TELECOM PARIS

Thèse

présentée pour obtenir le grade de docteur
de l'École Nationale Supérieure des Télécommunications
Spécialité : Électronique et Communications

Loïc BRUNEL

ALGORITHMES DE DÉCODAGE DE CANAL POUR L'ACCÈS MULTIPLE À ÉTALEMENT DE SPECTRE

Soutenue le 16 décembre 1999 devant le jury composé de

Sergio BENEDETTO	Président
Giuseppe CAIRE Emanuele VITERBO	Rapporteurs
Claude BERROU Robert VALLET	Examinateurs
Joseph BOUTROS	Directeur de thèse

École Nationale Supérieure des Télécommunications, Paris
Groupe communications numériques, département COMELEC

Channel decoding algorithms for spread spectrum multiple access

Loïc Brunel

École Nationale Supérieure des Télécommunications, Paris

Digital communications group, COMELEC department

Committee in charge:

Sergio Benedetto	chairman
Giuseppe Caire	reporters
Emanuele Viterbo	
Claude Berrou	
Robert Vallet	
Joseph Boutros	advisor

December 16, 1999

Remerciements

Je tiens à exprimer ici toute ma reconnaissance aux personnes qui m'ont aidé et encouragé au cours de ces trois années de thèse au département COMELEC de l'Ecole Nationale Supérieure des Télécommunications.

Je dois tout d'abord remercier le Professeur Sergio Benedetto, du *Politecnico* de Turin, pour m'avoir fait l'honneur de présider le jury de ma soutenance de thèse, Messieurs Giuseppe Caire, Professeur à l'institut EURECOM de Nice, et Emanuele Viterbo, Professeur au *Politecnico* de Turin, pour leur lecture attentive de mon manuscrit et leurs remarques constructives, et Messieurs Claude Berrou, Professeur à l'ENST de Bretagne, et Robert Vallet, Professeur à l'ENST, pour avoir pris part au jury et témoigné leur intérêt pour mon travail.

Ce travail de thèse n'aurait pu aboutir sans le soutien technique et financier de la Direction Scientifique et du département COMELEC de l'ENST. Mes remerciements vont notamment à Bernard Robinet, Directeur Scientifique de l'ENST, et Philippe Gallion, Directeur du département COMELEC.

Les résultats présentés dans ce document sont avant tout le fruit de l'encadrement de Joseph Boutros, qui a su exploiter son expérience déjà vaste de la recherche en communications numériques pour insuffler rigueur, innovation et dynamisme au sujet et qui a bien voulu me témoigner sa confiance durant ces trois années de travail en commun. Il a également réussi à développer une réelle coopération entre ses nombreux thésards, Céline Durand, Catherine Lamy, Sabine Leveiller, Sandrine Vialle, Francesc Boixadera, Christophe Brutel, Olivier Pothier et Hadi Sawaya. Le large éventail de connaissances couvert par leurs sujets de recherche et leur volonté de les transmettre ont beaucoup enrichi mon travail et mes connaissances et je les en remercie.

Ces remerciements s'adressent également à tous les autres thésards du département, notamment Amal Abou Hassan, Mohammad Aoude, François-Xavier Bergot, Ammar Chkeif, Christophe Gosset, Nicolas Ibrahim, Stefan Lauffenburger, Cédric Ware, Seyed Bahram Zahir Azami, qui ont contribué à créer une ambiance de travail amicale autour des deux symboles emblématiques du bureau A303 : la cafetière et la bouilloire.

Merci aussi à Daniela Boggio pour son aide précieuse, à Jean-Yves Chouinard, Professeur à l'université d'Ottawa en séjour sabbatique à l'ENST au début de ma thèse et à tous les permanents du département COMELEC.

Enfin je remercie Rose-Marie et toute ma famille pour leur soutien continu.

A Paris, le 20 décembre 1999.

Liste des abréviations

La signification d'une abréviation n'est souvent indiquée qu'à sa première apparition dans le texte. Il existe dans la plupart des cas une abréviation en français et une abréviation en anglais. Toutes deux sont indiquées une première fois puis nous employons l'abréviation la plus usuelle, qui est le plus souvent l'abréviation en anglais.

AMRC	Accès Multiple à Répartition par Codes
AMRF	Accès Multiple à Répartition en Fréquences
AMRT	Accès Multiple à Répartition en Temps
APP	*A Posteriori Probability*
AWGN	*Additive White Gaussian Noise*
BPSK	*Binary Phase Shift Keying*
BER	*Bit Error Rate*
CDMA	*Code Division Multiple Access*
DFE	*Decision Feedback Equalization*
DS-CDMA	*Direct Sequence Code Division Multiple Access*
DS-SSMA	*Direct Sequence Spread Spectrum Multiple Access*
EQM	Erreur Quadratique Moyenne
FDMA	*Frequency Division Multiple Access*
GLD	*Generalized Low Density Codes*
IES	Interférence Entre Symboles
ISI	*Intersymbol Interference*
LDPC	*Low Density Parity Check Codes*
LLR	*Log Likelihood Ratio*
LPI	*Low Probability of Interception*
MAP	Modulation d'Amplitude en Phase
MAQ	Modulation d'Amplitude en Quadrature
MDP	Modulation De Phase
ML	*Maximum Likelihood*
MMSE	*Minimum Mean Square Error*
PAM	*Phase Amplitude Modulation*
PIC	*Parallel Interference Canceller*

PN	*Pseudo-Noise*
QPSK	*Quadrature Phase Shift Keying*
RSMA	*Rate Splitting Multiple Access*
SIC	*Successive Interference Canceller*
SNR	*Signal to Noise Ratio*
TDMA	*Time Division Multiple Access*
ZF	*Zero Forcing*

Liste des notations

Les listes ci-dessous regroupent les notations employées dans les différents chapitres du document. Nous avons tenté dans la mesure du possible de conserver les mêmes notations d'un chapitre à l'autre. Seules les notations qui diffèrent de celles déjà définies pour les chapitres précédents sont présentes dans les listes. Certaines notations, qui apparaissent ponctuellement, ont été omises.

Détection multi-utilisateurs

$\mathcal{A}$	Alphabet des symboles $b_k(i)$
$b_k(i)$	Symbole i de l'utilisateur k
$\mathbf{b}(i)$	$(b_1(i)\ ,\ b_2(i)\ ,\ \ldots\ ,\ b_K(i))^T$
$\mathbf{b}$	$(\mathbf{b}(0)\ ,\ \ldots\ ,\ \mathbf{b}(N-1))$
$\overline{\mathbf{b}}$	$(\ b_1(0)\ ,\ b_2(0)\ ,\ \cdots\ ,\ b_K(0)\ ,\ b_1(1)\ ,\ \cdots\ ,\ b_K(N-1)\ )^T$
$\hat{\mathbf{b}}$	Séquence des symboles détectés
$d^2(\mathbf{b})$	Métrique à minimiser dans le détecteur ML
$\mathbf{D}_\omega$	$\mathrm{diag}(\omega_1\ ,\ \ldots\ ,\ \omega_K)$
$\overline{\mathbf{D}_\omega}$	$\mathrm{diag}(\ \omega_1\ ,\ \omega_2\ ,\ \cdots\ ,\ \omega_K\ ,\ \omega_1\ \cdots\ ,\ \omega_K\ )$
K	Nombre d'utilisateurs
L	Gain d'étalement
N	Nombre de symboles
N_0	Densité spectrale de puissance du bruit blanc gaussien
n_t	Bruit au temps t
$n_k(i)$	Bruit additif dans $y_k(i)$
$\overline{\mathbf{n}}$	$(\ n_1(0)\ ,\ n_2(0)\ ,\ \cdots\ ,\ n_K(0)\ ,\ n_1(1)\ ,\ \cdots\ ,\ n_K(N-1)\ )^T$
ω_k	Amplitude de l'utilisateur k
R	Débit d'information
$\mathbf{R}$	Matrice d'intercorrélation
r_t	Signal reçu au temps t
$s_k(t)$	Signature de l'utilisateur k
T	Période d'un symbole
T_c	Période d'un *chip*
τ_k	Retard de l'utilisateur k

σ^2	Variance du bruit
$S_t(\mathbf{b})$	Signal émis au temps t pour une séquence de symboles $\mathbf{b}$
$y_k(i)$	Sortie du filtre adapté à l'utilisateur k à l'instant i
$\mathbf{y}(i)$	$(y_1(i)\ ,\ y_2(i)\ ,\ \ldots\ ,\ y_K(i))^T$
$\mathbf{y}$	$(\mathbf{y}(0)\ ,\ \ldots\ ,\ \mathbf{y}(N-1))$
$\overline{\mathbf{y}}$	$(\ y_1(0)\ ,\ y_2(0)\ ,\ \cdots\ ,\ y_K(0)\ ,\ y_1(1)\ ,\ \cdots\ ,\ y_K(N-1)\ )^T$
$\tilde{\mathbf{y}}$	$\overline{\mathbf{y}}$ filtré

Codes indépendants et décodage itératif

$\mathcal{A}$	Ensemble des valeurs de $\tilde{\mathbf{a}}_j(i)$
$\tilde{\mathbf{a}}_j(i)$	Vecteur des $K-1$ bits codés interférant sur le bit i de l'utilisateur j
$b_j(\ell)$	Bit d'information ℓ de l'utilisateur j
$\mathbf{b}_j$	Vecteur des kN/n bits d'information de l'utilisateur j
C	Capacité en bits d'information par symbole
$c_j(i)$	Bit codé i de l'utilisateur j avant entrelacement
$\mathbf{c}_j$	Vecteur des N bits codés de l'utilisateur j avant entrelacement
$\tilde{c}_j(i)$	Bit codé i de l'utilisateur j après entrelacement
$\tilde{\mathbf{c}}_j$	Vecteur des N bits codés de l'utilisateur j après entrelacement
$\tilde{\mathbf{c}}_u^j$	$\Pi_j^{-1}[\tilde{\mathbf{c}}_u]$
$iter$	Indice de l'itération
E_b	Energie moyenne par bit d'information
E_{b_j}	Energie par bit d'information de l'utilisateur j
E_s	Energie totale par symbole
E_{s_j}	Energie par symbole de l'utilisateur j
k/n	Taux du code convolutif
$h^j(t)$	Réponse impulsionnelle du canal de l'utilisateur j
$\mathcal{M}$	Alphabet résultant de la superposition de deux modulations BPSK
ν	Mémoire du code convolutif
$\mathbf{p}_{C,s}^u$	Vecteur des N APPs sur les bits codés de l'utilisateur u à l'étape s
$\mathbf{p}_{I,s}^u$	Vecteur des kN/n APPs sur les bits d'information de l'utilisateur u à l'étape s
$\tilde{\mathbf{p}}_{C,s}^{u,j}$	$\Pi_j^{-1}\left[\Pi_u(\mathbf{p}_{C,s}^u)\right]$
Π_j	Entrelacement de l'utilisateur j
$\tilde{r}(i)$	Sortie du filtre adapté à l'instant i
$\tilde{\mathbf{r}}$	$(\tilde{r}(0)\ ,\ \tilde{r}(1)\ ,\ \ldots\ ,\ \tilde{r}(N-1))$
$\mathbf{r}_j$	$\Pi_j^{-1}(\tilde{\mathbf{r}})$
s	Indice de l'étape
SNR_j	Rapport signal sur bruit de l'utilisateur j
$S_t(\mathbf{b})$	Signal émis au temps t correspondant à la séquence de symboles $\mathbf{b}$
T_c	Période d'un bit codé

Décodage par sphères

$\mathcal{A}$	Alphabet des symboles $b_k(i)$
d_{Emin}	Distance minimale d'un réseau
$\det(\Lambda)$	Volume fondamental d'un réseau Λ
$\boldsymbol{\eta}$	Vecteur de bruit additif
$\mathbf{G}$	Matrice génératrice d'un réseau
$\boldsymbol{\Gamma}$	Matrice de Gram de $\mathbf{G}$
$\gamma(\Lambda)$	Gain fondamental d'un réseau Λ
$\gamma_s(S)$	Gain de forme d'une constellation S
Λ	Réseau de points
S	Constellation d'un réseau de points
$\tau(\Lambda)$	Coefficient d'erreur du réseau Λ
$\mathbf{v}_i$	Vecteur de base d'un réseau
$\mathbf{x}$	Point d'un réseau
$\mathbf{y}$	Vecteur reçu correspondant à un point bruité

Système pratique avec une modulation MAQ-16

$\{a_n\}$	Séquence q-aire
$\{\hat{a}_n\}$	Séquence polyphase telle que $\hat{a}_n = e^{j\frac{2\pi}{q}a_n}$
j	$\sqrt{-1}$
λ	Longueur des séquences
$R_{\max}$	Corrélation maximale non-triviale
$R_{i,\ell}^P(\tau)$	Intercorrélation périodique entre $\{\hat{a}_n^i\}$ et $\{\hat{a}_n^\ell\}$ pour un décalage τ
$\overline{R^P}$	Moyenne de $\lvert R_{i,\ell}^P(\tau)\rvert$
$R_{i,\ell}^A(\tau,b,c)$	Corrélation apériodique tenant compte des symboles b et c de l'utilisateur i
$\overline{R^A}$	Moyenne de $\lvert R_{i,\ell}^A(\tau,b,c)\rvert$
x^R	Partie réelle de x
x^I	Partie imaginaire de x

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

Le besoin grandissant de communications rapides et fiables, notamment dans le domaine radio-mobile, nécessite une utilisation de plus en plus efficace des ressources disponibles. Les concepteurs de nouveaux réseaux de communications numériques sont amenés à distribuer le canal de transmission entre des utilisateurs en nombre toujours croissant. Les systèmes d'accès multiple les plus intuitifs sont ceux qui permettent aux différents utilisateurs de transmettre sur des bandes de fréquences disjointes (*Accès Multiple à Répartition en Fréquences* (AMRF) ou plus communément *Frequency Division Multiple Access* (FDMA)) ou pendant des laps de temps différents (*Accès Multiple à Répartition en Temps* (AMRT) ou plus communément *Time Division Multiple Access* (TDMA)). On obtient ainsi une partition du canal d'accès multiple en sous-canaux mono-utilisateur indépendants [52]. Bien qu'il existe des techniques d'allocation dynamique des ressources, ces deux types d'accès multiple conservent une certaine rigidité, qui peut nuire à la capacité en nombre d'utilisateurs du système, surtout quand ceux-ci émettent des données de manière sporadique.

L'étalement de spectre donne une solution plus flexible au problème. La largeur de bande du signal émis dans un système d'étalement de spectre est très supérieure au débit d'information. L'étalement est effectué par un codage très redondant et le signal est multiplié par une séquence pseudo-aléatoire pour lui donner des propriétés proches de celles d'un bruit. L'étalement de spectre a des qualités attrayantes. Il est tout d'abord très résistant à de nombreux types de brouillages. Sa résistance au brouillage intentionnel fut la raison première de son développement pendant la deuxième guerre mondiale. Les bonnes propriétés d'autocorrélation des séquences étudiées confèrent au système une forte diversité sur les canaux multi-trajets à évanouissements et permettent la conception d'un récepteur de complexité réduite adapté à ces canaux. Finalement les signaux à étalement de spectre sont très résistants au brouillage par d'autres signaux du même type de par les bonnes propriétés d'intercorrélation des séquences d'étalement. C'est cette dernière propriété qui est exploitée pour l'accès multiple. Les signaux étalés ont une amplitude très inférieure à celle de leur signal d'origine et peuvent être confondus avec du bruit si on ignore la séquence d'étalement. Leur discrétion les rend difficiles à détecter et leur assure une faible probabilité d'interception (LPI pour *Low Probability of Interception*). En cas d'interception, la séquence pseudo-aléatoire maintient la confidentialité des données. Finalement, la présence de la séquence pseudo-aléatoire aux bonnes propriétés d'autocorrélation permet des mesures de délais et de vitesses dans les radars ou les systèmes de positionnement comme le *Global Positioning System* (GPS).

L'accès multiple à répartition par codes (AMRC) ou, plus communément, *Code Division Multiple Access* (CDMA) utilise le bon comportement des signaux à étalement de spectre en présence d'autres signaux du même type. Nous nous intéresserons uniquement aux signaux étalés par séquence directe et nous laisserons de côté les signaux étalés par sauts de fréquences (*frequency hopping*). On alloue à chaque utilisateur une signature différente, faiblement corrélée avec les signatures des autres utilisateurs, qui fait office de séquence d'étalement. Pour obtenir les performances optimales d'un canal mono-utilisateur, il faut choisir des séquences orthogonales entre elles. Les séquences d'Hadamard ont cette propriété lorsque les utilisateurs sont synchrones. Le synchronisme existe par exemple dans la liaison descendante d'un réseau radio-mobile, quand une même station de base émet vers les différents mobiles. Malheureusement les séquences d'Hadamard perdent leur propriété d'orthogonalité si elles sont décalées les unes par rapport aux autres. Ce décalage se produit immanquablement dans un système asynchrone, comme la liaison montante d'un réseau radio-mobile, où les différents mobiles émettent indépendamment vers la même station de base. Les séquences alors utilisées, par exemple des séquences de Gold, ne sont plus orthogonales mais conservent des intercorrélations très faibles quels que soient les décalages. Le CDMA a été choisi comme technique d'accès multiple pour les récentes normes radio-mobiles américaine (IS-95) et européenne (UMTS).

Le comportement du système n'est bien évidemment pas le même dans le cas d'une liaison descendante ou dans celui d'une liaison montante. Dans le premier cas, l'orthogonalité des séquences garantit l'absence d'interférence entre les utilisateurs. Par contre, dans la liaison montante, malgré les faibles valeurs prises par les intercorrélations, subsiste une interférence entre utilisateurs qu'il est bon de prendre en compte pour améliorer les performances du système. A cette fin, le système doit être *centralisé* ou *coopérant*, c'est-à-dire que chaque récepteur doit disposer des signatures des autres utilisateurs. Dans le cas contraire d'un système *non centralisé*, le détecteur doit être adaptatif et estimer lui-même toutes les caractéristiques de l'interférence. La détection multi-utilisateurs complique la réception mais celle-ci est heureusement assurée par la station de base. Une détection adaptative au niveau du mobile peut être intéressante pour estimer l'interférence provenant d'autres cellules.

D'autres méthodes d'accès multiple existent. L'accès multiple par partage des taux (RSMA pour *Rate Splitting Multiple Access*), présenté dans [56], emploie une séparation judicieuse des K utilisateurs du système en, au plus, $2K-1$ utilisateurs virtuels codés. Chaque utilisateur peut être décodé indépendamment des autres utilisateurs dans de bonnes conditions et son signal retiré du signal reçu pour décoder les utilisateurs restants. Dans [61], la limite en nombre d'utilisateurs imposée par le nombre de séquences orthogonales de Hadamard dans un système synchrone est dépassée grâce à l'ajout d'utilisateurs étalés par des séquences PN. Les auteurs envisagent également de remplacer les signaux étalés avec des séquences PN par des signaux d'un autre type, comme des signaux TDMA. On peut avec ces méthodes obtenir 30 à 40% d'utilisateurs en plus tout en gardant, grâce à un décodage itératif, des performances proches de celles du système totalement orthogonal.

Le chapitre 2 de ce document débute par une présentation du système CDMA et par une mise en place des notations nécessaires à sa compréhension. Après avoir constaté l'inefficacité du détecteur mono-utilisateur classique, à savoir le filtre adapté, nous passons en revue un certain nombre de détecteurs prenant en compte l'interférence multi-utilisateurs. Pour toutes ces méthodes, le système doit être *centralisé*. S'il ne l'est pas, on peut mettre en œuvre des versions adaptatives des détecteurs présentés.

Nous proposons au chapitre 3 une nouvelle méthode d'accès multiple inspirée des turbo-codes. La séquence d'étalement est remplacée par un entrelaceur placé en sortie du codeur de chaque utilisateur. On emploie ensuite un décodage itératif à sortie souple, qui tient compte de l'interférence dans le calcul de la métrique de décodage. On atteint ainsi de fortes efficacités spectrales. L'étude de la propagation des LLRs d'une itération à l'autre du détecteur nous permet d'obtenir les performances du système dans le cas asymptotique d'entrelaceurs de taille infinie.

Dans le chapitre 4, nous revenons à un système CDMA classique et nous montrons qu'un système synchrone peut être modélisé par un réseau de points. Cette modélisation nous permet d'appliquer un décodage optimal des réseaux de points, le décodage par sphères, à la détection conjointe de tous les utilisateurs. Pour les systèmes asynchrones et les canaux multi-trajets, le décodage par sphères est associé à une élimination d'interférences. Le décodage par sphères a pour grand intérêt d'avoir une complexité indépendante de la taille de la modulation employée. Grâce au calcul du gain total du réseau équivalent au système multi-utilisateurs synchrone, nous obtenons le gain théorique de celui-ci.

Le chapitre 5 s'intéresse finalement à la conception d'un système CDMA employant une modulation à forte efficacité spectrale, la MAQ-16. La détection de cette modulation est rendue possible par le décodage par sphères. L'emploi de séquences complexes pour l'étalement est envisagé pour réduire les intercorrélations et améliorer la suppression d'interférences du cas asynchrone.

Le chapitre 6 regroupe enfin différentes conclusions sur les études menées et indique quelques perspectives pour poursuivre les travaux et parfaire les résultats.

Chapitre 2

La détection multi-utilisateurs

2.1 Introduction

Jusqu'au milieu des années 80, la seule méthode employée pour détecter les différents utilisateurs d'un système CDMA était celle qui ne se basait que sur l'observation individuelle de chaque utilisateur. A cause des défauts d'orthogonalité des signatures employées, le signal de chaque utilisateur interfère avec celui des autres utilisateurs. Cette interférence est un bruit supplémentaire qui peut fortement perturber la détection. En 1986, Verdú est le premier à penser que l'observation et la détection conjointe de tous les utilisateurs peut sensiblement améliorer les performances du système. Le détecteur traite alors l'interférence en signal déterministe et non plus en simple bruit. Il tente d'utiliser au mieux le signal reçu et les informations dont il dispose sur le canal et les signaux émis pour retrouver la séquence initialement émise par chaque utilisateur. Le problème d'interférence entre utilisateurs est très proche de celui de l'interférence entre symboles (IES ou ISI pour *Intersymbol Interference*). Il n'est donc pas étonnant que la plupart des techniques utilisées s'inspirent fortement de celles existant déjà en égalisation pour la suppression de l'IES. Il suffit de les modifier pour les adapter à la notion d'interférence entre utilisateurs.

Après une brève présentation du signal CDMA dans la section 2.2 nous présenterons à la section 2.3 le détecteur conventionnel, qui ne tient pas compte de l'interférence entre les utilisateurs. Nous décrirons ensuite dans la section 2.4 le détecteur à maximum de vraisemblance (ML pour *Maximum Likelihood*), qui fut le premier détecteur multi-utilisateurs découvert. La section 2.5 présente des détecteurs sous-optimaux, dont la complexité est linéaire en fonction du nombre d'utilisateurs et non exponentielle comme l'algorithme ML, qui s'inspirent des égaliseurs linéaires tandis que la section 2.6 présente les détecteurs à retour de décision, semblables aux égaliseurs du même nom. La section 2.7 traite de détecteurs dont la complexité est toujours linéaire en fonction du nombre d'utilisateurs mais dont la technique, qui consiste à soustraire du signal reçu l'interférence estimée de chaque utilisateur, n'est plus inspirée de l'égalisation. Finalement la section 2.8 tire quelques conclusions sur la détection multi-utilisateurs.

2.2 Modélisation du système multi-utilisateurs sur un canal gaussien asynchrone

Dans un système à étalement de spectre, le signal transmis sur le canal a une largeur de bande W très grande devant le débit d'information R. On appelle *gain d'étalement* le rapport $L = W/R$. L'étalement, dans un système à séquence directe, est effectué par une signature, qui module l'information. Il existe un autre type d'étalement de spectre, par sauts de fréquences (*frequency hopping* en anglais), dans lequel la fréquence de transmission varie de façon pseudo-aléatoire plusieurs fois pendant l'émission d'un même symbole. Nous ne nous intéressons ici qu'à l'étalement par séquence directe. Dans les systèmes d'accès multiple à répartition par codes par séquence directe (DS-CDMA pour *Direct Sequence Code Division Multiple Access*), la signature, qui étale l'utilisateur mais doit aussi permettre de le distinguer des autres utilisateurs, est une séquence pseudo-aléatoire. Si on note T la période d'un symbole à étaler et T_c la période d'un élément de la séquence, nommé *bribe* ou plus communément *chip*, on peut écrire le gain d'étalement

$$L = W/R = T/T_c$$

La période d'un *chip* est donc L fois plus courte que la période d'un symbole quand le spectre est étalé L fois. Cependant la période de la séquence pseudo-aléatoire peut être bien plus longue qu'un temps symbole et on ne retrouvera donc pas la même signature pour l'utilisateur k d'un symbole à l'autre.

Comme dans [72], on se place dans le cas simple d'un canal gaussien asynchrone et d'une signature qui ne varie pas d'un symbole à l'autre. La signature $s_k(t)$ de l'utilisateur k peut être de n'importe quelle forme, pourvu qu'elle ait des propriétés d'intercorrélation avec les autres signatures qui permettent de séparer facilement les utilisateurs. Elle doit également avoir de bonnes propriétés d'autocorrélation pour permettre la synchronisation de la séquence au niveau du récepteur et l'emploi d'un récepteur RAKE tirant parti de la diversité inhérente au système sur les canaux à trajets multiples [51][52].

Soit $b_k(i)$ le symbole $i \in \{0,N-1\}$ de l'utilisateur k émis dans l'intervalle de temps $[iT,(i+1)T[$ et appartenant à l'alphabet $\mathcal{A}$. Le plus souvent $\mathcal{A} = \{\pm 1\}$, cependant on verra au chapitre 4 que l'on peut aussi employer des modulations de taille plus importante. On note $\mathbf{b}(i) = (b_1(i)\ , b_2(i)\ , \ \dots\ , b_K(i))^T$ le vecteur de taille K contenant les $i^{\text{èmes}}$ symboles des K utilisateurs et $\mathbf{b} = (\mathbf{b}(0)\ , \ \dots\ , \mathbf{b}(N-1))$ la matrice de taille $K \times N$ regroupant tous les symboles émis de tous les utilisateurs. On note τ_k le retard de l'utilisateur k ($0 \leq \tau_k < T$), E_{s_k} l'énergie du signal sur fréquence porteuse et $\omega_k = \sqrt{2E_{s_k}}$

son amplitude. Le signal émis est

$$S_t(\mathbf{b}) = \sum_{i=0}^{N-1}\sum_{k=1}^{K} \omega_k b_k(i) s_k(t - iT - \tau_k) \quad \text{avec} \quad s_k(t) = 0 \quad \text{si} \quad t \notin [0,T[$$

On classe les utilisateurs par ordre de retards croissants :

$$0 \leq \tau_1 \leq \tau_2 \leq \ldots \leq \tau_K < T$$

Soit r_t le signal reçu et n_t un bruit blanc gaussien de variance $\sigma^2 = N_0$:

$$r_t = S_t(\mathbf{b}) + n_t$$

Pour plus de détails sur l'étalement de spectre, le lecteur peut se reporter à [64][52][76]. Pour une définition plus complète sur un canal à trajets multiples, on pourra se référer à la thèse d'Eckhard Papproth [46]. Le signal reçu étant maintenant défini, nous allons pouvoir nous intéresser aux méthodes permettant de détecter les symboles des différents utilisateurs.

2.3 Détection conventionnelle

Pour un système CDMA, le détecteur le plus simple, le détecteur conventionnel, est celui qui fonctionne comme si chaque utilisateur était seul sur le canal. Il fait une détection symbole par symbole à la sortie du filtre adapté à chaque utilisateur pour fournir une décision sur le symbole i de l'utilisateur k [64][52][73]. La sortie $y_k(i)$ du filtre adapté s'écrit

$$\begin{aligned} y_k(i) &\triangleq \int_{-\infty}^{+\infty} {s_k}^*(t - iT - \tau_k) r_t \mathrm{d}t \\ &= \sum_{j=0}^{N-1}\sum_{\ell=1}^{K} \omega_\ell \, b_\ell(j) \int_{-\infty}^{+\infty} s_\ell(t - jT - \tau_\ell) \, {s_k}^*(t - iT - \tau_k) \, \mathrm{d}t + n_k(i) \\ &= \omega_k \, b_k(i) + \sum_{j=0}^{N-1}\sum_{\substack{\ell=1 \\ \ell\neq k}}^{K} \omega_\ell \, b_\ell(j) \int_{-\infty}^{+\infty} s_\ell(t - jT - \tau_\ell) \, {s_k}^*(t - iT - \tau_k) \, \mathrm{d}t + n_k(i) \end{aligned} \tag{2.1}$$

où $n_k(i)$ est un bruit blanc additif gaussien (AWGN pour *Additive White Gaussian Noise*). Cette opération de filtrage adapté est également appelée désétalement. Le deuxième terme dans (2.1) représente l'interférence des autres utilisateurs du système, qui peut être importante quand les signatures ne sont pas orthogonales. Elle l'est d'autant plus quand l'utilisateur interférent est très puissant : c'est l'effet d'éblouissement (*near-far effect* en anglais). Le détecteur conventionnel fait totalement abstraction des interférences provenant des autres utilisateurs, qui sont considérées comme un bruit supplémentaire. Quand le nombre des utilisateurs est suffisamment grand, l'interférence totale peut être modélisée comme un bruit blanc additif gaussien [53][80]. Négliger l'interférence nuit à la détection, qui est alors très vulnérable aux effets de l'éblouissement.

Pour démontrer son inefficacité, nous avons simulé ce type de détection pour un système synchrone avec 4 utilisateurs modulés par un modulation de phase binaire (MDP-2 ou BPSK (*Binary Phase Shift Keying*)). On pourrait bien sûr dans ce cas choisir des séquences d'Hadamard [31] de longueur 4 pour obtenir un système parfait mais le but est ici de mettre en évidence les effets de l'interférence multi-utilisateurs sur un exemple simple. Les signatures employées sont donc des séquences de Gold [21][28] [52], dont la période est la durée d'un symbole. La construction et les propriétés de ces séquences binaires sont brièvement exposées dans la section 5.3.1. Le rapport signal à bruit de l'utilisateur 1 est fixé à 7 dB et ceux des 3 autres utilisateurs sont égaux. En les faisant varier, on peut observer les effets de l'éblouissement sur l'utilisateur 1. La figure 2.1 présente les performances des 4 utilisateurs pour un facteur d'étalement 7 et la figure 2.2 pour un facteur d'étalement 31. Dans le premier cas les signatures ont une matrice d'intercorrélation qui s'écrit

$$\mathbf{R} = \frac{1}{7}\begin{pmatrix} 7 & 3 & -1 & -1 \\ 3 & 7 & -1 & 3 \\ -1 & -1 & 7 & -1 \\ -1 & 3 & -1 & 7 \end{pmatrix}$$

et dans le second cas la matrice d'intercorrélation est

$$\mathbf{R} = \frac{1}{31}\begin{pmatrix} 31 & -1 & -1 & -1 \\ -1 & 31 & -1 & 7 \\ -1 & -1 & 31 & -1 \\ -1 & 7 & -1 & 31 \end{pmatrix}$$

On constate que le détecteur conventionnel réagit très mal à l'éblouissement quand le facteur d'étalement est peu supérieur au nombre d'utilisateurs ($L = 7$, $K = 4$). Les performances des différents utilisateurs dépendent fortement des valeurs de leurs intercorrélations avec les autres utilisateurs. L'utilisateur 2, qui possède les intercorrélations les plus fortes, a les performances les plus éloignées du cas mono-utilisateur idéal et inversement pour l'utilisateur 4. Pour le facteur d'étalement 31, le détecteur a un comportement raisonnable. On ne doit donc pas systématiquement employer une détection multi-utilisateurs, tout dépend des paramètres du système. Cependant dans les cas les plus intéressants, où l'on essaie de maximiser l'efficacité spectrale totale, c'est-à-dire le nombre d'utilisateurs pour un facteur d'étalement donné, la détection conventionnelle a des performances décevantes.

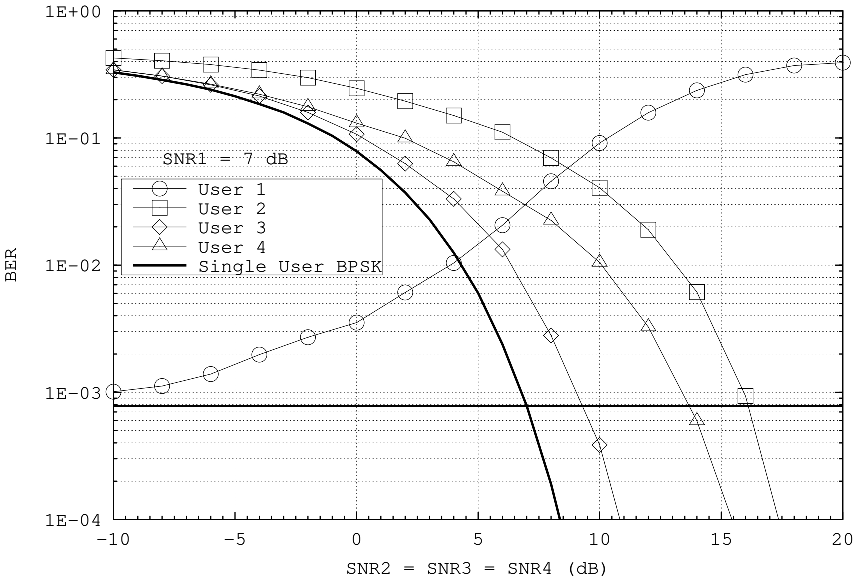


Fig. 2.1 – *Détecteur conventionnel pour* $K = 4$ *et* $L = 7$.

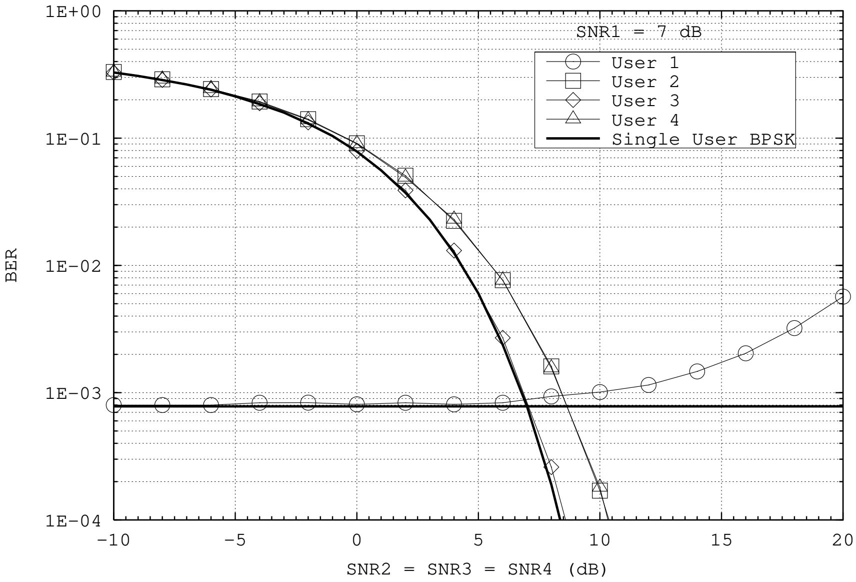


Fig. 2.2 – *Détecteur conventionnel pour* $K = 4$ *et* $L = 31$.

2.4 Détection à maximum de vraisemblance

Pour dépasser les performances de la détection conventionnelle, une détection plus intelligente doit être mise en œuvre, qui tient compte des interférences entre les utilisateurs.

2.4.1 Statistique suffisante

Le détecteur optimal au sens du maximum de vraisemblance pour un bruit n_t blanc additif gaussien détecte le signal $\hat{S}_t(\mathbf{b})$ de l'ensemble $\mathcal{S}$ de tous les signaux possibles tel que $\mathrm{P}\Big[\mathbf{b}|\left\{r_t, t \in \mathbb{R}\right\}\Big]$ soit maximale ou, comme on suppose l'équiprobabilité des séquences de symboles émises, que $\mathrm{P}\Big[\left\{r_t, t \in \mathbb{R}\right\}|\mathbf{b}\Big]$ soit maximale. Dans le cas gaussien, maximiser la probabilité $\mathrm{P}\Big[\left\{r_t, t \in \mathbb{R}\right\}|\mathbf{b}\Big]$ revient à minimiser la distance euclidienne entre le signal reçu r_t et le signal $S_t(\mathbf{b})$ supposé émis :

$$\begin{aligned}
\hat{S}_t(\mathbf{b}) &= \arg \min_{S_t(\mathbf{b}) \in \mathcal{S}} \|r_t - S_t(\mathbf{b})\|^2 \\
&= \arg \min_{S_t(\mathbf{b}) \in \mathcal{S}} \Big[2E_s - 2\mathrm{Re}\int_{-\infty}^{+\infty} r_t {S_t}^*(\mathbf{b})\mathrm{d}t\Big]
\end{aligned}$$

où E_s est l'énergie sur fréquence porteuse du signal $S_t(\mathbf{b})$.

