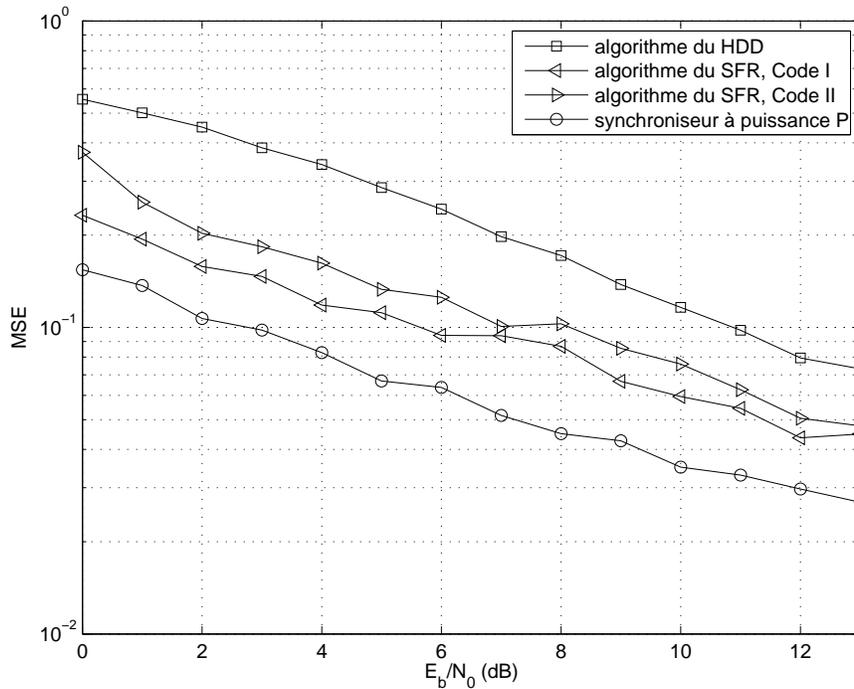


Figure 4.1 — Estimation aveugle du déphasage par l'algorithme du SFR. Le résidu de porteuse du système est supposé connu par le récepteur.

du déphasage proposées dans ce chapitre.

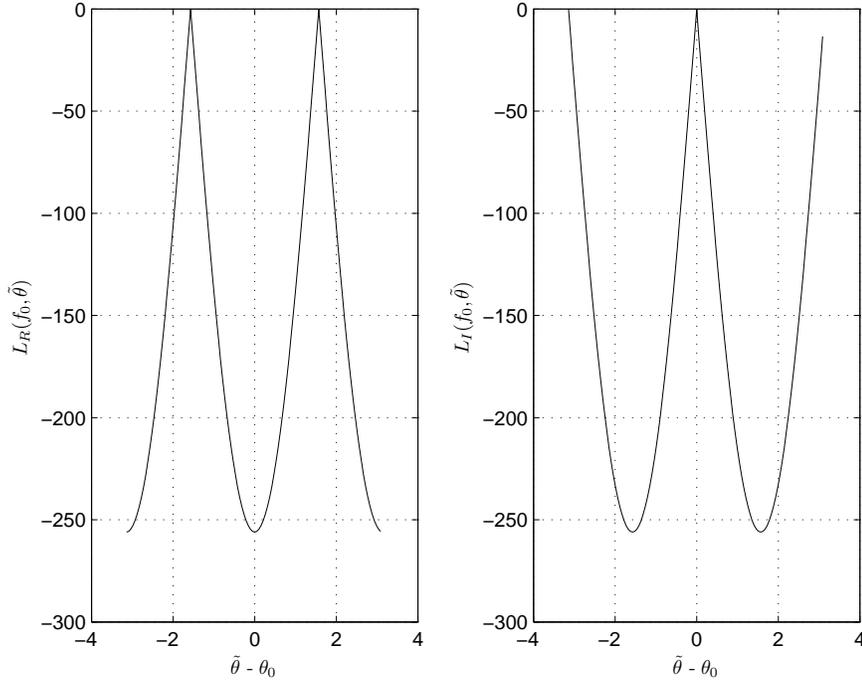
Nous proposons dans le paragraphe suivant une autre méthode d'estimation du déphasage introduit par le canal. Cette méthode d'estimation est basée sur le calcul et la minimisation d'une fonction du LLR du syndrome.

Dans le reste de ce chapitre et pour des raisons de simplicité, nous supposons que  $u_k$  est pair et constant pour toutes les lignes de la matrice de contrôle de parité.

### 4.3 Estimation du déphasage par minimisation d'une fonction de coût - Algorithme du P-SFM

Dans ce paragraphe, nous proposons une nouvelle technique d'estimation du déphasage introduit par le canal de propagation. Cette technique très performante en termes de minimisation de MSE de l'estimation, est décrite dans la suite.

Rappelons que nous sommes toujours dans le cas d'un résidu de porteuse connu par le récepteur. Une fois les échantillons  $r(k)$  reçus, nous compensons tout d'abord le résidu de porteuse qui leur est introduit par sa valeur exacte  $f_0$ . Les échantillons  $r_{f_0}(k)$  sont alors formés. Ensuite, nous appliquons à ces échantillons une phase quelconque  $\theta$ . Ainsi,



**Figure 4.2** — Les fonctions  $L_R(f_0, \tilde{\theta})$  et  $L_I(f_0, \tilde{\theta})$  en fonction de l'erreur d'estimation du déphasage ( $\tilde{\theta} - \theta_0$ ).

les fonctions de coût de (4.6) et (4.7) s'écrivent maintenant :

$$L_R(f_0, \tilde{\theta}) = E \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Re(r_{f_0}(k_j)e^{-j\tilde{\theta}})) \right) \min_{j=1, \dots, u_k} |\Re(r_{f_0}(k_j)e^{-j\tilde{\theta}})| \right) \right] \quad (4.25)$$

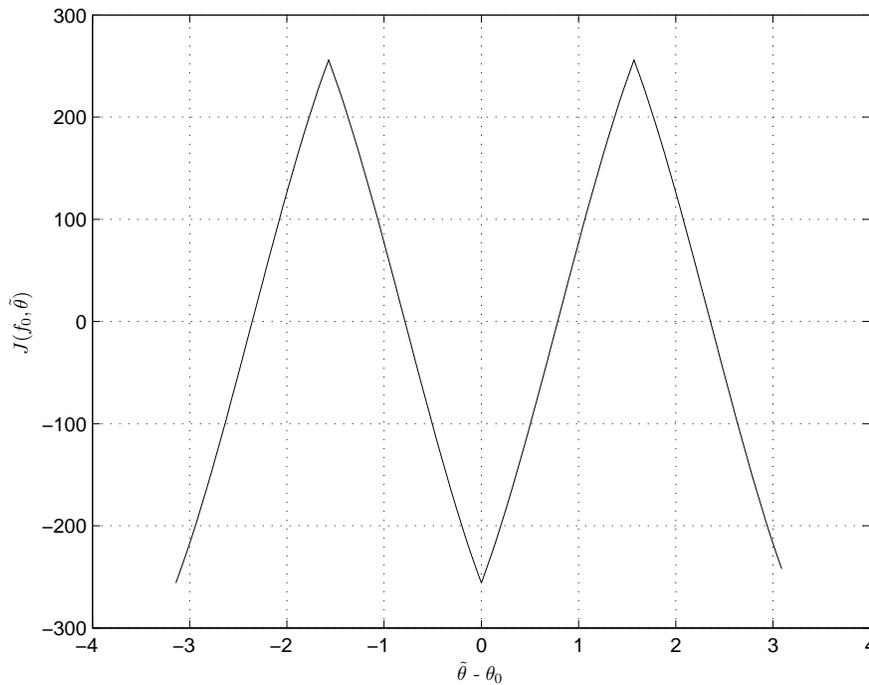
et

$$L_I(f_0, \tilde{\theta}) = E \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \left( \prod_{j=1}^{u_k} \text{sign}(\Im(r_{f_0}(k_j)e^{-j\tilde{\theta}})) \right) \min_{j=1, \dots, u_k} |\Im(r_{f_0}(k_j)e^{-j\tilde{\theta}})| \right) \right] \quad (4.26)$$

Nous traçons sur la figure 4.2 les variations de  $L_R(f_0, \tilde{\theta})$  et  $L_I(f_0, \tilde{\theta})$  en fonction de l'erreur d'estimation du déphasage ( $\tilde{\theta} - \theta_0$ ). Le code utilisé est un code LDPC de taille  $n_c = 512$  bits, rendement  $R = 0.5$  et ayant  $u_k = 4$  éléments non nuls dans chaque ligne de sa matrice de contrôle de parité. A partir de la figure 4.2 nous remarquons que  $L_R(f_0, \tilde{\theta})$  est minimale pour une erreur d'estimation nulle (i.e  $\tilde{\theta} = \theta_0$ ), alors que  $L_I(f_0, \tilde{\theta})$  est maximale. Ainsi, afin d'estimer le déphasage introduit par le canal, nous définissons une nouvelle fonction de coût donnée par :

$$J(f_0, \tilde{\theta}) = L_R(f_0, \tilde{\theta}) - L_I(f_0, \tilde{\theta}). \quad (4.27)$$

Les variations de  $J(f_0, \tilde{\theta})$  en fonction de ( $\tilde{\theta} - \theta_0$ ) sont tracées sur la figure 4.3 pour le même code LDPC déjà utilisé. Sur cette figure, nous observons que  $J(f_0, \tilde{\theta})$  est minimale pour  $\tilde{\theta} = \theta_0$  (modulo  $\pi$ ). Par conséquent, minimiser la fonction  $J(f_0, \tilde{\theta})$  par rapport à  $\tilde{\theta}$  donne une estimée du déphasage avec une ambiguïté de  $\pi$ .



**Figure 4.3** — La fonction de coût  $J(f_0, \tilde{\theta})$  en fonction de l'erreur d'estimation du déphasage  $(\tilde{\theta} - \theta_0)$ .

### 4.3.1 Minimisation de la fonction de coût par la méthode de descente du Gradient

A partir de l'allure de la courbe tracée sur la figure 4.3, il est clair que  $J(f_0, \tilde{\theta})$  a un seul minimum localisé à une phase  $\tilde{\theta} = \theta_0$ . Ainsi, le problème d'optimisation que nous avons n'est pas si compliqué et peut être résolu par la méthode de descente du Gradient [2].

#### Méthode de descente du Gradient

La méthode de descente du Gradient est une technique d'optimisation applicable aux fonctions dérivables ayant un seul minimum global. Soit  $h(x)$  la fonction à minimiser. Basée sur un algorithme itératif, la méthode de descente du Gradient est décrite ci-dessous :

- A l'itération 0 :
  - \* Initialiser  $x^{(0)}$ , où  $x^{(m)}$  est la solution du problème de minimisation après  $m$  itérations.
- A l'itération  $m$  :
  - \* Calculer :

$$x^{(m)} = x^{(m-1)} - \epsilon^{(m-1)} \left. \frac{\partial h(x)}{\partial x} \right|_{x=x^{(m-1)}} \quad (4.28)$$

où

$$\left. \frac{\partial h(x)}{\partial x} \right|_{x=x^{(m-1)}} \quad (4.29)$$

représente la dérivée partielle de  $h$  par rapport à  $x$ , calculée pour  $x = x^{(m-1)}$ . La variable  $\epsilon$  désigne le pas de la procédure de descente. Dans la suite de ce chapitre, nous considérons le cas d'un Gradient à pas variable. Ainsi, un exemple simple de  $\epsilon^{(m)}$  est :

$$\epsilon^{(m)} = \frac{1}{Cm}, \quad (4.30)$$

où  $C$  est une constante à régler.