On veut donc minimiser

$$d^2(\mathbf{b}) \triangleq \int_{-\infty}^{+\infty} |S_t(\mathbf{b})|^2 \mathrm{d}t - 2\mathrm{Re}\int_{-\infty}^{+\infty} {S_t}^*(\mathbf{b}) r_t \mathrm{d}t \tag{2.2}$$

Le produit scalaire peut s'écrire

$$\int_{-\infty}^{+\infty} {S_t}^*(\mathbf{b}) r_t \mathrm{d}t = \sum_{i=0}^{N-1} \sum_{k=1}^{K} \omega_k {b_k}^*(i) \int_{-\infty}^{+\infty} {s_k}^*(t - iT - \tau_k) r_t \mathrm{d}t$$

Puisque $y_k(i)$, la sortie du filtre adapté au $k^{\text{ième}}$ utilisateur pour le $i^{\text{ème}}$ symbole, s'écrit

$$y_k(i) = \int_{-\infty}^{+\infty} {s_k}^*(t - iT - \tau_k) r_t \mathrm{d}t$$

on obtient

$$\int_{-\infty}^{+\infty} {S_t}^*(\mathbf{b}) r_t \mathrm{d}t = \sum_{i=0}^{N-1} \sum_{k=1}^{K} \omega_k {b_k}^*(i) y_k(i)$$

ou matriciellement, en notant $\mathbf{y}(i) = (y_1(i) \ , \ \ldots \ , \ y_K(i))^T$ le vecteur des sorties des filtres adaptés aux K utilisateurs pour le symbole i et $\mathbf{D}_\omega = \mathrm{diag}(\omega_1, \ldots, \omega_K)$ la matrice diagonale de dimension $K \times K$ des amplitudes ω_k :

$$\int_{-\infty}^{+\infty} {S_t}^*(\mathbf{b}) r_t \mathrm{d}t = \sum_{i=0}^{N-1} \mathbf{b}^H(i) \mathbf{D}_\omega \mathbf{y}(i)$$

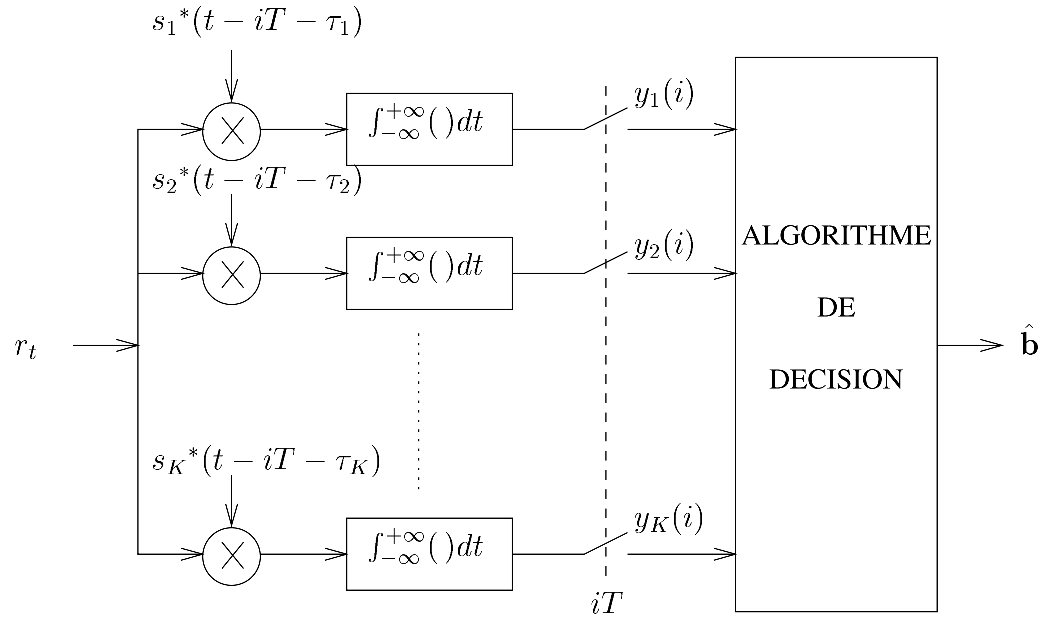


FIG. 2.3 – *Détecteur multi-utilisateurs à maximum de vraisemblance.*

On voit donc que $y_k(i)$ seul n'est pas une statistique suffisante pour la détection de $b_k(i)$. C'est pour cette raison que le détecteur conventionnel, qui se contente d'utiliser cette seule sortie, n'est pas performant. En revanche $\mathbf{y} = (\mathbf{y}(0)\ ,\ \ldots\ ,\ \mathbf{y}(N-1))$, la séquence complète des sorties des K filtres adaptés pour tous les symboles, est une statistique suffisante pour la détection de la séquence $\mathbf{b}$ la plus probable [72]. Le détecteur multi-utilisateurs à maximum de vraisemblance, représenté à la figure 2.3, est donc constitué d'un banc de filtres adaptés aux K utilisateurs suivi d'un algorithme de décision.

L'algorithme choisi doit avoir une complexité raisonnable pour pouvoir être utilisable. L'algorithme exhaustif, qui calcule $d^2(\mathbf{b})$ pour les 2^{KN} séquences $\mathbf{b}$ possibles dans le cas d'une modulation BPSK, dans le but de détecter KN symboles, calcule donc $2^{KN}/(KN)$ métriques par symbole. Sa complexité dépend exponentiellement de N, le nombre de symboles transmis !

La recherche exhaustive de $\hat{S}_t(\mathbf{b})$ parmi tous les $S_t(\mathbf{b})$ possibles est donc trop coûteuse. Les similarités entre les problèmes d'égalisation et de détection multi-utilisateurs, déjà évoquées au chapitre 1, suggèrent immédiatement l'emploi d'un algorithme communément utilisé en égalisation, l'algorithme de Viterbi [74][24]. Cette détection optimale a été proposée en 1986 par Sergio Verdú pour un canal gaussien asynchrone [72]. Elle a également été appliquée par la suite aux canaux multi-trajets [46], le principe étant exactement le même mais les écritures étant beaucoup plus complexes.

Avant de regarder l'algorithme proprement dit, intéressons-nous à la sortie du filtre adapté correspondant au $i^{\text{ème}}$ symbole de l'utilisateur k :

$$\begin{aligned}
y_k(i) &= \int_{-\infty}^{+\infty} r_t \; s_k{}^*(t - iT - \tau_k) \; \mathrm{d}t \\
&= \sum_{j=0}^{N-1} \sum_{\ell=1}^{K} \omega_\ell \; b_\ell(j) \; \int_{-\infty}^{+\infty} s_\ell(t-\tau_\ell) \; s_k{}^*(t-(i-j)T-\tau_k) \; \mathrm{d}t + n_k(i) \\
&= \sum_{j=0}^{N-1} \sum_{\ell=1}^{K} \omega_\ell \; R_{k\ell}(i-j) \; b_\ell(j) + n_k(i) \qquad (2.3)
\end{aligned}$$

où $R_{kl}(j) = \int_{-\infty}^{+\infty} s_\ell(t-\tau_\ell) s_k{}^*(t-jT-\tau_k)\mathrm{d}t$ est l'intercorrélation entre les séquences des utilisateurs ℓ et k décalées de j symboles et où $n_k(i)$ est le bruit en sortie du filtre adapté. Il faut réécrire les données de tous les utilisateurs en vecteurs colonnes de façon à faire apparaître une structure propice à l'emploi des méthodes classiques d'égalisation.

$$\begin{aligned}
\overline{\mathbf{b}} &= (\; b_1(0) \; , b_2(0) \; , \; \ldots \; , b_K(0) \; , b_1(1) \; , \; \ldots \; , b_K(N-1) \;)^T = (\; b_1 \; , \; \ldots \; , b_{KN} \;)^T \\
\overline{\mathbf{y}} &= (\; y_1(0) \; , y_2(0) \; , \; \ldots \; , y_K(0) \; , y_1(1) \; , \; \ldots \; , y_K(N-1) \;)^T = (\; y_1 \; , \; \ldots \; , y_{KN} \;)^T \\
\overline{\mathbf{n}} &= (\; n_1(0) \; , n_2(0) \; , \; \ldots \; , n_K(0) \; , n_1(1) \; , \; \ldots \; , n_K(N-1) \;)^T = (\; n_1 \; , \; \ldots \; , n_{KN} \;)^T \\
\overline{\mathbf{D}_\omega} &= \mathrm{diag}(\; \omega_1 \; , \omega_2 \; , \; \ldots \; , \omega_K \; , \omega_1 \; \ldots \; , \omega_K \;) = \mathrm{diag}(\; \omega_1 \; , \; \ldots \; , \omega_{KN} \;)
\end{aligned}$$

On écrit $\mathbf{R}(j)$ la matrice d'éléments $R_{kl}(j)$.

On peut alors, à partir de l'équation (2.3), écrire matriciellement la relation entre la séquence de symboles émis $\overline{\mathbf{b}}$ et l'ensemble des observations $\overline{\mathbf{y}}$ aux sorties du banc de filtres adaptés :

$$\overline{\mathbf{y}} = \mathbf{R}\overline{\mathbf{D}_\omega \mathbf{b}} + \overline{\mathbf{n}} \qquad (2.4)$$

avec

$$\mathbf{R} = \begin{bmatrix}
\mathbf{R}(0) & \mathbf{R}(-1) & \mathbf{0} & & \cdots & & \mathbf{0} \\
\mathbf{R}(1) & \mathbf{R}(0) & \mathbf{R}(-1) & \mathbf{0} & & & \mathbf{0} \\
\mathbf{0} & \mathbf{R}(1) & \mathbf{R}(0) & \mathbf{R}(-1) & \mathbf{0} & & \mathbf{0} \\
\vdots & & \ddots & \ddots & \ddots & & \vdots \\
\mathbf{0} & \cdots & & & \mathbf{0} & \mathbf{R}(1) & \mathbf{R}(0)
\end{bmatrix}$$

$\overline{\mathbf{b}}$, $\overline{\mathbf{y}}$ et $\overline{\mathbf{n}}$ sont des vecteurs colonnes de taille KN. $\overline{\mathbf{D}_\omega}$ est une matrice diagonale de taille $KN \times KN$ et $\mathbf{R}$ est la matrice des corrélations, également de taille $KN \times KN$. On considère que le canal est fixe et on se place dans le cas d'un seul trajet, ce qui explique que toutes les matrices $\mathbf{R}(j)$ avec $|j| > 1$ soient nulles. Le bruit $\overline{\mathbf{n}}$ est un bruit additif gaussien complexe de matrice de corrélation $\mathbf{\Phi}_{nn} = 2\mathbf{R}N_0$.

La figure 2.4 représente un système asynchrone simple à 3 utilisateurs. On peut y voir indiquées les valeurs des intercorrélations de l'utilisateur 2 avec les deux autres utilisateurs. La figure 2.5 permet de visualiser la notation matricielle de l'équation (2.4)

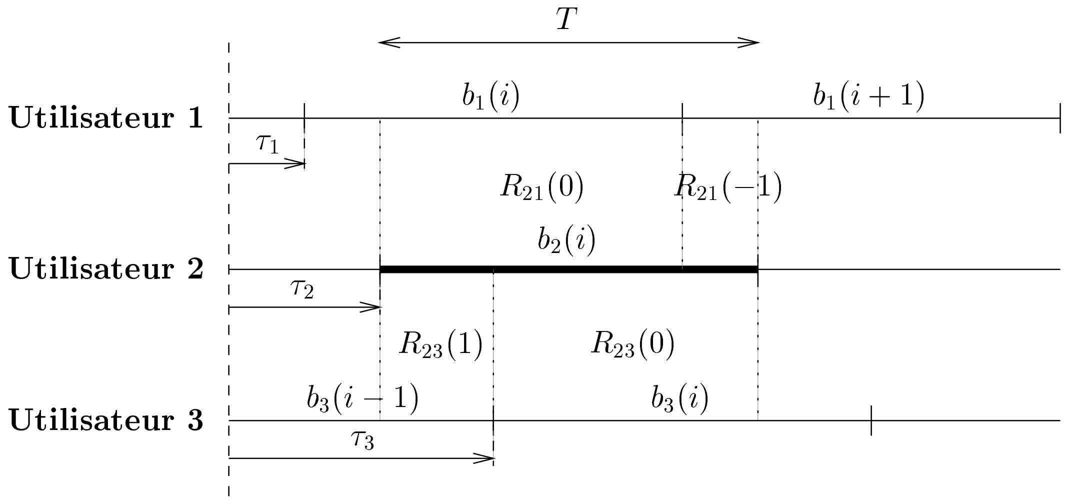


FIG. 2.4 – *Système asynchrone avec 3 utilisateurs : interférences sur l'utilisateur 2.*

et de comprendre les liens étroits entre égalisation et détection multi-utilisateurs. En effet la figure 2.5 pourrait représenter un système avec IES, dans lequel l'IES varierait d'un symbole à l'autre avec une périodicité K.

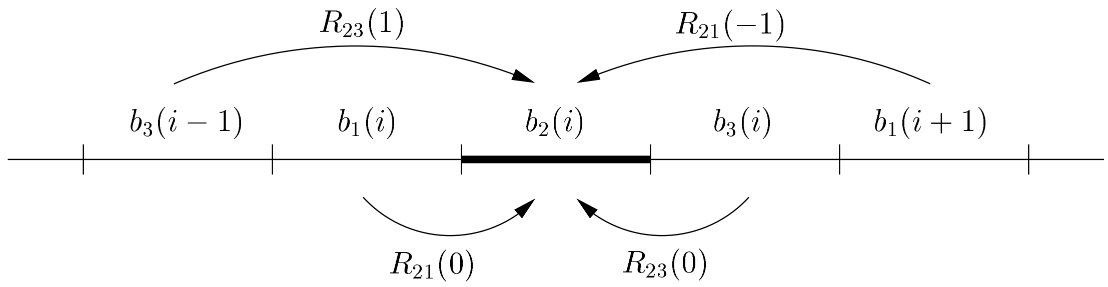


FIG. 2.5 – *Analogie entre détection multi-utilisateurs et égalisation.*

2.4.2 Algorithme de Verdú

Il faut trouver une façon plus efficace que la méthode exhaustive pour minimiser $d^2(\mathbf{b})$. On peut utiliser la dépendance séquentielle des $b_k(i)$, qui permet d'exprimer $d^2(\mathbf{b})$ par une somme de termes dépendant seulement d'un nombre limité de variables à un instant donné. Supposons que l'on puisse trouver

- un système discret markovien tel que $\mathbf{x}_{i+1} = f(\mathbf{x}_i, b_i)$ de condition initiale $\mathbf{x}_{i_0}$
- une métrique de transition $\lambda_i(\mathbf{x}_i, b_i)$
- une bijection entre l'ensemble des séquences transmises et l'ensemble des séquences de contrôle $\left\{b_i,\ i = i_0 \ldots i_f\right\}$

tels que

$$\Omega(\mathbf{b}) = -d^2(\mathbf{b}) = \sum_{i=i_0}^{i_f} \lambda_i(\mathbf{x}_i,b_i)$$

Cette structure en treillis, qui représente la structure markovienne du système (définition d'états, de transitions entre les états et de métriques sur les transitions), est celle qui convient parfaitement à l'algorithme de Viterbi. Il ne reste plus qu'à décrire les trois points ci-dessus pour le système étudié et on pourra appliquer l'algorithme de Viterbi sur le treillis correspondant, pour maximiser la métrique totale $\Omega(\mathbf{b})$, c'est-à-dire minimiser $d^2(\mathbf{b})$.

La bijection entre les symboles émis, qui sont regroupés dans la matrice $\mathbf{b}$, et la séquence de contrôle $\left\{b_i,\ i = i_0 \dots i_f\right\}$ qui se prête bien à la structure en treillis est décrite par le vecteur $\overline{\mathbf{b}}$. Cet arrangement permet de donner une structure séquentielle au système. Il s'écrit mathématiquement en faisant la division entière par K de l'entier i, de quotient $\eta(i)$ et de reste $\kappa(i)$:

$$i = \eta(i)K + \kappa(i) \quad \text{avec} \quad \kappa(i) = 1,\dots,K$$

Le symbole de contrôle b_i est le symbole $\eta(i)$ de l'utilisateur $\kappa(i)$.

$$b_i \triangleq b_{\kappa(i)}\left(\eta(i)\right) \in \mathcal{A} \tag{2.5}$$

Théorème 2.4.1 (Structure en treillis du système multi-utilisateurs) *Le système multi-utilisateurs étudié peut se mettre sous la forme recherchée de caractéristiques ci-dessous :*

$$\Omega(\mathbf{b}) = \sum_{i=1}^{KN} \lambda_i(x_i,b_i)$$

$$\begin{array}{rrcl}
\textit{avec} & \lambda_i(\mathbf{x}_i,b_i) & \triangleq & 2Re({b_i}^*\omega_i y_i) - |b_i|^2{\omega_i}^2 - 2Re\Big({b_i}^*\omega_i \mathbf{x}^{i^T}\mathbf{W}^{\kappa(i)}\mathbf{G}^{\kappa(i)}\Big) \\
 & \mathbf{W}^{\kappa(i)} & = & diag\Big(\omega_{\kappa(i+1)}\ ,\ \omega_{\kappa(i+2)}\ ,\ \dots\ ,\ \omega_{\kappa(i+K-1)}\Big) \\
 & \mathbf{x}^{i+1} & \triangleq & \Big(x_2^i\ x_3^i\ \dots\ x_{K-1}^i\ b_i\Big)^T \\
 & \mathbf{x}^1 & \triangleq & \Big(0\ 0\ \dots\ 0\ 0\Big)^T \\
 & \mathbf{G}^{\kappa(i)} & \triangleq & \Big(\ R_{\kappa(i),\kappa(i)+1}(1)\ ,\ \dots\ ,\ R_{\kappa(i),K}(1)\ ,\ R_{\kappa(i),1}(0)\ ,\ \dots\ ,\ R_{\kappa(i),\kappa(i)-1}(0)\ \Big)^T
\end{array}$$

$\mathbf{x}^{i+1}$ est un vecteur colonne obtenu en prenant les $K-1$ dernières lignes du vecteur colonne $\mathbf{x}^i$ et en rajoutant à la dernière ligne la nouvelle valeur b_i. La colonne $\mathbf{G}^{\kappa(i)}$

contient les intercorrélations du symbole de l'utilisateur $\kappa(i)$ avec les $K-1$ symboles précédemment reçus, chaque symbole appartenant à un utilisateur différent. On suppose que $\int_0^T |s_k(t)|^2 \mathrm{d}t = 1 \quad \text{pour} \quad k = 1,\ldots,K$.

Preuve 2.4.1 *On commence par réécrire le signal $S_t(\mathbf{b})$ en utilisant le réarrangement :*

$$\begin{aligned} S_t(\mathbf{b}) &= \sum_{i=0}^{N-1} \sum_{k=1}^{K} \omega_k b_k(i) s_k(t - iT - \tau_k) \\ &= \sum_{i=1}^{NK} \omega_i b_i s_{\kappa(i)} \left(t - \eta(i)T - \tau_{\kappa(i)}\right) \end{aligned}$$

On en déduit

$$\begin{aligned} \|S(\mathbf{b})\|^2 &= \int_{-\infty}^{+\infty} |S_t(\mathbf{b})|^2 dt \\ &= \int_{-\infty}^{+\infty} \sum_{i=1}^{KN} \Big[{\omega_i}^2 |b_i|^2 \Big| s_{\kappa(i)} \left(t - \eta(i)T - \tau_{\kappa(i)}\right) \Big|^2 \\ &\quad + \sum_{\substack{j \neq i \\ j=1}}^{NK} \omega_i \omega_j {b_i}^* b_j {s_{\kappa(i)}}^* \left(t - \eta(i)T - \tau_{\kappa(i)}\right) s_{\kappa(j)} \left(t - \eta(j)T - \tau_{\kappa(j)}\right) \Big] dt \end{aligned}$$

Une réorganisation de cette somme fait apparaître les doubles-produits du signal de l'utilisateur i avec les signaux précédents $(1 \leq j \leq i-1)$:

$$\begin{aligned} \|S(\mathbf{b})\|^2 = \sum_{i=1}^{NK} \Big[& {\omega_i}^2 |b_i|^2 \int_{-\infty}^{+\infty} \Big| s_{\kappa(i)} \left(t - \eta(i)T - \tau_{\kappa(i)}\right) \Big|^2 dt \\ & + 2Re\Big(\sum_{j=1}^{i-1} \omega_i \omega_j {b_i}^* b_j \int_{-\infty}^{+\infty} {s_{\kappa(i)}}^* \left(t - \eta(i)T - \tau_{\kappa(i)}\right) s_{\kappa(j)} \left(t - \eta(j)T - \tau_{\kappa(j)}\right) dt \Big) \Big] \end{aligned}$$

On introduit ainsi une notion de passé *de l'instant i, caractérisée par la somme de 1 à $i-1$, qui est primordiale dans l'algorithme de Viterbi.*

Comme on a supposé la fonction $s_k(t)$ nulle en dehors de l'intervalle $[0,T[$, on peut montrer que

$$ {s_{\kappa(i)}}^* \left(t - \eta(i)T - \tau_{\kappa(i)}\right) s_{\kappa(i-\ell)} \left(t - \eta(i-\ell)T - \tau_{\kappa(i-\ell)}\right) \neq 0 \quad \Longleftrightarrow \quad \ell \in \left\{0,\ldots,K-1\right\} $$

En supposant $b_i = 0 \quad \text{pour} \quad i < i_0$, on peut écrire

$$\begin{aligned} \|S(\mathbf{b})\|^2 = \sum_{i=1}^{NK} \Big[& |b_i|^2 {\omega_i}^2 + 2Re\Big(\sum_{\ell=1}^{K-1} \omega_i \omega_{i-\ell} {b_i}^* b_{i-\ell} \\ & \times \int_{-\infty}^{+\infty} {s_{\kappa(i)}}^* \left(t - \eta(i)T - \tau_{\kappa(i)}\right) s_{\kappa(i-\ell)} \left(t - \eta(i-\ell)T - \tau_{\kappa(i-\ell)}\right) dt \Big) \Big] \end{aligned} \tag{2.6}$$

et

$$\int_{-\infty}^{+\infty} s_{\kappa(i)}{}^{*}\left(t-\eta(i)T-\tau_{\kappa(i)}\right) s_{\kappa(i-\ell)}\left(t-\eta(i-\ell)T-\tau_{\kappa(i-\ell)}\right) dt = \mathbf{G}_{K-\ell}^{\kappa(i)}$$
$$= R_{\kappa(i),\kappa(i-\ell)}\left(\eta(i)-\eta(i-\ell)\right)$$

Comme on peut le voir dans l'équation (2.6), pour un i *donné de la première sommation, le terme sommé n'est fonction que de* b_i*, le symbole de contrôle, et des* $K-1$ *symboles de contrôle précédents, qui forment l'état* $\mathbf{x}^i$*. L'état* $\mathbf{x}^{i+1}$ *est donc formé de* b_i *et des* $K-2$ *derniers symboles de contrôle :*

$$\mathbf{x}^{i+1} = \left(x_1^{i+1}\ x_2^{i+1}\ \ldots\ x_{K-1}^{i+1}\right)^T = \left(x_2^{i}\ x_3^{i}\ \ldots\ x_{K-1}^{i}\ b_i\right)^T = \left(b_{i-K+2}\ b_{i-K+3}\ \ldots\ b_{i-1}\ b_i\right)^T$$

On en déduit que $x_{K-\ell}^i = b_{i-\ell}$ *et on peut donc réécrire (2.6)*

$$\begin{aligned}\|S(\mathbf{b})\|^2 &= \sum_{i=1}^{NK}\left[|b_i|^2{\omega_i}^2 + 2Re\Big(\sum_{\ell=1}^{K-1}\omega_i\omega_{i-\ell}{b_i}^{*}x_{K-\ell}^{i}\mathbf{G}_{K-\ell}^{\kappa(i)}\Big)\right]\\ &= \sum_{i=1}^{NK}\left[|b_i|^2{\omega_i}^2 + 2Re\Big(\sum_{j=1}^{K-1}\omega_i\omega_{i+j}{b_i}^{*}x_{j}^{i}\mathbf{G}_{j}^{\kappa(i)}\Big)\right] \qquad (2.7)\end{aligned}$$

De même

$$\int_{-\infty}^{+\infty} {S_t}^{*}(\mathbf{b})r_t dt = \sum_{i=1}^{NK}\omega_i {b_i}^{*} y_i \qquad (2.8)$$

A remplaçant (2.7) et (2.8) dans (2.2), on obtient

$$\Omega(\mathbf{b}) = \sum_{i=1}^{NK}\left[2Re({b_i}^{*}\omega_i y_i) - |b_i|^2{\omega_i}^2 - 2Re\Big({b_i}^{*}\omega_i {\mathbf{x}^i}^T\mathbf{W}^{\kappa(i)}\mathbf{G}^{\kappa(i)}\Big)\right] = \sum_{i=1}^{NK}\lambda_i(\mathbf{x}_i, b_i)$$

Le système multi-utilisateurs peut donc se mettre sous une forme exploitable par un algorithme de Viterbi. Si chaque utilisateur emploie une modulation binaire, le treillis associé à un tel système comporte 2^{K-1} états, chacun de ces états étant relié à deux états de l'étape précédente et à deux états de l'étape suivante. A chaque étape, il faut tester 2^K métriques et le treillis contient NK étapes. En tout ce détecteur calcule $NK2^K$ métriques pour détecter NK symboles, soit 2^K métriques par symbole.

Nous avons simulé ce détecteur à maximum de vraisemblance avec le même système que dans la section 2.3. Cependant ce système est ici asynchrone. En effet, dans le cas synchrone, l'utilisation de l'algorithme de Viterbi ne s'impose pas puisque chaque symbole n'interfère qu'avec un symbole de chacun des autres utilisateurs et que le détecteur ML exhaustif n'a donc que 2^K métriques différentes à calculer. Les résultats présentés sur les

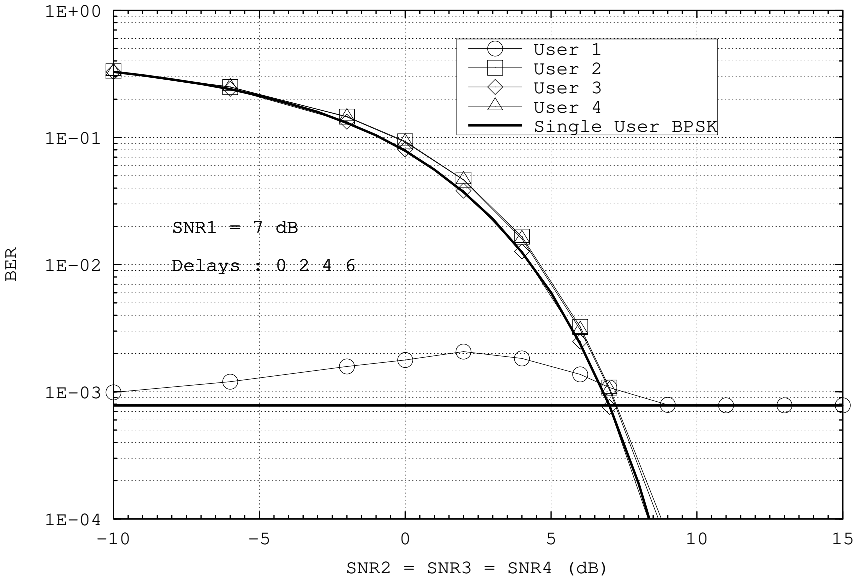


FIG. 2.6 – *Détecteur à maximum de vraisemblance pour $K = 4$ et $L = 7$.*

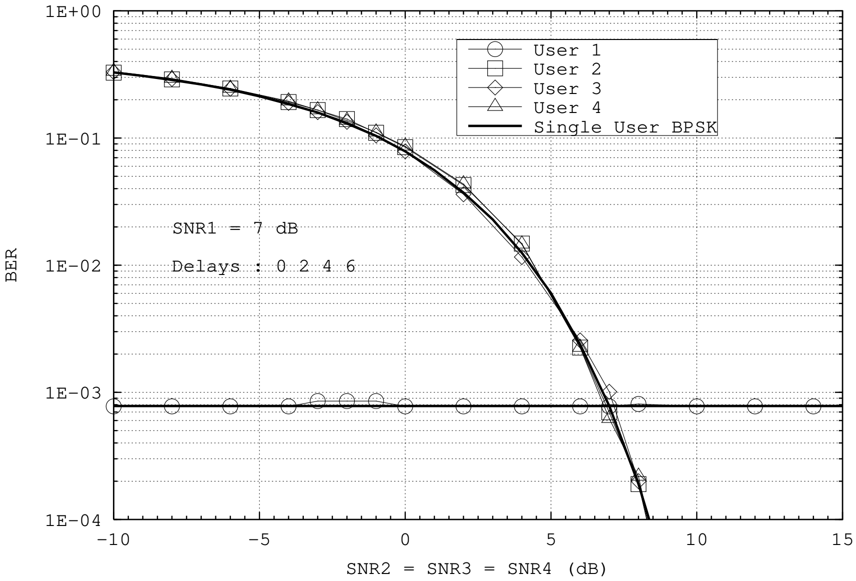


FIG. 2.7 – *Détecteur à maximum de vraisemblance pour $K = 4$ et $L = 31$.*

figures 2.6 et 2.7 montrent bien que la détection optimale est très résistante au problème de l'éblouissement. Les performances de chaque utilisateur sont quasiment identiques à celles d'un utilisateur qui serait seul à transmettre sur le canal et d'autant plus proches que les séquences sont longues et ont de bonnes propriétés d'intercorrélation.

La complexité du détecteur à maximum de vraisemblance de Verdú, qui ne dépend plus exponentiellement que du nombre des utilisateurs, est évidemment bien meilleure que celle d'une recherche exhaustive dans un système asynchrone. Néanmoins cette dépendance exponentielle reste prohibitive : elle limite le nombre d'utilisateurs à un nombre $K < 10$ [52]. Pour augmenter le nombre d'utilisateurs traités conjointement, d'autres types de détecteurs, eux aussi inspirés des égaliseurs utilisés dans les systèmes mono-utilisateur, ont été développés. Leur complexité croît linéairement avec K.

2.5 Méthodes linéaires

De même qu'en égalisation, on dispose d'une série d'observations regroupées dans le vecteur $\overline{\mathbf{y}}$. Si on se contente de faire une détection à seuil sur les éléments de $\overline{\mathbf{y}}$, on obtient les résultats désastreux de la détection conventionnelle. Comme en égalisation, on peut filtrer ces observations ou, de manière équivalente, multiplier le vecteur par une matrice $\mathbf{C}$ pour obtenir un vecteur d'observations $\tilde{\mathbf{y}}$ de meilleure qualité, c'est-à-dire qui se prête mieux à la détection à seuil. La structure, très simple, de ce type de détecteurs est présentée à la figure 2.8.

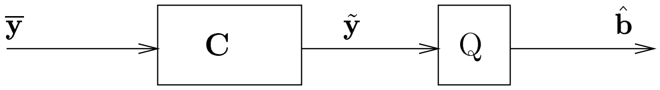


FIG. 2.8 – *Détecteur multi-utilisateurs linéaire.*

2.5.1 Critère de forçage à zéro (décorrélateur)

Le détecteur linéaire vérifiant le critère de forçage à zéro (ZF pour *Zero Forcing*), inspiré de son homonyme en égalisation, cherche à éliminer entièrement les interférences, c'est pourquoi il est aussi nommé *décorrélateur* [38][39]. Pour éliminer totalement les interférences, il suffit d'inverser la matrice des corrélations $\mathbf{R}$ et de multiplier $\overline{\mathbf{y}}$ par cet inverse, que l'on note $\mathbf{C}_{\mathrm{ZF}}$. On obtient alors

$$\tilde{\mathbf{y}} = \mathbf{C}_{\mathrm{ZF}}\overline{\mathbf{y}} = \mathbf{R}^{-1}\overline{\mathbf{y}} = \overline{\mathbf{D}_\omega \mathbf{b}} + \mathbf{R}^{-1}\overline{\mathbf{n}}$$

On remarque immédiatement que, dans le cas le plus courant d'une modulation BPSK, toute estimation d'amplitude est ici inutile, puisque le facteur multiplicatif ω_k ne perturbe pas la détection à seuil. Le décorrélateur est le détecteur qui résiste le mieux à l'éblouissement puisque l'interférence des autres utilisateurs, quelle que soit leur puissance, est entièrement supprimée. Malheureusement, comme en égalisation, il élimine l'interférence sans tenir compte du bruit et augmente ainsi le niveau de celui-ci. Les performances s'en trouvent dégradées.

2.5.2 Critère de minimisation de l'erreur quadratique moyenne (*MMSE*)

Pour réduire l'augmentation de bruit due à la suppression de l'interférence, on a recours, comme en égalisation classique, à un deuxième critère, qui va permettre de trouver le juste milieu entre suppression de l'interférence et minimisation du bruit [40][79]. Ce critère est le critère de minimisation de l'Erreur Quadratique Moyenne (EQM), MMSE (*Minimum Mean Square Error*) en anglais. On cherche la matrice $\mathbf{C}_{\mathrm{MMSE}}$ qui minimise $E\Big\{\|\tilde{\mathbf{y}} - \overline{\mathbf{D}_\omega}\mathbf{b}\|^2\Big\}$ et on obtient

$$\mathbf{C}_{\mathrm{MMSE}} = \Big(\frac{2N_0}{\overline{\mathbf{D}_\omega}^2} + \mathbf{R}\Big)^{-1} \tag{2.9}$$

On peut faire les deux mêmes remarques qu'en égalisation. Quand le niveau de l'interférence augmente et que le bruit devient négligeable, le détecteur MMSE tend vers le décorrélateur. Par contre, quand l'interférence est négligeable, il ne tient plus compte que du bruit et ses performances rejoignent celles du détecteur conventionnel, alors que les performances du décorrélateur, handicapé par un bruit plus fort, s'en éloignent. L'emploi du détecteur MMSE, contrairement à celui du décorrélateur, nécessite une estimation des amplitudes puisque $\overline{\mathbf{D}_\omega}$ apparaît dans l'équation (2.9).

L'estimation et l'inversion de la matrice $\mathbf{R}$ de taille $KN \times KN$ est très coûteuse, c'est pourquoi d'autres structures évitant l'inversion, comme la structure polynomiale, ou travaillant sur des matrices plus petites, comme la structure employant un filtre matriciel, ont été développées.

2.5.3 Structure polynomiale

Pour supprimer l'inversion de la matrice $\mathbf{R}$, on calcule la matrice $\mathbf{C}_{\mathrm{PE}}$ grâce aux puissances de $\mathbf{R}$:

$$\mathbf{C}_{\mathrm{PE}} = \sum_{i=0}^{N_C-1} c_i \mathbf{R}^i$$

où N_C représente le nombre de coefficients c_i, qui permettent d'approcher le critère choisi [44], par exemple $\mathbf{C}_{\mathrm{PE}} \simeq \mathbf{R}^{-1}$.

La matrice $\mathbf{R}$ modélise les effets combinés de l'étalement, du canal et du filtre adapté. On peut donc l'implanter directement comme sur la figure 2.9. Cette structure peut

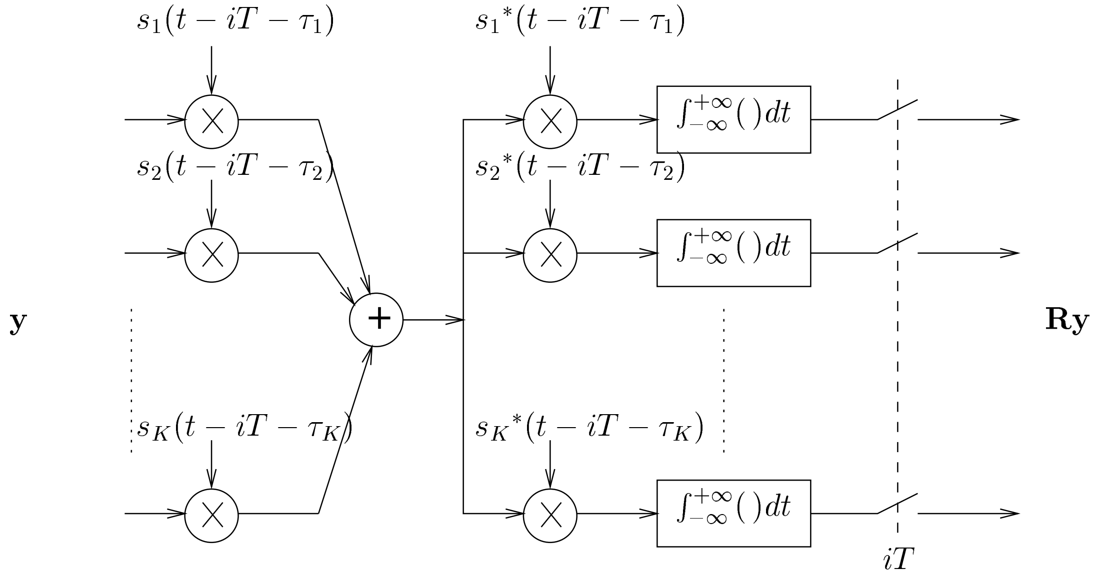


FIG. 2.9 – *Implantation de la matrice* $\mathbf{R}$ *de la structure polynomiale.*

satisfaire le critère de forçage à zéro de manière exacte si la longueur N de la séquence des symboles est finie mais le nombre d'étages N_C est alors prohibitif. Les coefficients c_i restent valables pour une plage de variation des paramètres du système relativement importante et la taille de cette plage augmente avec le nombre d'étages. Enfin, le nombre de composants nécessaires à l'implantation de cette structure augmente linéairement avec KN_C.