- Sortir quand le nombre maximal d'itérations est atteint ou bien qu'un critère d'arrêt est vérifié.

Dans notre cas, nous voulons minimiser la fonction  $J(f_0, \tilde{\theta})$  donnée par (4.27). Comme indiqué dans (4.28), la méthode du Gradient nécessite le calcul de la dérivée de la fonction à minimiser. Cependant, les fonctions  $L_R(f_0, \tilde{\theta})$  et  $L_I(f_0, \tilde{\theta})$  données respectivement par (4.25) et (4.26) ne sont pas dérivables. Toutefois, ces fonctions sont des approximations du LLR du syndrome qui ont été obtenues à partir de (1.27). En utilisant l'expression exacte du LLR d'un syndrome donnée en (1.25), nous définissons :

$$L_{R_e}(f_0, \tilde{\theta}) = E \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \operatorname{atanh} \left( \prod_{j=1}^{u_k} \tanh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right) \right) \right] \quad (4.31)$$

et

$$L_{I_e}(\tilde{\theta}) = E \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \operatorname{atanh} \left( \prod_{j=1}^{u_k} \tanh \left( \Im \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right) \right) \right]. \quad (4.32)$$

Les fonctions  $L_{R_e}(f_0, \tilde{\theta})$  et  $L_{I_e}(f_0, \tilde{\theta})$  sont des approximations respectives de  $L_R(f_0, \tilde{\theta})$  et  $L_I(f_0, \tilde{\theta})$ . Ainsi, minimiser (4.27) revient à minimiser

$$J_e(f_0, \tilde{\theta}) = L_{R_e}(f_0, \tilde{\theta}) - L_{I_e}(f_0, \tilde{\theta}). \quad (4.33)$$

Le calcul détaillé de la dérivée partielle de  $J_e(f_0, \tilde{\theta})$  est donné en Annexe C.1.

Remarquons que (4.31) et (4.32) sont estimées par :

$$\hat{L}_{R_e}(f_0, \tilde{\theta}) = \frac{1}{K} \sum_{i=0}^{K-1} \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \operatorname{atanh} \left( \prod_{j=1}^{u_k} \tanh \left( \Re \left( \frac{r_{f_0}(k_j + in_c)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right) \right) \right] \quad (4.34)$$

et

$$\hat{L}_{I_e}(f_0, \tilde{\theta}) = \frac{1}{K} \sum_{i=0}^{K-1} \left[ \sum_{k=1}^{n_r} \left( (-1)^{u_k+1} \operatorname{atanh} \left( \prod_{j=1}^{u_k} \tanh \left( \Im \left( \frac{r_{f_0}(k_j + in_c)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right) \right) \right]. \quad (4.35)$$

Ainsi, la fonction de coût à minimiser devient :

$$\hat{J}_e(f_0, \tilde{\theta}) = \hat{L}_{R_e}(f_0, \tilde{\theta}) - \hat{L}_{I_e}(f_0, \tilde{\theta}). \quad (4.36)$$

Finalement, la méthode que nous proposons et qui sera désignée par l'algorithme du P-SFM (Phase estimation-Syndrome Function Minimization), estime le déphasage par :

$$\hat{\theta}_{PSFM} = \underset{\tilde{\theta}}{\operatorname{argmin}} \hat{J}_e(f_0, \tilde{\theta}). \quad (4.37)$$

Notons que la variance  $\sigma^2$  introduite dans (4.34) et (4.35) est inconnue. Ainsi, pour des raisons de simplicité et comme proposé dans [49], la valeur de cette variance peut être remplacée par 1.

### 4.3.2 Résultats des simulations

Nous présentons dans ce paragraphe les résultats des simulations après application de la méthode du P-SFM pour l'estimation du déphasage. Le code testé est un code LDPC de longueur  $n_c = 512$  bits, ayant un rendement  $R = 0.5$  et  $u_k = 4$ . Les courbes de MSE en fonction du  $E_b/N_0$  sont tracées sur la figure 4.4. Sur cette même figure, nous traçons aussi les courbes obtenues après application de l'algorithme du HDD, du synchroniseur à puissance  $P$  et de la méthode du SFR proposée au paragraphe 4.2. Notons que pour l'algorithme du gradient utilisé dans la procédure d'estimation du P-SFM, nous avons choisi un pas

$$\epsilon^{(m)} = \frac{1}{30m} \quad (4.38)$$

et nous avons fixé le nombre d'itérations à 50. A l'itération 0 du Gradient, la phase  $\tilde{\theta}$  est initialisée aléatoirement. En observant la figure 4.4, il est clair que la méthode d'estimation du P-SFM proposée au paragraphe 4.3 est très puissante et ses performances dépassent largement celles de toutes les autres méthodes testées. En appliquant cette méthode d'estimation, une MSE d'environ  $4.10^{-3}$  est atteinte pour un  $E_b/N_0$  égal à 3 dB.

Afin d'étudier l'efficacité de l'algorithme du P-SFM pour l'estimation aveugle du déphasage introduit par le canal de transmission, nous traçons sur la figure 4.5 les courbes de BER obtenues après estimation du déphasage puis décodage du code LDPC par un décodeur BP. Huit itérations de ce décodeur ont été effectuées et le code LDPC testé est le même que celui que nous venons d'utiliser. A partir de la figure 4.5, il est clair que si nous appliquons le synchroniseur à puissance  $P$ , une dégradation importante des performances est notée par rapport au cas d'une détection cohérente. Par contre, lorsque l'on applique la technique du P-SFM, on obtient une courbe très proche de celle du cas d'une détection cohérente. Pour un BER égal à  $10^{-3}$ , la différence entre les deux courbes est inférieure à 0.2 dB.

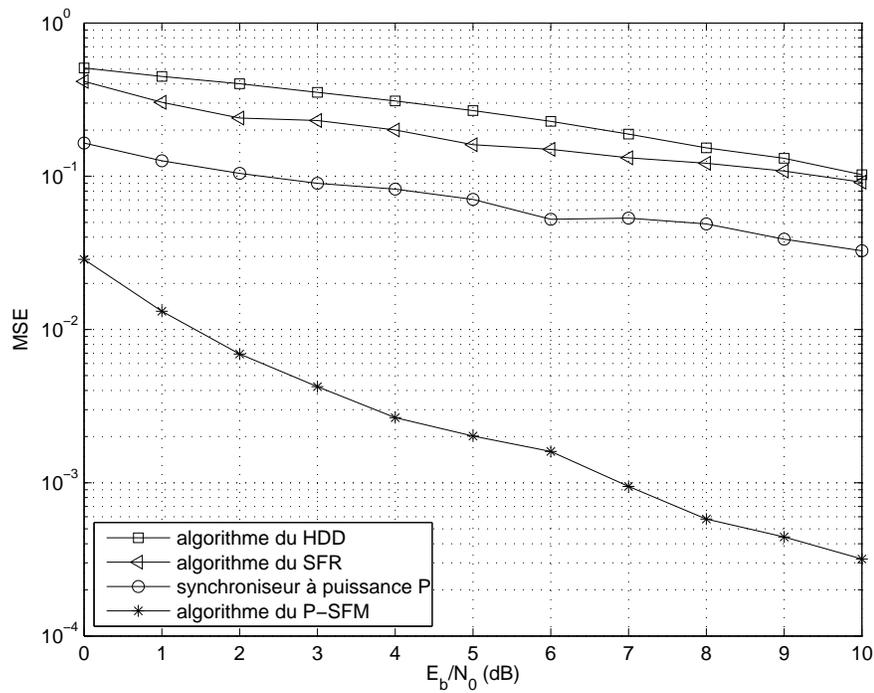


Figure 4.4 — MSE de l'estimation du déphasage en présence d'un résidu de porteuse connu par le récepteur.

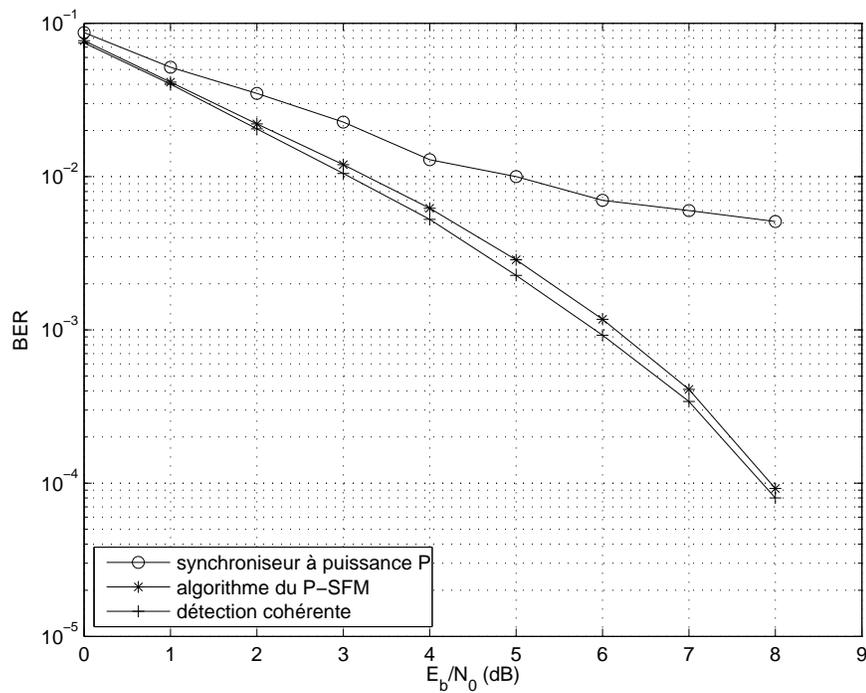


Figure 4.5 — Courbes de BER obtenues après estimation du déphasage puis application du décodeur BP. Les courbes sont tracées en présence d'un résidu de porteuse connu par le récepteur.

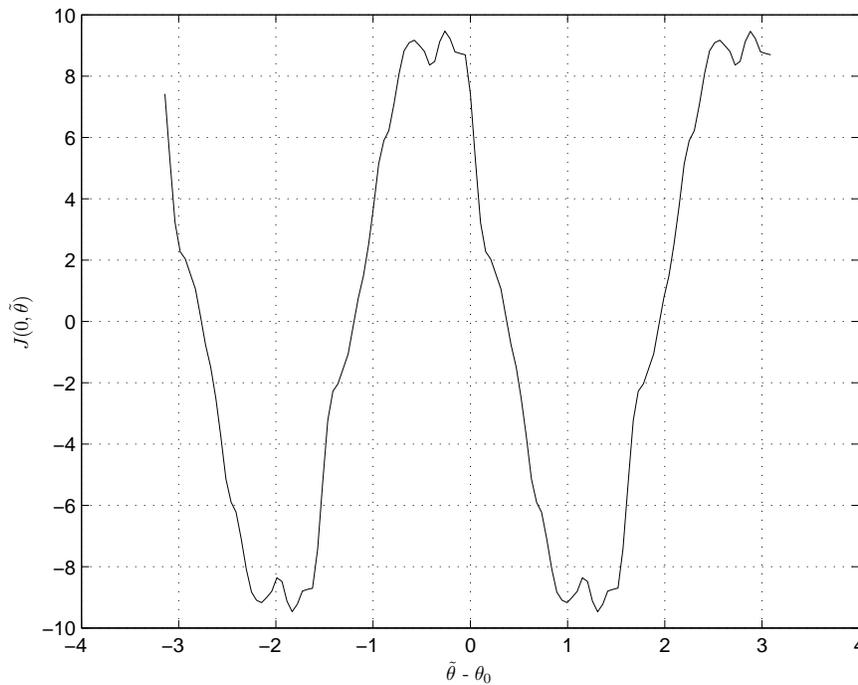
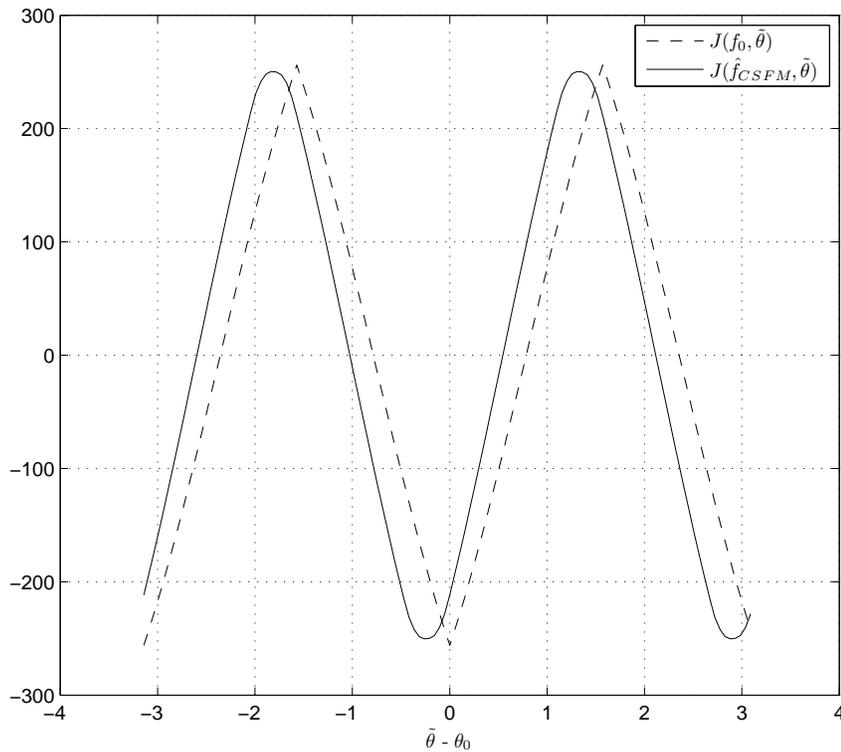


Figure 4.6 — La fonction de coût  $J(0, \tilde{\theta})$  en présence d'un résidu de porteuse inconnu.

## 4.4 Performances de l'algorithme du P-SFM après estimation du résidu de porteuse par le C-SFM

Jusqu'à présent, nous avons supposé dans ce chapitre que le résidu de porteuse du système est connu par le récepteur. Ceci n'est pas toujours le cas. Nous considérons maintenant qu'un résidu de porteuse inconnu est présent dans le système de transmission et notre but est d'estimer le déphasage introduit par le canal de propagation. Dans le chapitre précédent, nous avons montré qu'un déphasage inconnu n'affecte pas l'estimation du CFO. Etudions maintenant l'effet d'un CFO inconnu sur l'estimation du déphasage. Pour cela, nous traçons sur la figure 4.6 la fonction  $J(0, \tilde{\theta})$  donnée dans (4.27), en présence d'un résidu de porteuse inconnu et aléatoirement choisi entre  $-0.1/T_s$  et  $0.1/T_s$ . A partir de cette figure, il est clair qu'en présence d'un résidu de porteuse inconnu, notre méthode basée sur la minimisation de  $J(0, \tilde{\theta})$  (critère du P-SFM) n'est plus capable d'estimer le déphasage (pour  $\tilde{\theta} = \theta_0$ ,  $J(0, \tilde{\theta})$  n'est pas minimale). Donc, avant d'estimer le déphasage, il faut tout d'abord estimer le résidu de porteuse du système.

Soit  $\hat{f}_{CSFM}$  l'estimée du résidu de porteuse obtenue par la méthode du C-SFM proposée au chapitre 3. Etudions maintenant la fonction  $J(\hat{f}_{CSFM}, \tilde{\theta})$  obtenue après compensation respective du résidu de porteuse et du déphasage présents dans les échantillons reçus par  $\hat{f}_{CSFM}$  et  $\tilde{\theta}$ . Cette fonction est tracée sur la figure 4.7 où elle est comparée à  $J(f_0, \tilde{\theta})$ . Comme nous le voyons sur cette figure, minimiser maintenant la fonction  $J(\hat{f}_{CSFM}, \tilde{\theta})$  en fonction de  $\tilde{\theta}$  aboutit à une estimée du déphasage. Cependant, un biais



**Figure 4.7** — La fonction de coût  $J(\hat{f}_{CSFM}, \tilde{\theta})$  après compensation par une estimée du CFO et un déphasage quelconque  $\tilde{\theta}$ .

apparaît lors de l'estimation de la phase et ceci est dû à l'erreur d'estimation du résidu de porteuse. Ce biais tend vers zéro quand l'erreur d'estimation du CFO tend vers zéro.

Afin d'évaluer l'influence de ce biais sur les performances de la méthode du P-SFM, nous traçons sur la figure 4.8 la courbe correspondante de MSE. Sur cette même figure, nous traçons aussi les courbes de MSE de l'algorithme HDD et du synchroniseur à puissance  $P$ . A partir de cette figure, la différence qui existe entre les performances du P-SFM et des algorithmes existants est claire.

Nous traçons maintenant les courbes de BER obtenues à la sortie du décodeur BP appliqué après les procédures d'estimation du CFO et du déphasage. Les résultats des simulations sont présentés sur la figure 4.9. Concernant nos techniques d'estimations, 1500 itérations du Recuit Simulé ont été effectuées pour estimer le CFO et 50 itérations du Gradient pour estimer le déphasage. Pour l'algorithme du BP, 8 itérations étaient suffisantes. Comparées au cas d'une détection cohérente, une petite dégradation est observée dans les performances du décodeur BP après application des techniques d'estimation du C-SFM et P-SFM. Pour un BER de  $10^{-3}$ , la différence entre les deux courbes est seulement de  $0.2 \text{ dB}$ . D'un autre côté, il est clair qu'appliquer les méthodes conventionnelles d'estimation du CFO et de la phase donne de très mauvais résultats.

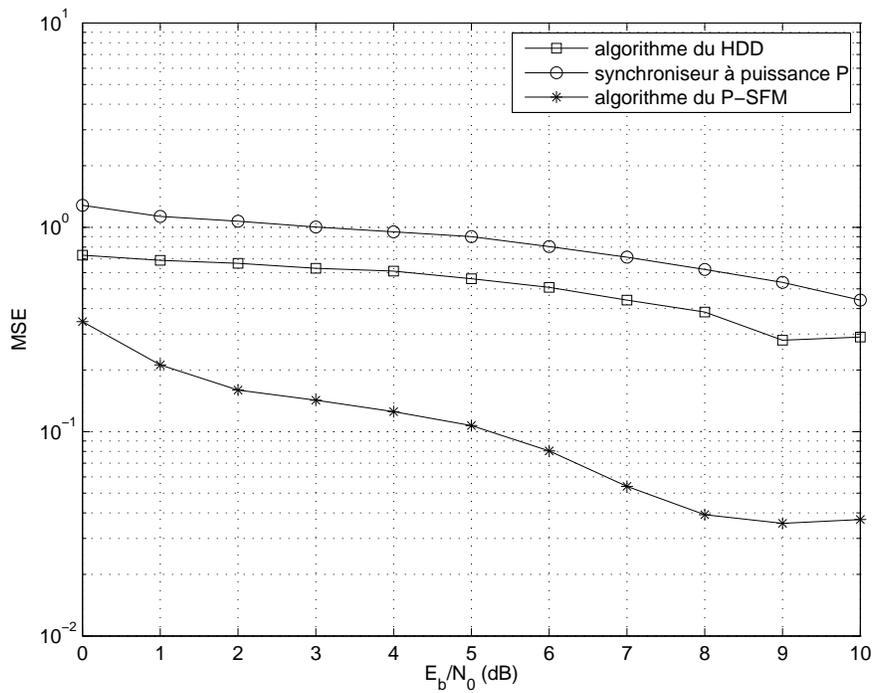


Figure 4.8 — MSE de l'estimation de phase obtenu après estimation du résidu de porteuse.

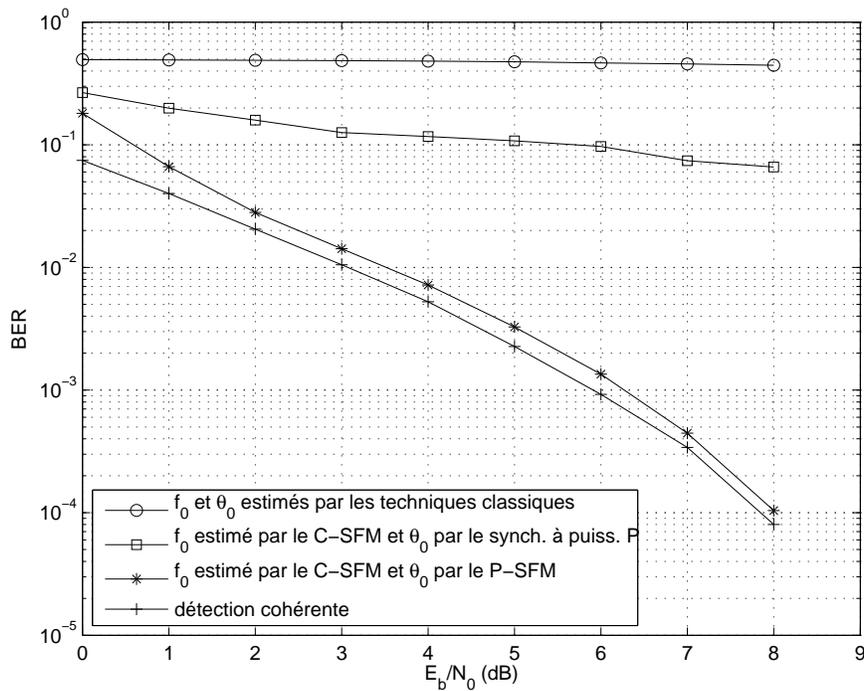


Figure 4.9 — Courbes de BER obtenues après estimation du résidu de porteuse et du déphasage puis application du décodeur BP.

## 4.5 Algorithme du P-SFM dans le cas de systèmes utilisant des modulations à grand nombre d'états

Les techniques aveugles d'estimation du déphasage que nous avons proposées dans ce chapitre sont seulement applicables dans le cas d'une modulation BPSK. Nous proposons dans ce paragraphe une autre technique d'estimation qui peut être appliquée quand des modulations à grand nombre d'états sont utilisées. Cette technique itérative, basée elle aussi sur l'algorithme du P-SFM, est décrite ci-dessous.