2.5.4 Détection multi-utilisateurs linéaire avec un filtre matriciel

On travaille maintenant sur des matrices plus petites. En effet, l'emploi de la matrice $\mathbf{R}$ de taille $KN \times KN$ va être remplacé par un filtre matriciel $\mathbf{C}$, dont les $2N_C+1$ coefficients $\mathbf{C}(j)$, $j=-N_C,\ldots,N_C$ sont des matrices $K \times K$ (cf. Figure 2.10) [39][40][46]. Son entrée $\mathbf{y}(i)$ et sa sortie $\tilde{\mathbf{y}}(i)$ à l'instant i sont des vecteurs de dimension K. La relation

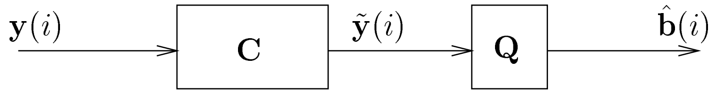


FIG. 2.10 – *Détecteur multi-utilisateurs linéaire construit à l'aide d'un filtre matriciel.*

entre l'entrée et la sortie du filtre est :

$$\tilde{\mathbf{y}}(i) = (\mathbf{C} * \mathbf{y})(i) = \sum_{j=-N_C}^{N_C} \mathbf{C}(j)\mathbf{y}(i-j)$$

On peut optimiser le filtre $\mathbf{C}$ selon le critère de forçage à zéro ou le critère MMSE grâce à la méthode notamment présentée dans [46].

2.6 Détection multi-utilisateurs à retour de décision (*DFE*)

Toujours en s'inspirant de l'égalisation classique, on peut réaliser le détecteur sous-optimal à retour de décision (DFE pour *Decision Feedback Equalization*) décrit à la figure 2.11 et fonctionnant sur le même principe que son homonyme en égalisation [18][19][34][79]. Comme en égalisation, la décision s'appuie non seulement sur le vecteur d'observations $\mathbf{y}(i)$ filtré par un filtre transverse matriciel $\mathbf{C}_f$, mais aussi sur les symboles déjà détectés $\hat{\mathbf{b}}(i)$, qui, après filtrage par un filtre matriciel de rétroaction $\mathbf{C}_b\mathbf{D}_\omega$, où $\mathbf{D}_\omega = \mathrm{diag}(\omega_1, \ldots, \omega_K)$, sont soustraits de la sortie du filtre transverse. Le

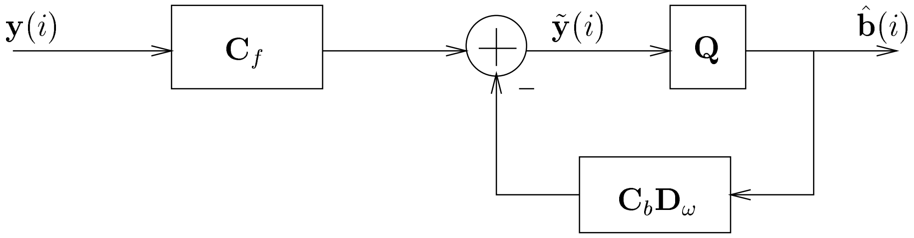


FIG. 2.11 – *Détecteur multi-utilisateurs à retour de décision.*

vecteur d'entrée $\tilde{\mathbf{y}}(i)$ de l'organe de décision $\mathbf{Q}$ s'écrit :

$$\begin{aligned} \tilde{\mathbf{y}}(i) &= (\mathbf{C}_f * \mathbf{y})(i) - (\mathbf{C}_b\mathbf{D}_\omega * \hat{\mathbf{b}})(i) \\ &= \sum_j \left[\mathbf{C}_f(j)\mathbf{y}(i-j) - \mathbf{C}_b(j)\mathbf{D}_\omega\hat{\mathbf{b}}(i-j)\right] \end{aligned} \tag{2.10}$$

Par définition, le filtre de rétroaction doit être causal puisqu'il supprime l'interférence des symboles déjà détectés. La définition du *passé* en détection multi-utilisateurs est cruciale. Elle est évidemment temporelle mais dépend aussi des utilisateurs. On ne doit pas choisir au hasard l'ordre de détection des utilisateurs. Le plus puissant d'entre eux sera le plus facile à détecter. Cette détection une fois effectuée, on peut supprimer la puissante interférence de cet utilisateur pour détecter plus facilement les autres. Le dernier utilisateur détecté bénéficie des détections de tous les autres utilisateurs, ce qui compense le handicap que représente sa faible puissance. Cette définition du *passé* entraîne immédiatement une définition de la *causalité* [19][46] :

Définition 2.6.1 (Causalité) *Soit $u_k(i)$ la composante du vecteur $\mathbf{u}(i)$ à l'instant i correspondant à l'utilisateur k.*

– *Le* présent *de cette composante est lui-même.*

– *Le* passé *de cette composante est formé de celles de tous les utilisateurs aux instants précédents $i-1$, $i-2$, ... et de celles des utilisateurs $1, \ldots, k-1$ à l'instant i.*

Un filtre matriciel dont l'entrée et la sortie ont même dimension, est dit causal *si chaque composante de la séquence de ses vecteurs de sortie ne dépend que de son passé et de son présent. Il est* strictement causal *si elle ne dépend que de son passé.*

Si l'on accepte cette définition du passé et que l'on veut détecter en priorité les utilisateurs les plus puissants, on doit classer ceux-ci du plus puissant au moins puissant ($\omega_1 \geq \omega_2 \geq \ldots \geq \omega_K$). L'ordre de détection est donc maintenant complètement indépendant des retards relatifs des utilisateurs, à la différence de l'ordre de détection employé par le détecteur ML de la section 2.4.

Le filtre $\mathbf{C}_b$ doit être strictement causal donc, pour tout j négatif, $\mathbf{C}_b(j)$ est une matrice nulle. Si on prend un filtre de taille N_b, le coefficient est une matrice non nulle pour tout j compris entre 1 et N_b-1. Le coefficient $\mathbf{C}_b(0)$ est un cas spécial : on doit avoir $\mathbf{C}_{b,k\ell}(0)$ nul si $k \leq \ell$. Le filtre transverse doit supprimer l'interférence entre utilisateurs due au *futur*. Il est anti-causal et on doit donc introduire un délai de décision dans le processus pour qu'il soit réalisable en pratique. Pour tout j positif $\mathbf{C}_f(j)$ est une matrice nulle. Si on prend un filtre de taille N_f, le coefficient est une matrice non nulle pour tout j compris entre -1 et $-N_f+1$. Le coefficient $\mathbf{C}_f(0)$ est un cas spécial : on doit avoir $\mathbf{C}_{f,k\ell}(0)$ nul si $k > \ell$.

On peut remarquer que les filtres ont des tailles finies. Un filtre transverse de taille infinie nécessiterait un délai de décision infini. Le détecteur sera sous-optimal. On peut trouver les coefficients des filtres selon le critère ZF ou le critère MMSE en utilisant la méthode exposée dans [46].

Les figures 2.12 et 2.13 permettent de comparer les différents types de détecteurs multi-utilisateurs basés sur des techniques connues en égalisation. Le système synchrone simulé est le même qu'à la section 2.3.

Le détecteur linéaire ZF décorrèle complètement les signaux des utilisateurs. Il est donc totalement résistant à l'éblouissement. La probabilité d'erreur de l'utilisateur 1 reste parfaitement constante mais à un niveau d'erreur très supérieure à celui du détecteur optimal et même à celui du détecteur conventionnel pour un niveau d'interférence faible car la puissance du bruit est augmentée par la décorrélation.

Le détecteur linéaire MMSE prend en compte la puissance des utilisateurs. Pour une interférence multi-utilisateurs faible, quand le bruit est prédominant, il a les performances du détecteur conventionnel et pour une interférence forte, quand le bruit est négligeable, il a les performances d'un détecteur linéaire ZF. On retrouve donc la propriété des performances de l'égaliseur MMSE, qui rejoignent les performances de l'égaliseur ZF quand le bruit devient négligeable.

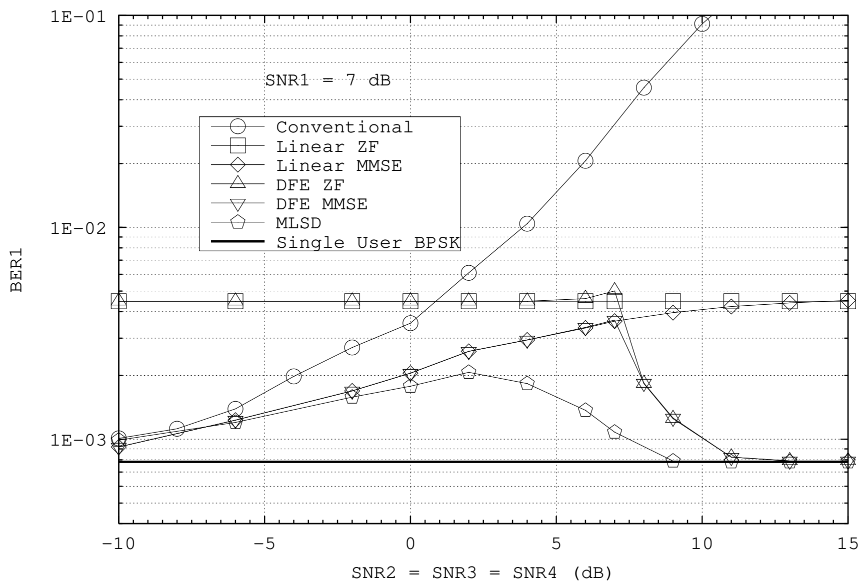


FIG. 2.12 – *Comparaison des performances de différents détecteurs pour l'utilisateur 1 ($K = 4$, $L = 7$).*

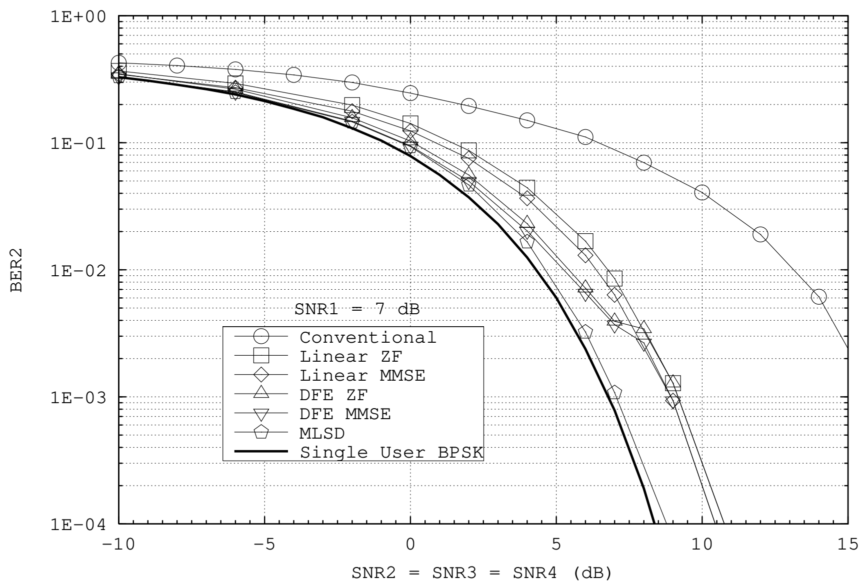


FIG. 2.13 – *Comparaison des performances de différents détecteurs pour l'utilisateur 2 ($K = 4$, $L = 7$).*

Le détecteur à retour de décision ZF est meilleur que le détecteur ZF linéaire à partir du moment où des décisions correctes sont réinjectées dans le filtre de rétroaction. Tant que l'utilisateur 1 a la plus forte puissance, il ne bénéficie pas de la suppression de l'interférence des autres utilisateurs dans le cas AWGN et synchrone. En effet, dans ce cas, $N_b = N_f = 1$. Dès que les autres utilisateurs deviennent plus puissants, ses performances s'améliorent et tendent vers celles du détecteur optimal.

Le détecteur à retour de décision MMSE réalise, comme le détecteur MMSE linéaire, un compromis entre le détecteur conventionnel et le détecteur à retour de décision ZF.

2.7 Détection par soustraction des interférences

Les techniques de détection par soustraction des interférences utilisent le même principe que la détection à retour de décision en évitant les inversions de matrices ou l'emploi de filtres matriciels. En effet, ces détecteurs vont se contenter d'évaluer l'interférence des utilisateurs les uns sur les autres pour l'éliminer au moment de la détection. On peut distinguer deux grands types de détecteurs par soustraction des interférences : le *détecteur par éliminations successives des interférences* (SIC) [36][47][48][75] et le *détecteur par éliminations parallèles des interférences* (PIC) [22][35][48][70][71]. Dans [45], la détection itérative est également considérée du point de vue de l'algorithme EM (*Expectation-Maximisation*) [69].

2.7.1 Le détecteur par éliminations successives des interférences

Le détecteur par éliminations successives des interférences (SIC pour *Successive Interference Canceller*) a une structure très proche de celle du DFE.

On désétale tout d'abord le signal reçu à l'aide du banc de filtres adaptés. On sélectionne la sortie correspondant au signal le plus puissant, on prend une décision sur cet utilisateur à l'aide d'un détecteur à seuil puis on réétale le signal détecté pour recréer le signal effectivement présent dans le signal reçu et l'en soustraire. On peut ensuite effectuer la détection des utilisateurs moins puissants dans les itérations ultérieures en utilisant le signal reçu débarrassé de l'interférence des utilisateurs déjà traités. La figure 2.14 représente une itération de la détection durant laquelle le symbole i de l'utilisateur k est détecté. On peut itérer plusieurs fois de manière à traiter tous les utilisateurs ou seulement certains. Dans ce dernier cas les utilisateurs restants sont détectés de manière conventionnelle.

On peut éliminer l'interférence de deux manières. L'*élimination dure* consiste à réétaler le symbole détecté $\hat{b}_k(i)$ et à le multiplier par la puissance ω_k de l'utilisateur k pour recréer l'interférence, tandis que l'*élimination souple* se contente de réétaler direc-

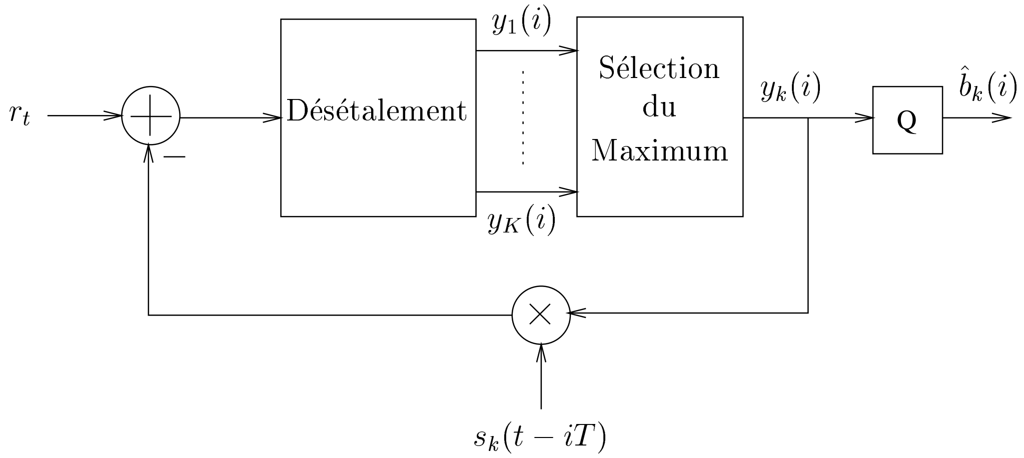


FIG. 2.14 – *Détecteur par éliminations successives souples des interférences (système synchrone).*

tement la sortie $y_k(i)$ du filtre adapté. L'élimination souple est plus facile à implanter car elle évite une estimation des puissances. Dans les situations délicates où tous les utilisateurs ont la même puissance elle est plus performante car elle ne retire pas exactement toute l'interférence causée par l'utilisateur k et minimise donc les effets d'une erreur d'estimation. En effet, dans l'élimination dure, on prend plus de risques car une erreur dans la décision ou dans l'estimation de l'amplitude peut entraîner une grave baisse des performances. Une erreur sur un symbole BPSK quadruple l'énergie de l'interférence due à l'utilisateur k au lieu de l'éliminer. La figure 2.14 représente une élimination souple pour des utilisateurs synchrones. Dans le cas d'un système asynchrone, il faut travailler sur un bloc de plusieurs symboles successifs et moyenner la puissance de chaque utilisateur sur ces symboles pour choisir celui que l'on va supprimer en priorité. L'élimination d'interférence s'effectue ensuite bloc par bloc.

Chaque étage de détection apporte un délai supplémentaire, il existe donc un compromis entre le nombre d'utilisateurs traités et le délai consenti. Pour réduire ce délai, on peut supprimer l'interférence de plusieurs utilisateurs d'un coup. On diminue ainsi la complexité de l'implantation sans nuire gravement aux performances.

2.7.2 Le détecteur par éliminations parallèles des interférences

Le détecteur par éliminations parallèles des interférences (PIC pour *Parallel Interference Canceller*) est plus compliqué à implanter que le SIC. En effet, il travaille en parallèle sur tous les utilisateurs au lieu de les traiter les uns après les autres comme fait le SIC. Pour chaque utilisateur, on va supprimer l'interférence totale due à tous les autres utilisateurs. La figure 2.15 représente la partie du premier étage du détecteur concernant la détection de l'utilisateur 1. Elle commence par un filtrage adapté du signal reçu. Chaque sortie des filtres adaptés $y_\ell(i)$ avec ℓ compris entre 2 et K, correspondant aux utilisateurs interférant avec l'utilisateur 1, est réétalée pour produire une estimation

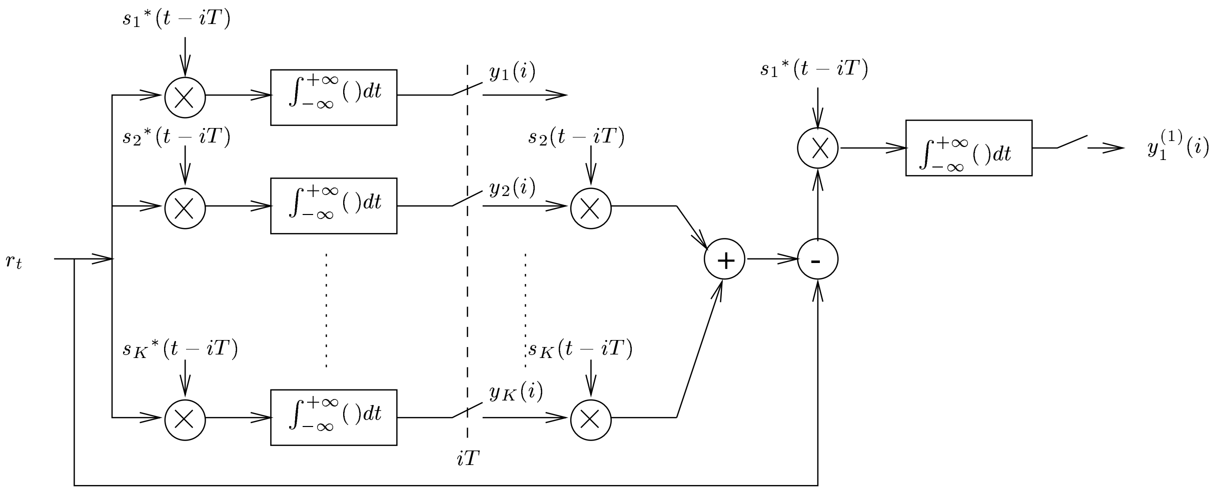


FIG. 2.15 – *Premier étage du détecteur par éliminations parallèles souples (système synchrone).*

souple de l'interférence totale subie par l'utilisateur 1. Cette interférence est soustraite au signal reçu et le signal épuré ainsi obtenu est désétalé par le filtre adapté à l'utilisateur 1. En reprenant l'opération pour les K utilisateurs, on obtient un nouvel ensemble d'observations $(y_1^{(1)}(i) \dots y_K^{(1)}(i))$ dans lesquelles l'interférence est sensée être moins présente. On peut ensuite rajouter d'autres étages, dans lesquels le filtrage adapté initial n'est plus nécessaire puisqu'on utilise maintenant les sorties de l'étage précédent. La qualité de ces sorties va croître avec le nombre d'étages et il suffira de faire une détection à seuil sur les sorties du dernier étage.

Le PIC est plus compliqué à implanter que le SIC. Il est plus performant que le SIC quand tous les utilisateurs ont même puissance. Par contre, quand l'éblouissement est fort, on aura tout intérêt à utiliser un SIC, qui convergera plus rapidement. De même que pour ce dernier, l'élimination dans le PIC peut être dure ou souple. La figure 2.15 représente une élimination souple dans un système synchrone. La sortie du premier étage pour l'utilisateur k à l'instant i s'écrit

$$\begin{aligned} y_k^{(1)}(i) &= \int_{iT}^{(i+1)T} s_k{}^*(t-iT)\Big(r_t - \sum_{\substack{\ell=1\\ \ell\neq k}}^{K} s_\ell(t-iT)y_\ell(i)\Big)\mathrm{d}t \\ &= y_k(i) - \sum_{\substack{\ell=1\\ \ell\neq k}}^{K} y_\ell(i)\int_0^T s_k{}^*(t)s_\ell(t)\mathrm{d}t \\ &= y_k(i) - \sum_{\substack{\ell=1\\ \ell\neq k}}^{K} R_{k\ell}(0)y_\ell(i) \end{aligned}$$

On peut également l'écrire sous forme matricielle :

$$\mathbf{y}^{(1)}(i) = \mathbf{I}_K\mathbf{y}(i) + \Big(\mathbf{I}_K - \mathbf{R}(0)\Big)\mathbf{y}(i)$$

où $\mathbf{y}^{(1)}(i) = (y_1^{(1)}(i)\ ,\ \ldots\ ,\ y_K^{(1)}(i))^T$ et $\mathbf{I}_K$ est la matrice unité de taille $K \times K$. Au deuxième étage, on utilise la sortie $\mathbf{y}^{(1)}(i)$ du premier étage :

$$\mathbf{y}^{(2)}(i) = \mathbf{I}_K\mathbf{y}(i) + \Big(\mathbf{I}_K - \mathbf{R}(0)\Big)\mathbf{y}^{(1)}(i) = \Big(\mathbf{I}_K + \mathbf{B} + \mathbf{B}^2\Big)\mathbf{y}(i)$$

où $\mathbf{B} = \mathbf{I}_K - \mathbf{R}(0)$. On peut montrer par récurrence qu'à l'itération n la sortie du détecteur est

$$\mathbf{y}^{(n)}(i) = \Big(\mathbf{I}_K + \sum_{\ell=1}^{n} \mathbf{B}^{\ell}\Big)\mathbf{y}(i)$$

Or

$$\mathbf{R}(0)^{-1} = \mathbf{I}_K + \sum_{\ell=1}^{n} \Big(\mathbf{I}_K - \mathbf{R}(0)\Big)^{\ell}$$

et donc, quand le nombre d'itérations n tend vers l'infini, on obtient en sortie du détecteur

$$\mathbf{y}^{(\infty)}(i) = \mathbf{R}(0)^{-1}\mathbf{y}(i)$$

qui est la sortie d'un décorrélateur. La démonstration peut facilement se généraliser à un système asynchrone. Le PIC employant l'élimination souple tend donc vers le décorrélateur quand le nombre d'itérations tend vers l'infini.

De nombreuses variantes des détecteurs SIC et PIC existent. On peut par exemple utiliser un décorrélateur au premier étage pour améliorer la qualité de la première décision [71] ou insérer un étage de décodage-recodage avant l'élimination de l'interférence [59][60].

Les deux méthodes de soustraction des interférences ont été testées sur le même système qu'aux sections précédentes. Les figures 2.16 et 2.17 présentent les performances d'un détecteur qui combine les deux méthodes SIC et PIC. En effet la première itération applique exactement la méthode SIC mais, comme aux itérations suivantes on dispose déjà d'une décision pour tous les utilisateurs, on peut supprimer l'interférence totale sur chaque utilisateur et appliquer ainsi la méthode PIC. L'élimination est ici dure tandis que sur les figures 2.18 et 2.19, les performances du même détecteur sont présentées avec une élimination souple. Les figures 2.20, 2.21, 2.22 et 2.23 présentent les résultats pour un détecteur PIC. On constate que l'élimination dure fonctionne moins bien dans les cas où les utilisateurs ont des puissances proches. Par contre elle fonctionne beaucoup mieux que l'élimination souple si l'interférence est très forte. On voit également que les détecteurs à élimination souple ont un comportement proche de celui du détecteur linéaire avec forçage à zéro quand le nombre d'itérations est grand. Le détecteur PIC fonctionne mieux quand tous les utilisateurs ont la même puissance tandis que le détecteur hybride SIC-PIC l'emporte si les puissances sont disparates. Dans un système employant un contrôle de puissance, où tous les utilisateurs ont des puissances proches, un détecteur PIC avec éliminations souples est plus indiqué.

Nous avons également testé un système asynchrone plus réaliste avec $K = 20$ utilisateurs et un facteur d'étalement $L = 40$ sur canal AWGN. La séquence d'étalement de chaque utilisateur est obtenue par l'addition modulo 2 d'une séquence de Kasami de longueur 960 et d'une séquence d'Hadamard de longueur 40 répétée 24 fois. La séquence

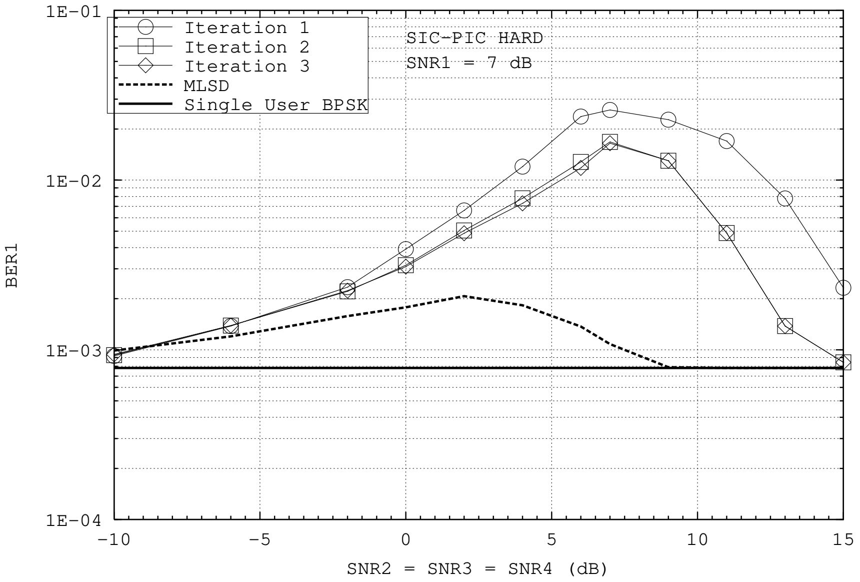


FIG. 2.16 – *Détecteur SIC-PIC avec éliminations dures pour l'utilisateur 1 ($K = 4$, $L = 7$).*

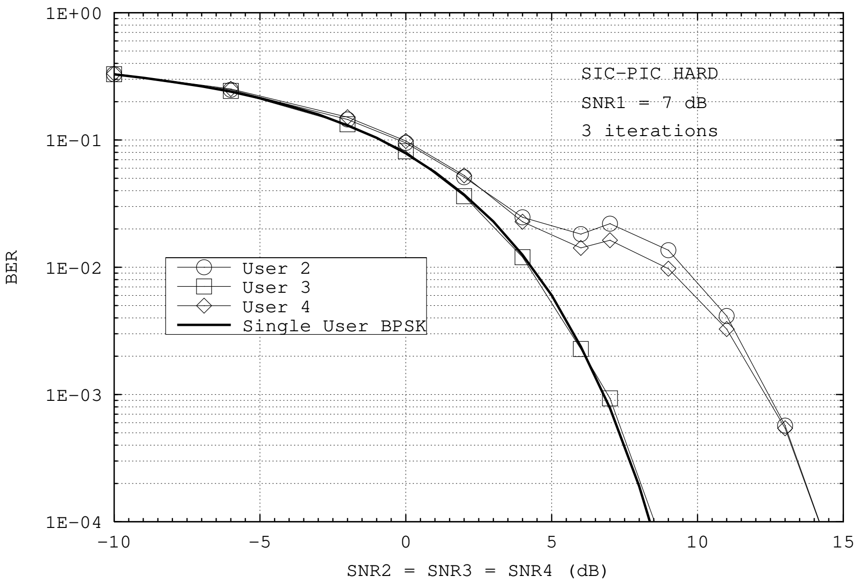


FIG. 2.17 – *Détecteur SIC-PIC avec éliminations dures pour les utilisateurs 2, 3, 4 ($K = 4$, $L = 7$).*

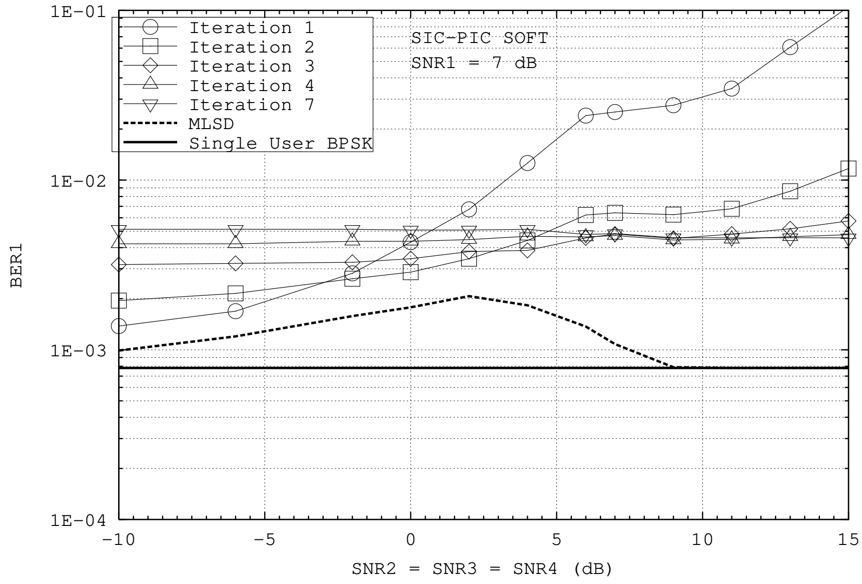


FIG. 2.18 – *Détecteur SIC-PIC avec éliminations souples pour l'utilisateur 1 ($K = 4$, $L = 7$).*

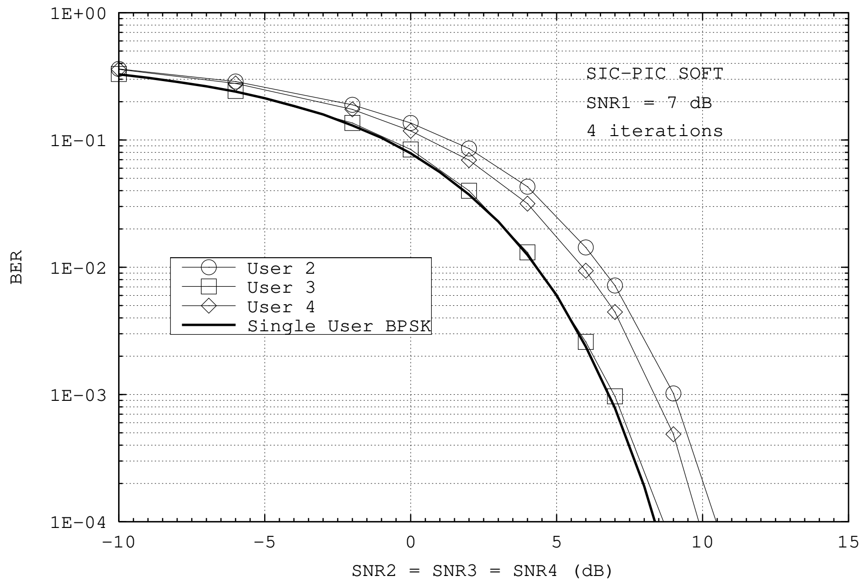


FIG. 2.19 – *Détecteur SIC-PIC avec éliminations souples pour les utilisateurs 2, 3, 4 ($K = 4$, $L = 7$).*

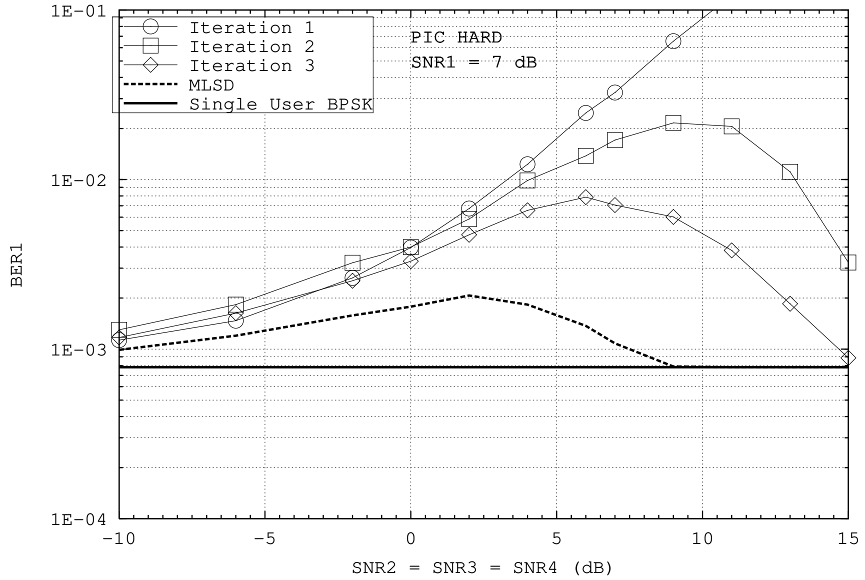


FIG. 2.20 – *Détecteur PIC avec éliminations dures pour l'utilisateur 1 ($K = 4$, $L = 7$).*

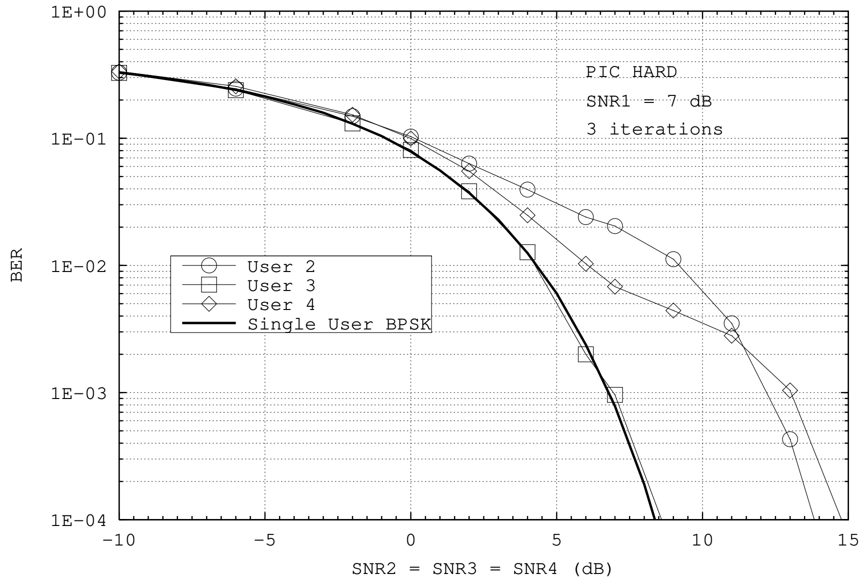


FIG. 2.21 – *Détecteur PIC avec éliminations dures pour les utilisateurs 2, 3, 4 ($K = 4$, $L = 7$).*

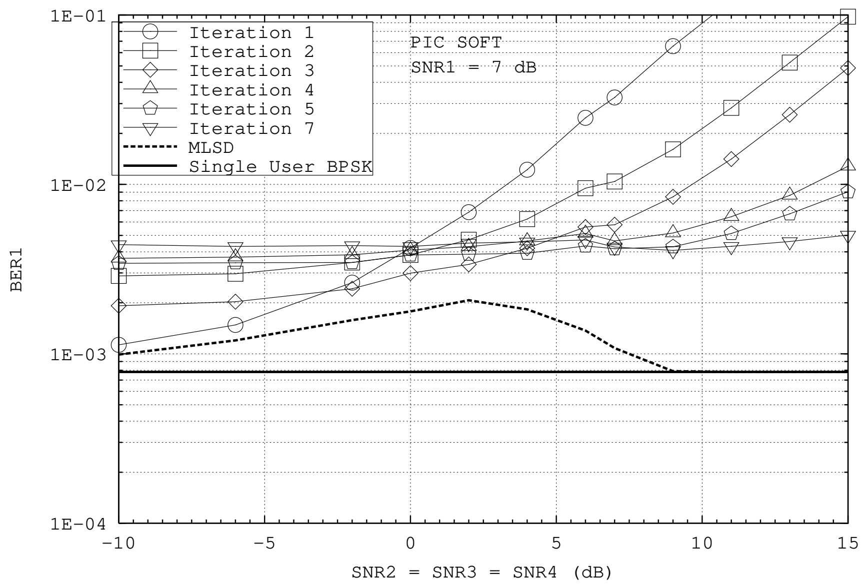


FIG. 2.22 – *Détecteur PIC avec éliminations souples pour l'utilisateur 1 ($K = 4$, $L = 7$).*

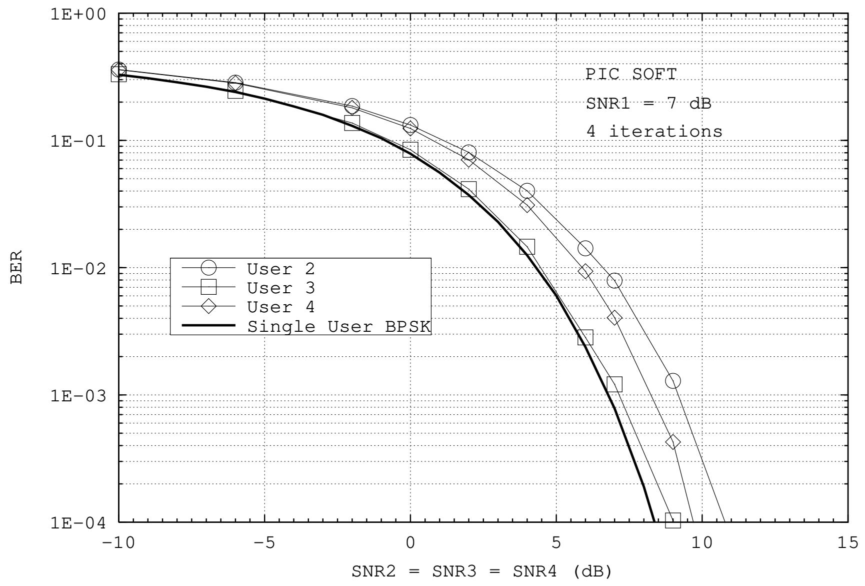


FIG. 2.23 – *Détecteur PIC avec éliminations souples pour les utilisateurs 2, 3, 4 ($K = 4$, $L = 7$).*

de Kasami est obtenue à partir d'une des 32 séquences de Kasami de longueur 1023 [33][21][52] tronquée. Une période de la séquence totale couvre 24 symboles des utilisateurs. Ceux-ci sont modulés par une QPSK, le premier ayant un rapport signal à bruit fixe égal à 7 dB et tous les autres ayant le même rapport signal à bruit. On peut voir sur les figures 2.24, 2.25 et 2.26 les performances de différents détecteurs multi-utilisateurs pour l'utilisateur 1, pour l'utilisateur qui présente les meilleures performances et pour l'utilisateur qui présente les plus mauvaises performances parmi les autres utilisateurs. Les performances des détecteurs itératifs sont présentées après la cinquième itération. Les mêmes remarques que pour le système avec $K = 4$ et $L = 7$ s'appliquent. Il faut noter le seuil dans les performances du détecteur hybride SIC-PIC à éliminations dures pour les faibles niveaux de bruit. L'interférence multi-utilisateurs est alors le seul facteur limitant et diminuer encore le niveau de bruit ne change plus rien aux performances. Le détecteur hybride SIC-PIC à éliminations souples se comporte mieux puisqu'il ne possède pas ce seuil, les valeurs souples retirées évoluant avec le niveau de bruit. Il faut aussi noter que l'on obtient des performances proches de celles de l'élimination dure idéale en estimant la puissance des utilisateurs par moyennage de la sortie des filtres adaptés sur plusieurs symboles (ici 24 symboles).

2.8 Conclusions

Nous nous sommes intéressés dans ce chapitre aux méthodes de détection multi-utilisateurs centralisées, qui supposent une connaissance parfaite des signatures des différents utilisateurs, de leurs retards relatifs et de leurs puissances. On sait estimer ces deux derniers paramètres et sur les canaux variant dans le temps, le détecteur doit être régulièrement mis à jour pour fournir une variable de décision de bonne qualité. Cependant cette mise à jour est coûteuse en capacité de calcul et peut être évitée grâce à des algorithmes adaptatifs, comme le détecteur linéaire MMSE adaptatif [40], qui calculent eux-mêmes, à partir d'une séquence d'apprentissage connue, les transformations qu'ils doivent appliquer au signal reçu. L'émission régulière de telles séquences d'apprentissage permet ensuite au détecteur de s'adapter aux variations du système. Il peut également utiliser, entre deux émissions de séquences d'apprentissage, ses propres décisions pour affiner son calcul. On dit alors qu'il est en mode *decision-directed*. Malheureusement, dans un système qui varie très vite, la qualité des données peut se dégrader soudainement et le mode *decision-directed* devient inefficace. Des algorithmes dits aveugles (*blind*), qui peuvent se passer des séquences d'apprentissage pour s'adapter au système, sont alors nécessaires. Certains algorithmes totalement décentralisés peuvent même ignorer les séquences des autres utilisateurs. Ils estiment alors à partir du signal reçu la loi de la variable aléatoire issue de la superposition de l'interférence entre les utilisateurs et du bruit blanc additif gaussien [10][11].

L'emploi de modulations à forte efficacité spectrale en CDMA reste marginal. La complexité du détecteur optimal, déjà prohibitive avec une modulation BPSK, est encore accrue par l'augmentation du nombre de symboles de la modulation. Si la complexité des

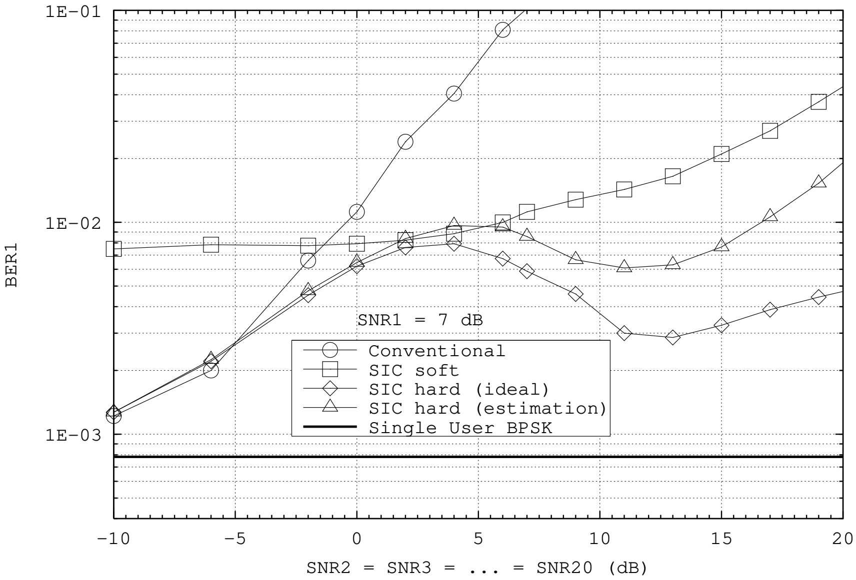


FIG. 2.24 – *Comparaison de différents détecteurs hybrides SIC-PIC pour l'utilisateur 1 ($K = 20,\ L = 40$).*

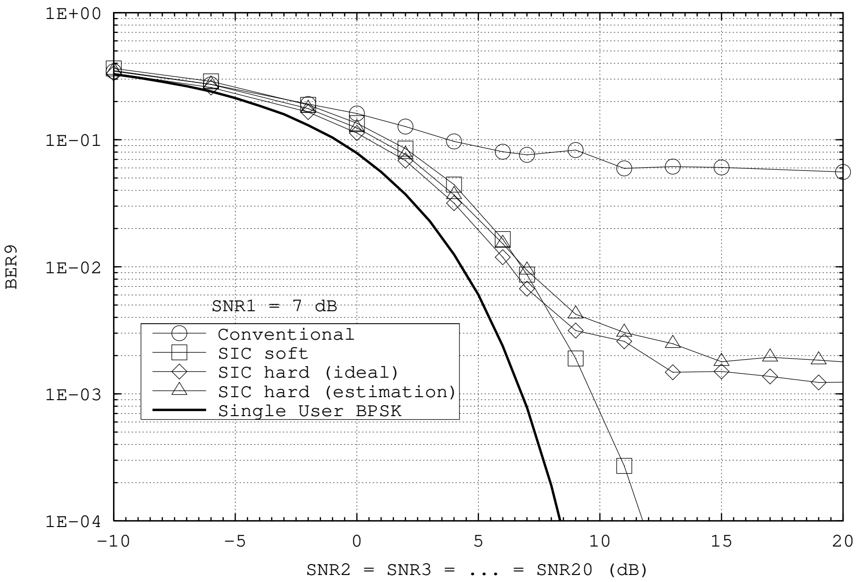


FIG. 2.25 – *Comparaison de différents détecteurs hybrides SIC-PIC pour l'utilisateur le meilleur ($K = 20,\ L = 40$).*

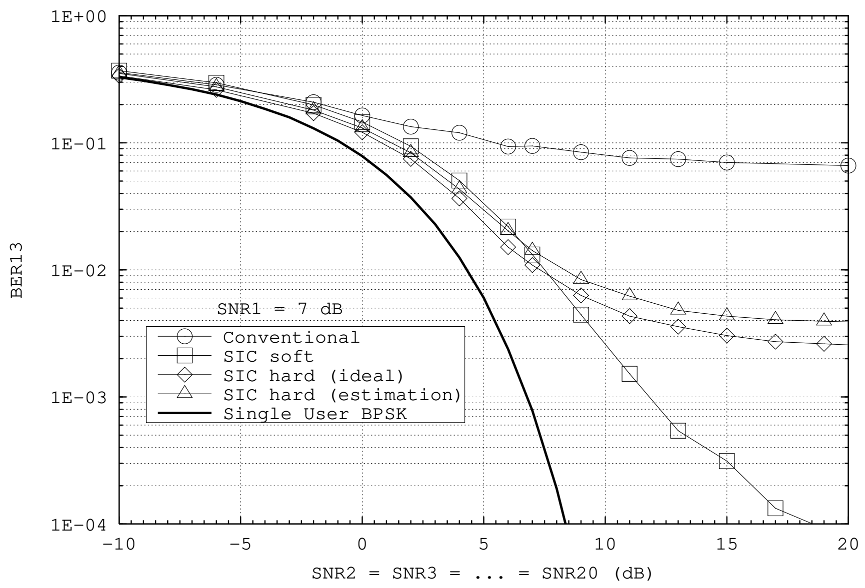


FIG. 2.26 – *Comparaison de différents détecteurs hybrides SIC-PIC pour l'utilisateur le pire ($K = 20$, $L = 40$).*

détecteurs par élimination des interférences ne varie pas avec la taille de la modulation employée, leur performances, par contre, risquent d'être fortement dégradées. Seules les méthodes linéaires et DFE semblent pouvoir s'adapter à l'emploi de modulations de grande taille. Nous proposons au chapitre 4 un détecteur dont la complexité ne varie pas avec la taille de la modulation et qui est employé au chapitre 5 avec une modulation MAQ-16 et des séquences d'étalement complexes.

Chapitre 3

Codes indépendants et décodage itératif*

3.1 Introduction

Différentes méthodes de détection multi-utilisateurs ont été présentées au chapitre précédent en l'absence de codage. Les méthodes par soustraction des interférences, présentées à la section 2.7, permettent de tirer facilement profit du codage pour améliorer les performances. Ainsi Sanada et Wang ont proposé d'utiliser la sortie dure du décodeur de canal pour reconstruire l'interférence d'un utilisateur et la soustraire de l'observation [59][60].

Nous proposons un algorithme de codage/décodage pour l'accès multiple basé sur un décodage itératif (turbo) applicable à la fois aux systèmes synchrones et asynchrones. Les deux idées principales de ce travail sont les suivantes :

- Les séquences pseudo-aléatoires quasi-orthogonales du système d'accès multiple par étalement de spectre à séquence directe sont remplacées par des codes de canal indépendants. Ces derniers sont construits en combinant des codes convolutifs et des entrelaceurs aléatoires. Les entrelaceurs détruisent la structure en treillis et donnent au signal un aspect aléatoire, remplissant ainsi le rôle des séquences pseudo-aléatoires.

- L'interférence entre les différents utilisateurs est reconstruite en utilisant l'information souple de la sortie d'un décodeur. Le décodage itératif utilise l'observation du canal et les probabilités a priori sur les bits codés des autres utilisateurs pour

∗ Le contenu de ce chapitre a été présenté au *Mediterranean Workshop on Coding and Information Integrity*, Ein-Boqeq, Israël, en Octobre 1997, et a paru dans les *Annales des Télécommunications*, tome 54, no. 7-8, pp. 401-410, en Juillet/Août 1999.

supprimer leur interférence et calculer la probabilité a posteriori sur les bits codés de l'utilisateur en cours de détection.

Le décodage conjoint itératif est effectué par l'algorithme aller-retour (*forward-backward*) [1]. Notons que le système multi-utilisateurs fondé sur des codes indépendants et une information de sortie souple présente de fortes similarités avec les systèmes mono-utilisateur employant des turbo-codes [4][2]. Les performances du système multi-utilisateurs décrit ici dépendent de la taille de l'entrelaceur et du taux et de la mémoire des codes de canal utilisés.

D'autres études ont été menées indépendamment sur le décodage itératif pour les systèmes d'accès multiple [66][32][43][54][15]. Dans [66] Tarköy limite son étude aux canaux non bruités et son décodeur itératif est radicalement différent de celui présenté dans ce chapitre. Ibrahim, dans [32], applique ce même algorithme au canal AWGN. Moher, dans [43], utilise des séquences pseudo-aléatoires (PN pour *Pseudo-Noise*) dans les signaux transmis et son décodage itératif est basé sur la minimisation de l'information mutuelle. Reed *et al.* utilisent aussi des séquences PN dans les signaux modulés dans [54] mais ils emploient une structure de décodage similaire à celle proposée dans ce chapitre. Enfin, dans [15], un système semblable à celui présenté ici est testé sur un canal synchrone. Damen et Belfiore y étudient également l'utilisation de codes en blocs avec le décodage souple des codes en blocs de Fossorier-Lin. Il faut souligner que la technique d'accès multiple décrite ici n'utilise aucune séquence PN et que l'efficacité spectrale du système, c'est-à-dire le nombre de bits par échantillon réel, est relativement élevée.

Le chapitre est organisé comme suit : la section 3.2 décrit la structure de l'émetteur tandis que la section 3.3 montre comment l'algorithme forward-backward peut être utilisé dans le récepteur du système multi-utilisateurs. Nous calculons ensuite la capacité multi-utilisateurs dans la section 3.4 afin d'évaluer l'efficacité du système. La section 3.5 présente les résultats de simulations de Monte Carlo sur un canal non bruité et montre qu'une probabilité d'erreur nulle peut être atteinte avec 4/3 bits d'information par symbole malgré les ambiguïtés. La section 3.6 montre les taux d'erreur sur canal AWGN pour un système avec 2 utilisateurs et un entrelaceur de taille infinie et la section 3.7 pour des systèmes avec 2 et 4 utilisateurs et des entrelaceurs de taille finie. En utilisant de simples codes convolutifs et pour une valeur de rapport signal à bruit située à 1.75 dB de la capacité multi-utilisateurs, les performances du système d'accès multiple sont pratiquement égales à celles d'un système mono-utilisateur employant les mêmes codes. Finalement la section 3.8 tente de donner une explication aux performances décevantes des différents codes pseudo-aléatoires dans le système et la section 3.9 regroupe quelques conclusions.

3.2 Description du codeur à accès multiple

La structure de l'émetteur utilisé est inspirée de celle des turbo-codes. Dans ces derniers, les bits d'information sont traités par différents codeurs concaténés en parallèle. Ces codeurs sont rendus indépendants par un entrelacement aléatoire de l'entrée de chaque niveau de concaténation. Notre système traite des données provenant de différents utilisateurs et associe à chaque utilisateur un codeur différent. Les bits d'information à l'entrée de chaque codeur sont différents, ce qui rend leur entrelacement totalement inutile. Par contre les signaux émis en sortie des différents codeurs sont sommés lors de la transmission sur le canal. De manière à *casser* la mémoire du code de chaque utilisateur, on entrelace sa sortie avant la sommation finale et on garantit ainsi une certaine orthogonalité entre les utilisateurs. L'émetteur multi-utilisateurs obtenu de cette façon est l'image par réflexion d'un turbo-code.

On note K le nombre d'utilisateurs dans le système d'accès multiple et N la taille des entrelaceurs utilisés. Tous les utilisateurs emploient des codes convolutifs identiques de taux k/n et de mémoire ν. Le treillis des codes convolutifs est régulièrement fermé de manière à diviser le flot des données en blocs de longueur N. Chaque codeur est initialisé à l'état zéro, lit $kN/n - k\nu$ bits d'information, ajoute $k\nu$ bits de fermeture pour revenir à l'état initial et génère au total N bits codés par bloc.
Soient $b_j(\ell)$ le $\ell^{\text{ième}}$ bit d'information de l'utilisateur j, $c_j(i)$ son $i^{\text{ème}}$ bit codé avant l'entrelacement et $\tilde{c}_j(i)$ son $i^{\text{ème}}$ bit codé après l'entrelacement, avec $0 \leq \ell < kN/n$, $0 \leq i < N$ et $1 \leq j \leq K$. Ces bits sont regroupés dans les vecteurs $\mathbf{b}_j$, $\mathbf{c}_j$ et $\tilde{\mathbf{c}}_j$, respectivement de taille kN/n, N et N. La permutation aléatoire utilisée pour l'entrelacement de l'utilisateur j est notée Π_j et on a donc la relation $\tilde{\mathbf{c}}_j = \Pi_j(\mathbf{c}_j)$. Les bits entrelacés $\tilde{\mathbf{c}}_j$ sont modulés par une modulation de phase binaire (MDP-2 ou BPSK pour *Binary Phase Shift Keying*) avant d'être transmis sur le canal de réponse impulsionnelle $h^j(t)$ et de fonction de transfert $H^j(f)$. Le signal reçu est $r_t = S_t + n_t$ où S_t est la somme des signaux de tous les utilisateurs et où $n(t)$ est un bruit additif.

$$S_t = \sum_{i=0}^{N-1} \sum_{j=1}^{K} \omega_j \ (2\tilde{c}_j(i) - 1) \ h^j(t - iT_c)$$

où ω_j est l'amplitude de l'utilisateur j et T_c la période d'un bit codé. Le délai de transmission τ_j de l'utilisateur j est compris dans la réponse impulsionnelle du canal $h^j(t)$. La figure 3.1 montre le diagramme complet de l'émetteur multi-utilisateurs.

Le codeur à accès multiple et le canal décrits ci-dessus correspondent au modèle de la liaison montante asynchrone d'un réseau radio-mobile, dans lequel les utilisateurs ont des amplitudes ω_j différentes et transmettent sur des canaux de réponses impulsionnelles différentes. On peut ainsi étudier les effets de l'éblouissement sur les performances. Pour des raisons de simplicité, on supposera dans la suite de ce chapitre que tous les utilisateurs transmettent sur le même canal idéal de réponse impulsionnelle rectangulaire de durée T_c, d'amplitude $1/\sqrt{T_c}$ et de retard τ_j, i.e. $h^j(t) = \text{rect}_{T_c}(t - \tau_j)/\sqrt{T_c}$. Le bruit additif est blanc et gaussien de densité spectrale de puissance N_0. La généralisation aux

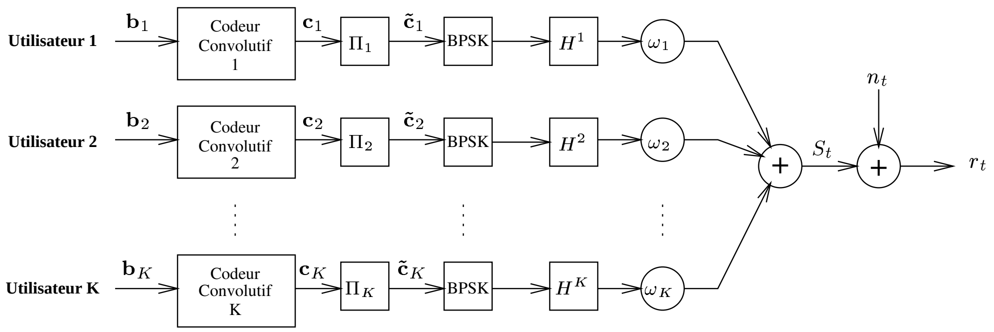


FIG. 3.1 – *Codeur à accès multiple.*

canaux multi-trajets à évanouissements ne présente pas de difficultés et est rapidement évoquée en conclusion de ce chapitre.

3.3 Description du décodeur

Tous les utilisateurs sont décodés conjointement afin de tenir compte des interférences mutuelles. L'utilisation de l'algorithme forward-backward [1], qui fournit une information souple sur les bits émis par chaque utilisateur, nous permet d'itérer le processus de décodage et d'approcher ainsi les performances du décodeur conjoint à maximum de vraisemblance (décodeur ML pour *Maximum Likelihood*). L'algorithme forward-backward est décrit dans l'annexe A.

3.3.1 Système multi-utilisateurs synchrone

On considère tout d'abord une transmission synchrone ($\tau_j = 0, \forall j$). Après filtrage adapté, tous les utilisateurs sont successivement décodés à chaque itération. Afin de rendre possible le décodage de l'utilisateur j, on doit désentrelacer la sortie du filtre adapté $\tilde{\mathbf{r}} = (\tilde{r}(0)\ , \tilde{r}(1)\ , \ \ldots\ , \tilde{r}(N-1))$ par l'entrelaceur réciproque Π_j^{-1}. On reconstruit ainsi le treillis du code de l'utilisateur j sans reconstruire ceux des autres utilisateurs. Le décodage d'un des utilisateurs est nommé *étape*. Une étape ne permet de déterminer que les probabilités a posteriori (APPs pour *A Posteriori Probabilities*) des bits de l'utilisateur j. Elles sont produites par l'algorithme forward-backward, qui utilise les probabilités a posteriori des autres utilisateurs calculées aux étapes précédentes pour prendre en compte l'interférence multi-utilisateurs dans le calcul de la métrique. Une itération complète englobe K étapes, une pour chaque utilisateur. La figure 3.2 représente l'étape s effectuant le décodage de l'utilisateur j à l'itération $iter$. La relation liant le numéro de l'étape au numéro de l'itération est

$$s = (iter - 1)K + j \quad \text{avec} \quad iter \geq 1 \quad \text{et} \quad j = 1, \ldots, K$$

Malgré la grande similarité entre l'émetteur multi-utilisateurs proposé et un turbo-code, les informations a priori utilisées pour itérer et les métriques calculées dans l'algorithme forward-backward sont très différentes. Dans les turbo-codes, les codes constituants sont systématiques et chaque étape utilise l'observation du canal et les probabilités a priori sur les bits systématiques. L'information a priori sur ces bits est l'information extrinsèque fournie par le décodeur à l'étape précédente. Les étapes successives de décodage des codes constituants du turbo-code mettent à jour les APPs des bits d'informations, qui sont communs à tous les codes constituants. L'information extrinsèque est donc une partie essentielle de l'APP, qui change d'une itération à l'autre. Les bits de parité des codes constituants sont émis successivement et n'interfèrent pas entre eux : les observations et les probabilités a priori sur les bits de parité d'un code constituant sont donc totalement inutiles dans le décodage d'un autre code constituant. Par contre, dans le système d'accès multiple présenté ici, les bits codés de tous les utilisateurs interfèrent les uns avec les autres. Leur information a priori est donc utile pour le décodage de l'utilisateur j car l'observation de la sortie du canal contient tous ces bits codés. D'autre part, les probabilités extrinsèques sur les bits d'information de l'utilisateur j n'existent pas parce que le décodage des autres utilisateurs ne fournit aucune information sur l'entrée du codeur de l'utilisateur j. En effet les bits d'information des différents utilisateurs sont entièrement indépendants.

Nous utilisons des codes convolutifs non récursifs non systématiques de taux k/n. On note $\mathbf{p}^u_{C,s}$ et $\mathbf{p}^u_{I,s}$ les vecteurs, respectivement de taille N et kN/n, contenant les APPs sur les bits codés et sur les bits d'information de l'utilisateur u à l'étape s. Les APPs de tous les utilisateurs sont considérées comme probabilités a priori mises à part celles de l'utilisateur j, qui est le seul utilisateur décodé à l'étape s. Les APPs de l'utilisateur j prendront les valeurs données par l'algorithme forward-backward après décodage. Les probabilités a priori de tous les utilisateurs à l'étape s sont les APPs fournies par l'étape $s-1$. Initialement, à l'étape 1, aucune information a priori n'est disponible et toutes les probabilités sont donc égales à 1/2. On obtient les relations suivantes :

$$\begin{aligned} \forall u \in \{1,K\} \ ; \ \forall i \in \{0,N-1\} \quad & p^u_{C,1}(i) = 1/2 \\ \forall s > 1 \ ; \ \forall u \neq j \quad & \mathbf{p}^u_{C,s} = \mathbf{p}^u_{C,s-1} \end{aligned}$$

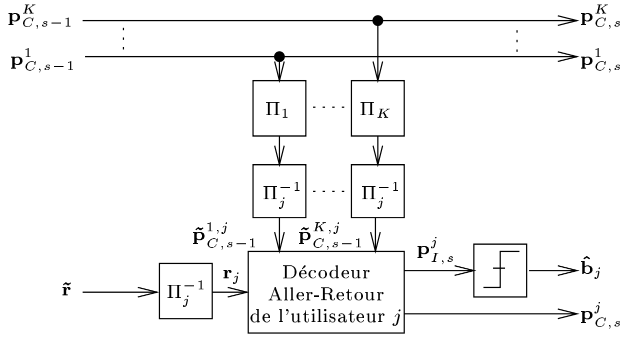


FIG. 3.2 – *Etape s effectuant le décodage de l'utilisateur j.*

A l'étape s et pour l'utilisateur j, on calcule les APPs sur les bits codés $\mathbf{p}_{C,s}^{j}$, qui seront utilisées comme informations a priori à l'étape suivante, ainsi que les APPs sur les bits d'information $\mathbf{p}_{I,s}^{j}$, qui permettent de prendre une décision dans l'itération courante. Le $i^{\text{ème}}$ bit entrelacé de l'utilisateur j est $\tilde{c}_j(i)$. L'observation du canal associée à $\tilde{c}_u(i)$ est $\tilde{r}(i)$ pour tous les utilisateurs u. Pour décoder l'utilisateur j, on doit reconstruire le treillis de son code en désentrelaçant la séquence $\tilde{\mathbf{r}}$ avec Π_j^{-1} pour obtenir

$$\mathbf{r}_j = \Pi_j^{-1}(\tilde{\mathbf{r}}) = (r_j(0)\ ,\ \ldots\ ,\ r_j(N-1))$$

Soit le vecteur $(c_j(\ell_1)\ ,\ c_j(\ell_2)\ ,\ \ldots\ ,\ c_j(\ell_n))$ représentant les n bits désentrelacés successifs correspondant à l'observation $(r_j(\ell_1)\ ,\ r_j(\ell_2),\ \ldots\ ,\ r_j(\ell_n))$ et associés à une branche du treillis du code liée aux bits d'information $\{b_j(\ell)\}$. Ces bits désentrelacés interfèrent avec les bits codés de l'utilisateur u via les permutations Π_u and Π_j^{-1}. La séquence binaire de l'utilisateur u interférant avec l'utilisateur j est

$$\tilde{\mathbf{c}}_u^j = \Pi_j^{-1}\Big[\Pi_u(\mathbf{c}_u)\Big] = \Pi_j^{-1}[\tilde{\mathbf{c}}_u] \quad \text{pour} \quad 1 \leq u \leq K \quad \text{et} \quad u \neq j$$

Le décodeur itératif tient compte de l'interférence des autres utilisateurs dans le calcul de la probabilité conditionnelle $p(r_j(\ell_i) \mid c_j(\ell_i))$. Soit $\tilde{\mathbf{a}}_j(\ell_i)$ le vecteur des $K-1$ bits pris à la position ℓ_i des $K-1$ vecteurs $\tilde{\mathbf{c}}_u^j$ avec $u \neq j$:

$$\tilde{\mathbf{a}}_j(\ell_i) = (\tilde{c}_1^j(\ell_i)\ ,\ \ldots\ ,\ \tilde{c}_{j-1}^j(\ell_i)\ ,\ \tilde{c}_{j+1}^j(\ell_i)\ ,\ \ldots\ ,\ \tilde{c}_K^j(\ell_i))$$

On note $\mathcal{A}$ l'ensemble des 2^{K-1} valeurs possibles de $\tilde{\mathbf{a}}_j(\ell_i)$ et on peut donc écrire

$$p\Big(r_j(\ell_i) \mid c_j(\ell_i)\Big) = \sum_{\tilde{\mathbf{a}}_j(\ell_i)\in\mathcal{A}} \Pr\Big\{r_j(\ell_i) \mid \tilde{\mathbf{a}}_j(\ell_i)\ ,\ c_j(\ell_i)\Big\} \times \Pr\Big\{\tilde{\mathbf{a}}_j(\ell_i) \mid c_j(\ell_i)\Big\} \tag{3.1}$$

Comme les séquences binaires sont toutes indépendantes,

$$\Pr\Big\{\tilde{\mathbf{a}}_j(\ell_i) \mid c_j(\ell_i)\Big\} = \prod_{u=1;u\neq j}^{K} \Pr\Big\{\tilde{c}_u^j(\ell_i)\Big\} = \prod_{u=1;u\neq j}^{K} \tilde{p}_{C,s-1}^{u,j}(\ell_i) \tag{3.2}$$

Les éléments du produit ci-dessus sont extraits de $\tilde{\mathbf{p}}_{C,s-1}^{u,j}$, le vecteur des probabilités a priori sur les bits $\tilde{\mathbf{c}}_u^j$ obtenu par entrelacement des probabilités a posteriori fournies par l'étape précédente :

$$\tilde{\mathbf{p}}_{C,s-1}^{u,j} = \Pi_j^{-1}\Big[\Pi_u(\mathbf{p}_{C,s-1}^{u})\Big]$$

Si le canal est gaussien avec une variance de bruit $\sigma^2 = N_0$,

$$\Pr\Big\{r_j(\ell_i) \mid \tilde{\mathbf{a}}_j(\ell_i)\ ,\ c_j(\ell_i)\Big\} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{\Big(r_j(\ell_i) - S_i^j\Big)^2}{2\sigma^2}\right\} \tag{3.3}$$

où

$$S_i^j = \omega_j\Big[2c_j\ (\ell_i) - 1\Big] + \sum_{u=1;u\neq j}^{K} \omega_u\Big[2\tilde{c}_u^j\ (\ell_i) - 1\Big]$$

Les probabilités a posteriori $\mathbf{p}^j_{C,s}$ et $\mathbf{p}^j_{I,s}$ sont calculées à l'aide de l'algorithme forward-backward avec une métrique sur chaque branche égale au produit de n probabilités conditionnelles définies par l'équation (3.1). On calcule donc les APPs pour l'utilisateur j à l'étape s et on peut itérer ce processus pour approcher les performances du décodeur conjoint ML.

3.3.2 Système multi-utilisateurs asynchrone

Considérons maintenant un canal asynchrone, dans lequel le délai τ_j de l'utilisateur j n'est pas nécessairement un multiple de la période d'un bit codé T_c. Sans perte de généralité, on peut supposer que

$$0 = \tau_1 \leq \tau_2 \leq \ldots \leq \tau_L < NT_c$$

Chaque symbole d'un utilisateur donné interfère avec un ou deux symboles de chacun des autres utilisateurs comme on peut le voir sur la figure 3.3. Le premier symbole (respectivement le dernier symbole) de l'utilisateur j dans un bloc donné de taille N interfère avec des symboles du bloc précédent (respectivement du bloc suivant). Afin de minimiser les effets de bord, le décodage itératif est appliqué sur un seul bloc de taille N si N est grand devant les délais de transmission, et il est appliqué sur plusieurs blocs successifs si N est petit. Cette approche bloc par bloc est celle employée habituellement dans les techniques d'élimination soustractive d'interférences avec des canaux asynchrones.

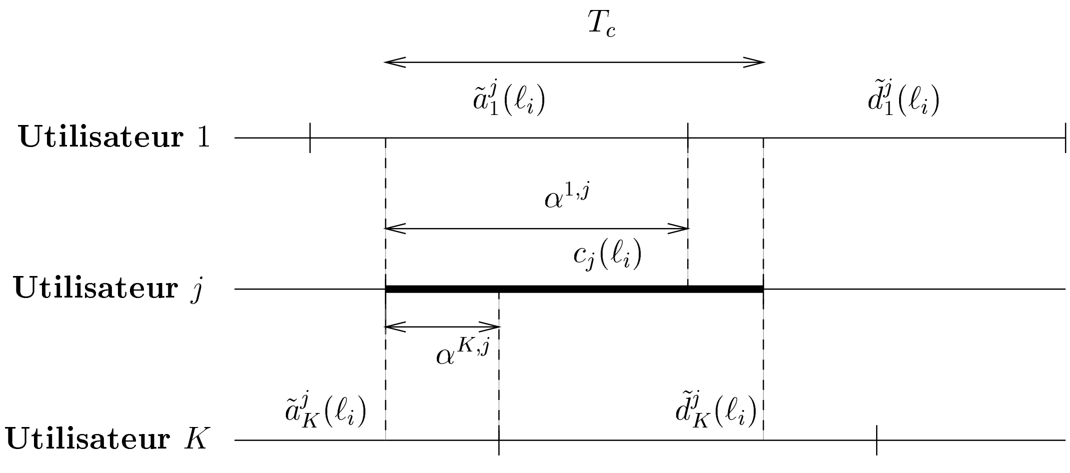


FIG. 3.3 – *Interférences sur l'utilisateur j dans un système à accès multiple asynchrone.*

La méthode de décodage est globalement la même que pour le canal synchrone. Les opérations d'entrelacement et de désentrelacement au niveau du décodeur doivent être synchronisées avec les opérations d'entrelacement effectuées en sortie des différents codeurs et donc présentes dans le signal reçu. Dans le calcul de la probabilité conditionnelle $p(r_j(\ell_i) \mid c_j(\ell_i))$, l'interférence de chaque utilisateur est maintenant constituée de deux bits. Soit $\tilde{\mathbf{a}}_j(\ell_i)$ (respectivement $\tilde{\mathbf{d}}_j(\ell_i)$) le vecteur de $K-1$ bits, chaque bit appartenant à un vecteur $\tilde{\mathbf{c}}_u^j$ différent, avec $u \neq j$, et interférant avec le commencement (resp. la fin) du symbole $c_j(\ell_i)$ comme illustré par la figure 3.3.

$$\begin{aligned}
\tilde{\mathbf{a}}_j(\ell_i) &= \left(\tilde{a}_1^j(\ell_i)\ ,\ \ldots\ ,\ \tilde{a}_{j-1}^j(\ell_i)\ ,\ \tilde{a}_{j+1}^j(\ell_i)\ ,\ \ldots\ ,\ \tilde{a}_K^j(\ell_i)\right) \\
\tilde{\mathbf{d}}_j(\ell_i) &= \left(\tilde{d}_1^j(\ell_i)\ ,\ \ldots\ ,\ \tilde{d}_{j-1}^j(\ell_i)\ ,\ \tilde{d}_{j+1}^j(\ell_i)\ ,\ \ldots\ ,\ \tilde{d}_K^j(\ell_i)\right)
\end{aligned}$$

Les équations (3.1) and (3.2) deviennent alors

$$p\Big(r_j(\ell_i) \mid c_j(\ell_i)\Big) = \sum_{\substack{\tilde{\mathbf{a}}_j(\ell_i)\in\mathcal{A} \\ \tilde{\mathbf{d}}_j(\ell_i)\in\mathcal{A}}} \Pr\Big\{r_j(\ell_i) \mid \tilde{\mathbf{a}}_j(\ell_i)\ ,\ \tilde{\mathbf{d}}_j(\ell_i)\ ,\ c_j(\ell_i)\Big\} \times \Pr\Big\{\tilde{\mathbf{a}}_j(\ell_i)\ ,\ \tilde{\mathbf{d}}_j(\ell_i) \mid c_j(\ell_i)\Big\} \tag{3.4}$$

$$\Pr\Big\{\tilde{\mathbf{a}}_j(\ell_i)\ ,\ \tilde{\mathbf{d}}_j(\ell_i) \mid c_j(\ell_i)\Big\} = \prod_{u=1;u\neq j}^{K} \Pr\Big\{\tilde{a}_u^j(\ell_i)\Big\}\Pr\Big\{\tilde{d}_u^j(\ell_i)\Big\}$$

Cette dernière égalité est possible grâce aux entrelaceurs, qui rendent indépendants deux bits interférents consécutifs d'un même utilisateur. La probabilité conditionnelle de l'équation (3.3) peut maintenant s'écrire

$$\Pr\Big\{r_j(\ell_i) \mid \tilde{\mathbf{a}}_j(\ell_i)\ ,\ \tilde{\mathbf{d}}_j(\ell_i)\ ,\ c_j(\ell_i)\Big\} = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{\Big(r_j(\ell_i) - S_i^j\Big)^2}{2\sigma^2}\right\} \tag{3.5}$$

où

$$S_i^j = \omega_j\Big[2c_j\left(\ell_i\right) - 1\Big] + \sum_{u=1;u\neq j}^{K} \omega_u\left\{\frac{\alpha^{u,j}}{T_c}\Big[2\tilde{a}_u^j\left(\ell_i\right) - 1\Big] + \frac{T_c - \alpha^{u,j}}{T_c}\Big[2\tilde{d}_u^j\left(\ell_i\right) - 1\Big]\right\} \tag{3.6}$$

$\alpha^{u,j}/T_c$ (respectivement $(T_c - \alpha^{u,j})/T_c$) représente la fraction de la période d'un bit codé durant laquelle le bit $\tilde{a}_u^j\left(\ell_i\right)$ (respectivement $\tilde{d}_u^j\left(\ell_i\right)$) de l'utilisateur u interfère avec le symbole $c_j\left(\ell_i\right)$.

$$\begin{aligned}
\alpha^{u,j} &= T_c - (\tau_j - \tau_u) \text{ modulo } T_c \qquad \text{si} \quad u < j \\
\alpha^{u,j} &= (\tau_u - \tau_j) \text{ modulo } T_c \qquad \text{sinon}
\end{aligned}$$

Evidemment, l'expression (3.6) n'est valable que si la réponse impulsionnelle $h^j(t)$ est rectangulaire. Pour d'autres types de mise en forme spectrale, les coefficients d'inter-corrélation doivent être calculés en intégrant le produit des deux réponses impulsionnelles se superposant respectivement pendant les durées $\alpha^{u,j}$ et $(T_c - \alpha^{u,j})$.