Nous supposons tout d'abord que le résidu de porteuse du système est parfaitement estimé au récepteur. Ceci nous permet de compenser les échantillons reçus et de former ainsi les échantillons  $r_{f_0}(k)$ . A chaque itération de la méthode d'estimation du déphasage proposée, nous compensons le déphasage présent dans les échantillons  $r_{f_0}(k)$  par une phase quelconque  $\tilde{\theta}$ . Ensuite, nous utilisons l'approximation (2.17) et nous estimons le LLR de chaque bit associé à un échantillon  $r(k)$  par :

$$\hat{\Gamma}(a_{(f_0, \tilde{\theta})}((k-1)q+i)) = \min_{\substack{\gamma \in \mathcal{Q} \\ a(i)=0}} \frac{|r_{f_0}(k)e^{-j\tilde{\theta}} - \gamma|^2}{\sigma^2} - \min_{\substack{\gamma \in \mathcal{Q} \\ a(i)=1}} \frac{|r_{f_0}(k)e^{-j\tilde{\theta}} - \gamma|^2}{\sigma^2}, \quad i = 1, \dots, q \quad (4.39)$$

où nous rappelons que  $\mathcal{Q}$  est l'ensemble des symboles modulés,  $\gamma$  est un symbole possible de  $\mathcal{Q}$  et  $q$  est le nombre de bits par symbole. La variable  $a_{(f_0, \tilde{\theta})}(k)$  représente le  $k^{\text{ième}}$  bit codé obtenu après compensation du CFO et du déphasage présents dans les échantillons reçus par  $f_0$  et  $\tilde{\theta}$ . Quand les LLR de tous les bits sont calculés, nous évaluons la fonction de coût suivante :

$$J'(f_0, \tilde{\theta}) = E \left[ \sum_{k=1}^{n_r} \left[ (-1)^{u_k+1} \prod_{j=1}^{u_k} \text{sign} \left( \hat{\Gamma}(a_{(f_0, \tilde{\theta})}(k_j)) \right) \min_{j=1, \dots, u_k} |\hat{\Gamma}(a_{(f_0, \tilde{\theta})}(k_j))| \right] \right]. \quad (4.40)$$

La figure 4.10 montre les variations de  $J'(f_0, \tilde{\theta})$  en fonction de l'erreur d'estimation  $(\tilde{\theta} - \theta_0)$ , pour un canal supposé sans bruit. Le code LDPC est le même que celui déjà utilisé dans ce chapitre mais le type de modulation considérée est une 16-QAM. Pour cette simulation, nous avons fait varier le déphasage  $\theta_0$  de  $-2\pi$  à  $2\pi$ . A partir de la figure 4.10, il est clair qu'optimiser la fonction  $J'(f_0, \tilde{\theta})$  en fonction de  $\tilde{\theta}$  donne une estimée du déphasage. En outre, nous remarquons que le minimum de  $J'(f_0, \tilde{\theta})$  est atteint pour une phase  $\tilde{\theta} = \theta_0$  modulo  $2\pi$ . Ainsi en minimisant  $J'(f_0, \tilde{\theta})$ , nous sommes en mesure d'estimer le déphasage introduit par le canal sans aucune ambiguïté.

Cependant, d'après (4.39) et (4.40), la fonction  $J'(f_0, \tilde{\theta})$  n'est pas dérivable. Ainsi, la méthode de descente du Gradient ne peut pas être appliquée pour résoudre le problème d'optimisation en question. Dans la suite, nous utilisons l'algorithme du Recuit Simulé pour minimiser la fonction  $J'(f_0, \tilde{\theta})$ . Cet algorithme est expliqué en détail au paragraphe 3.2.

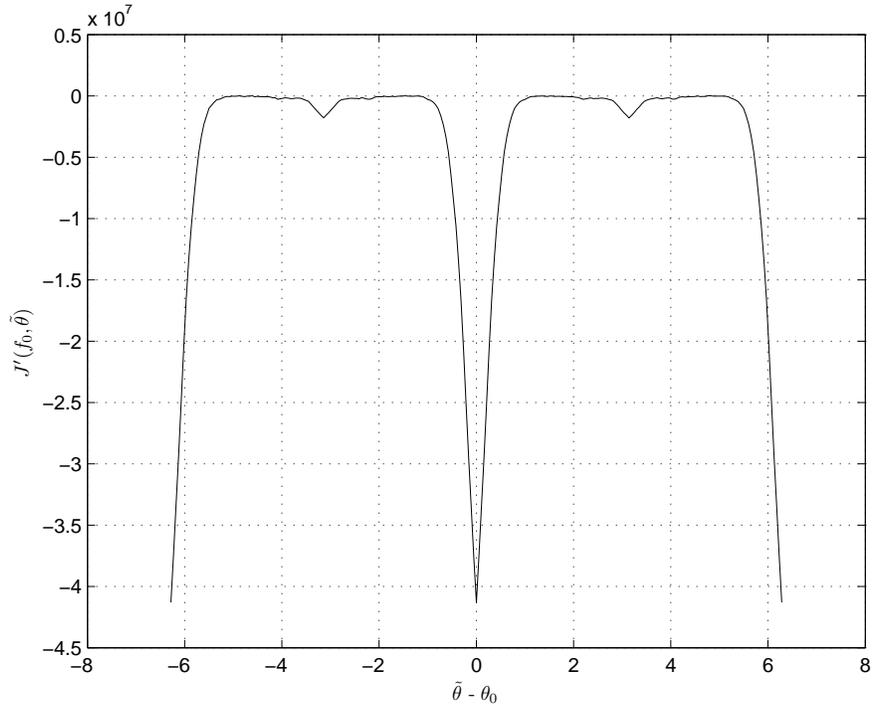


Figure 4.10 — La fonction de coût  $J'(f_0, \tilde{\theta})$  en fonction de l'erreur d'estimation du déphasage  $(\tilde{\theta} - \theta_0)$ .

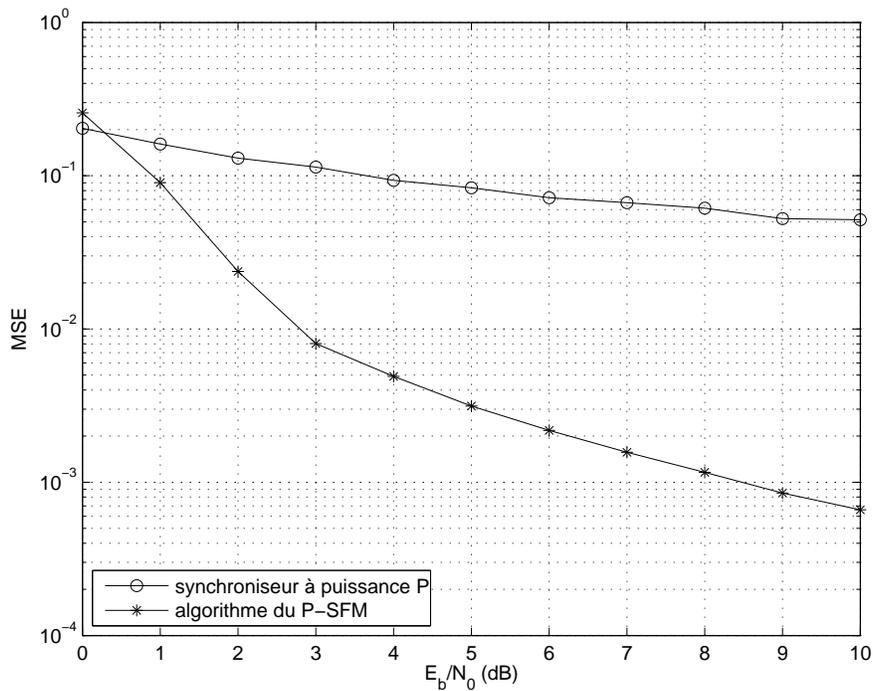


Figure 4.11 — MSE de l'estimation du déphasage dans le cas d'une modulation 16-QAM.

### 4.5.1 Résultats des simulations

Nous évaluons dans ce paragraphe les performances de l'algorithme du P-SFM proposé pour des systèmes utilisant des modulations à grand nombre d'états. Le type de modulation considérée est une 16-QAM. Les performances du P-SFM sont aussi comparées à celles du synchroniseur à puissance  $P$  qui, pour une modulation QAM, estime le déphasage par :

$$\hat{\theta}_P = \frac{1}{4} \arg \left( \sum_{k=1}^N (r(k))^4 \right), \quad (4.41)$$

où  $r(k)$  désigne dans ce cas un échantillon 16-QAM reçu.

Les courbes de MSE de l'estimation du déphasage sont tracées sur la figure 4.11. Encore une fois, notre méthode d'estimation du déphasage basée sur l'algorithme P-SFM se montre très performante. Pour 100 itérations de l'algorithme du Recuit Simulé, une MSE d'environ  $5 \cdot 10^{-3}$  est atteinte pour un  $E_b/N_0$  égal à 4 dB.

De plus, remarquons que pour le cas d'une modulation 16-QAM, le synchroniseur à puissance  $P$  donne une estimée du déphasage avec une ambiguïté de  $\pi/2$ . Cependant, comme montré sur la figure 4.10, la méthode proposée dans ce paragraphe calcule une estimée précise de la phase, sans aucune ambiguïté.

## 4.6 Conclusion

Nous avons considéré dans ce chapitre le problème d'estimation du déphasage introduit par le canal de transmission. Nous avons proposé deux méthodes d'estimation du déphasage basées sur des fonctions du LLR du syndrome calculées à partir des parties réelles et imaginaires des échantillons reçus. La première méthode (algorithme du SFR) donne une équation analytique du déphasage en absence de bruit. Malheureusement, cette expression n'est pas robuste à la présence de bruit. La deuxième méthode d'estimation (algorithme du P-SFM) est basée sur la minimisation d'une fonction de coût. La procédure d'optimisation est effectuée en utilisant l'algorithme du Gradient pour une modulation BPSK et le Recuit Simulé pour des modulations à plus grand nombre d'états. Comparée à d'autres méthodes d'estimation du déphasage, la méthode du P-SFM se montre très performante. Un écart de plusieurs dB est observé par rapport aux méthodes existantes. Notons que, dans le cas où des modulations à grand nombre d'états sont considérées, la méthode du P-SFM est capable de donner une estimation précise du déphasage sans aucune ambiguïté.

Nous avons aussi appliqué la méthode d'estimation du déphasage en présence d'un résidu de porteuse que nous estimons en appliquant la technique proposée au chapitre précédent. La courbe de BER obtenue après estimation du résidu de porteuse et du déphasage par les techniques proposées (C-SFM et P-SFM) colle presque parfaitement avec celle d'une détection cohérente.

Notons finalement que des codes LDPC ont été exclusivement testés dans les simulations de ce chapitre. Toutefois, les méthodes d'estimation proposées sont applicables à des systèmes utilisant n'importe quel type de codes correcteurs d'erreurs à condition qu'ils aient des matrices de contrôle de parité creuses. Dans le cas de codes ayant des

matrices non creuses, on peut adapter ces critères d'estimation en suivant l'approche du m-SSS proposée au paragraphe 2.3.2 du chapitre 2.



# Conclusion et perspectives

CETTE thèse est consacrée à la synchronisation aveugle des systèmes de transmission. Nous avons étudié en particulier :

- la synchronisation trame,
- l'estimation du résidu de porteuse,
- l'estimation du déphasage.

Les méthodes de synchronisation que nous avons proposées sont basées sur des techniques aveugles qui tirent profit de la présence d'un code correcteur d'erreurs dans la chaîne de transmission.