3.3.3 Réduction de la complexité

La complexité des deux décodeurs conjoints synchrone et asynchrone décrits par les équations (3.1) et (3.4) dépend exponentiellement du nombre des utilisateurs. Le nombre de sommations à chaque transition de l'algorithme forward-backward est 2^{K-1} dans le système synchrone et $2^{2(K-1)}$ dans le système asynchrone. Cette forte complexité est le point faible du système et il serait bienvenu de la réduire.

Application des algorithmes Max-Log-MAP et Log-MAP

Une première méthode de réduction de la complexité consiste à éviter le calcul de la fonction exponentielle dans les équations (3.3) et (3.5). Cette simplification est rendue possible par l'approximation *Max-Log-MAP*: $\log(\sum e^{x_i}) \approx \max(x_i)$ [57][58]. On choisit de se restreindre au cas synchrone, la généralisation au cas asynchrone étant évidente. On note LLR (*Log-Likelihood Ratio*) le rapport de vraisemblance logarithmique défini comme suit pour une variable aléatoire X pouvant prendre deux valeurs -1 ou 1 :

$$\mathrm{LLR}(X) = \ln\left(\frac{\Pr(X=1)}{\Pr(X=-1)}\right)$$

L'algorithme Max-Log-MAP est décrit en détails dans l'annexe B. Il faut calculer le logarithme de la métrique et on doit donc transformer (3.1) en

$$\ln\Big(p\big(r_j(\ell_i) \mid c_j(\ell_i)\big)\Big) = \ln\left(\sum_{\tilde{\mathbf{a}}_j(\ell_i)\in\mathcal{A}} \Pr\Big\{r_j(\ell_i) \mid \tilde{\mathbf{a}}_j(\ell_i)\ ,\ c_j(\ell_i)\Big\} \times \Pr\Big\{\tilde{\mathbf{a}}_j(\ell_i) \mid c_j(\ell_i)\Big\}\right)$$

où

$$\Pr\Big\{r_j(\ell_i) \mid \tilde{\mathbf{a}}_j(\ell_i)\ ,\ c_j(\ell_i)\Big\} = \frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{\left(r_j(\ell_i)-S_i^j\right)^2}{2\sigma^2}\right\} = \frac{1}{\sqrt{2\pi}\sigma}e^{-\delta^2\left(\tilde{\mathbf{a}}_j(\ell_i)\right)}$$

On obtient donc

$$\begin{aligned}\ln\Big(p\big(r_j(\ell_i) \mid c_j(\ell_i)\big)\Big) &= \ln\left(\sum_{\tilde{\mathbf{a}}_j(\ell_i)\in\mathcal{A}} \exp\Big[-\delta^2(\tilde{\mathbf{a}}_j(\ell_i)) + \ln\Big(\Pr\{\tilde{\mathbf{a}}_j(\ell_i) \mid c_j(\ell_i)\}\Big)\Big]\right)\\ &\approx \max_{\tilde{\mathbf{a}}_j(\ell_i)\in\mathcal{A}}\Big[-\delta^2(\tilde{\mathbf{a}}_j(\ell_i)) + \ln\Big(\Pr\{\tilde{\mathbf{a}}_j(\ell_i) \mid c_j(\ell_i)\}\Big)\Big]\end{aligned}$$

Le facteur $\frac{1}{\sqrt{2\pi}\sigma}$ a été volontairement omis dans les calculs car il se simplifie dans le calcul des LLRs. D'après (3.2)

$$\ln\Big(\Pr\{\tilde{\mathbf{a}}_j(\ell_i) \mid c_j(\ell_i)\}\Big) = \sum_{u=1;u\neq j}^{K} \ln\Big(\Pr\Big\{\tilde{c}_u^j(\ell_i)\Big\}\Big) \tag{3.7}$$

Comme les itérations précédentes ne fournissent plus des APPs mais des LLRs, il vaut mieux exprimer (3.7) en fonction de $\mathrm{LLR}(\tilde{c}_u^j(\ell_i))$. On peut montrer que

$$\begin{aligned}\ln\Pr\Big\{\tilde{c}_u^j(\ell_i)=1\Big\} &= \mathrm{LLR}\Big(\tilde{c}_u^j(\ell_i)\Big) - \ln\Big(1+\exp\Big\{\mathrm{LLR}\Big(\tilde{c}_u^j(\ell_i)\Big)\Big\}\Big)\\ &\approx \mathrm{LLR}\Big(\tilde{c}_u^j(\ell_i)\Big) - \max\Big\{0, \mathrm{LLR}\Big(\tilde{c}_u^j(\ell_i)\Big)\Big\}\end{aligned}$$

$$\ln \Pr\Big\{\tilde{c}_u^j(\ell_i) = 1\Big\} \approx \min\Big\{0,\mathrm{LLR}\Big(\tilde{c}_u^j(\ell_i)\Big)\Big\} \tag{3.8}$$

De même

$$\ln \Pr\Big\{\tilde{c}_u^j(\ell_i) = -1\Big\} \approx -\max\Big\{0,\mathrm{LLR}\Big(\tilde{c}_u^j(\ell_i)\Big)\Big\} \tag{3.9}$$

On peut donc déduire des formules (3.7), (3.8) et (3.9) la valeur de $\ln \Pr\{\tilde{\mathbf{a}}_j(\ell_i) \mid c_j(\ell_i)\}$, qui est le résultat d'une somme dépendant des valeurs contenues dans le vecteur $\tilde{\mathbf{a}}_j(\ell_i)$.

L'algorithme Max-Log-MAP est sous-optimal. Pour tenter d'améliorer les résultats, on peut utiliser l'algorithme Log-MAP, qui reproduit exactement l'algorithme forward-backward dans le domaine logarithmique. On peut voir dans l'annexe C que l'approximation (B.6) peut être parfaitement corrigée par une fonction de correction f_c. En remplaçant toutes les approximations ci-dessus par la valeur exacte ainsi obtenue, on remplace l'algorithme Max-Log-MAP par l'algorithme Log-MAP. La complexité de celui-ci peut être réduite en discrétisant la fonction f_c en un nombre fini de valeurs pré-calculées (cf. annexe C).

Simplification de la métrique

L'algorithme Max-Log-MAP, bien que nous évitant l'emploi de la fonction exponentielle, nécessite encore l'évaluation de 2^{K-1} métriques euclidiennes par transition dans le treillis. Le nombre de ces évaluations peut être considérablement réduit si l'on ne garde que les K métriques correspondant aux combinaisons de symboles interférents ayant les probabilités a priori les plus fortes. On dispose en effet des $K-1$ probabilités a priori $\tilde{p}_{C,s-1}^{u,j}(\ell_i)$ pour $u \neq j$. La combinaison la plus probable a priori est celle regroupant les valeurs de $\tilde{c}_u^j(\ell_i)$ (1 ou -1) maximisant $\tilde{p}_{C,s-1}^{u,j}(\ell_i)$. Les $K-1$ combinaisons restantes sont obtenues en changeant la valeur d'un des symboles interférents. On obtient ainsi les K combinaisons les plus probables et on peut restreindre la sommation de l'équation (3.1) à ces K termes et considérer ainsi que les autres termes sont négligeables car la valeur de leur probabilité a priori est très faible.

Cette métrique simplifiée peut être intégrée à l'algorithme forward-backward classique mais on peut aussi l'employer avec l'algorithme Max-Log-MAP. $\ln \Pr\{\tilde{\mathbf{a}}_j(\ell_i) \mid c_j(\ell_i)\}$ s'exprime alors très simplement. En effet, considérons tout d'abord la combinaison la plus probable :

- Si $\mathrm{LLR}\Big(\tilde{c}_u^j(\ell_i)\Big) \leq 0$ alors on choisit $\tilde{c}_u^j(\ell_i) = -1$ et

$$\ln \Pr\Big\{\tilde{c}_u^j(\ell_i) = -1\Big\} \approx -\max\Big\{0,\mathrm{LLR}\Big(\tilde{c}_u^j(\ell_i)\Big)\Big\} = 0$$

- Si $\mathrm{LLR}\Big(\tilde{c}_u^j(\ell_i)\Big) > 0$ alors on choisit $\tilde{c}_u^j(\ell_i) = 1$ et

$$\ln \Pr\Big\{\tilde{c}_u^j(\ell_i) = 1\Big\} \approx \min\Big\{0,\mathrm{LLR}\Big(\tilde{c}_u^j(\ell_i)\Big)\Big\} = 0$$

Dans tous les cas, on n'obtient, pour la configuration maximale, que des valeurs nulles et donc

$$\ln \Pr\left\{\tilde{\mathbf{a}}_j(\ell_i) \mid c_j(\ell_i)\right\} = 0$$

Pour les $K-1$ autres configurations, on change un des termes $\tilde{c}^j_m(\ell_i)$.

- Si dans la configuration maximale $\tilde{c}^j_m(\ell_i)$ était égal à -1 $\Big(\mathrm{LLR}\big(\tilde{c}^j_m(\ell_i)\big) \leq 0\Big)$, dans la nouvelle configuration, $\tilde{c}^j_m(\ell_i) = 1$. On obtient donc

$$\ln \Pr\Big\{\tilde{c}^j_m(\ell_i) = 1\Big\} \approx \min\Big\{0{,}\mathrm{LLR}\big(\tilde{c}^j_m(\ell_i)\big)\Big\} = \mathrm{LLR}\big(\tilde{c}^j_m(\ell_i)\big)$$

$$\ln \Pr\Big\{\tilde{c}^j_m(\ell_i) = 1\Big\} \approx -\Big|\mathrm{LLR}\big(\tilde{c}^j_m(\ell_i)\big)\Big|$$

- Si dans la configuration maximale $\tilde{c}^j_m(\ell_i)$ était égal à 1 $\Big(\mathrm{LLR}\big(\tilde{c}^j_m(\ell_i)\big) > 0\Big)$, dans la nouvelle configuration, $\tilde{c}^j_m(\ell_i) = -1$. On obtient donc

$$\ln \Pr\Big\{\tilde{c}^j_m(\ell_i) = -1\Big\} \approx -\max\Big\{0{,}\mathrm{LLR}\big(\tilde{c}^j_m(\ell_i)\big)\Big\} = -\mathrm{LLR}\big(\tilde{c}^j_m(\ell_i)\big)$$

$$\ln \Pr\Big\{\tilde{c}^j_m(\ell_i) = -1\Big\} \approx -\Big|\mathrm{LLR}\big(\tilde{c}^j_m(\ell_i)\big)\Big|$$

Donc, dans tous les cas, si on modifie la valeur de $\tilde{c}^j_m(\ell_i)$ dans la combinaison de probabilité a priori maximale, on obtient une combinaison de probabilité a priori telle que

$$\ln \Pr\left\{\tilde{\mathbf{a}}_j(\ell_i) \mid c_j(\ell_i)\right\} \approx -\Big|\mathrm{LLR}\big(\tilde{c}^j_m(\ell_i)\big)\Big| \tag{3.10}$$

Dans le cas où on corrige l'approximation par la fonction f_c, le logarithme de la probabilité a priori de la combinaison la plus probable n'est plus nul. Il faut donc ajouter sa valeur à celle de l'approximation (3.10) pour obtenir la valeur exacte des logarithmes des probabilités a priori des $K-1$ autres combinaisons considérées.

3.4 La capacité du système d'accès multiple

De manière à évaluer l'efficacité de notre codage pour l'accès multiple, nous calculons la capacité du canal gaussien quand plusieurs signaux de modulation BPSK superposés sont utilisés.

Etudions le cas simple de deux modulations BPSK ($K = 2$). Si les deux signaux ont des énergies différentes, l'alphabet $\mathcal{M}$ résultant de leur superposition, c'est-à-dire de leur sommation réelle, contient 4 symboles :

$$\mathcal{M} = \Big\{\ -\omega_1 - \omega_2\ ,\ -\omega_1 + \omega_2\ ,\ \omega_1 - \omega_2\ ,\ \omega_1 + \omega_2\ \Big\}$$

Par contre, si les deux signaux ont la même énergie, $-\omega_1+\omega_2 = \omega_1-\omega_2 = 0$ et l'alphabet $\mathcal{M}$ ne contient plus que 3 symboles distincts :

$$\mathcal{M} = \left\{ \ -2\omega_1 \ , \ 0 \ , \ 2\omega_1 \ \right\}$$

Dans ce cas le symbole '0' est ambigu puisqu'il peut résulter de l'émission d'un '+1' par l'utilisateur 1 et d'un '−1' par l'utilisateur 2 ou de l'émission d'un '−1' par l'utilisateur 1 et d'un '+1' par l'utilisateur 2. L'amplitude ω_i est définie comme

$$\omega_i = \sqrt{2E_{s_i}}$$

où E_{s_i} est l'énergie moyenne par bit codé ou par symbole pour un utilisateur (le facteur 2 vient de la représentation sur fréquence porteuse). Soit $M = |\mathcal{M}|$ le nombre de symboles dans l'alphabet $\mathcal{M}$. Les symboles sont nommés s_i pour $i \in \{0,\ldots,M-1\}$. On note $\Pr\{s_i\}$ la probabilité a priori du symbole s_i. Pour deux modulations de puissances différentes, les quatre symboles ont la même probabilité a priori :

$$\Pr\{s_i\} = \frac{1}{4}, \quad \forall i = 0,\ldots,3$$

Pour deux modulations de même puissance , le symbole '0' correspond à deux paires de symboles émis possibles, il a donc une probabilité a priori qui est le double de celles des symboles '$-2\omega_1$' et '$2\omega_1$', qui ne correspondent qu'à une seule paire de symboles émis possible chacun :

$$\Pr\{-2\omega_1\} = \frac{1}{4} \ ; \ \Pr\{0\} = \frac{1}{2} \ ; \ \Pr\{2\omega_1\} = \frac{1}{4}$$

La capacité C en bits d'information par symbole du canal à bruit additif blanc gaussien (AWGN) quand on transmet des symboles s_i pris dans un alphabet de taille M et de probabilité a priori $\Pr\{s_i\}$, est [27]

$$C = \sum_{i=0}^{M-1} \Pr\{s_i\} \int_{-\infty}^{+\infty} p(x|s_i) \log_2 \left\{ \frac{p(x|s_i)}{\sum_{j=0}^{M-1} \Pr\{s_j\} p(x|s_j)} \right\} \mathrm{d}x \tag{3.11}$$

où $p(x|s_i)$ est la vraisemblance de l'obtention de x en sortie du canal sachant que l'on a émis le symbole s_i. Pour des modulations (et donc des utilisateurs) ayant des puissances différentes, en utilisant

$$\Pr\{s_i\} = \frac{1}{M} \quad \forall i = 0,\ldots,M-1$$

on obtient [68]

$$C = \log_2(M) - \frac{1}{M} \sum_{i=0}^{M-1} \int_{-\infty}^{+\infty} \frac{\mathrm{e}^{-x^2/2\sigma^2}}{\sqrt{2\pi}\sigma} \times \log_2 \left[\sum_{j=0}^{M-1} \exp\left(-\frac{[s_i + x - s_j]^2 - x^2}{2\sigma^2} \right) \right] \mathrm{d}x \tag{3.12}$$

Pour des utilisateurs de même puissance, l'équation (3.11) ne peut pas être simplifiée si on l'écrit seulement avec 3 symboles. Cependant, si on considère que l'alphabet contient 4 symboles avec 2 symboles égaux de probabilité 1/4, on peut appliquer la formule (3.12).

Soient E_b l'énergie moyenne par bit d'information et E_s l'énergie totale par symbole.

$$\begin{aligned} E_s &= E_{s_1} + E_{s_2} \\ E_b &= \frac{R_1 E_{b_1} + R_2 E_{b_2}}{R_1 + R_2} \end{aligned}$$

où R_i et E_{b_i} sont le taux de codage et l'énergie moyenne par bit d'information pour l'utilisateur i. On veut calculer la valeur minimale en décibels du rapport signal à bruit $SNR = E_b/N_0$ qui permet d'atteindre une capacité donnée et on suppose donc évidemment que le taux total R est égal à la capacité :

$$C = R = R_1 + R_2$$

La variance du bruit $\sigma^2 = N_0$ peut être exprimée de la manière suivante :

$$\sigma^2 = 10^{-\frac{SNR}{10}} \times E_b = 10^{-\frac{SNR}{10}} \times \frac{E_s}{C} \tag{3.13}$$

En substituant (3.13) dans (3.11) ou (3.12) et en utilisant une méthode numérique du point fixe avec une valeur initiale $C_0 = 1$, on obtient la capacité C en fonction du rapport signal à bruit SNR. Asymptotiquement, quand SNR est suffisamment grand, la capacité globale du pire cas correspondant à deux utilisateurs de même puissance tend vers la valeur de 1.5 bits d'information par symbole. Cette valeur est évidemment plus faible que la capacité correspondant à trois symboles équiprobables ($\log_2(3) = 1.58$ bits d'information par symbole). Pour des utilisateurs ayant des puissances différentes par symbole, la capacité tend toujours vers la valeur $\log_2(4) = 2$ et le meilleur cas, c'est-à-dire celui pour lequel on tend le plus vite vers cette valeur, correspond à une différence de puissances $\Delta E_s(\text{dB}) = 10\log_{10}(E_{s_2}/E_{s_1}) = 6$ dB. Dans cette situation particulière, la modulation BPSK de l'utilisateur 2 a deux fois l'amplitude de celle de l'utilisateur 1. On obtient un alphabet $\mathcal{M} = \{-3\omega_1, -\omega_1, \omega_1, 3\omega_1\}$, qui est le plus régulier (il correspond à l'alphabet d'une modulation d'amplitude à 4 symboles) et est donc celui qui transporte le plus d'information. Si la différence de puissances ΔE_s est trop forte, la capacité chute pour les faibles valeurs de SNR.

La figure 3.4 montre les limites de la capacité multi-utilisateurs pour deux modulations BPSK ayant la même énergie par bit d'information ($\Delta E_b = 0$) sur le canal gaussien. Le taux d'information de chaque modulation BPSK doit être plus faible que la capacité du canal gaussien sur lequel on transmet une unique modulation BPSK. De même la somme des deux taux doit être plus faible que la capacité du canal si on transmet les symboles de l'alphabet $\mathcal{M}$ obtenu après superposition des deux modulations BPSK.

La figure 3.5 représente les valeurs minimales de $SNR_1 = E_{b_1}/N_0$ requises pour atteindre une probabilité d'erreur nulle en fonction du nombre de bits par symbole pour chaque utilisateur. On a supposé ici que les deux utilisateurs ont la même efficacité spectrale ($R_1 = R_2$ et $\Delta E_s(\text{dB}) = \Delta E_b(\text{dB})$). Pour une efficacité spectrale donnée, si

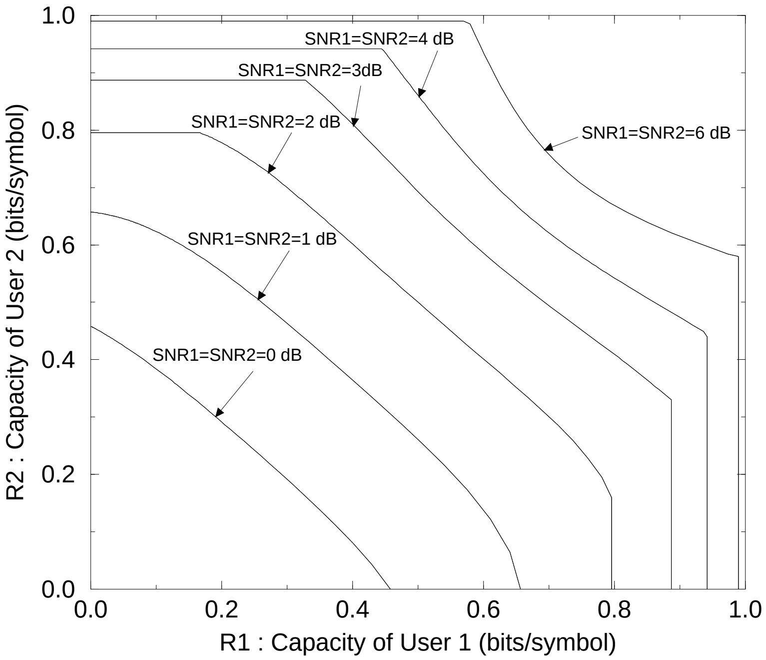


FIG. 3.4 – *Diagramme de capacité avec $\Delta E_b = 0$ dB ($K = 2$, canal AWGN).*

SNR_1 est plus faible que la valeur minimale donnée par la courbe, le système fonctionne en dehors de la zone de capacité et on ne peut pas atteindre une probabilité d'erreur nulle. Pour de faibles rapports signal à bruit et pour les valeurs de ΔE_s strictement positives, toutes les courbes sont confondues car la modulation BPSK seule sur le canal est la modulation limitante.

On s'intéresse maintenant au cas où les deux utilisateurs traversent deux canaux de Rayleigh différents. Les deux utilisateurs, qui empruntent deux canaux différents, sont généralement asynchrones mais nous faisons le calcul de la capacité dans le cas synchrone, qui est le pire cas. On note α_1 l'évanouissement sur l'utilisateur 1 et α_2 l'évanouissement sur l'utilisateur 2. Ces deux évanouissements suivent une loi de Rayleigh, qui a pour densité de probabilité

$$p_R(\alpha) = 2\alpha e^{-\alpha^2}, \quad \forall \alpha \in [0, +\infty[$$

Soit $C_{|\alpha_1,\alpha_2}$ la capacité du canal sachant les évanouissements α_1 et α_2 fixés. L'alphabet $\mathcal{M}$ reçu est alors

$$\mathcal{M} = \left\{ \ -\alpha_1\omega_1 - \alpha_2\omega_2 \ , \ -\alpha_1\omega_1 + \alpha_2\omega_2 \ , \ \alpha_1\omega_1 - \alpha_2\omega_2 \ , \ \alpha_1\omega_1 + \alpha_2\omega_2 \ \right\}$$

où deux des symboles peuvent parfois avoir la même valeur. La capacité $C_{|\alpha_1,\alpha_2}$ se calcule par la formule (3.12) et il faut ensuite moyenner cette capacité conditionnelle sur toutes les valeurs des évanouissements pour obtenir la capacité C du canal :

$$C = \int_0^{+\infty} \int_0^{+\infty} p_R(\alpha_1) p_R(\alpha_2) C_{|\alpha_1,\alpha_2} \mathrm{d}\alpha_1 \mathrm{d}\alpha_2$$

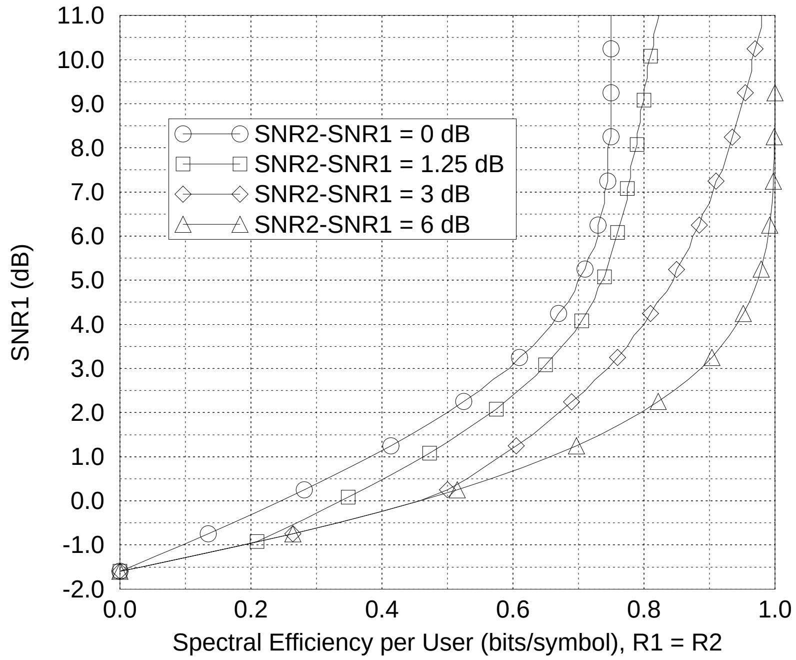


FIG. 3.5 – *E_{b1}/N_0 minimal requis pour atteindre une probabilité d'erreur nulle ($K = 2$, canal AWGN).*

L'emploi de l'algorithme du point fixe s'avère ici difficile. Il est plus facile de calculer la capacité pour une valeur de E_s/N_0 donnée et d'en déduire E_b/N_0.

La figure 3.6 montre les limites de la capacité multi-utilisateurs pour deux modulations BPSK ayant la même énergie par bit d'information ($\Delta E_b = 0$) et traversant deux canaux de Rayleigh distincts. Le taux d'information de chaque modulation BPSK doit être plus faible que la capacité du canal de Rayleigh sur lequel on transmet une unique modulation BPSK et la somme des deux taux doit être plus faible que la capacité du canal si on transmet les symboles de l'alphabet $\mathcal{M}$.

Finalement, la figure 3.7 représente les valeurs minimales de $SNR_1 = E_{b_1}/N_0$ requises pour atteindre une probabilité d'erreur nulle en fonction du nombre de bits par symbole pour chaque utilisateur. Les deux utilisateurs ont la même efficacité spectrale ($R_1 = R_2$ et $\Delta E_s(\text{dB}) = \Delta E_b(\text{dB})$). On remarque qu'ici la capacité totale du système où les deux utilisateurs ont la même puissance tend vers la valeur 2 et non vers la valeur 1.5 comme sur le canal gaussien. Cette amélioration asymptotique est due aux évanouissements, qui rendent la probabilité que les deux utilisateurs aient la même puissance à un instant donné quasiment nulle.

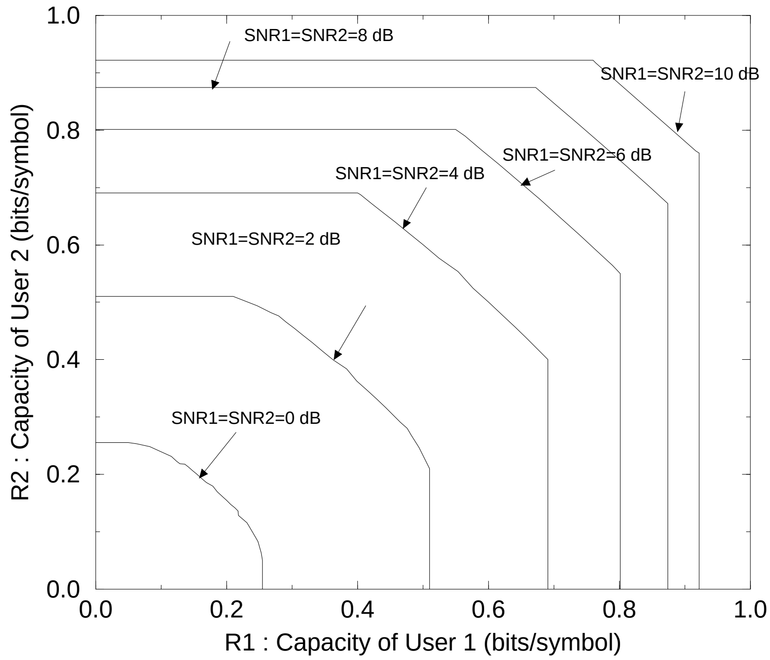


FIG. 3.6 – *Diagramme de capacité avec $\Delta E_b = 0$ dB ($K = 2$, canaux de Rayleigh).*

3.5 Performances sur un canal non bruité

Nous présentons dans cette section les performances du système d'accès multiple sur un canal non bruité avec deux utilisateurs ayant le même taux : $R_1 = R_2$. Le taux d'erreur par bit (*BER* pour *Bit Error Rate*) est obtenu en fonction du rapport signal à bruit par des simulations de Monte Carlo.

Le cas de deux utilisateurs avec des énergies différentes est trivial puisque l'alphabet obtenu contient 4 symboles distincts et qu'en l'absence de bruit une probabilité d'erreur nulle est atteinte sans codage. Par contre, l'alphabet de taille 3 du cas où les deux utilisateurs ont la même énergie n'est pas uniquement décodable puisqu'il existe une ambiguïté sur le symbole '0'. Pour combler cette lacune, on doit utiliser un code correcteur d'erreurs. Les simulations présentées dans la suite de cette section vont bien sûr porter sur ce dernier cas non trivial.

Sur la figure 3.8 on peut voir les performances obtenues en employant des codes (7,5) et (17,15) de taux 1/2 et respectivement de longueurs de contrainte 3 et 5 après 4, 6 et 10 itérations de décodage. Ces codes sont poinçonnés de manière à obtenir un taux de 2/3. Le taux global obtenu est donc 4/3 et on voit que la transmission est quasiment sans erreurs pour des tailles d'entrelaceurs supérieures à 5000 pour le code (7,5) et à 2000 pour le code (17,15) après 10 itérations de décodage. On constate donc que la taille des entrelaceurs peut être réduite si la longueur de contrainte des codes est légèrement

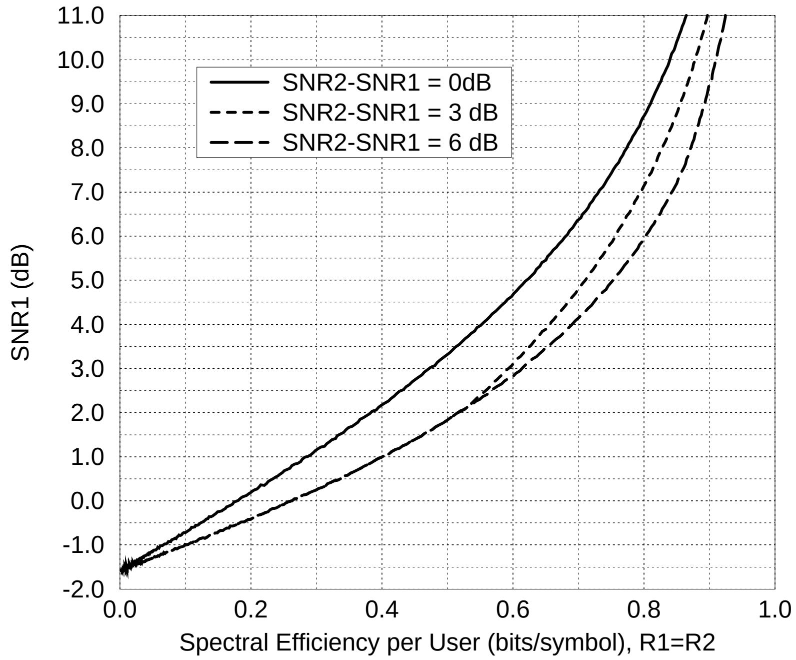


FIG. 3.7 – *E_{b1}/N_0 minimal requis pour atteindre une probabilité d'erreur nulle ($K = 2$, canaux de Rayleigh).*

augmentée. Cependant, à la première itération, l'interférence est complètement inconnue et le décodage est, de ce fait, confronté à un rapport signal à bruit très faible. Il doit donc être performant pour les faibles valeurs de SNR. Par conséquent, les codes convolutifs avec une grande longueur de contrainte donnent des résultats décevants car, bien qu'ils soient très efficaces à fort SNR, ils présentent des performances trop mauvaises pour les faibles valeurs de rapport signal à bruit.

Grâce à la figure 3.9, on peut comparer, après 10 itérations de décodage, les performances de quatre schémas de codage différents sur le canal non bruité : le code (7,5) poinçonné à un taux 2/3, le code (17,15) également poinçonné à un taux 2/3, le code (17,06,15) non poinçonné de taux 2/3 avec un entrelacement sur les bits et finalement ce même code (17,06,15) avec un entrelacement sur les branches. L'entrelaceur par branches considère les 3 bits codés correspondant au franchissement d'une étape dans le treillis comme un bloc unique. Ces 3 bits resteront donc groupés après entrelacement contrairement à ce qui se passe avec l'entrelacement par bits. Cet entrelacement par branches a pour intérêt de ne pas corrompre les autres étapes du treillis en cas de mauvaise estimation de l'interférence sur une transition donnée. En effet, dans ce cas, l'entrelacement par bits va corrompre 1, 2 ou 3 étapes du treillis lors du décodage de l'autre utilisateur. Les simulations ont néanmoins montré que de fréquentes erreurs se produisaient toujours sur les mêmes étapes du treillis à cause de la structure de l'entrelaceur. L'entrelacement par bits ne souffre pas de ce phénomène car les 3 bits d'une transition dans le treillis sont

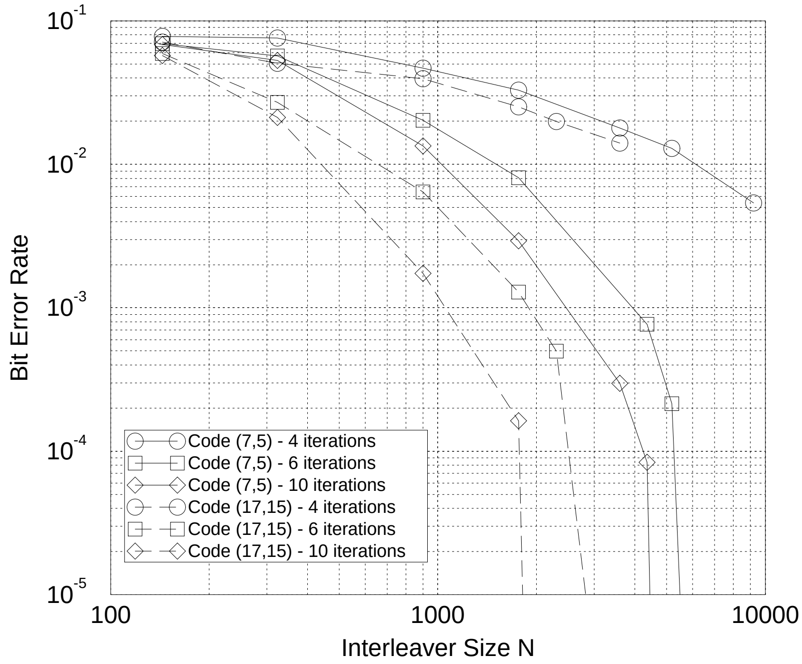


FIG. 3.8 – *Codes (7,5) et (17,15) poinçonnés ($K = 2$, canal non bruité)*

traités séparément, ce qui réduit les risques d'une erreur irréparable. Pour améliorer les performances de l'entrelacement par branches, nous avons repéré ces branches *à risques* par simulation et nous avons supprimé les erreurs répétitives en forçant à '0' les bits d'information correspondants. On peut alors utiliser cette valeur connue dans le décodage et les étoiles sur la figure 3.9 montrent le gain en taux d'erreur par bit après élimination de quatre branches sensibles pour des entrelaceurs de tailles 900 et 3600. La perte en taux d'information est négligeable.

Les résultats présentés dans cette section sont meilleurs que ceux donnés par Tarköy dans [66] pour de grands entrelaceurs et avec seulement 10 itérations de décodage. Tarköy propose le même codeur d'accès multiple mais utilise un décodeur différent présentant une complexité moindre en détriment de la flexibilité. En effet, au lieu d'utiliser les probabilités a posteriori des bits codés des autres utilisateurs comme information a priori, il s'en sert pour calculer la moyenne conditionnelle et la variance de l'interférence modélisée comme un bruit gaussien. La moyenne est ensuite soustraite du signal reçu et la variance utilisée dans la métrique de l'algorithme de décodage.

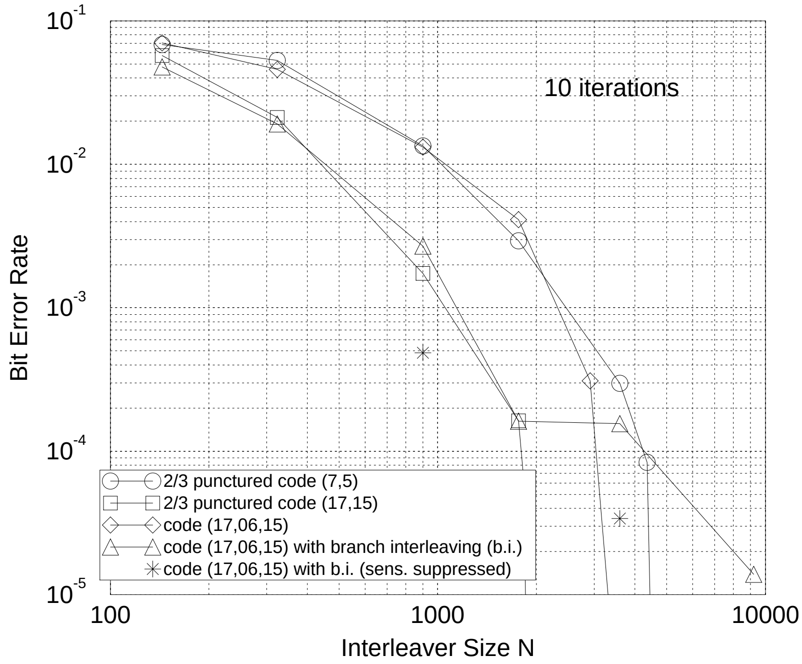


FIG. 3.9 – *Comparaison de différents codes et types d'entrelacements ($K = 2$, canal non bruité).*

3.6 Performances avec des entrelaceurs infinis sur un canal bruité

Avant de nous intéresser aux résultats des simulations pour des tailles d'entrelaceurs finies, nous présentons les performances du système dans le cas asymptotique d'un entrelaceur de taille infinie sur canal AWGN. Richardson et Urbanke [55] ont récemment découvert une méthode pour obtenir de manière analytique les performances du décodage itératif des codes LDPC (*Low Density Parity Check Codes*) [26] lorsque ceux-ci ont une taille infinie. Cette méthode a ensuite été généralisée dans [9] et [50] aux turbo-codes [4][2] et aux codes GLD (*Generalized Low Density Codes*) [8][85]. Nous nous proposons maintenant d'appliquer cette méthode dite de propagation des distributions des LLRs à notre décodage itératif multi-utilisateurs pour trouver ses performances dans le cas idéal d'entrelaceurs de taille infinie.

3.6.1 Calcul de la formule de propagation de la densité avec 2 utilisateurs

Décodage optimal

Nous nous limitons au cas simple d'un système synchrone avec deux utilisateurs sur canal AWGN. Supposons qu'à l'étape s on décode l'utilisateur 1. Intéressons-nous plus particulièrement au décodage de son $j^{\text{ème}}$ bit codé $c_1(j)$. Il faut remarquer que, comme l'entrelaceur employé est de taille infinie, le code convolutif n'est plus régulièrement fermé et l'ensemble des bits de l'utilisateur est décodé d'un coup.

Soit $\text{APP}\Big(c_1(j)\Big)$ la probabilité a posteriori du bit $c_1(j)$ après décodage. On note $\mathcal{C}_1$ le code de l'utilisateur 1 et $\mathbf{c}_1$ un mot de code de $\mathcal{C}_1$.

$$\begin{aligned}
\text{APP}\Big(c_1(j)\Big) &= p^1_{C,s}(j) = \Pr\Big\{c_1(j) \mid \mathbf{r}_1\Big\} \\
&= \sum_{\mathbf{c}_1 \in \mathcal{C}_1} \Pr\Big\{c_1(j)\ ,\ \mathbf{c}_1 \mid \mathbf{r}_1\Big\} \\
&= \sum_{\mathbf{c}_1 \in \mathcal{C}_1} \frac{\Pr\Big\{\mathbf{r}_1 \mid c_1(j)\ ,\ \mathbf{c}_1\Big\}\Pr\Big\{c_1(j),\mathbf{c}_1\Big\}}{\Pr\{\mathbf{r}_1\}} \\
&\propto \sum_{\mathbf{c}_1 \in \mathcal{C}_1 | c_1(j)} \Pr\Big\{\mathbf{r}_1|\mathbf{c}_1\Big\}\Pr\{\mathbf{c}_1\} \\
&\propto \sum_{\mathbf{c}_1 \in \mathcal{C}_1 | c_1(j)} \prod_{\ell=0}^{W-1} p\Big(r_1(\ell) \mid c_1(\ell)\Big)\Pr\Big\{c_1(\ell)\Big\} \qquad (3.14)
\end{aligned}$$

On ne dispose d'aucune information a priori sur les bits codés de l'utilisateur 1 donc $\Pr\Big\{c_1(\ell)\Big\} = 0.5$. W est la taille en nombre de bits codés d'une fenêtre autour du bit $c_1(j)$ sur laquelle on fait fonctionner l'algorithme forward-backward. Les simulations de Monte-Carlo ont montré qu'une taille W égale à $16n$ fois la mémoire du code suffisait pour égaler, grâce à un décodage localisé, les performances du décodage du code entier, qui est de taille infinie [9].

Les équations (3.1) et (3.2) appliquées à un système avec deux utilisateurs donnent

$$\begin{aligned}
p\Big(r_1(\ell) \mid c_1(\ell)\Big) &= p\Big(r_1(\ell) \mid c_1(\ell)\ ,\ \tilde{c}^1_2(\ell) = 1\Big)\Pr\{\tilde{c}^1_2(\ell) = 1\} \\
&\quad + p\Big(r_1(\ell) \mid c_1(\ell)\ ,\ \tilde{c}^1_2(\ell) = 0\Big)\Pr\{\tilde{c}^1_2(\ell) = 0\} \qquad (3.15)
\end{aligned}$$

On note A^1_{ji} (respectivement A^0_{ji}) le nombre de mots de code de poids de Hamming i du code convolutif $\mathcal{C}_1$ dans la fenêtre de taille W tels que le $j^{\text{ème}}$ bit codé $c_1(j)$ ait la valeur 1 (respectivement la valeur 0). Les polynômes énumérateurs de poids conditionnels sont $A^1_j(x) = \sum_i A^1_{ji}x^i$ et $A^0_j(x) = \sum_i A^0_{ji}x^i$. Dans [2], Benedetto et Montorsi définissaient déjà un polynôme énumérateur de poids conditionnel mais la condition était alors le poids de Hamming de la séquence des bits d'information. Le code est isotrope si $A^1_j(x) = A^1(x)$

et $A^0_j(x) = A^0(x)$ pour toutes les positions j, c'est-à-dire si tous les bits du mot de code sont protégés de la même façon par un décodeur APP [9]. Les bits codés situés à une position donnée sur une branche du treillis sont protégés de la même manière sur toutes les branches du fait de la périodicité du treillis. Par contre, d'un bit à l'autre d'une même branche, la protection peut varier. Nous avons donc vérifié que les codes que nous employions étaient isotropes et qu'ils se prêtaient donc bien à l'étude de cette section, dans laquelle on suppose que les codes sont isotropes. On peut donc omettre l'indice j dans les notations. Le code $\mathcal{C}_1$ contient 2^{k-1} mots de code tels que $c_1(j)$ ait la valeur 1. Le mot de code $\mathbf{c}_1$ de poids de Hamming $w(\mathbf{c}_1)$ est formé de $w(\mathbf{c}_1)$ bits à 1 et $W - w(\mathbf{c}_1)$ bits à 0. Si on note $\ell^1_{\mathbf{c}_1,1}, \ldots, \ell^1_{\mathbf{c}_1,w(\mathbf{c}_1)}$ les positions correspondant à des bits à 1 et $\ell^0_{\mathbf{c}_1,1}, \ldots, \ell^0_{\mathbf{c}_1,W-w(\mathbf{c}_1)}$ les positions des autres bits, la formule (3.14) devient :

$$\begin{aligned}
\mathrm{APP}\Big(c_1(j)=1\Big) &\propto \sum_{\substack{\mathbf{c}_1\in\mathcal{C}_1\\ c_1(j)=1}} \prod_{p=1}^{w(\mathbf{c}_1)} p\Big(r_1(\ell^1_{\mathbf{c}_1,p}) \mid c_1(\ell^1_p)=1\Big) \prod_{p=1}^{W-w(\mathbf{c}_1)} p\Big(r_1(\ell^0_{\mathbf{c}_1,p}) \mid c_1(\ell^0_p)=0\Big) \\
&\propto \sum_{\substack{\mathbf{c}_1\in\mathcal{C}_1\\ c_1(j)=1}} \prod_{p=1}^{w(\mathbf{c}_1)} \frac{p\Big(r_1(\ell^1_{\mathbf{c}_1,p}) \mid c_1(\ell^1_p)=1\Big)}{p\Big(r_1(\ell^1_{\mathbf{c}_1,p}) \mid c_1(\ell^1_p)=0\Big)} \prod_{\ell=1}^{W} p\Big(r_1(\ell) \mid c_1(\ell)=0\Big) \qquad (3.16)
\end{aligned}$$

On obtient une expression de $\mathrm{APP}\Big(c_1(j)=0\Big)$ très proche de (3.16) :

$$\mathrm{APP}\Big(c_1(j)=0\Big) \propto \sum_{\substack{\mathbf{c}_1\in\mathcal{C}_1\\ c_1(j)=0}} \prod_{p=1}^{w(\mathbf{c}_1)} \frac{p\Big(r_1(\ell^1_{\mathbf{c}_1,p}) \mid c_1(\ell^1_p)=1\Big)}{p\Big(r_1(\ell^1_{\mathbf{c}_1,p}) \mid c_1(\ell^1_p)=0\Big)} \prod_{\ell=1}^{W} p\Big(r_1(\ell) \mid c_1(\ell)=0\Big) \qquad (3.17)$$

Pour ne pas avoir à déterminer les constantes de proportionnalité, on introduit les LLRs suivants, qui s'expriment grâce aux relations (3.16) et (3.17) :

$$\begin{aligned}
LLR_s(j) &= \ln\left[\frac{\mathrm{APP}\Big(c_1(j)=1\Big)}{\mathrm{APP}\Big(c_1(j)=0\Big)}\right] \\
&= \ln\left[\frac{\displaystyle\sum_{\substack{\mathbf{c}_1\in\mathcal{C}_1\\ c_1(j)=1}} \prod_{p=1}^{w(\mathbf{c}_1)} \frac{p\Big(r_1(\ell^1_{\mathbf{c}_1,p}) \mid c_1(\ell^1_p)=1\Big)}{p\Big(r_1(\ell^1_{\mathbf{c}_1,p}) \mid c_1(\ell^1_p)=0\Big)}}{\displaystyle\sum_{\substack{\mathbf{c}_1\in\mathcal{C}_1\\ c_1(j)=0}} \prod_{p=1}^{w(\mathbf{c}_1)} \frac{p\Big(r_1(\ell^1_{\mathbf{c}_1,p}) \mid c_1(\ell^1_p)=1\Big)}{p\Big(r_1(\ell^1_{\mathbf{c}_1,p}) \mid c_1(\ell^1_p)=0\Big)}}\right] \qquad (3.18) \\
LLR_0(j) &= \ln\left(\frac{0.5}{0.5}\right) = 0
\end{aligned}$$

Pour deux variables aléatoires X et Y indépendantes et identiquement distribuées (iid), le produit XY est noté $X^{\otimes 2}$. Pour deux variables aléatoires X et Z dépendantes et identiquement distribuées (id), la somme $X+Z$ est notée $2\odot X$. Les variables aléatoires $p\Big(r_1(\ell^1_p) \mid c_1(\ell^1_p)=1\Big)/p\Big(r_1(\ell^1_p) \mid c_1(\ell^1_p)=0\Big)$ sont iid, donc

$$\prod_{p=1}^{w(\mathbf{c}_1)} \frac{p\Big(r_1(\ell^1_{\mathbf{c}_1,p}) \mid c_1(\ell^1_p)=1\Big)}{p\Big(r_1(\ell^1_{\mathbf{c}_1,p}) \mid c_1(\ell^1_p)=0\Big)} = \left[\frac{p\Big(r_1(0) \mid c_1(0)=1\Big)}{p\Big(r_1(0) \mid c_1(0)=0\Big)}\right]^{\otimes w(\mathbf{c}_1)} \qquad (3.19)$$

Il existe A_i^1 (resp. A_i^0) mots de code de poids $w(\mathbf{c}_1) = i$ tels que $c_1(j) = 1$ (resp. $c_1(j) = 0$) mais les termes correspondants dans la somme ne sont pas indépendants car les produits (3.19) sont calculés à partir du même mot reçu de longueur W. On peut donc écrire LLR_s à partir de (3.18) sous la forme

$$\begin{aligned} LLR_s &= \ln \left[\frac{\sum_{i=0}^{W} A_i^1 \odot \left[\frac{p\Big(r_1(0) \mid c_1(0) = 1\Big)}{p\Big(r_1(0) \mid c_1(0) = 0\Big)} \right]^{\otimes i}}{\sum_{i=0}^{W} A_i^0 \odot \left[\frac{p\Big(r_1(0) \mid c_1(0) = 1\Big)}{p\Big(r_1(0) \mid c_1(0) = 0\Big)} \right]^{\otimes i}} \right] \\ &= \ln \left[\frac{\sum_{i=0}^{W} A_i^1 \odot [\exp(\mathcal{O}_s)]^{\otimes i}}{\sum_{i=0}^{W} A_i^0 \odot [\exp(\mathcal{O}_s)]^{\otimes i}} \right] \end{aligned} \tag{3.20}$$

avec

$$\mathcal{O}_s = \ln \left[\frac{p\Big(r_1(0) \mid c_1(0) = 1\Big)}{p\Big(r_1(0) \mid c_1(0) = 0\Big)} \right] \tag{3.21}$$

qui représente l'observation pour le décodage de l'utilisateur 1 à l'étape s. Tous les numéros de bits ont été remplacés par le numéro 0 pour bien souligner que toutes les probabilités suivent la même loi, indépendamment du bit concerné, mais la dépendance entre les produits est liée à la position des '1'. Il faut noter que l'isotropie, qui explique la disparition de l'indice j dans (3.20), ne dépend pas seulement des polynômes énumérateurs de poids conditionnels. Les positions des '1' dans les mots de code inclus dans ces polynômes doivent être également indépendantes de j. Cette condition est vérifiée d'une branche à l'autre du treillis d'un code convolutif grâce à sa périodicité. Les 4 observations de l'utilisateur 1 sont :

$$\begin{aligned} O_{00} &= p\Big(r_1(0) \mid c_1(0) = 0 \ , \ \tilde{c}_2^1(0) = 0\Big) \quad ; \quad O_{01} = p\Big(r_1(0) \mid c_1(0) = 0 \ , \ \tilde{c}_2^1(0) = 1\Big) \\ O_{10} &= p\Big(r_1(0) \mid c_1(0) = 1 \ , \ \tilde{c}_2^1(0) = 0\Big) \quad ; \quad O_{11} = p\Big(r_1(0) \mid c_1(0) = 1 \ , \ \tilde{c}_2^1(0) = 1\Big) \end{aligned}$$

L'introduction de ces observations dans (3.21) via l'équation (3.15) donne

$$\mathcal{O}_s = \ln \left[\frac{O_{11}.\Pr\{\tilde{c}_2^1(0) = 1\} \ + \ O_{10}.\Pr\{\tilde{c}_2^1(0) = 0\}}{O_{01}.\Pr\{\tilde{c}_2^1(0) = 1\} \ + \ O_{00}.\Pr\{\tilde{c}_2^1(0) = 0\}} \right] \tag{3.22}$$

Il faut à chaque itération déterminer la distribution de $\mathcal{O}_s$ à partir de la distribution a priori de LLR_{s-1} des bits codés de l'utilisateur 2 fournie par l'itération précédente. La relation de propagation (3.20) décrit donc comment la distribution de probabilité des LLR_s a posteriori des bits codés de l'utilisateur 1 est reliée aux observations et à la distribution de probabilité des LLR_{s-1} a priori des bits codés de l'utilisateur 2. A l'étape suivante, les rôles des deux utilisateurs seront bien sûr inversés mais la relation restera la même en intervertissant les indices 1 et 2. On trouve la distribution de probabilité des LLRs a posteriori des bits d'information de l'utilisateur 1 en remplaçant dans l'équation (3.20) A_i^1 (respectivement A_i^0) par le nombre de mots de codes de poids i tels que le bit d'information qui nous intéresse soit égal à 1 (respectivement 0). Les polynômes énumérateurs de poids conditionnels pour les bits codés et les bits d'information sont obtenus grâce à la matrice de transfert du code convolutif [16]. Cette méthode est expliquée dans l'annexe D.

Nous supposons que le mot de code tout à zéro est émis par l'utilisateur 1. Les bits de l'utilisateur 2 interférents font partie d'un mot de code entrelacé par un entrelaceur de taille infinie. On peut donc considérer qu'ils prennent les valeurs '0' ou '1' de manière équiprobable. Sur un canal gaussien, le signal reçu est donc soit le symbole $-\omega_1 - \omega_2$ bruité, soit le symbole $-\omega_1+\omega_2$ bruité et sa distribution est la somme de deux gaussiennes centrées sur ces deux symboles et de variance N_0. La distribution de $\mathcal{O}$ est plus délicate à obtenir. Supposons tout d'abord que le bit codé $\tilde{c}_2^1(0)$ de l'utilisateur 2 prenne la valeur '1'. On reçoit donc $r_1(0) = -\omega_1+\omega_2+n(0)$ où $n(0)$ est un bruit blanc additif gaussien de moyenne nulle et de variance N_0. Comme LLR_{s-1} a été calculée à l'itération précédente en supposant que l'utilisateur 2 n'avait émis que des bits à '0', les probabilités a priori s'obtiennent de la manière suivante :

$$\Pr\{\tilde{c}_2^1(0) = 1\} = \frac{1}{1 + e^{LLR_{s-1}}} \quad ; \quad \Pr\{\tilde{c}_2^1(0) = 0\} = \frac{1}{1 + e^{-LLR_{s-1}}}$$

Les 4 observations sont :

$$O_{00|\tilde{c}_2^1(0)=1} \propto \exp\left(-\frac{(2\omega_2 + n(0))^2}{2N_0}\right) \quad ; \quad O_{01|\tilde{c}_2^1(0)=1} \propto \exp\left(-\frac{n(0)^2}{2N_0}\right)$$
$$O_{10|\tilde{c}_2^1(0)=1} \propto \exp\left(-\frac{(-2\omega_1 + 2\omega_2 + n(0))^2}{2N_0}\right) \quad ; \quad O_{11|\tilde{c}_2^1(0)=1} \propto \exp\left(-\frac{(-2\omega_1 + n(0))^2}{2N_0}\right)$$

On obtient donc à partir de (3.22)

$$\mathcal{O}_{s|\tilde{c}_2^1(0)=1} = \ln\left[\frac{\exp\left(-\frac{(-2\omega_1+n(0))^2}{2N_0}\right) + \exp\left(LLR_{s-1} - \frac{(-2\omega_1+2\omega_2+n(0))^2}{2N_0}\right)}{\exp\left(-\frac{n(0)^2}{2N_0}\right) + \exp\left(LLR_{s-1} - \frac{(2\omega_2+n(0))^2}{2N_0}\right)}\right]$$

En procédant de même quand l'utilisateur 2 émet un '0', on obtient

$$\mathcal{O}_{s|\tilde{c}_2^1(0)=0} = \ln\left[\frac{\exp\left(LLR_{s-1} - \frac{(-2\omega_1-2\omega_2+n(0))^2}{2N_0}\right) + \exp\left(-\frac{(-2\omega_1+n(0))^2}{2N_0}\right)}{\exp\left(LLR_{s-1} - \frac{(-2\omega_2+n(0))^2}{2N_0}\right) + \exp\left(-\frac{n(0)^2}{2N_0}\right)}\right]$$

La moyenne des distributions de $\mathcal{O}_{s|\tilde{c}_2^1(0)=1}$ et $\mathcal{O}_{s|\tilde{c}_2^1(0)=0}$ donne la distribution de $\mathcal{O}_s$. On peut obtenir simplement ces distributions en tirant un grand nombre d'événements de LLR_{s-1} et $n(0)$ selon leurs lois respectives. La figure 3.10 permet de comparer les distributions de $\mathcal{O}_1$, l'observation utilisée pour le décodage dans la première étape ($s = 1$), selon le niveau de bruit. Les deux utilisateurs ont la même puissance (-5 dB, 0 dB, 3 dB ou 6 dB) et on voit bien les pics autour de $\mathcal{O}_1 = 0.0$, qui trahissent la présence des ambiguïtés.

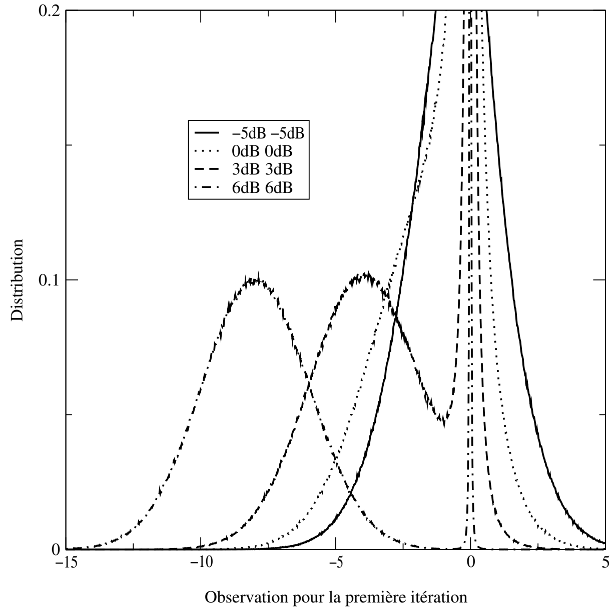


FIG. 3.10 – *Distribution de l'observation $\mathcal{O}_1$ pour la première étape de décodage.*

Soit $p_{LLR_s}(x)$ la densité de probabilité de LLR_s à l'étape s. Puisque le mot de code nul a été émis, LLR_s doit être négatif pour que la décision soit correcte. On peut donc en déduire que la probabilité d'erreur par bit à l'étape s est

$$P_{e_s} = \int_0^{+\infty} p_{LLR_s}(x)\mathrm{d}x \tag{3.23}$$

Si le code convolutif utilisé n'est pas isotrope, il faut calculer n distributions des LLRs en sortie du codeur correspondant chacune à une position sur la branche du treillis. D'une position à l'autre, seuls les polynômes énumérateurs conditionnels changent. L'entrelaceur étant de taille infinie, on peut considérer que le bit codé interférent a des chances égales de venir d'une position ou d'une autre sur la branche du treillis. Il faut donc prendre comme distribution a priori la moyenne des n distributions a posteriori calculées.

Cette étude peut se généraliser à un système de K utilisateurs, le nombre de termes dans le numérateur et dans le dénominateur de $\exp(\mathcal{O}_s)$ étant alors 2^{K-1}.

Décodage sous-optimal par éliminations souples

Nous nous proposons maintenant de calculer la formule de propagation pour le système sous-optimal de détection par éliminations souples proposé par Tarköy sur canal non bruité [66] et Ibrahim sur canal AWGN [32]. Les APPs servent dans ce cas à calculer la moyenne de l'interférence, utilisée comme information souple, que l'on retire du signal reçu. La variance du signal après élimination d'interférence est également estimée pour être utilisée dans la métrique.

Quand la modulation utilisée est une BPSK, l'information souple sur le symbole émis par l'utilisateur 2 est

$$1.\Pr\{\tilde{c}_2^1(0)=1\}+(-1).\Pr\{\tilde{c}_2^1(0)=0\} \;=\; 2.\Pr\{\tilde{c}_2^1(0)=1\}-1$$

Le signal utilisé pour le décodage est donc

$$r_1'(0)=r_1(0)-\omega_2\Big(2.\Pr\{\tilde{c}_2^1(0)=1\}-1\Big)$$

On peut toujours exploiter la formule de propagation (3.20) mais l'observation est maintenant

$$\mathcal{O}_s'=\ln\left[\frac{p\Big(r_1'(0)\mid c_1(0)=1\Big)}{p\Big(r_1'(0)\mid c_1(0)=0\Big)}\right]=\frac{r_1(0)\omega_1-\omega_1\omega_2\Big(2.\Pr\{\tilde{c}_2^1(0)=1\}-1\Big)}{\sigma'^2}$$

La variance σ'^2 peut se calculer directement à partir de la distribution de $r_1'(0)$. Elle est aussi égale à la somme de la variance du bruit N_0 et de la variance de $2.\Pr\{\tilde{c}_2^1(0)=1\}-1$.

Décodage sous-optimal par éliminations dures

Dans le cas de l'élimination dure, où on retranche directement la valeur du bit obtenu par décodage de l'utilisateur interférent à l'étape précédente, le signal utilisé pour le décodage est

$$\begin{aligned} r_1''(0) &= r_1(0)-\omega_2 \quad \text{si } \Pr\{\tilde{c}_2^1(0)=1\}>0.5 \\ r_1''(0) &= r_1(0)+\omega_2 \quad \text{sinon} \end{aligned}$$

La formule de propagation (3.20) est toujours valable mais l'observation est maintenant

$$\mathcal{O}_s''=\ln\left[\frac{p\Big(r_1''(0)\mid c_1(0)=1\Big)}{p\Big(r_1''(0)\mid c_1(0)=0\Big)}\right]=\frac{\omega_1 r_1''(0)}{N_0}$$

Le bit retranché est correct avec une probabilité $1-P_{e_s}$ qui peut être calculée par la formule (3.23). Dans ce cas la distribution de $r_1''(0)$ est une gaussienne centrée sur $-\omega_1$, puisque l'utilisateur 1 n'émet que des '0', et de variance N_0. Par contre, si le bit retranché est erroné, la distribution de $r_1''(0)$ est la somme de deux gaussiennes centrées sur $-\omega_1+2\omega_2$ et $-\omega_1-2\omega_2$ de variance N_0 et pondérées par $1/2$. La distribution de $\mathcal{O}_s''$ est donc la somme de ces deux distributions, la première pondérée par $1-P_{e_s}$ et la deuxième pondérée par P_{e_s}.

3.6.2 Performances

Nous avons appliqué la relation de propagation (3.20) à un système de 2 utilisateurs codés par un code convolutif (23,35) et employant un entrelaceur de taille infinie. La

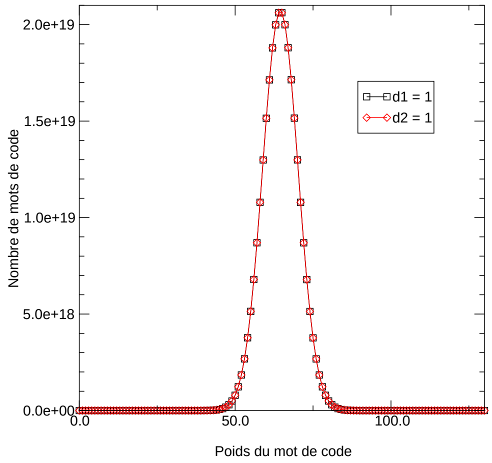


FIG. 3.11 – *Distributions conditionnelles du poids des mots de code selon que le bit est le premier ou le deuxième bit d'une branche du code (23,35) non systématique.*

taille de la fenêtre considérée pour le décodage est $W = 130$. Nous avons vérifié que les polynômes énumérateurs de poids conditionnels étaient bien les mêmes, que le bit concerné soit le premier sur la branche ou le deuxième. Sur la figure 3.11 on peut voir que les deux distributions sont effectivement confondues.

Appliquer directement la relation de propagation sur un code aussi long serait trop coûteux en calculs. C'est pourquoi nous avons opté pour une approche plus pragmatique : nous avons remplacé les polynômes énumérateurs de poids conditionnels par un algorithme forward-backward, qui énumère lui-même les mots du code. En effet, à l'étape s, nous disposons de la distribution de LLR_{s-1} et de N_0 la variance du bruit. Nous pouvons donc tirer W échantillons reçus et W LLRs a priori sur le bit interférent selon leurs lois respectives. Il faut cependant tenir compte du fait que, selon que l'on a émis les bits '0 0' ou '0 1', la distribution de LLR_{s-1} n'est pas la même. C'est pourquoi nous avons toujours travaillé avec les deux distributions en parallèle pour éviter par exemple qu'un symbole victime d'une ambiguïté soit associé avec une probabilité assez forte à une information a priori de bonne qualité. On fait ensuite fonctionner l'algorithme forward-backward sur la fenêtre de taille W et on construit les deux distributions conditionnelles de LLR_s en observant les probabilités a posteriori fournies en sortie de l'algorithme pour la branche située au centre de la fenêtre. On répète cette opération suffisamment de fois pour obtenir une distribution fiable. Si le code est isotrope, la distribution obtenue sera identique quel que soit le bit de la branche considéré.

La figure 3.12(a) représente les distributions des LLRs sur les bits d'information, sur les bits codés victimes d'une ambiguïté et sur les bits codés qui n'en sont pas victimes au fur et à mesure des itérations d'un système optimal dans lequel les 2 utilisateurs ont un rapport signal sur bruit de 3 dB. Une droite en traits interrompus à la verticale de la valeur 0 permet de se faire une idée de la valeur de la probabilité d'erreur par bit. Le décodeur éprouve quelques difficultés à lever les ambiguïtés mais il y arrive enfin après 7 itérations. La figure 3.12(b) présente les mêmes distributions pour un système dans lequel l'utilisateur 1 a un rapport signal à bruit de 3.5 dB et l'utilisateur 2 un rapport signal à bruit de 3 dB. Le mot ambiguïté n'est plus vraiment exact dans ce cas et on désigne par ce mot le symbole $-\omega_1 + \omega_2$, qui est moins fiable que le symbole $-\omega_1 - \omega_2$. Seules les distributions de l'utilisateur 2 sont ici tracées pour éviter d'avoir des changements de rapport signal à bruit d'une distribution à l'autre. On constate qu'ici le système fonctionne très bien puisque les distributions s'éloignent vers les valeurs négatives pour un faible nombre d'itérations.

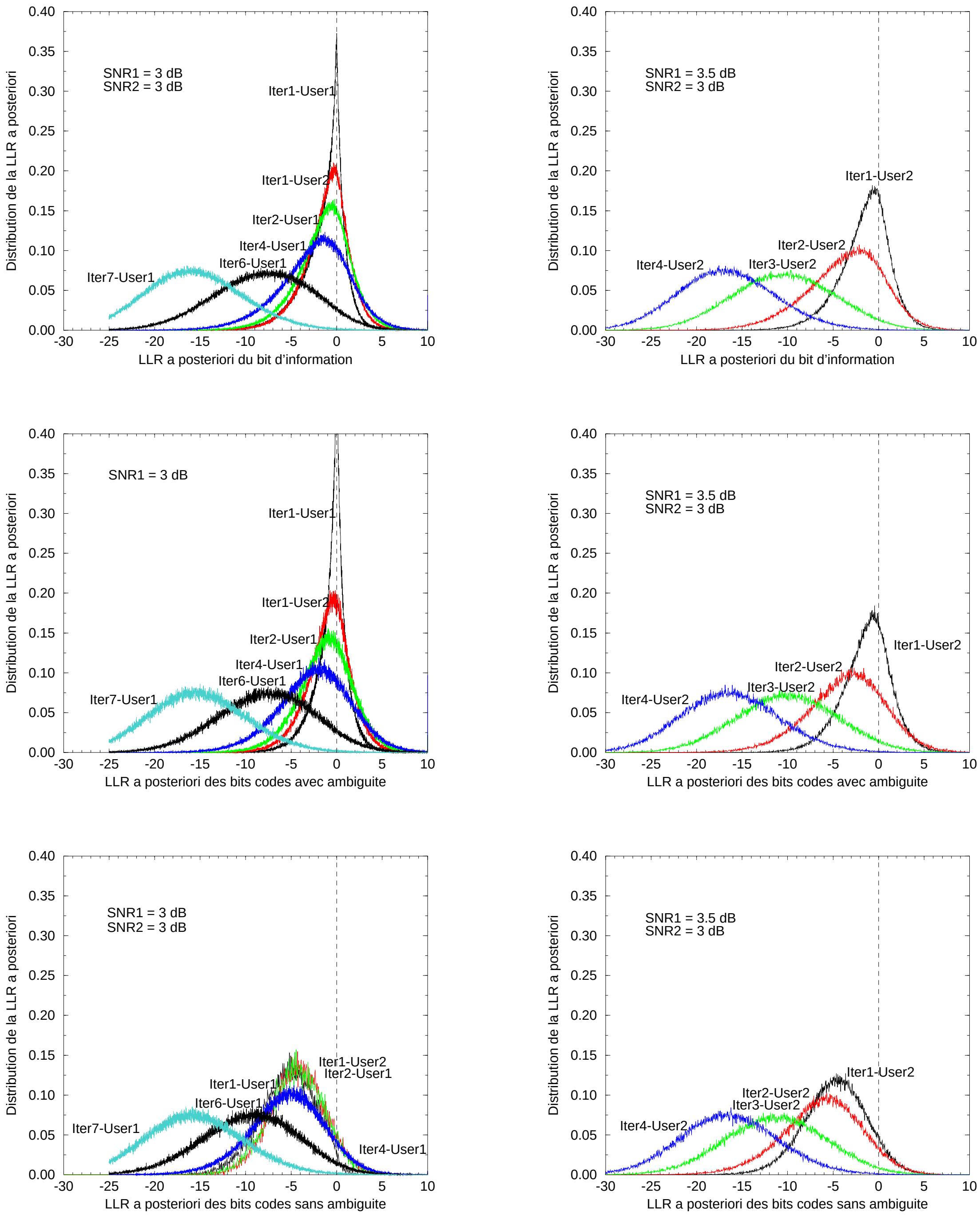
Les figures 3.13(a) et 3.13(b) présentent les performances d'un système synchrone avec 2 utilisateurs codés par un code (23,35) et entrelacés par des entrelaceurs de taille infinie. SNR_1 est fixe, égal à 3 dB, et SNR_2 varie. Le décodage itératif permet d'atteindre les performances mono-utilisateur à partir de 8 itérations. Ces résultats confirment bien la capacité de l'entrelaceur infini à orthogonaliser les signaux des utilisateurs. Les figures 3.14(a) et 3.14(b) présentent les performances du même système avec la détection sous-optimale par éliminations souples. Les performances sont moins bonnes puisque que les performances mono-utilisateur sont atteintes à partir de $SNR_2 = 4.5$ dB et pour la $8^{\text{ième}}$ itération. La complexité de cette méthode est néanmoins son grand atout puisqu'elle est linéaire en fonction du nombre d'utilisateurs. Finalement nous avons testé le même système avec des éliminations dures de l'interférence. Avec cette méthode d'élimination très simple, le système atteint les performances mono-utilisateur à partir de 5 dB et 5 itérations. Pour faciliter la comparaison, les performances de l'itération 8 du système avec éliminations souples et celles de l'itération 5 du système avec éliminations dures sont reportées en traits interrompus épais sur les figures 3.13(a) et 3.13(b).