Dans la première partie de cette thèse (chapitres 1 et 2), nous avons considéré le problème de la synchronisation trame en supposant que le système de transmission n'est affecté ni d'un résidu de porteuse ni d'un déphasage dans le canal de transmission. Nous avons proposé une nouvelle technique de synchronisation aveugle (algorithme du SSS) basée sur un critère MAP qui maximise la probabilité qu'une position donnée soit la bonne position de synchronisation sachant les échantillons reçus. Nous avons démontré que maximiser cette probabilité est équivalent, après quelques suppositions et approximations, à la minimisation du LLR du syndrome calculé à partir de la matrice de contrôle de parité du code correcteur d'erreurs. Les principaux résultats que nous avons obtenus dans cette partie sont mentionnés ci-dessous :

- Les performances de la synchronisation trame étant mesurées en calculant la probabilité de fausse synchronisation, nous avons effectué une étude théorique qui nous a permis de trouver une équation analytique de la probabilité de fausse synchronisation de la méthode du SSS. La validité de cette équation a été vérifiée en comparant les résultats théoriques à ceux des simulations.
- Les résultats des simulations ont montré que les performances de la méthode de synchronisation du SSS surpassent toutes les autres techniques de synchronisation aveugles existantes, à notre connaissance, dans la littérature.
- Nous avons démontré que la méthode de synchronisation proposée (méthode du SSS) est plus performante quand elle est appliquée à des systèmes utilisant des codes correcteurs d'erreurs ayant des matrices de contrôle de parité creuses, comme par exemple les codes LDPC.
- La méthode du SSS est aussi très performante pour les codes convolutifs qui peuvent être aussi définis par des matrices de contrôle de parité creuses. Les courbes de FER obtenues après synchronisation et décodage ont montré l'efficacité de cette méthode de synchronisation.
- Les codes BCH et les codes produits sont des codes correcteurs d'erreurs ayant des

matrices de contrôle parité non creuses. Pour pouvoir synchroniser des systèmes utilisant ces types de codes, nous avons modifié notre critère de synchronisation en adaptant la matrice de contrôle de parité à la fiabilité des échantillons reçus. Les performances de l'algorithme de synchronisation ainsi obtenu (algorithme du m-SSS) sont très satisfaisantes, surtout pour les codes produits qui fonctionnent à des faibles SNR.

- Nous avons aussi calculé une expression analytique du FER obtenu après synchronisation et décodage, en fonction de la probabilité de fausse synchronisation et du FER obtenu dans le cas d'une synchronisation parfaite. Ainsi, en utilisant cette équation, on n'est plus obligé d'appliquer le décodeur à chaque fois qu'on veut tester l'efficacité d'une méthode de synchronisation en termes de FER.

Nous avons voulu utiliser la même approche pour corriger d'autres défauts de synchronisation. La deuxième partie de ce mémoire (chapitre 3) est consacrée à l'estimation du résidu de porteuse. Les principaux travaux réalisés dans cette partie sont les suivants :

- Nous avons supposé tout d'abord que le déphasage introduit par le canal de propagation est parfaitement connu du récepteur et nous avons proposé une méthode d'estimation du résidu de porteuse (méthode du C-SFM) basée sur le calcul et la minimisation d'une fonction du LLR du syndrome obtenue à partir des parties réelles des échantillons reçus. Cette fonction de coût présentant un seul minimum global et plusieurs minima locaux, la procédure d'optimisation est effectuée en utilisant l'algorithme du Recuit Simulé. Comparée à une méthode classique d'estimation du résidu de porteuse, la méthode du C-SFM est clairement plus performante. La différence entre les performances de ces deux méthodes est de plusieurs  $dB$ .
- Afin de réduire la complexité de la méthode d'estimation du C-SFM, nous avons proposé d'appliquer en premier temps la méthode classique d'estimation du résidu de porteuse. La fréquence estimée par cette méthode sert de fréquence d'initialisation pour notre technique d'estimation. L'algorithme ainsi obtenu (algorithme du C-ISFM) est moins complexe que le C-SFM d'origine.
- En présence d'un déphasage inconnu dans le canal de transmission, nous avons modifié nos critères d'estimation du résidu de porteuse (C-SFM et C-ISFM) en calculant une nouvelle fonction de coût. Cette fonction est basée sur le calcul des LLR du syndrome à partir des parties réelles et imaginaires des échantillons reçus. Les résultats des simulations ont montré que minimiser cette fonction de coût aboutit à de très bonnes performances en termes de MSE et BER. Ainsi, même en présence d'un déphasage inconnu, la méthode proposée d'estimation aveugle du résidu de porteuse s'est montrée très performante.

La dernière partie de ce document (chapitre 4) traite le problème de l'estimation du déphasage introduit par la canal de transmission. Dans cette partie, nous avons proposé deux techniques d'estimation aveugle du déphasage basées sur le même concept étudié dans cette thèse : le calcul des fonctions du LLR du syndrome et ceci à partir de la matrice de contrôle de parité du code correcteur d'erreurs utilisé dans la chaîne de transmission.

- Dans la première partie de ce chapitre, nous avons supposé que le résidu de porteuse du système est connu au récepteur. La première méthode d'estimation du déphasage que nous avons proposée (algorithme du SFR) calcule une équation analytique de l'estimée du déphasage. En comparant cette méthode à des algorithmes d'estimation proposés dans la littérature, le synchroniseur à puissance  $P$  reste le plus performant.
- Nous avons alors proposé une autre technique d'estimation du déphasage (algorithme du P-SFM) basée cette fois-ci sur le calcul et la minimisation d'une fonction de coût. Cette fonction est calculée à partir du LLR du syndrome des parties réelles et imaginaires des échantillons reçus. Comme la fonction de coût en question contient un seul minimum global, l'algorithme d'optimisation utilisé est le Gradient. Les résultats des simulations ont montré que la méthode d'estimation du déphasage du P-SFM est très performante et est largement plus puissante que toutes les autres méthodes d'estimation testées.
- Nous avons aussi appliqué cette technique d'estimation quand un résidu de porteuse inconnu est présent dans le système. Après estimation du résidu de porteuse par le C-SFM et du déphasage par le P-SFM, les courbes de BER obtenues étaient presque les mêmes que celle d'une détection cohérente.
- Nous avons finalement étudié le cas des modulations à grand nombre d'états et les résultats des simulations ont montré, encore une fois, que la technique d'estimation du P-SFM est très efficace. De plus, cette technique est capable d'estimer le déphasage sans aucune ambiguïté de phase.

## Perspectives

Les principales perspectives découlant de cette thèse sont les suivantes :

Au cours du développement du critère de synchronisation trame du SSS proposé au chapitre 1, nous avons supposé que les éléments du syndrome sont tous indépendants. Cette supposition est d'autant plus valide que la matrice de contrôle de parité du code est creuse. Malheureusement, ceci n'est pas le cas pour plusieurs types de codes correcteurs d'erreurs. Ainsi, il serait intéressant d'étudier la dépendance des éléments de syndrome et son influence sur la méthode de synchronisation du SSS. Un nouveau critère de synchronisation pourrait ainsi être établi.

Nous avons calculé au paragraphe 1.3.2 du chapitre 1 une équation analytique de la probabilité de fausse synchronisation. Cette équation analytique peut servir à développer un algorithme sous optimal par rapport à celui proposé, mais moins complexe. En effet, connaissant les lois de probabilité suivies par les critères de synchronisation, nous pouvons définir un seuil de synchronisation. Ainsi, il ne serait plus nécessaire de tester les  $n_c$  positions possibles de la fenêtre de synchronisation. La position de synchronisation serait la première position pour laquelle la valeur du critère est inférieure au seuil. Un gain important en termes de complexité et de temps

de calcul est alors atteint.

La méthode de synchronisation trame du SSS a été appliquée à des systèmes utilisant différents types de codes correcteurs d'erreurs. Nous citons les codes LDPC, les codes convolutifs, les codes BCH et les code produits. Suite à ces travaux, d'autres codes pourraient être testés comme par exemple les turbocodes caractérisés par leurs très bonnes performances de décodage. Il serait aussi intéressant d'appliquer cette méthode de synchronisation trame aux codes RS qui sont les codes à symboles non binaires les plus connus et les plus utilisés dans la pratique.

Bien que les critères de synchronisation étudiés dans cette thèse sont basés sur les propriétés des codes correcteurs d'erreurs, l'étape de synchronisation est indépendante de celle du décodage. Cependant on pourrait effectuer la synchronisation conjointement avec le décodage en insérant nos critères étudiés dans des procédures de turbo-synchronisation. Ceci pourrait aboutir à de meilleures performances.

Le problème de la synchronisation trame étudié dans cette thèse a été séparé de celui de l'estimation du résidu de porteuse et de l'estimation du déphasage. Cependant, comme le même critère basé sur le calcul des fonctions du LLR du syndrome a été utilisé pour les trois types de synchronisation (avec sûrement des modifications et des adaptations), on pourrait peut-être estimer les paramètres  $t_0$ ,  $f_0$  et  $\theta_0$  conjointement en résolvant des problèmes d'optimisation multidimensionnels. Bien que le critère résultant serait un peu complexe à résoudre, il serait intéressant d'explorer cette voie de recherche.

Nous avons supposé tout au long de cette thèse que la synchronisation rythme (c'est à dire l'instant optimal d'échantillonnage des signaux) est parfaitement effectuée. Ceci n'est pas toujours le cas. Il serait alors intéressant d'appliquer les critères proposés dans cette thèse sur ce genre de synchronisation.

Les résultats des simulations présentés dans ce manuscrit ont montré que les techniques de synchronisation proposées sont très performantes. Cependant, la complexité et le temps de calcul que demandent ces techniques n'ont pas été étudiés. Ainsi, pour une comparaison juste, il serait utile d'étudier ces paramètres.