3.7 Performances avec des entrelaceurs finis sur un canal bruité

Nous présentons dans cette section les performances du système d'accès multiple sur le canal gaussien pour $K = 2$ et $K = 4$ utilisateurs.

3.7.1 Système synchrone

Tout d'abord nous considérons un système synchrone avec 2 utilisateurs employant un code de taux 1/2. La figure 3.16 représente le taux d'erreur par bit pour les deux utilisateurs en fonction du rapport signal à bruit SNR_2 de l'utilisateur 2, le rapport signal à bruit SNR_1 de l'utilisateur 1 étant fixé à 3 dB. Les résultats sont présentés pour trois codes différents : le code convolutif (23,35) de longueur de contrainte 5, le code convolutif (133,171) de longueur de contrainte 7 et finalement le turbo-code résultant



(a) (b)

FIG. 3.12 – *Evolution de la distribution des LLRs sur les bits d'information, sur les bits codés victimes d'une ambiguïté et sur les autres bits codés au cours du décodage itératif d'un système avec 2 utilisateurs : (a) $E_{b_1}/N_0 = E_{b_2}/N_0 = 3dB$ et (b) $E_{b_1}/N_0 = 3.5dB$; $E_{b_2}/N_0 = 3dB$.*

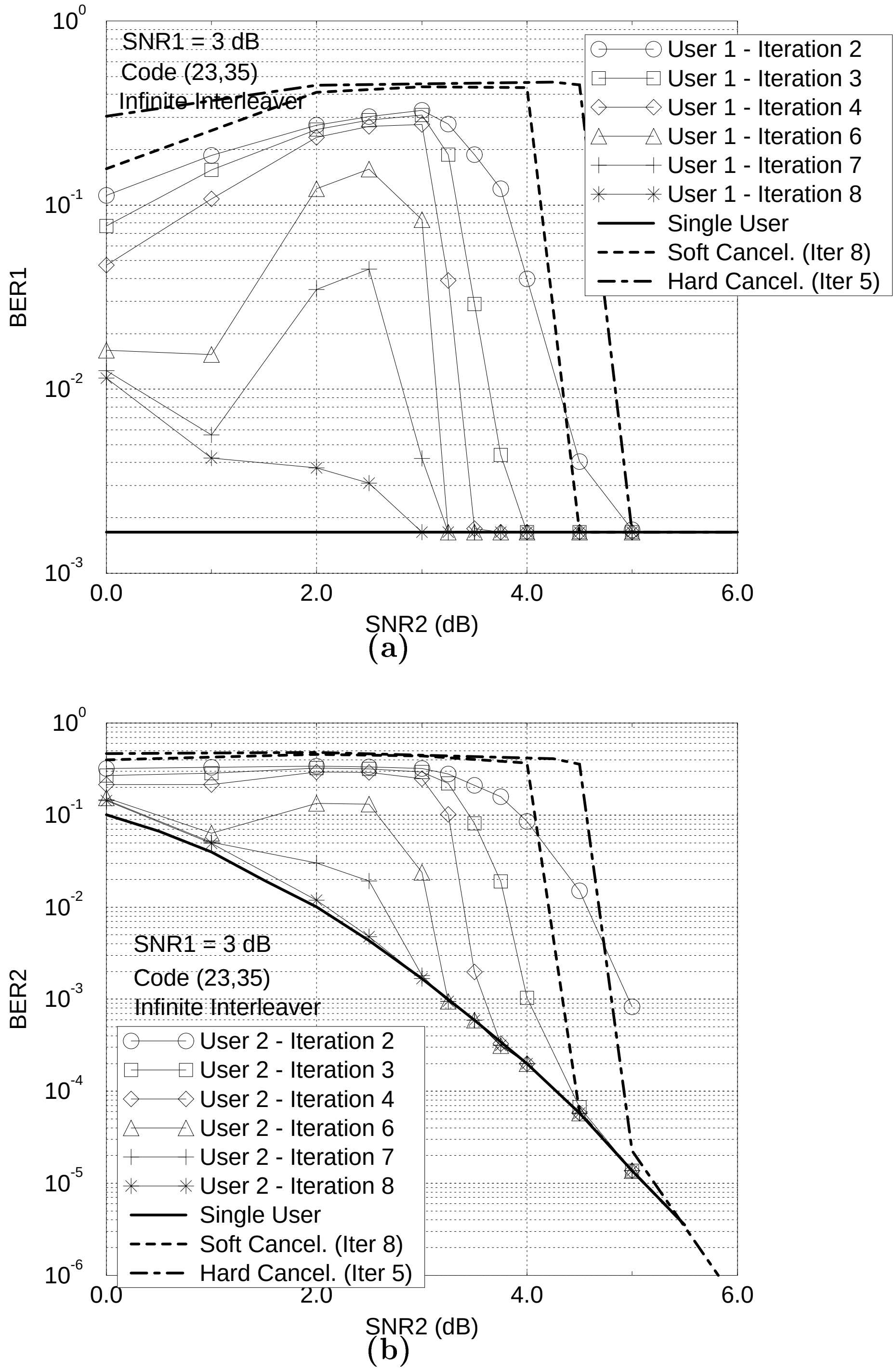


FIG. 3.13 – *Influence du nombre d'itérations de décodage dans le cas d'un entrelaceur de taille infinie, code convolutif (23,35) de taux 1/2, 2 utilisateurs avec $SNR_1 = 3$ dB, canal AWGN synchrone : (a) Utilisateur 1 et (b) Utilisateur 2. Les performances mono-utilisateur ont été ajoutées en lignes épaisses sur les deux figures.*

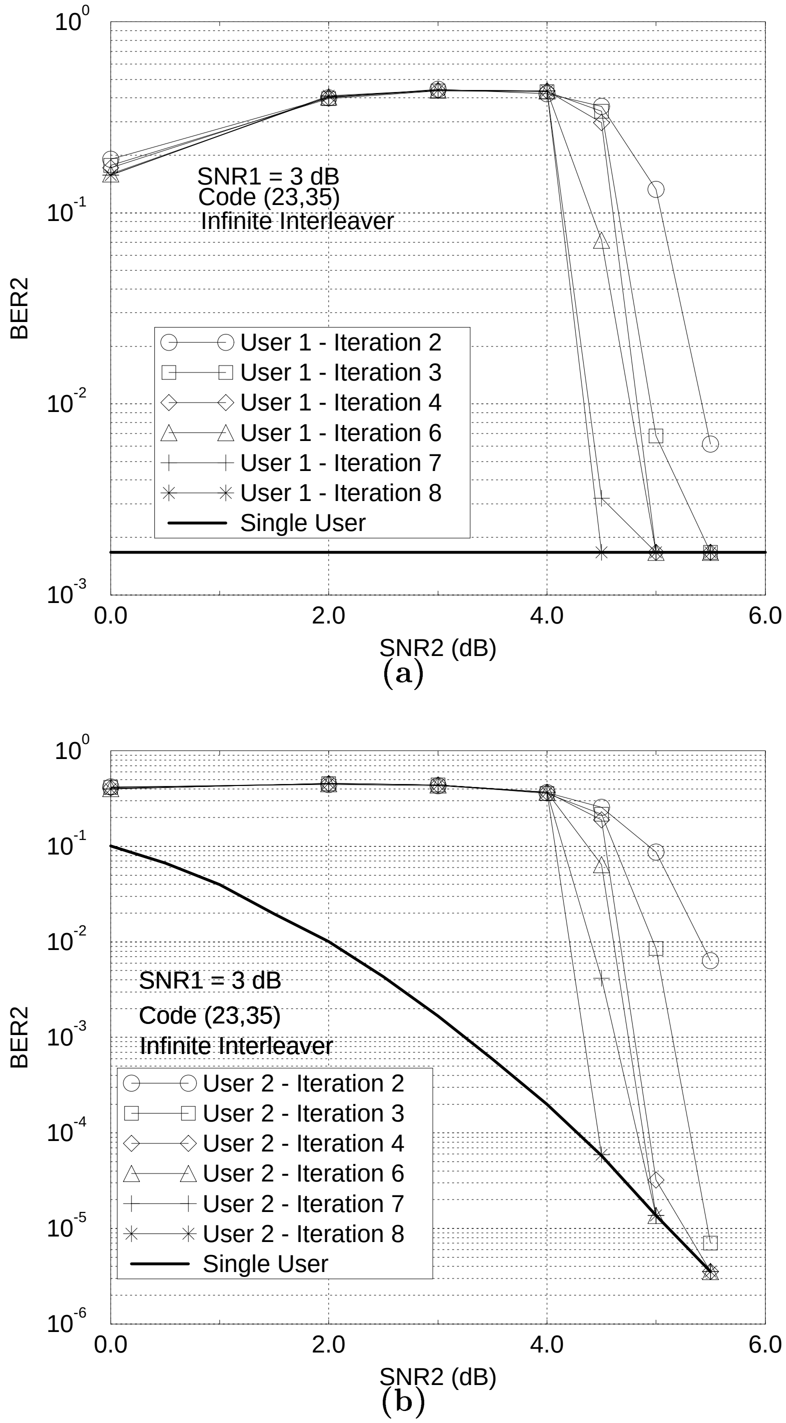
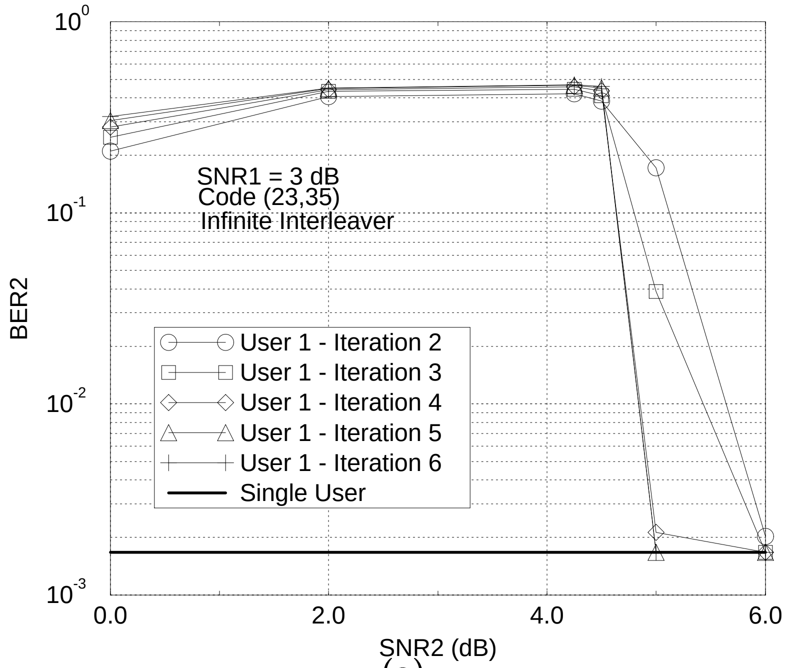
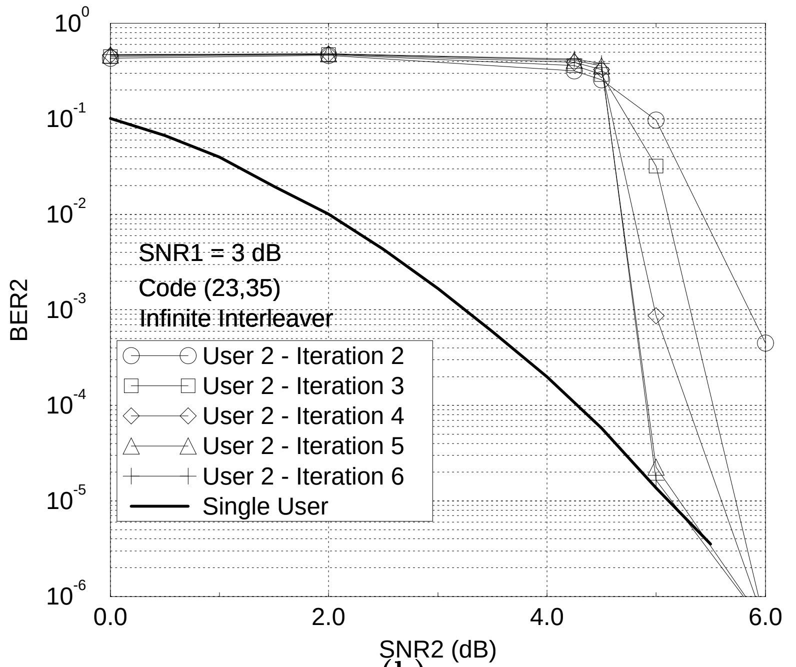


FIG. 3.14 – *Influence du nombre d'itérations de décodage dans le cas d'un entrelaceur de taille infinie avec élimination souple sur le signal reçu, code convolutif (23,35) de taux 1/2, 2 utilisateurs avec $SNR_1 = 3$ dB, canal AWGN synchrone : (a) Utilisateur 1 et (b) Utilisateur 2. Les performances mono-utilisateur ont été ajoutées en lignes épaisses sur les deux figures.*



(a)



(b)

FIG. 3.15 – *Influence du nombre d'itérations de décodage dans le cas d'un entrelaceur de taille infinie avec élimination dure sur le signal reçu, code convolutif (23,35) de taux 1/2, 2 utilisateurs avec $SNR_1 = 3$ dB, canal AWGN synchrone : (a) Utilisateur 1 et (b) Utilisateur 2. Les performances mono-utilisateur ont été ajoutées en lignes épaisses sur les deux figures.*

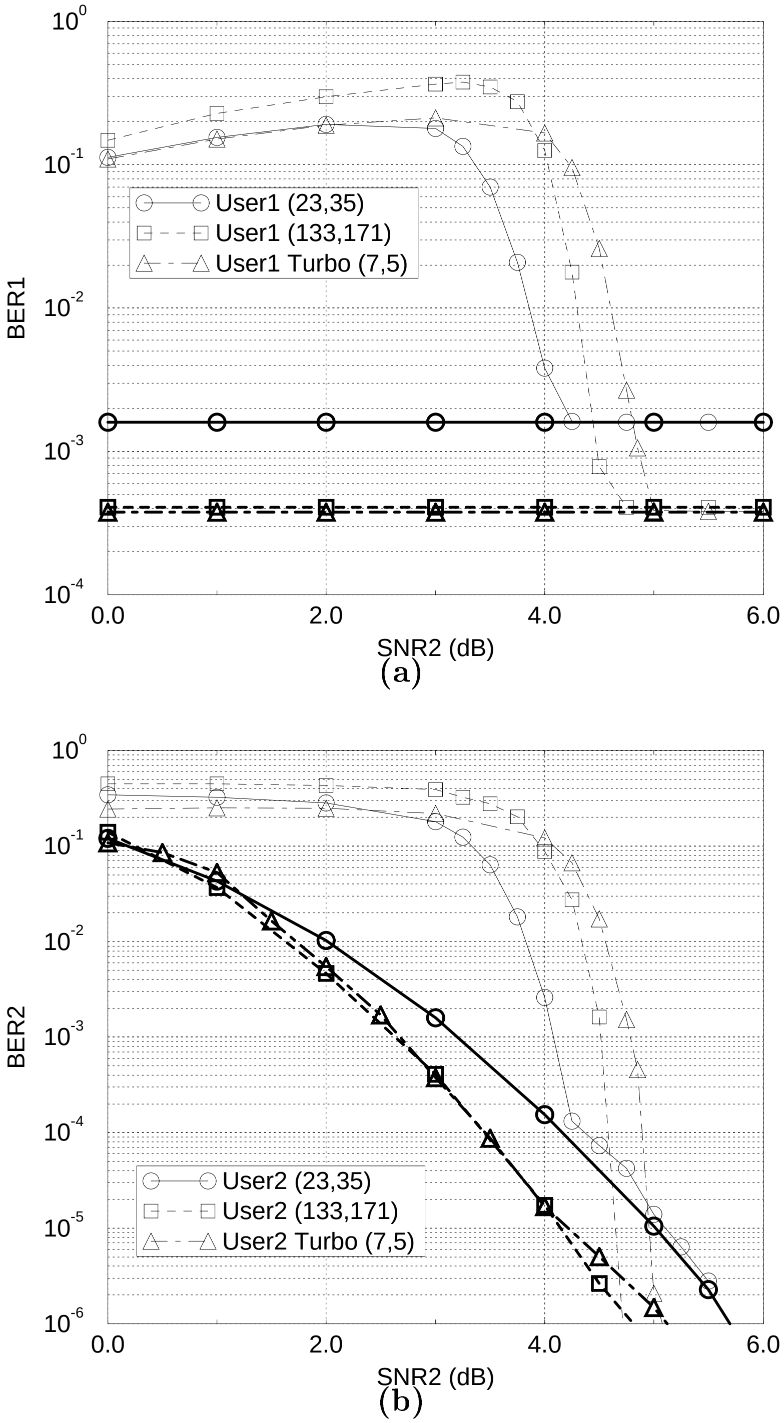


Fig. 3.16 – *Comparaison de codes de taux 1/2 pour 2 utilisateurs avec $SNR_1 = 3$ dB, canal AWGN synchrone : (a) Utilisateur 1 et (b) Utilisateur 2. Les performances mono-utilisateur ont été ajoutées en lignes épaisses sur les deux figures.*

de la concaténation parallèle de deux codes récursifs systématiques (7,5) utilisant un entrelacement de taille 100. Nous avons intentionnellement choisi des codes constituants de longueur de contrainte faible et un entrelaceur petit pour que le turbo-code ainsi obtenu n'ait pas des performances trop mauvaises à faible rapport signal à bruit. La taille des deux entrelaceurs dédiés à l'accès multiple est de 10000 bits codés. Le processus de décodage multi-utilisateurs est itéré 5 fois. A chaque étape, le décodage du turbo-code est itéré 15 fois. Les lignes épaisses sur les figures représentent les performances mono-utilisateur de chaque code. Les lignes épaisses horizontales représentent notamment le taux d'erreur par bit constant de chaque code sur un canal mono-utilisateur pour un rapport signal à bruit SNR_1 constant.

La figure 3.16 illustre clairement la capacité du décodage conjoint itératif à éliminer complètement l'interférence multi-utilisateurs, particulièrement quand les codes sont bien choisis. Par exemple, dans le cas du code (23,35), les performances du système d'accès multiple sont quasiment égales à celles du système mono-utilisateur employant le même code quand $SNR_2 = 4.25$ dB. Pour ce point précis ($SNR_1 = 3$ dB et $SNR_2 = 4.25$ dB), les probabilités d'erreurs des deux utilisateurs avoisinent (2.10^{-3},10^{-4}). Pour deux utilisateurs tels que $\Delta E_s = 1.25$ dB, la paire de rapports signal à bruit minimaux qui permet d'atteindre une probabilité d'erreur nulle est (1.25 dB, 2.5 dB) (cf. figure 3.5). Ce système d'accès multiple est donc à 1.75 dB de la capacité multi-utilisateurs.

Le turbo-code (7,5) a de meilleures performances que le code (23,35) pour des valeurs de SNR_2 élevées mais se comporte mal pour les faibles rapports signal à bruit, à cause de la faible fiabilité des APPs fournies par le décodeur turbo. La figure 3.16 montre également que le code convolutif (133,171) a de moins bonnes performances que le code (23,35) pour de faibles rapports signal à bruit parce que ses performances en présence du très fort niveau de bruit à la première itération sont mauvaises et ne permettent pas d'avoir une amélioration significative à l'itération suivante. La figure 3.17 illustre l'influence du processus de décodage itératif sur le taux d'erreur par bit. Les deux utilisateurs sont codés avec un code convolutif (23,35) de taux 1/2.

Les performances d'un système à 4 utilisateurs sont données à la figure 3.18. Le rapport signal à bruit du premier utilisateur est constant ($SNR_1 = 3$ dB) et les taux d'erreur par bit des utilisateurs 1 et 2 (BER_1 et BER_2) sont tracés en fonction des rapports signal à bruit identiques des utilisateurs 2, 3 et 4 ($SNR_2 = SNR_3 = SNR_4$). Chaque utilisateur est codé par le code convolutif (37, 33, 25, 25, 35, 33, 27, 37) de longueur de contrainte 6 [52] et entrelacé par des entrelaceurs de taille 400 et 10000. Comme on peut le voir sur la figure 3.18, le décodage conjoint itératif supprime entièrement l'interférence multi-utilisateurs. Pour des rapports signal à bruit faibles et modérés, les performances du système d'accès multiple sont plus proches de celles du système mono-utilisateur si on augmente la taille de l'entrelaceur.

Sur la figure 3.19, nous comparons les performances des algorithmes sous-optimaux avec celles de l'algorithme forward-backward pour un entrelaceur de taille 10000 et 5 itérations. Nous constatons que les performances ne sont pas dégradées par la simpli-

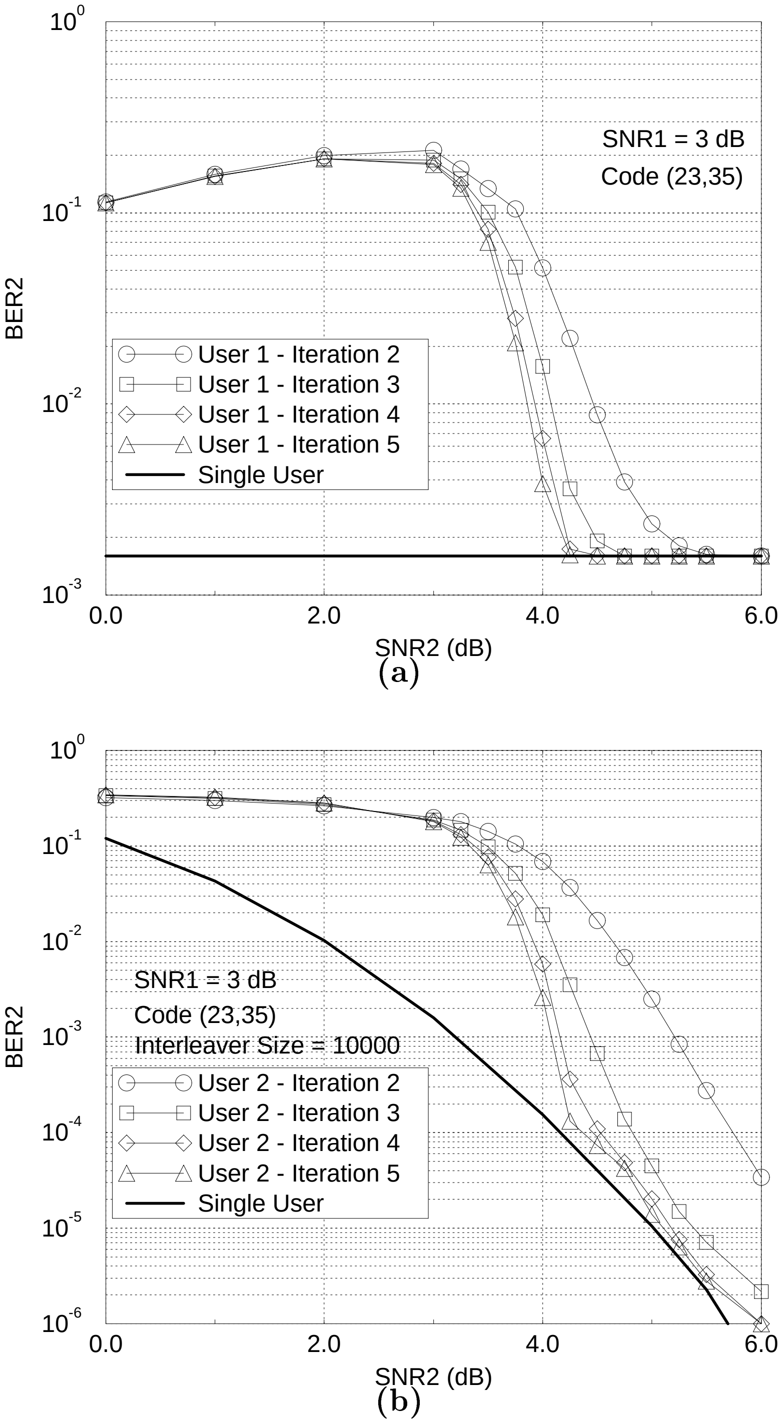
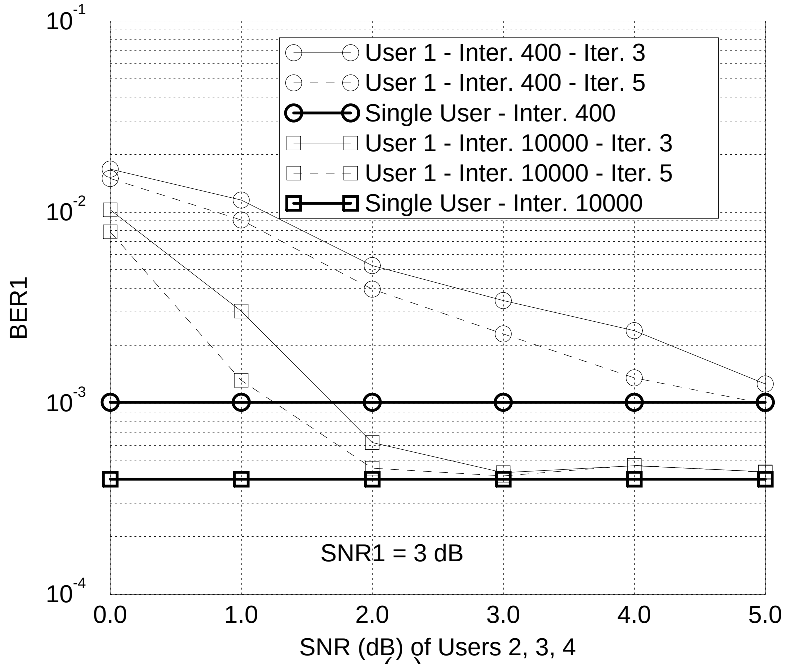
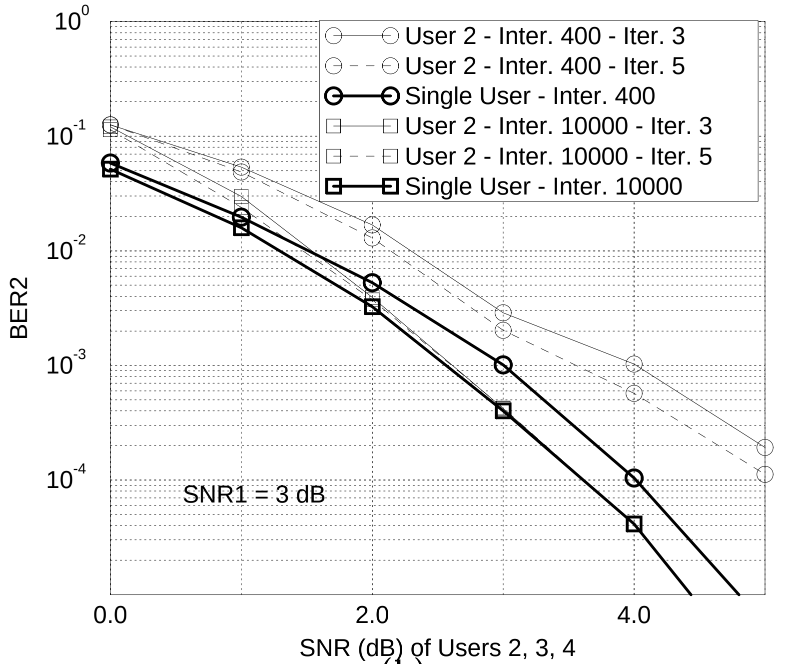


FIG. 3.17 – *Influence du nombre d'itérations de décodage, code convolutif (23,35) de taux 1/2, 2 utilisateurs avec SNR_1 = 3 dB, canal AWGN synchrone : (a) Utilisateur 1 et (b) Utilisateur 2. Les performances mono-utilisateur ont été ajoutées en lignes épaisses sur les deux figures.*



(a)



(b)

FIG. 3.18 – *Comparaison des tailles d'entrelaceur 400 et 10000 pour un code de taux 1/8 et 4 utilisateurs avec SNR_1 = 3 dB, canal AWGN synchrone : (a) Utilisateur 1 et (b) Utilisateur 2. Les performances mono-utilisateur ont été ajoutées en lignes épaisses sur les deux figures.*

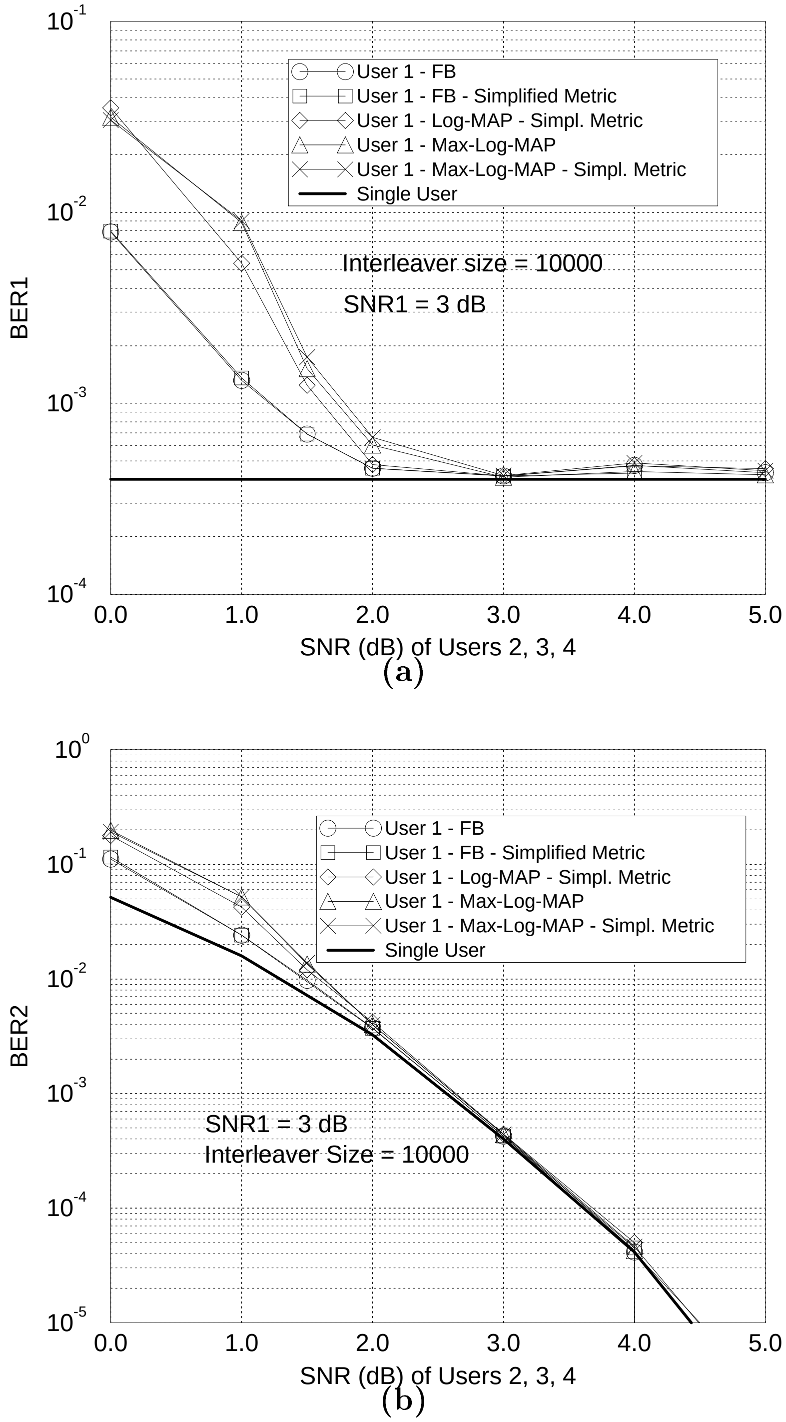


FIG. 3.19 – *Comparaison des différents algorithmes pour un code de taux 1/8, un entrelaceur de taille 10000 et 4 utilisateurs avec SNR_1 = 3 dB, canal AWGN synchrone : (a) Utilisateur 1 et (b) Utilisateur 2. Les performances mono-utilisateur ont été ajoutées en lignes épaisses sur les deux figures.*

fication de la métrique transformant la somme de $2^{K-1} = 8$ termes en une somme de $K = 4$ termes et divisant ainsi la complexité par un facteur 2. L'emploi de l'algorithme Log-MAP ou de l'algorithme Max-Log-MAP ne dégrade pas fortement les performances. Pour l'algorithme Log-MAP, nous avons utilisé une table de 8 valeurs pré-calculées de $f_c(x)$ pour x compris entre 0 et 3.

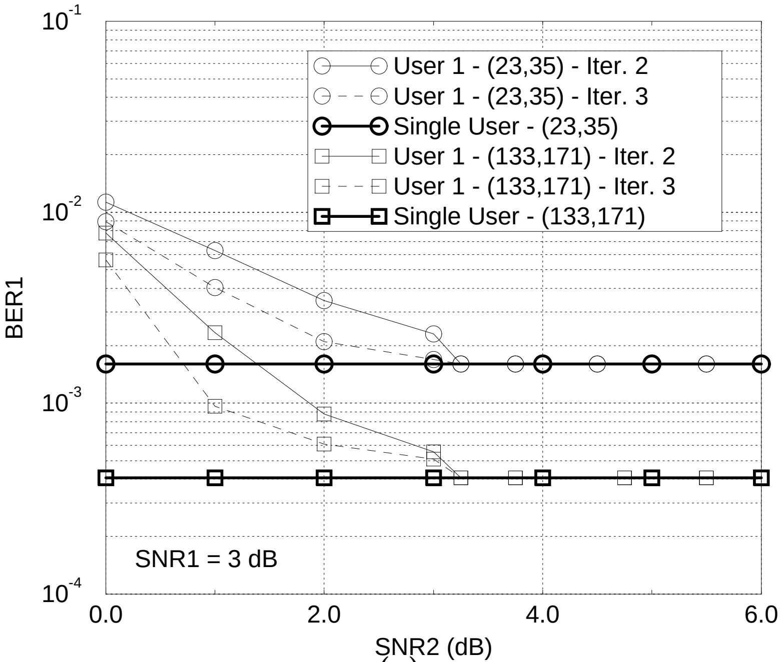
3.7.2 Système asynchrone

On considère finalement les effets de l'asynchronisme sur le même système à deux utilisateurs que pour la figure 3.16. Dans les simulations présentées sur la figure 3.20, l'utilisateur 2 a un retard de 2.5 périodes de bit codé ($5/2\ T_c$) par rapport à l'utilisateur 1. Les performances du système sont excellentes avec seulement 2 ou 3 itérations de décodage conjoint. Ce bon comportement est dû à l'asynchronisme, qui engendre une double diversité sur l'information a priori. En effet, chaque symbole interfère avec deux symboles de chacun des autres utilisateurs et la probabilité d'avoir une ambiguïté avec ces deux symboles et donc d'avoir une information a priori mauvaise sur ces deux symboles est beaucoup plus faible.

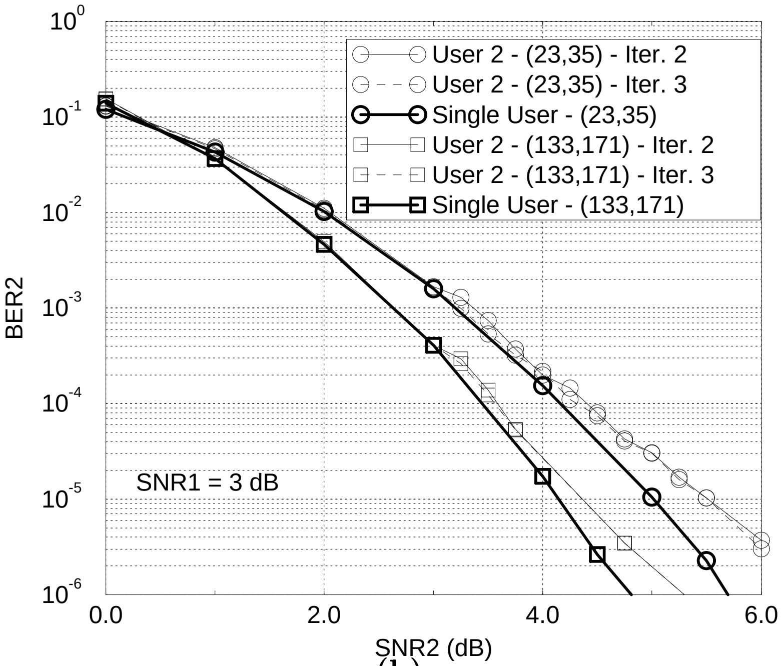
3.8 Problème des codes pseudo-aléatoires

L'utilisation de codes pseudo-aléatoires pour le système d'accès multiple proposé est séduisante. Le principe de décodage itératif permet à ces codes d'atteindre des performances excellentes, proches de la capacité. Ces codes pseudo-aléatoires sont construits à partir de codes convolutifs (turbo-codes [4]), d'équations de parité (codes LDPC [26]) ou de codes en blocs de petite taille (codes GLD [8][85]). Si l'on utilise un turbo-code, on doit en choisir un dont la longueur est suffisamment faible pour que les performances à faible rapport signal à bruit n'entravent pas la bonne marche du processus itératif d'élimination souple de l'interférence. La figure 3.16 montre clairement que, malgré ce choix, les performances du système d'accès multiple employant des turbo-codes restent mauvaises pour des utilisateurs ayant des puissances proches. Les codes LDPC et GLD, qui ont pourtant des performances meilleures à faible rapport signal à bruit, donnent des résultats tout aussi décevants. Il semblerait donc que l'APP fournie en sortie du décodeur itératif des codes pseudo-aléatoires soit de mauvaise qualité, c'est-à-dire que sa connaissance, bien qu'elle suffise pour trouver la bonne valeur du bit émis, donne par contre une mauvaise idée de sa fiabilité, compromettant ainsi le processus itératif d'élimination souple des interférences entre les utilisateurs.