# Annexe A

## Annexe du Chapitre 1

### A.1 Calcul de l'équation analytique de la probabilité de fausse synchronisation de la technique du HSS

Nous calculons dans ce paragraphe l'équation analytique de la probabilité de fausse synchronisation  $P_{FHSS}$ . La méthode du HSS cherche la position de synchronisation qui minimise la variable  $\varphi$ . Nous avons donc :

$$\begin{aligned} P_{FHSS} &= Pr[\varphi(t_0) > \min_{t=0, \dots, n_c-1, t \neq t_0} \varphi(t)] \\ &= 1 - Pr[\varphi(t_0) < \min_{t=0, \dots, n_c-1, t \neq t_0} \varphi(t)]. \end{aligned} \quad (A.1)$$

Pour  $t \neq t_0$ , les variables  $\varphi(t)$  sont indépendantes d'une position de la fenêtre de synchronisation à une autre. D'où :

$$P_{FHSS} = 1 - \prod_{t=0, t \neq t_0}^{n_c-1} Pr[\varphi(t_0) < \varphi(t)]. \quad (A.2)$$

A une position de non synchronisation, nous avons vu au chapitre 1 que toutes les variables  $\varphi(t)_{t \neq t_0}$  suivent la même distribution Binomiale. D'où :

$$\begin{aligned} P_{FHSS} &= 1 - \left[ Pr[\varphi(t_0) < \varphi(t)_{t \neq t_0}] \right]^{n_c-1} \\ &= 1 - \left[ \sum_{k=0}^{n_r-1} Pr[\varphi(t_0) = k \ \& \ \varphi(t)_{t \neq t_0} > k] \right]^{n_c-1} \\ &\stackrel{ind}{=} 1 - \left[ \sum_{k=0}^{n_r-1} \left( Pr[\varphi(t_0) = k] Pr[\varphi(t)_{t \neq t_0} > k] \right) \right]^{n_c-1} \\ &= 1 - \left[ \sum_{k=0}^{n_r-1} \left( Pr[\varphi(t_0) = k] \sum_{j=k+1}^{n_r} Pr[\varphi(t)_{t \neq t_0} = j] \right) \right]^{n_c-1}. \end{aligned} \quad (A.3)$$

Nous avons démontré au chapitre 1 que :

$$\begin{aligned} \varphi(t) &\sim \mathcal{B}(n_r, 1/2) & t \neq t_0 \\ \text{et } \varphi(t_0) &\sim \mathcal{B}(n_r, p). \end{aligned} \quad (\text{A.4})$$

D'où

$$Pr[\varphi(t) = j] = \binom{n_r}{j} \left(\frac{1}{2}\right)^j \left(1 - \frac{1}{2}\right)^{n_r-j} = \frac{\binom{n_r}{j}}{2^{n_r}} \quad (\text{A.5})$$

$$\text{et } Pr[\varphi(t_0) = k] = \binom{n_r}{k} p^k (1-p)^{n_r-k} \quad (\text{A.6})$$

En remplaçant (A.5) et (A.6) dans (A.3), nous obtenons une expression analytique de la probabilité de fausse synchronisation :

$$P_{F_{HSS}} = 1 - \left[ \sum_{k=0}^{n_r-1} \left( \binom{n_r}{k} p^k (1-p)^{n_r-k} \sum_{j=k+1}^{n_r} \left( \frac{\binom{n_r}{j}}{2^{n_r}} \right) \right) \right]^{n_c-1}. \quad (\text{A.7})$$

## A.2 Calcul des paramètres $m_{t_0}$ et $\sigma_{t_0}^2$ de $\hat{\phi}(t_0)$

Nous avons démontré dans (1.45) que

$$\hat{\phi}(t_0) \sim \mathcal{N}(n_r m_{t_0}, n_r \sigma_{t_0}^2). \quad (\text{A.8})$$

Trouvons maintenant les expressions des variables  $m_{t_0}$  et  $\sigma_{t_0}^2$  introduites dans cette distribution.

Nous avons déjà vu au chapitre 1 que  $\hat{L}(S_t(k))$  peut s'écrire sous la forme :

$$\hat{L}(S_t(k)) = (2S_t(k) - 1)Y_t(k). \quad (\text{A.9})$$

A la position de synchronisation  $t_0$ ,  $S_{t_0}(k)$  et  $Y_{t_0}(k)$  ne sont pas indépendantes. Dans ce cas, nous écrivons  $m_{t_0}$  sous la forme :

$$m_{t_0} = Pr[S_{t_0}(k) = 0] E[\hat{L}(S_{t_0}(k))_{/S_{t_0}(k)=0}] + Pr[S_{t_0}(k) = 1] E[\hat{L}(S_{t_0}(k))_{/S_{t_0}(k)=1}]. \quad (\text{A.10})$$

Calculons tout d'abord

$$E[\hat{L}(S_{t_0}(k))_{/S_{t_0}(k)=1}] = E[Y_{t_0}(k)_{/S_{t_0}(k)=1}]. \quad (\text{A.11})$$

Rappelons que nous sommes toujours dans le cas d'une modulation BPSK. Pour cela, un échantillon reçu s'écrit :

$$r(k) = b(k - t_0) + w(k) \quad (\text{A.12})$$

où  $b(k) = \pm 1$  est un symbole modulé et  $w(k)$  désigne le bruit.

Dire que  $S_{t_0}(k) = 1$  signifie qu'il existe un nombre impair d'erreurs parmi les  $u_k$  échantillons impliqués dans le calcul du  $k^{\text{ième}}$  élément du syndrome. En d'autres termes

et selon (A.12), il y a un nombre impair d'éléments  $w(k)$  qui, une fois ajoutés à  $b(k)$ , changent son signe.

Notons qu'une erreur parvient quand :

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

De plus, comme

$$Pr[b(k) = 1] = Pr[b(k) = -1] = 1/2 \quad (\text{A.13})$$

et  $w(k)$  est une variable Gaussienne de moyenne nulle,

$$E[Y_{t_0}(k)/S_{t_0}(k)=1] = E[\min_{j=1,\dots,u_k} |1 + w(t_0 + k_j)|_{/B_1}], \quad (\text{A.14})$$

où  $B_1$  est l'événement : { parmi  $u_k$  échantillons de bruit, un nombre impair de ces éléments ont des valeurs inférieures à  $-1$  }.

En répétant la même procédure pour le cas  $S_{t_0}(k) = 0$ , nous obtenons :

$$E[\hat{L}(S_{t_0}(k))/S_{t_0}(k)=0] = -E[Y_{t_0}(k)/S_{t_0}(k)=0] \quad (\text{A.15})$$

et

$$E[Y_{t_0}(k)/S_{t_0}(k)=0] = E[\min_{j=1,\dots,u_k} |1 + w(t_0 + k_j)|_{/B_2}], \quad (\text{A.16})$$

où  $B_2$  est l'événement : { parmi  $u_k$  échantillons de bruit, un nombre pair de ces éléments ont des valeurs inférieures à  $-1$  }.

Finalement, la moyenne  $m_{t_0}$  de  $\hat{L}_{t_0}(k)$  est égale à :

$$m_{t_0} = -(1-p)E[\min_{j=1,\dots,u_k} |1 + w(t_0 + k_j)|_{/B_2}] + pE[\min_{j=1,\dots,u_k} |1 + w(t_0 + k_j)|_{/B_1}]. \quad (\text{A.17})$$

Concernant la variance  $\sigma_{t_0}^2$ , nous pouvons facilement démontrer que :

$$\sigma_{t_0}^2 = E[(\min_{j=1,\dots,u_k} |r(t_0 + k_j)|)^2] - m_{t_0}^2. \quad (\text{A.18})$$

À partir de (A.17) et (A.18), nous observons que les expressions de  $m_{t_0}$  et  $\sigma_{t_0}^2$  dépendent principalement du niveau de bruit dans le canal de transmission. Afin de calculer ces valeurs, nous générons des échantillons de bruit et appliquons (A.17) and (A.18) en utilisant n'importe quel logiciel numérique. Une fois la moyenne  $m_{t_0}$  et la variance  $\sigma_{t_0}^2$  sont estimées, leurs valeurs sont tabulées pour être utilisées ultérieurement dans notre étude.



# Annexe B

## Annexe du Chapitre 3

### B.1 Calcul de l'expression analytique de la variance

$$\sigma_{class}^2$$

L'échantillon reçu de (3.1) est statistiquement équivalent (s.e) à :

$$r(k) \stackrel{s.e}{=} (b(k) + w(k))e^{j(2\pi k f_0 T_s + \theta_0)}. \quad (\text{B.1})$$

Pour la méthode existante qui estime le résidu de porteuse par (3.2) et en choisissant un coefficient  $D = 1$ , nous introduisons les deux variables :

$$x = \frac{1}{N-1} \sum_{k=1}^{N-1} (r(k)r^*(k-1))^2, \quad (\text{B.2})$$

$$\text{et } y = E[(r(k)r^*(k-1))^2] = e^{j4\pi f_0 T_s}. \quad (\text{B.3})$$

Selon [24], la variance de l'erreur d'estimation du résidu de porteuse est approximée par :

$$\sigma_{class}^2 \approx \frac{1}{4\pi^2 T_s^2} \frac{1}{8} \Re \left( \frac{E[|x-y|^2]}{|y|^2} - \frac{E[(x-y)^2]}{y^2} \right). \quad (\text{B.4})$$

Dans le cas d'une modulation BPSK

$$r(k)^2 \stackrel{s.e}{=} (1 + v(k))e^{2j(2\pi k f_0 T_s + \theta_0)}, \quad (\text{B.5})$$

où

$$v(k) = 2b(k)w(k) + w(k)^2. \quad (\text{B.6})$$

$w(k)$  étant un bruit complexe Gaussien de moyenne nulle et une variance totale  $\sigma^2 = 2\sigma_e^2$ , nous avons :

$$\begin{aligned} E[w(k)^2] &= 0, & E[|w(k)|^2] &= 2\sigma_e^2, & E[|w(k)|^4] &= 8\sigma_e^4, \\ E[v(k)] &= 0, & E[v(k)^2] &= 0, & E[|v(k)|^2] &= 8\sigma_e^2 + 8\sigma_e^4. \end{aligned} \quad (\text{B.7})$$

En prenant en considération les équations ci-dessus et en substituant (B.2) et (B.3) dans (B.4) nous obtenons :

$$\sigma_{class}^2 \approx \frac{N-2}{\pi^2 T_s^2 (N-1)^2} (2\sigma_e^4 + 4\sigma_e^6 + 2\sigma_e^8). \quad (\text{B.8})$$

Pour des grandes valeurs de  $N$ , (B.8) est approximativement égale à :

$$\sigma_{class}^2 \approx \frac{1}{\pi^2 T_s^2 N} (2\sigma_e^4 + 4\sigma_e^6 + 2\sigma_e^8). \quad (\text{B.9})$$

# Annexe C

## Annexe du Chapitre 4

### C.1 Calcul de la dérivée partielle de $J_e(f_0, \tilde{\theta})$

Afin de trouver

$$\frac{\partial J_e(f_0, \tilde{\theta})}{\partial \tilde{\theta}} = \frac{\partial L_{R_e}(f_0, \tilde{\theta})}{\partial \tilde{\theta}} - \frac{\partial L_{I_e}(f_0, \tilde{\theta})}{\partial \tilde{\theta}}, \quad (\text{C.1})$$

nous devons calculer les dérivées partielles de  $L_{R_e}(f_0, \tilde{\theta})$  et  $L_{I_e}(f_0, \tilde{\theta})$ . Conformément à (4.31) nous avons :

$$\begin{aligned} \frac{\partial L_{R_e}(f_0, \tilde{\theta})}{\partial \tilde{\theta}} &= E \left[ \sum_{k=1}^{n_r} \frac{\partial}{\partial \tilde{\theta}} \left( (-1)^{u_k+1} \operatorname{atanh} \left( \prod_{j=1}^{u_k} \tanh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right) \right) \right] \\ &= E \left[ \sum_{k=1}^{n_r} (-1)^{u_k+1} \frac{\sum_{j=1}^{u_k} \left( \frac{\partial}{\partial \tilde{\theta}} \left[ \tanh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right] \right) \prod_{\substack{i=1 \\ i \neq j}}^{u_k} \tanh \left( \Re \left( \frac{r_{f_0}(k_i)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right)}{1 - \left( \prod_{j=1}^{u_k} \tanh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right)^2} \right] \end{aligned} \quad (\text{C.2})$$

Or

$$\frac{\partial}{\partial \tilde{\theta}} \left[ \tanh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right] = \frac{\frac{\partial}{\partial \tilde{\theta}} \left[ \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right]}{\left( \cosh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right)^2} \quad (\text{C.3})$$

et

$$\frac{\partial}{\partial \tilde{\theta}} \left[ \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right] = \frac{1}{\sigma^2} \left[ -b(k_j) \cos \theta_0 \sin \tilde{\theta} + b(k_j) \sin \theta_0 \cos \tilde{\theta} - \Re(w(k_j)) \sin \tilde{\theta} \right]. \quad (\text{C.4})$$

L'équation (C.4) ne peut pas être calculée en raison du nombre de variables inconnues qui y participent. Cependant, cette équation peut être approchée par :

$$\frac{\partial}{\partial \tilde{\theta}} \left[ \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right] \approx \frac{1}{\sigma^2} \left[ -\Re(r_{f_0}(k_j)) \sin \tilde{\theta} + \Im(r_{f_0}(k_j)) \cos \tilde{\theta} \right]. \quad (\text{C.5})$$

L'approximation précédente devient exacte en absence de bruit. En substituant (C.3) et (C.5) dans (C.2), nous obtenons :

$$\frac{\partial L_{R_e}(f_0, \tilde{\theta})}{\partial \tilde{\theta}} = E \left[ \sum_{k=1}^{n_r} (-1)^{u_k+1} \frac{\sum_{j=1}^{u_k} \left( \frac{-\Re(r_{f_0}(k_j)) \sin \tilde{\theta} + \Im(r_{f_0}(k_j)) \cos \tilde{\theta}}{\sigma^2 \left( \cosh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right)^2} \prod_{\substack{i=1 \\ i \neq j}}^{u_k} \tanh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right)}{1 - \left( \prod_{j=1}^{u_k} \tanh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right)^2} \right] \quad (\text{C.6})$$

Une procédure similaire est effectuée pour calculer la dérivée de  $L_{I_e}(f_0, \tilde{\theta})$ . Finalement, (C.1) est égale à :

$$\begin{aligned} \frac{\partial J_e(f_0, \tilde{\theta})}{\partial \tilde{\theta}} = & E \left[ \sum_{k=1}^{n_r} (-1)^{u_k+1} \left( \frac{\sum_{j=1}^{u_k} \left( \frac{-\Re(r_{f_0}(k_j)) \sin \tilde{\theta} + \Im(r_{f_0}(k_j)) \cos \tilde{\theta}}{\sigma^2 \left( \cosh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right)^2} \prod_{\substack{i=1 \\ i \neq j}}^{u_k} \tanh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right)}{1 - \left( \prod_{j=1}^{u_k} \tanh \left( \Re \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right)^2} \right. \\ & \left. - \frac{\sum_{j=1}^{u_k} \left( \frac{-\Im(r_{f_0}(k_j)) \sin \tilde{\theta} - \Re(r_{f_0}(k_j)) \cos \tilde{\theta}}{\sigma^2 \left( \cosh \left( \Im \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right)^2} \prod_{\substack{i=1 \\ i \neq j}}^{u_k} \tanh \left( \Im \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right)}{1 - \left( \prod_{j=1}^{u_k} \tanh \left( \Im \left( \frac{r_{f_0}(k_j)}{\sigma^2} e^{-j\tilde{\theta}} \right) \right) \right)^2} \right) \right]. \quad (\text{C.7}) \end{aligned}$$

# Publications

## Revues internationales :

- [R<sub>1</sub>] R. Imad, S. Houcke, Theoretical analysis of a MAP based blind frame synchronizer, IEEE Transactions on Wireless Communications, vol. 8, no. 10, Octobre 2009.
- [R<sub>2</sub>] R. Imad, G. Sicot, S. Houcke, Blind frame synchronization for error correcting codes having a sparse parity check matrix, IEEE Transactions on Communications, vol. 57, no. 6, Juin 2009.

## Conférences internationales :

- [C<sub>1</sub>] R. Imad, S. Houcke, M. Ghogho, Effective techniques for blind phase offset estimation in LDPC coded systems, SPAWC, Pérouse, Italie, Juin 2009.
- [C<sub>2</sub>] R. Imad, S. Houcke, C. Jogo, Blind frame synchronization of product codes based on the adaptation of the parity check matrix, ICC, Dresde, Allemagne, Juin 2009.
- [C<sub>3</sub>] R. Imad, S. Houcke, Blind frame synchronization and phase offset estimation for coded systems, SPAWC, Recife, Brésil, Juillet 2008.
- [C<sub>4</sub>] R. Imad, S. Houcke, C. Douillard, Blind frame synchronization on Gaussian channel, EUSIPCO, Poznan, Pologne, Septembre 2007.

## Conférences nationales :

- [C<sub>5</sub>] R. Imad, S. Houcke, M. Ghogho, Estimation aveugle du résidu de porteuse dans les systèmes utilisant des codes LDPC, GRETSI, Dijon, France, Septembre 2009.
- [C<sub>6</sub>] R. Imad, S. Houcke, C. Douillard, Synchronisation Trame Aveugle sur Canal Gaussien, GRETSI, Troyes, France, Septembre 2007.



# Bibliographie

- [1] TurboConcept DVB-S2 IP Core.
- [2] G. Arfken. *Mathematical Methods for Physicists*. FL : Academic Press, third edition, 1985.
- [3] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Transactions on Information Theory*, 20(2) :284–287, March 1974.
- [4] R. H. Barker. Group synchronization of binary digital systems. *Communication Theory*, pages 273–287, 1953.
- [5] C. Bergogne, P. Sehier, and M. Bousquet. Reduced complexity frequency estimator for burst transmission. In *Proc. of GLOBECOM*, pages 1318–1322, 1995.
- [6] C. Berrou. *Codes et turbocodes*. Springer, 2007.
- [7] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding : turbo-codes. In *Proc. of ICC*, pages 1067–1070, May 1993.
- [8] E. Cacciamani and C. Jr. Wolejsza. Phase-ambiguity resolution in a four-phase PSK communications system. *IEEE Transactions on Communication Technology*, 19(6) :1200–1210, December 1971.
- [9] T. M. Cassaro and C. N. Georghiades. Frame synchronization for coded systems over AWGN channels. *IEEE Transactions on Communications*, 52(3) :484–489, March 2004.
- [10] K. Cavalec-Amis. *Optimisation des turbocodes pour les systèmes à grande efficacité spectrale*. PhD thesis, Université de Rennes 1, 2001.
- [11] V. Cerny. Thermodynamical approach to the traveling salesman problem : an efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1), January 1985.
- [12] D. Chase. A class of algorithms for decoding block codes with channel measurement information. *IEEE Transactions on Information theory*, IT-18(1) :170–182, January 1972.
- [13] J. Chen and M. P. C. Fossorier. Near optimum universal belief propagation based decoding of low-density parity check codes. *IEEE Transactions on Communications*, 50(3) :406–414, March 2002.
- [14] P. Ciblat and M. Ghogho. Blind NLLS carrier frequency-offset estimation for QAM, PSK, and PAM modulations : Performance at low SNR. *IEEE Transactions on Communications*, 54(10) :1725–1730, October 2006.

- [15] F. Classen, H. Meyr, and P. Sehier. Maximum likelihood open loop carrier synchronizer for digital radio. In *Proc. of ICC*, 1993.
- [16] A. N. d'Andrea, U Mengali, and Reggiannini. R. Carrier phase recovery for narrow-band polyphase-shift keyed signals. *Alta Frequenza*, 57(10) :575–581, December 1988.
- [17] M. Dervin, M. L. Boucheret, G. Mesnager, and A. Ducasse. Phase detection involving parity-check equations and suited to transmissions at low signal to noise ratio. In *Proc. of ICC*, 2005.
- [18] P. Elias. Error-free coding. *IRE Trans. Inf. Theory*, 4(4) :29–37, January 1954.
- [19] M. P. Fitz. Further results in the fast estimation of a single frequency. *IEEE Transactions on Communications*, 42(234) :862–864, March 1994.
- [20] R. G. Gallager. Low-density parity-check codes. *IRE Trans. Inform. Theory*, 8(1) :21–28, January 1962.
- [21] F. M. Gardner. Demodulator reference recovery techniques suited for digital implementation. *European Space Agency Final Report, Contract 6847/86/NL/DG*, August 1988.
- [22] F. M. Gardner. Frequency detectors for digital demodulators via maximum likelihood derivation. *ESA-ESTEC Final Report Part 2*, March 1990.
- [23] C. N. Georghiadis. Blind carrier phase acquisition for qam constellations. *IEEE Transactions on Communications*, 45(11) :1477–1486, November 1997.
- [24] M. Ghogho, A. Swami, and T. S. Durrani. Frequency estimation in the presence of doppler spread : Performance analysis. *IEEE Transactions on Signal Processing*, 49(4) :777–789, April 2001.
- [25] S. W. Golomb and R. A. Scholtz. Generalized Barker sequences. *IEEE Transactions on Information theory*, IT-11(4) :817–824, October 1965.
- [26] J. Hagenauer, E. Offer, and L. Papke. Iterative decoding of binary block and convolutional codes. *IEEE trans. on Information Theory*, 42(2) :429–445, March 1996.
- [27] P. Hoeher and J. Lodge. “Turbo DPSK” : Iterative differential psk demodulation and channel decoding. *IEEE Transactions on Communications*, 47(6) :837–843, June 1999.
- [28] S. Houcke and G. Sicot. Procédé et dispositif de synchronisation trame. Institut TELECOM/TELECOM Bretagne IP. Patent No 0501748, FT R&D. February 2005.
- [29] S. Houcke and G. Sicot. Blind frame synchronization for block code. In *Proc. of EUSIPCO, European Sig. Proc.*, September 2006.
- [30] S. Houcke, G. Sicot, and M. Debbah. Blind detection for block interleaved division multiple access. In *Proc. of GLOBECOM*, 2006.
- [31] M. M. K. Howlader and B. D. Woerner. Decoder-assisted frame synchronization for packet transmission. *IEEE J. Sel. Areas Comm.*, 19(12) :2231–2345, December 2001.

- [32] H. Huh and J. V. Krogmeier. A unified approach to optimum frame synchronization. *IEEE Trans. Wireless Comm.*, 5(12) :3700–3711, December 2006.
- [33] R. Imad and S. Houcke. Blind frame synchronization and phase offset estimation for coded systems. In *Proc. of SPAWC*, July 2008.
- [34] R. Imad, S. Houcke, and C. Douillard. Blind frame synchronization on gaussian channel. In *Proc. of EUSIPCO, European Sig. Proc.*, September 2007.
- [35] R. Imad, S. Houcke, and Mounir Ghogho. Blind estimation of the carrier frequency and phase offsets for ldpc coded systems. *submitted to IEEE Transactions on Wireless Communications*, March 2009.
- [36] R. Imad, S. Houcke, and C. Jego. Blind frame synchronization of product codes based on the adaptation of the parity check matrix. In *Proc. of ICC*, June 2009.
- [37] R. Imad, G. Sicot, and S. Houcke. Blind frame synchronization for error correcting codes having a sparse parity check matrix. *IEEE Transactions on Communications*, 57(6) :1574–1577, June 2009.
- [38] C Jego and W. J. Gross. Turbo decoding of product codes based on the modified adaptive belief propagation algorithm. In *Proc. of ISIT*, pages 641–644, June 2007.
- [39] J. Jiang and K. R. Narayanan. Iterative soft decoding of reed-solomon codes. *IEEE Communication Letters*, 8(4) :244–246, April 2004.
- [40] J. Jiang and K. R. Narayanan. Iterative soft-input-soft-output decoding of reed-solomon codes by adapting the parity check matrix. *IRE Trans. Inf. Theory*, 52(8) :3746–3756, August 2006.
- [41] R. Johannesson and K. Sh. Zigangirov. *Fundamentals of Convolutional Coding*. IEEE series on Digital and Mobile Communication. IEEE Press, 1999.
- [42] S. Kay. A fast and accurate single frequency estimator. *IEEE Transactions on Accouctics Speech and Signal Processing*, 37(12) :1987–1990, December 1989.
- [43] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization with simulated annealing. *Science*, 1983.
- [44] F. Kschischang, B. Frey, and H. A. Loeliger. Factor graphs and the sum product algorithm. *IEEE Transactions on Information theory*, 47(2) :498–519, February 2001.
- [45] D. Lee, H. Kim, C. R. Jones, and J. D. Villasenor. Pilotless frame synchronization for LDPC-coded transmission systems. *IEEE Transactions on Signal Processing*, 56(7) :2865–2874, July 2008.
- [46] V. Lottici and M. Luise. Carrier phase recovery for turbo-coded linear modulations. In *Proc. of ICC*, 2002.
- [47] M. G. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman. Analysis of low density codes and improved designs using irregular graphs. In *Proc. of the ACM STOC*, 1998.
- [48] G. L. Lui and H. H. Tan. Frame synchronization for gaussian channels. *IEEE Transactions on Communications*, 35(8) :818–829, August 1987.
- [49] C. Ma and B. P. Ng. Comparison of LDPC decoding algorithms with and without estimation of noise variance. In *Proc. of the 3rd conference on Mobile Technology, Applications and Systems*, 2006.

- [50] H. H. Ma and J. K. Wolf. On tail biting convolutional codes. *IEEE Transactions on Communications*, 34(2) :104–111, February 1986.
- [51] D. J. C. MacKay and R. M. Neal. Near shannon-limit performance of low-density parity-check codes. *Electronics Letters*, 33(6) :457–458, March 1997.
- [52] J. L. Massey. Optimum frame synchronization. *IEEE Transactions on Communications*, 20(2) :115–119, April 1972.
- [53] W. Matsumoto and H. Imai. Blind synchronization with enhanced sum-product algorithm for low-density parity-check codes. In *Proc. of the 5<sup>th</sup> Int. Symp. Wireless Personal Multimedia Communications*, pages 966–970, 2002.
- [54] M. H. Meyers and L. E. Franks. Joint carrier phase and symbol timing recovery for pam systems. *IEEE Transactions on Communications*, COM-28(8) :1121–1129, August 1980.
- [55] H. Meyr, M. Moeneclaey, and S. A. Fechtel. *Digital Communication Receivers. Synchronization, Channel Estimation and Signal Processing*. J. Wiley and Sons, 1998.
- [56] B. Mielczarek and A. Svensson. Joint synchronization and decoding of turbo codes on AWGN channels. In *Proc. of VTC*, volume 3, pages 1886–1890, May 1999.
- [57] M. Moeneclaey and G. de Jonghe. ML-oriented NDA carrier synchronization for general rotationally symmetric signal constellations. *IEEE Transactions on Communications*, 42(8) :2531–2533, August 1994.
- [58] R. H. Morelos-Zaragoza. *The Art of Error Correcting Coding*. John Wiley & Sons, 2006.
- [59] W. Oh and K. Cheun. Joint decoding and carrier phase recovery algorithm for turbo codes. *IEEE Communications Letters*, 5(9) :375–377, September 2001.
- [60] R. Pyndiah. Near-optimum decoding of product codes : Block turbo codes. *IEEE Transactions on Communications*, 46(8) :1003–1010, August 1998.
- [61] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq. Near optimum decoding of product codes. In *Proc. of GLOBECOM*, pages 339–343, November 1994.
- [62] P. Robertson. A generalized frame synchronizer. In *Proc. of GLOBECOM*, pages 365–369, December 1992.
- [63] J. Sun. *Synchronization for Capacity-Approaching Coded Communication Systems*. PhD thesis, College of Engineering and Mineral Resources at West Virginia University, 2004.
- [64] J. Sun and M. C. Valenti. Optimum frame synchronization for preamble-less packet transmission of turbo codes. *Asilomar Conference on Signals, Systems, and Computers*, 1 :1126–1130, November 2004.
- [65] S. A. Tretter. Estimating the frequency of a noisy sinusoid by linear regression. *IEEE Transactions on Information theory*, IT-31(6) :832–835, November 1985.
- [66] A. J. Van Wijngaarden and T. J. Willink. Frame synchronization using distributed sequences. *IEEE Transactions on Communications*, 48(12) :2127–2138, December 2000.

- [67] A. J. Viterbi and A. M. Viterbi. Nonlinear estimation of psk-modulated carrier phase with application to burst digital transmission. *IEEE Transactions on Information theory*, 29(4) :543–551, July 1983.
- [68] Y. Wang, E. Serpedin, and P. Ciblat. Optimal blind carrier recovery for MPSK burst transmissions. *IEEE Transactions on Communications*, 51(9) :1571–1581, September 2003.
- [69] H. Wymeersch and M. Moeneclaey. Iterative code-aided ML phase estimation and phase ambiguity resolution. *Eurasip Journal on Applied Signal Processing. Special Issue on Turbo Processing*, 2005(6) :981–988, May 2005.
- [70] L. Zhang and A. Burr. A novel carrier phase recovery method for turbo coded QPSK systems. In *Proc. of EW*, 2000.



# Index

- A**
- algorithme
- ABP ..... 61
  - BCJR ..... 53
  - BP ..... 49, 61, 97, 100
  - C-ISFM ..... 81, 85
  - C-SFM ..... 75, 84
  - Chase-Pyndiah (de) ..... 60
  - HDD ..... 88
  - HSS ..... 27, 37, 41, 50, 55, 111
  - m-ABP ..... 61, 66
  - m-SSS ..... 61, 64, 66
  - P-SFM ..... 93, 97, 104
  - SFR ..... 89, 92, 97
  - SSS ..... 31–34, 38, 40, 41, 50, 63
  - Viterbi (de) ..... 53
- ambiguïté de phase ..... 88, 91, 94, 104
- B**
- BABG ..... 26, 29
- BER ..... 41, 62, 97, 100
- C**
- CFO ..... voir résidu de porteuse
- code
- BCH ..... 47, 60
  - convolutif ..... 53, 54, 56, 57
  - cyclique ..... 59
  - étendu ..... 60
  - LDPC ..... 33, 45, 48, 77, 92
  - produit ..... 47, 60
- D**
- déphasage du canal ..... 87
- dimension d'un code ..... 27, 60
- distribution
- Binomiale ..... 38
  - de Bernoulli ..... 37
  - Gaussienne ..... 38, 39
- DVB-S2 ..... 52
- E**
- équation de parité ..... 27, 29, 37
- F**
- fenêtre de synchronisation ... 28, 31, 62
- FER ..... 52, 58, 66
- G**
- Gradient(descente du) ..... 95
- L**
- LLR
- d'un échantillon ..... 33, 67, 102
  - du syndrome . 29, 31, 63, 75, 83, 89, 96
- longueur d'un code ..... 27, 37
- longueur de contrainte ..... 53
- M**
- MAP ..... 25, 30, 35, 45
- matrice de contrôle de parité 23, 27, 31, 44, 47, 54, 60, 63, 73, 89
- modulation
- 16-QAM ..... 68, 104
  - BPSK ..... 29, 67, 73
  - QPSK ..... 67
- MSE ..... 79, 92, 97, 104
- MV ..... 26, 88
- P**
- polynôme générateur ..... 53, 56
- probabilité
- a posteriori* ..... 30, 35
  - d'erreur du canal ..... 37, 42
  - d'indépendance des éléments du syndrome ..... 44
  - de fausse synchronisation 37, 38, 40
- R**
- Recuit Simulé ..... 73, 78, 102
- rendement de codage ..... 27, 41

résidu de porteuse ..... 73, 77, 99

### S

séquence d'apprentissage.....26, 52, 57

simulations de Monte Carlo ..47, 79, 92

synchroniseur à puissance  $P$  88, 92, 104

syndrome ..... 27, 28, 42, 54, 87

### T

tail-biting.....54

## Résumé

Les dernières années ont vu une augmentation dans la demande de systèmes de communications numériques efficaces et fiables. Afin de protéger les données transmises contre le bruit, les codes correcteurs d'erreurs doivent être introduits dans le système de transmission. Les codes LDPC (Low Density Parity Check), les codes produits et les turbocodes ont prouvé leur efficacité dans la détection et la correction des erreurs, même à des faibles SNR (Signal to Noise Ratio). Cependant, une dégradation importante dans les performances de ces codes est atteinte dans le cas d'une mauvaise synchronisation au niveau du récepteur. Ainsi, l'utilisation de méthodes de synchronisation efficaces est nécessaire pour le bon fonctionnement d'un système de transmission. Les techniques classiques de synchronisation sont basées sur le rajout de bits pilotes dans la séquence de données à transmettre. La détection de ces bits à la réception nous permet de réaliser la synchronisation. Cependant, ces méthodes ont pour effet de diminuer l'efficacité spectrale de la transmission surtout quand des codes correcteurs d'erreurs de petite taille sont utilisés dans le système. Pour cela, nous sommes intéressés dans cette thèse par la conception de nouvelles techniques de synchronisation aveugles, qui sont capables de synchroniser à des faibles SNR. Nous considérons dans cette thèse le problème de la synchronisation trame, de l'estimation du résidu de porteuse et de l'estimation du déphasage. Nous proposons des techniques de synchronisation originales qui sont basées sur le calcul et la minimisation de fonctions du LLR (Log-Likelihood Ratio) du syndrome calculées à partir de la matrice de contrôle de parité du code correcteur d'erreurs. Les résultats des simulations ont montré que les techniques proposées sont très efficaces et leurs performances surpassent celles de plusieurs méthodes existant dans la littérature.

**Mots clés :** Synchronisation trame, estimation du déphasage, estimation du résidu de porteuse, codage de canal, algorithmes d'optimisation.

---

## Abstract

The past few years have seen an increasing demand for efficient and reliable digital communication systems. In order to protect the transmitted data from noise, error correcting codes should be used in the transmission scheme. LDPC (Low Density Parity Check) codes, product codes and turbocodes have proven their efficiency in detecting and correcting errors, even at low SNR (Signal to Noise Ratio). However, a degradation in the performance of these codes is expected in the case of a bad synchronization. Therefore, the use of efficient synchronization methods is essential to keep the system in a good working order. Conventional synchronization techniques involve the insertion of pilot bits in the transmitted sequence of data. The detection of these bits at the reception enables us to carry out synchronization. However, applying these methods decreases the spectral efficiency of the transmission especially when error correcting codes of small sizes are used in the system. For that reason, we are interested in this thesis by the design of novel methods of blind synchronization, which are able to synchronize even on low SNR. We consider in this thesis the problem of frame synchronization, CFO (Carrier Frequency Offset) estimation and phase estimation. We propose powerful synchronization techniques that involve the calculation and the minimization of functions of the LLR (Log-Likelihood Ratio) of the syndrome calculated from the parity check matrix of the error correcting code. Simulation results have shown that the proposed techniques are very powerful and their performance clearly exceeds those of several existing methods.

**Keywords :** Frame synchronization, phase offset estimation, CFO estimation, channel coding, optimization algorithms.