Nous avons également essayé d'itérer entre les codes supérieurs des turbo-codes des différents utilisateurs, puis entre leurs codes inférieurs, au lieu de décoder de manière itérative classique le turbo-code d'un utilisateur avant d'utiliser les APPs fournies pour le décodage des autres utilisateurs. Cette méthode, qui avait pour but de tirer parti des



(a)



(b)

FIG. 3.20 – *Comparaison de codes de taux 1/2 pour 2 utilisateurs avec $SNR_1 = 3$ dB, canal AWGN asynchrone, délai relatif égal à $5/2\ T_c$: (a) Utilisateur 1 et (b) Utilisateur 2. Les performances mono-utilisateur ont été ajoutées en lignes épaisses sur les deux figures.*

bonnes performances en décodage multi-utilisateurs des codes convolutifs pris individuellement, n'a pas permis d'améliorer les résultats.

3.9 Conclusions

Ce chapitre montre que l'accès multiple à répartion par codes peut être implanté sans utilisation de signaux quasi-orthogonaux. L'orthogonalité entre les signaux est remplacée par l'indépendance entre les codes, garantie par l'entrelacement. En d'autres termes, l'orthogonalité est transférée de l'espace des signaux vers l'espace des probabilités. Le décodage conjoint itératif est capable d'éliminer complètement l'interférence entre les utilisateurs et ce d'autant mieux que les codes sont bien choisis. Un bon code pour notre système d'accès multiple est un code dont les performances sont satisfaisantes pour un faible rapport signal à bruit correspondant environ à un taux d'erreur par bit de 10^{-1} quand l'interférence est totalement inconnue. Pour des rapports signal à bruit faibles et modérés, les performances multi-utilisateurs sont d'autant plus proches des performances mono-utilisateur du code que la taille de l'entrelaceur est élevée.

Pour des raisons de simplicité, les résultats présentés dans ce chapitre se limitent au cas idéal d'un canal AWGN. Dans le cas d'un canal à évanouissements de Rayleigh parfaitement connus avec un seul trajet, on peut facilement appliquer le décodeur itératif en introduisant les coefficients d'évanouissement dans le signal reconstruit S_i^j des équations (3.3) et (3.5). Le décodage sur un canal de Rayleigh multi-trajets est également possible puisque les signaux provenant des différents trajets d'un même utilisateur peuvent être vus comme des signaux de différents utilisateurs traversant un canal de Rayleigh avec un seul trajet. L'utilisation de l'algorithme forward-backward avec métrique simplifiée est très utile dans ce dernier cas pour linéariser la complexité du système comme proposé dans la section 3.3.3.

Duan, Rimoldi et Urbanke [17] ont montré que pour des systèmes mono-utilisateur, la capacité du canal AWGN pouvait être atteinte en utilisant des constellations indépendantes multi-niveaux, des modulations BPSK codées par des turbo-codes par exemple. Cette propriété est également vérifiée pour un système d'accès multiple quand les utilisateurs ont des énergies distinctes. Cependant, dans le cas particulier où tous les utilisateurs ont la même puissance et le même taux, nous avons remarqué par simulation que les turbo-codes, les LPDC et les codes GLD ne permettaient pas d'atteindre les performances obtenues grâce aux codes convolutifs non seulement à cause de leurs performances parfois mauvaises à faible rapport signal à bruit mais aussi, peut-être, à cause de la mauvaise qualité des APPs que leur décodeur itératif fournit. A partir de cette constatation, Chayat et Shamai [12] ont construit un système d'accès multiple utilisant des turbo-codes et une détection itérative souple dans lequel on force les puissances des différents utilisateurs à des valeurs distinctes.

Après avoir étudié dans ce chapitre des systèmes à forte efficacité spectrale se passant de l'emploi de séquences pseudo-aléatoires, nous revenons maintenant au système CDMA classique par séquences directes. Nous allons tenter d'augmenter son efficacité spectrale en augmentant la taille des modulations utilisées et pour ce faire, nous commençons par décrire un détecteur multi-utilisateurs dont la complexité est indépendante du nombre d'utilisateurs.

Chapitre 4

Décodage par sphères*

4.1 Introduction

Un nouvel algorithme de détection conjointe de faible complexité pour les systèmes d'accès multiple par séquences directes (DS-SSMA pour *Direct Sequence Spread Spectrum Multiple Access*) est proposé dans ce chapitre. Cet algorithme est optimal au sens du maximum de vraisemblance (ML) pour les systèmes CDMA synchrones. Le récepteur modélise la sortie du désétalement comme le point d'un réseau multi-dimensionnel (empilement de sphères) perturbé par un bruit et applique un algorithme de décodage universel des réseaux de points pour détecter conjointement tous les utilisateurs. Dans le cas d'un système asynchrone, le décodage de réseaux de points est combiné à une élimination d'interférences et les performances du détecteur ainsi créé restent excellentes malgré sa sous-optimalité.

Le chapitre est organisé comme suit : la section 4.2 fait tout d'abord quelques rappels essentiels sur les réseaux de points puis la section 4.3 décrit la structure du système de transmission à accès multiple synchrone et montre sa représentation par un réseau de points. La section 4.4 explique l'algorithme de décodage par sphères (*Sphere Decoding*), qui est un décodeur ML de faible complexité pour des constellations de réseaux de points. Le décodage par sphères est ensuite appliqué à la détection ML dans un système DS-SSMA à la section 4.5. La section 4.6 présente la combinaison du décodage par sphère et de l'élimination d'interférences pour la démodulation conjointe dans un système DS-SSMA asynchrone et la section 4.7 décrit l'application de la même idée dans le cas d'un canal à trajets multiples. Dans la section 4.8, nous calculons une approximation analytique pour le gain du système en déterminant les paramètres du réseau. Les résultats des simulations pour des systèmes synchrones et asynchrones sur canal AWGN sont présentés dans la section 4.9 et comparés avec ceux du détecteur optimal de Verdú

* Le contenu de ce chapitre a été présenté au *Information Theory Workshop*, Kruger National Park, Afrique du Sud, Juin 1999, et soumis à *IEEE Transactions on Information Theory*, Mars 1999.

et ceux d'un détecteur hybride SIC-PIC présentés respectivement aux sections 2.4 et 2.7. La section 4.10 dresse finalement quelques conclusions.

4.2 Les réseaux de points

Avant d'expliquer comment un système multi-utilisateurs peut être vu comme un réseau de points, il est bon de faire quelques rappels sur ces derniers. Ils sont connus dans les mathématiques depuis plus d'un siècle et ont trouvé une application dans les communications numériques avec l'apparition des modulations codées au début des années 80. Les réseaux de points permettent de construire des modulations de grande taille dont l'énergie est minimale, ayant des propriétés géométriques permettant de minimiser la probabilité d'erreur lors de la transmission sur un canal donné.

Un réseau de points de dimension K est un empilement de sphères dans un espace à K dimensions. Les meilleurs empilements sont ceux qui maximisent le nombre de sphères dans un espace donné. Les sphères garantissent une certaine distance minimale entre les points du réseau, qui ne sont autres que leurs centres. Cette distance minimale fixe la borne supérieure du taux d'erreur par symbole obtenu quand on transmet les symboles sur un canal gaussien. Le fait de maximiser la densité de sphères dans un espace donné permet de maximiser l'efficacité spectrale pour une énergie moyenne des symboles fixée.

On note $\mathbb{R}$ le corps des réels et $\mathbb{Z}^K$ l'anneau des entiers relatifs. Un réseau de points de $\mathbb{R}^K$ est mathématiquement défini comme un sous-groupe discret de rang K de $\mathbb{R}^K$. $\mathbb{Z}^K$ est donc un exemple simple de réseau.

4.2.1 Définitions des paramètres

Tout point $\mathbf{x}$ d'un réseau Λ peut s'exprimer par la combinaison linéaire de K vecteurs de base $\{\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_K\}$:

$$\mathbf{x} = b_1\mathbf{v}_1 + b_2\mathbf{v}_2 + \ldots + b_K\mathbf{v}_K \quad \text{où } b_i \in \mathbb{Z},\ \forall i = 1,\ldots,K$$

Ces vecteurs de base forment les lignes de la *matrice génératrice* $\mathbf{G}$ du réseau. On peut donc écrire $\mathbf{x} = \mathbf{bG}$ où $\mathbf{b} = (b_1,\ldots,b_K) \in \mathbb{Z}^K$.

La région délimitée par les vecteurs de base est appelée *parallélotope fondamental* et son volume, noté $\mathrm{vol}(\Lambda)$ ou $\det(\Lambda)$, est nommé *volume fondamental.* Ce volume fondamental, qui n'est autre que le module du produit vectoriel des K vecteurs de base, est donc égal à $|\det(\mathbf{G})|$. S'il existe plusieurs choix possibles pour la matrice génératrice d'un même réseau, il n'existe par contre qu'une unique valeur pour le volume fondamental.

La *région de Voronoi* ou *cellule de Dirichlet* d'un point $\mathbf{x}$ d'un réseau Λ est l'ensemble

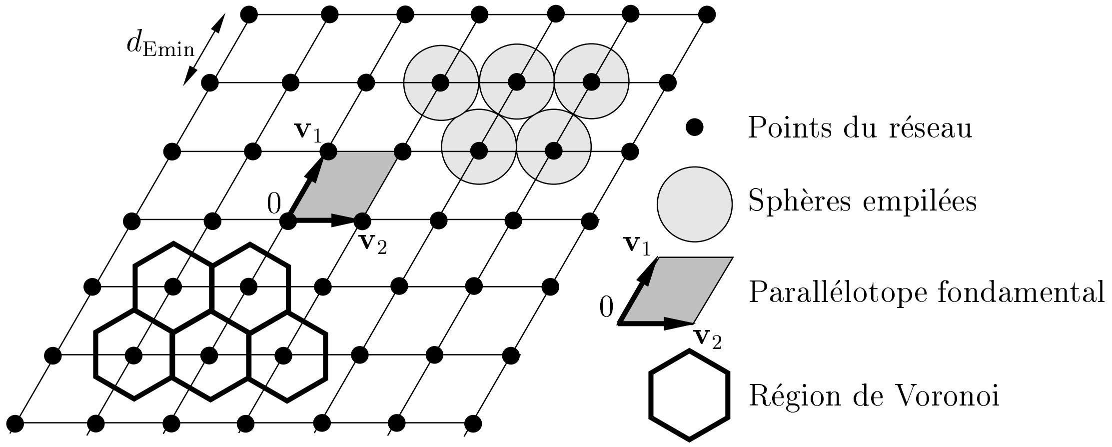


FIG. 4.1 – *Un réseau bidimensionnel et ses paramètres.*

des points de $\mathbb{R}^K$ plus proches de $\mathbf{x}$ que de tout autre point du réseau. Le volume de cette région est égal au volume fondamental.

Le *rayon d'empilement* ρ du réseau est le rayon de la plus grande sphère inscrite dans la région de Voronoi et le *rayon de recouvrement* celui de la plus petite sphère circonscrite à cette même région. Le rayon d'empilement est donc le rayon des sphères dont l'empilement constitue le réseau de points et le rayon de recouvrement est celui des sphères les plus petites qui, centrées sur les points du réseau, permettent de recouvrir tout l'espace $\mathbb{R}^K$. La *densité* du réseau est le rapport entre le volume de la sphère de rayon ρ et le volume fondamental. Enfin, le coefficient d'erreur (*kissing number*) $\tau(\Lambda)$ du réseau est le nombre de sphères tangentes à une même sphère dans l'empilement ou, en d'autres termes, le nombre de voisins d'un point du réseau, situés à la distance minimale $d_{\mathrm{Emin}} = 2\rho$.

La figure 4.1 représente le réseau hexagonal A_2, qui est le réseau le plus dense dans $\mathbb{R}^2$. Certains des paramètres définis ci-dessus y sont indiqués. Pour plus de détails et des définitions plus mathématiques, le lecteur pourra se reporter par exemple aux références [7] et [14].

4.2.2 Performances des réseaux de points sur canal gaussien

Nous nous intéressons maintenant à l'application des réseaux de points aux communications numériques, à savoir à la transmission de symboles d'un réseau Λ sur un canal AWGN.

Supposons que le point émis est le point $\mathbf{x}$ appartenant à une constellation S, c'est-à-dire à un sous-ensemble fini de Λ, d'énergie finie, de forme quelconque et centré sur l'origine pour minimiser l'énergie. Le décodeur à maximum de vraisemblance recherche le point du réseau le plus proche du point reçu $\mathbf{y} = \mathbf{x} + \boldsymbol{\eta}$. Le vecteur de bruit $\boldsymbol{\eta}$ a des composantes réelles indépendantes suivant une distribution gaussienne, de moyenne nulle et de variance σ^2. L'application de la borne de l'union au réseau donne une borne supérieure de la probabilité d'erreur par point :

$$P_e \lesssim \frac{\tau(\Lambda)}{2}\mathrm{erfc}\left(\frac{\rho}{\sigma\sqrt{2}}\right) \tag{4.1}$$

qui devient presque une égalité quand le bruit devient très faible.

Supposons que la constellation S est le cube $[-A,+A]^K$. Son volume est donc $(2A)^K$. Chaque point de la constellation occupe un volume égal au volume fondamental si on néglige les effets de bord. La constellation contient n points avec

$$n \simeq \frac{(2A)^K}{\det(\Lambda)}$$

Le nombre de bits par point est $m = \log_2(n)$. L'énergie moyenne d'un symbole de S est

$$E_p = \int\int\cdots\int_{[-A,+A]^K} \|\mathbf{x}\|^2 \frac{\mathrm{d}\mathbf{x}}{\mathrm{Vol}(S)} = \frac{KA^2}{3}$$

L'énergie moyenne par bit sur fréquence porteuse est donc

$$E_b = \frac{E_p}{2m} = \frac{KA^2}{6m} = \frac{K}{24m}\sqrt[K/2]{2^m\det(\Lambda)}$$

et le rapport signal à bruit en présence d'un bruit gaussien de variance $\sigma^2 = N_0$ s'écrit :

$$\frac{E_b}{N_0} = \frac{K}{24m\sigma^2}\sqrt[K/2]{2^m\det(\Lambda)} \tag{4.2}$$

Le remplacement de (4.2) dans (4.1) donne l'expression de la probabilité d'erreur par point d'une constellation cubique de n points d'un réseau Λ à K dimensions [6] :

$$P_e \approx \frac{\tau(\Lambda)}{2}\mathrm{erfc}\left(\sqrt{\frac{3m/K}{2^{2m/K}}\frac{E_b}{N_0}\frac{d^2_{\mathrm{Emin}}}{\sqrt[K/2]{\det(\Lambda)}}}\right) \tag{4.3}$$

Pour le réseau $\mathbb{Z}^K$, $d_{\mathrm{Emin}} = 1$, $\det(\Lambda) = 1$ et donc

$$P_e(\mathbb{Z}^K) \approx \frac{\tau(\Lambda)}{2}\mathrm{erfc}\left(\sqrt{\frac{3m/K}{2^{2m/K}}\frac{E_b}{N_0}}\right)$$

Le rapport énergétique

$$\gamma(\Lambda) = \frac{d^2_{\mathrm{Emin}}}{\sqrt[K/2]{\det(\Lambda)}} \tag{4.4}$$

est le *gain fondamental* du réseau Λ par rapport au réseau $\mathbb{Z}^K$ quand on fixe le nombre de bits par dimension. Ce gain ne dépend que des caractéristiques du réseau. A ce gain vient s'ajouter le *gain de forme* $\gamma_s(S)$, qui dépend de la constellation choisie. Il est égal au rapport entre l'énergie de la constellation et l'énergie d'une constellation cubique de même volume [25]. Les constellations qui maximisent le gain de forme sont les constellations sphériques. Le *gain total* d'une constellation est le produit du gain fondamental et du gain de forme.

Ces quelques définitions sur les réseaux de points à l'esprit, nous pouvons nous intéresser à leur application à la détection multi-utilisateurs.

4.3 Représentation en réseau d'un système synchrone

Considérons tout d'abord un système d'accès multiple avec étalement de spectre à séquence directe comportant K utilisateurs synchrones. Le symbole $b_k(i)$ de l'utilisateur k émis à l'instant i appartient à l'alphabet entier $\mathcal{A}$ de cardinalité $|\mathcal{A}|$. Chaque utilisateur k transmet un bloc de N symboles avec une amplitude du signal ω_k. Les symboles sont étalés par une signature réelle $s_k(t)$ de durée la période symbole T :

$$s_k(t) = 0 \quad \text{si} \quad t \notin [0,T[$$

Les K symboles transmis à l'instant i sont placés dans un vecteur ligne $\mathbf{b}(i)$ défini comme $\mathbf{b}(i) = (b_1(i),\ldots,b_K(i))$. Le signal modulé correspondant est, en fonction du temps t :

$$S_t = \sum_{i=0}^{N-1} \sum_{k=1}^{K} \omega_k b_k(i) s_k(t - iT)$$

Nous supposons que le canal est un canal idéal à bruit blanc additif gaussien. Soit $r_t = S_t + \eta_t$ le signal reçu au temps t et η_t un bruit gaussien réel de moyenne nulle et de variance N_0. Nous ne nous intéressons pour l'instant qu'au cas de signaux réels. La généralisation au cas complexe est présentée dans le chapitre suivant.

Soit $\mathbf{y}(i) = (y_1(i),\ldots,y_K(i))$ le vecteur ligne dans lequel $y_k(i)$ est la sortie à l'instant i du filtre adapté à l'utilisateur k :

$$\begin{aligned} y_k(i) &\triangleq \int_{-\infty}^{+\infty} s_k(t - iT) r_t \mathrm{d}t + n_k(i) \\ &= \sum_{\ell=1}^{K} \omega_\ell b_\ell(i) \int_0^T s_\ell(t) s_k(t) \mathrm{d}t + n_k(i) \end{aligned} \tag{4.5}$$

$\mathbf{y}(i)$ est une statistique suffisante pour la détection à maximum de vraisemblance (ML) de $\mathbf{b}(i)$. Les coefficients d'intercorrélation du vecteur de bruit $\mathbf{n}(i) = (n_1(i),\ldots,n_K(i))$

ont pour valeurs

$$E[n_\ell(i)n_k(i)] = R_{\ell k}N_0 \quad \text{avec} \quad R_{\ell k} = \int_0^T s_\ell(t)s_k(t)\mathrm{d}t \quad \text{pour} \quad k,\, \ell = 1,\ldots,K \tag{4.6}$$

Soit $\mathbf{D}_\omega$ la matrice diagonale $\mathrm{diag}(\omega_1,\ldots,\omega_K)$ formée par les amplitudes des différents utilisateurs et $\mathbf{R} = [R_{\ell k}]$ la matrice de taille $K \times K$ des intercorrélations entre les signatures. L'équation (4.5) devient alors

$$\mathbf{y}(i) = \mathbf{b}(i)\mathbf{M} + \mathbf{n}(i) \tag{4.7}$$

où la matrice $\mathbf{M}$ de taille $K \times K$ est définie par

$$\mathbf{M} = \mathbf{D}_\omega\mathbf{R}$$

Le vecteur $\mathbf{y}(i)$ dans l'équation (4.7) peut être vu comme un point du réseau Λ de dimension K et de matrice génératrice $\mathbf{M}$, corrompu par un bruit $\mathbf{n}(i)$. Si les signatures sont bien choisies et si toutes les amplitudes sont strictement positives, le réseau Λ est un $\mathbb{Z}$-module de rang K de l'espace réel $\mathbb{R}^K$ de dimension K. Les lignes de $\mathbf{M}$ forment une base de Λ. Le système à accès multiple génère un point $\mathbf{b}(i)\mathbf{M}$ appartenant à une constellation, c'est-à-dire à un sous-ensemble fini de Λ, de taille $|\mathcal{A}|^K$.

La représentation en réseaux des systèmes multi-utilisateurs nous permet d'utiliser un algorithme de décodage ML des réseaux de points très efficace, nommé le *décodage par sphères* [7][77][78]. Le décodeur par sphères est un algorithme universel capable de décoder n'importe quel réseau défini par une matrice génératrice $\mathbf{M}$ arbitraire. Il est directement inspiré d'un algorithme permettant de trouver le point du réseau le plus proche d'un point quelconque de l'espace (*closest vector problem*) [23].

4.4 Décodage par sphères avec un bruit blanc gaussien

Nous décrivons tout d'abord le décodage ML d'un réseau K-dimensionnel Λ quelconque utilisé sur un canal à bruit blanc additif gaussien et généré par une matrice réelle $\mathbf{G}$ de taille $K \times K$. Le décodeur doit trouver le point du réseau le plus proche du vecteur reçu, ce qui revient à minimiser la métrique

$$m(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{K} |y_i - x_i|^2 = \|\mathbf{y} - \mathbf{x}\|^2 \tag{4.8}$$

où $\mathbf{y} = \mathbf{x}+\boldsymbol{\eta}$ est le vecteur reçu, $\boldsymbol{\eta}= (\eta_1,\ldots,\eta_K)$ le vecteur de bruit et $\mathbf{x} = (x_1,\ldots,x_K)$ un point appartenant à Λ. Le vecteur de bruit $\boldsymbol{\eta}$ a des composantes réelles indépendantes suivant une distribution gaussienne de moyenne nulle et de variance σ^2. Les points du réseau $\{\mathbf{x} = \mathbf{b}\mathbf{G}\}$ sont obtenus à partir des vecteurs de données $\mathbf{b} = (b_1,\ldots,b_K)$ dans lesquels les composantes b_i appartiennent à l'anneau des entiers $\mathbb{Z}$. On note $\{\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_K\}$ les lignes de la matrice $\mathbf{G}$ formant une base.

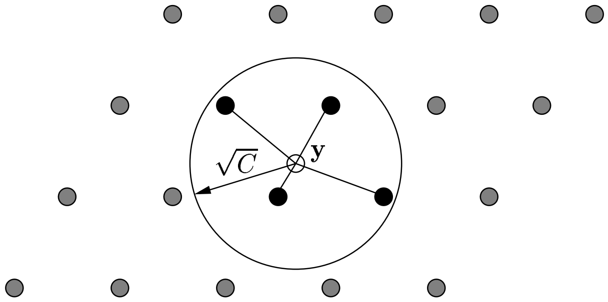


FIG. 4.2 – *Représentation géométrique de l'algorithme de décodage par sphères.*

En pratique l'ensemble des vecteurs de données est limité à un alphabet de taille finie $\mathcal{A}^K \subset \mathbb{Z}^K$ et un décodeur ML exhaustif recherche le meilleur point $\mathbf{x}$ dans la totalité de la constellation finie. Le décodeur par sphères restreint son calcul aux points qui sont situés à l'intérieur d'une sphère de rayon donné $\sqrt{C}$ centrée sur le point reçu comme on peut le voir sur la figure 4.2. Seuls les points du réseau situés à une distance quadratique inférieure à C du point reçu sont donc considérés pour la minimisation de la métrique (4.8). Le décodeur effectue la minimisation suivante :

$$\min_{\mathbf{x}\in\Lambda} \|\mathbf{y}-\mathbf{x}\| = \min_{\mathbf{w}\in\mathbf{y}-\Lambda} \|\mathbf{w}\| \tag{4.9}$$

L'égalité ci-dessus montre que nous devons trouver le vecteur $\mathbf{w}$ le plus court dans l'ensemble translaté $\mathbf{y}-\Lambda$. Nous écrivons le vecteur reçu $\mathbf{y}$ et la différence $\mathbf{w}$ comme

$$\begin{aligned} \mathbf{y} =& \boldsymbol{\rho}\mathbf{G} \qquad \text{avec} \qquad \boldsymbol{\rho} = (\rho_1,\ldots,\rho_K) \in \mathbb{R}^K \\ \mathbf{w} =& \boldsymbol{\xi}\mathbf{G} \qquad \text{avec} \qquad \boldsymbol{\xi} = (\xi_1,\ldots,\xi_K) \in \mathbb{R}^K \end{aligned}$$

Il faut bien noter que $\boldsymbol{\rho}$ et $\boldsymbol{\xi}$ sont des vecteurs réels. Comme $\mathbf{w} = \mathbf{y}-\mathbf{x}$ où $\mathbf{x}$ appartient au réseau Λ, on a la relation $\xi_i = \rho_i - u_i$ pour $i = 1,\ldots,K$ avec $\mathbf{w} = \sum_{i=1}^K \xi_i \mathbf{v}_i$. $\mathbf{w}$ est un point du réseau dont les coordonnées ξ_i sont exprimées dans le repère translaté centré sur le point reçu $\mathbf{y}$. On veut que $\mathbf{w}$ appartienne à une sphère de rayon quadratique C centrée en $\mathbf{y}$, c'est-à-dire en $\mathbf{0}$ dans le nouveau repère, donc

$$\|\mathbf{w}\|^2 = Q(\boldsymbol{\xi}) = \boldsymbol{\xi}\mathbf{G}\mathbf{G}^T\boldsymbol{\xi}^T \leq C \tag{4.10}$$

Dans le nouveau système de coordonnées défini par $\boldsymbol{\xi}$, la sphère de rayon quadratique C centrée en $\mathbf{y}$ est transformée en une ellipse centrée à l'origine. La factorisation de Cholesky [13] de la matrice de Gram $\boldsymbol{\Gamma} = \mathbf{G}\mathbf{G}^T$ donne $\boldsymbol{\Gamma} = \mathbf{A}\mathbf{A}^T$, où $\mathbf{A}$ est une matrice triangulaire inférieure d'éléments a_{ij}. On a alors

$$Q(\boldsymbol{\xi}) = \boldsymbol{\xi}\mathbf{A}\mathbf{A}^T\boldsymbol{\xi}^T = \|\mathbf{A}^T\boldsymbol{\xi}^T\|^2 = \sum_{i=1}^{K}\left(a_{ii}\xi_i + \sum_{j=i+1}^{K} a_{ji}\xi_j\right)^2 \leq C$$

En posant

$$\begin{aligned} q_{ii} &= a_{ii}^2 && \text{pour} \quad i = 1,\ldots,K \\ q_{ij} &= \frac{a_{ij}}{a_{ii}} && \text{pour} \quad j = 1,\ldots,K, \quad i = j+1,\ldots,K \end{aligned}$$

on obtient

$$Q(\boldsymbol{\xi}) = \sum_{i=1}^{K} q_{ii}\Big(\xi_i + \sum_{j=i+1}^{K} q_{ji}\xi_j\Big)^2$$

En s'intéressant tout d'abord à la plage de variations possibles de ξ_K, puis en rajoutant une à une les composantes, on obtient les K inégalités suivantes, qui définissent tous les points à l'intérieur de l'ellipse :

$$\begin{aligned} &q_{KK}{\xi_K}^2 \le C \\ &q_{K-1,K-1}\Big(\xi_{K-1} + q_{K,K-1}\xi_K\Big)^2 + q_{KK}{\xi_K}^2 \le C \\ &\forall 1 \le \ell \le K, \quad \sum_{i=l}^{K}\left(a_{ii}\xi_i + \sum_{j=i+1}^{K} a_{ji}\xi_j\right)^2 \le C \end{aligned} \tag{4.11}$$

Les bornes (4.11) nous permettent de montrer que les composantes entières de $\mathbf{b}$ doivent satisfaire les inégalités suivantes [77][78] :

$$\begin{aligned} \left\lceil -\sqrt{\frac{C}{q_{KK}}} + \rho_K \right\rceil \le \quad & b_K \quad \le \left\lfloor \sqrt{\frac{C}{q_{KK}}} + \rho_K \right\rfloor \\ \left\lceil -\sqrt{\frac{C - q_{KK}{\xi_K}^2}{q_{K-1,K-1}}} + \rho_{K-1} + q_{K,K-1}\xi_K \right\rceil \le \quad & b_{K-1} \quad \le \left\lfloor \sqrt{\frac{C - q_{KK}{\xi_K}^2}{q_{K-1,K-1}}} + \rho_{K-1} + q_{K,K-1}\xi_K \right\rfloor \end{aligned}$$

$$\begin{aligned} &\left\lceil -\sqrt{\frac{1}{q_{ii}}\left(C - \sum_{\ell=i+1}^{K} q_{\ell\ell}\left(\xi_\ell + \sum_{j=\ell+1}^{K} q_{j\ell}\xi_j\right)^2\right)} + \rho_i + \sum_{j=i+1}^{K} q_{ji}\xi_j \right\rceil \le b_i \\ &b_i \le \left\lfloor \sqrt{\frac{1}{q_{ii}}\left(C - \sum_{\ell=i+1}^{K} q_{\ell\ell}\left(\xi_\ell + \sum_{j=\ell+1}^{K} q_{j\ell}\xi_j\right)^2\right)} + \rho_i + \sum_{j=i+1}^{K} q_{ji}\xi_j \right\rfloor \end{aligned} \tag{4.12}$$

$\lceil x \rceil$ est le plus petit entier supérieur au réel x et $\lfloor x \rfloor$ est le plus grand entier inférieur au réel x. Les bornes inférieure et supérieure dans (4.12) nous apprennent que le décodeur par sphères possède K compteurs internes, à savoir un compteur par dimension. Il suffit ensuite de faire varier les valeurs des différents compteurs à l'intérieur des bornes sans oublier que les valeurs de celles-ci varient avec les valeurs des différents compteurs. En pratique ces bornes peuvent être mises à jour de manière récursive [78]. On énumère donc toutes les valeurs du vecteur $\mathbf{b}$ pour lesquelles le point correspondant du réseau

$\mathbf{x} = \mathbf{bG}$ est situé en deça de la distance quadratique C du point reçu. Les points du réseau situés en dehors de la sphère considérée ne sont jamais testés. Par conséquent le complexité de décodage ne dépend pas de la taille $|\mathcal{A}|^K$ de la constellation du réseau. De plus, la recherche à l'intérieur de la sphère peut être considérablement accélérée en mettant à jour le rayon $\sqrt{C}$ avec la dernière norme euclidienne calculée $\|\mathbf{w}\|$. Finalement, on sélectionne comme meilleur point $\mathbf{x}$ celui associé à la norme minimale $\|\mathbf{w}\|$.

Le rayon de recherche $\sqrt{C}$ doit être choisi de manière appropriée. En effet, le nombre de points du réseau situés à l'intérieur de la sphère de décodage augmente avec C. C'est pourquoi le choix d'une grande valeur de C ralentit l'algorithme, tandis que la sphère de recherche peut être vide si C est trop faible. De manière à être sûr que le décodeur par sphères trouve au moins un point du réseau, on doit choisir un rayon de recherche supérieur au rayon de recouvrement du réseau. On peut par exemple le choisir égal à la borne supérieure de Rogers [14] :

$$\sqrt{C}^K = (K \log K + K \log\log K + 5K) \times \frac{|\det(\mathbf{G})|}{V_K}$$

où V_K est le volume d'une sphère de rayon 1 dans l'espace réel $\mathbb{R}^K$.

4.5 Décodage par sphères d'un système synchrone

Les échantillons de bruit additif inclus dans le modèle du système caractérisé par l'équation (4.7) sont corrélés. L'équation (4.6) montre que cette corrélation est due aux intercorrélations non nulles entre les signatures des différents utilisateurs. Le décodeur ML du réseau doit minimiser la métrique suivante :

$$m'(\mathbf{y}(i)|\mathbf{x}(i)) = (\mathbf{y}(i) - \mathbf{x}(i))\mathbf{R}^{-1}(\mathbf{y}(i) - \mathbf{x}(i))^T \tag{4.13}$$

Les équations du décodeur par sphères peuvent facilement être adaptées à l'optimisation décrite ci-dessus. Cette optimisation est équivalente au décodage ML d'un réseau Λ' de matrice génératrice $\mathbf{M}$ en présence d'un bruit coloré $\mathbf{n}(i)$. Néanmoins, nous préférons blanchir le bruit en sortie du banc de filtres adaptés de manière à utiliser la même procédure de décodage que celle décrite à la section précédente. Il faut noter que toutes les études de performances des réseaux de points ont été menées dans le cas d'un bruit blanc additif gaussien. Le blanchissement du bruit va également nous aider à simplifier l'étude analytique de l'impact des paramètres du réseau sur le taux d'erreur du système CDMA présentée dans la section 4.8.

L'opération de blanchissement du bruit effectuée avant le décodage du réseau est similaire à celle qui est largement connue en théorie de l'égalisation. La factorisation de Cholesky de la matrice d'intercorrélation $\mathbf{R}$ donne

$$\mathbf{R} = \mathbf{W}\mathbf{W}^T$$

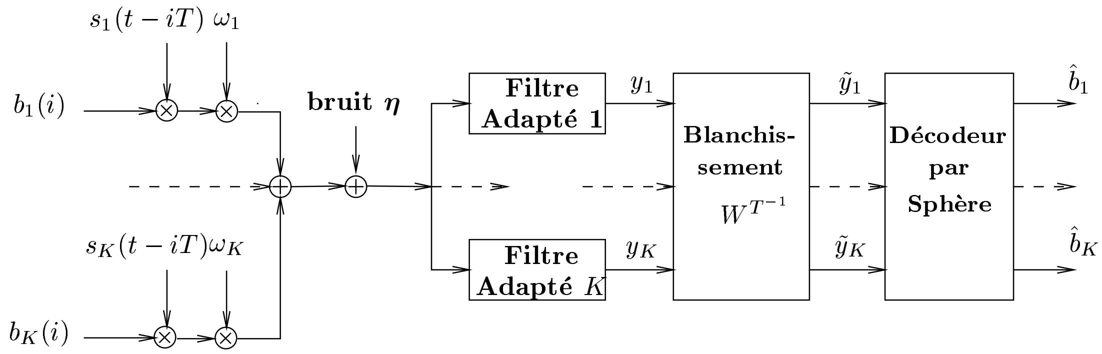


FIG. 4.3 – *Modèle d'un système CDMA avec détection conjointe du réseau associé.*

où $\mathbf{W}$ est une matrice triangulaire inférieure. L'équation (4.13) devient alors

$$m'(\mathbf{y}(i)|\mathbf{x}(i)) = (\mathbf{y}(i) - \mathbf{x}(i))\mathbf{W}^{T^{-1}}\mathbf{W}^{-1}(\mathbf{y}(i) - \mathbf{x}(i))^T \quad (4.14)$$

L'observation blanchie est définie comme

$$\tilde{\mathbf{y}}(i) = \mathbf{y}(i)\mathbf{W}^{T^{-1}}$$

et le nouveau point du réseau est donné par

$$\tilde{\mathbf{x}}(i) = \mathbf{x}(i)\mathbf{W}^{T^{-1}}$$

L'équation (4.14) est transformée en une métrique similaire à (4.8). On peut facilement montrer que la matrice de covariance du bruit filtré $\mathbf{n}(i) \times \mathbf{W}^{T^{-1}}$ est $N_0\mathbf{I}_K$, où $\mathbf{I}_K$ est la matrice identité de taille $K \times K$. Le modèle complet du système CDMA est illustré par la figure 4.3.

Nous écrivons maintenant la relation entre le point du réseau $\tilde{\mathbf{x}}(i)$ et le vecteur de données $\mathbf{b}(i)$:

$$\tilde{\mathbf{x}}(i) = \mathbf{x}(i)\mathbf{W}^{T^{-1}} = \mathbf{b}(i)\mathbf{M}\mathbf{W}^{T^{-1}} = \mathbf{b}(i)\mathbf{D}_\omega\mathbf{W} \quad (4.15)$$

L'équation (4.15) montre que l'opération de blanchissement se traduit par la formation d'un nouveau réseau de matrice génératrice $\mathbf{G} = \mathbf{D}_\omega\mathbf{W}$. C'est pourquoi le nouveau point reçu $\tilde{\mathbf{y}}(i)$ est traité par le décodeur par sphères associé à cette nouvelle matrice. Puisque $\mathbf{D}_\omega\mathbf{W}$ est déjà une matrice triangulaire inférieure, la factorisation de Cholesky précédant la recherche dans la sphère définie par les inégalités (4.12) peut être omise ($\mathbf{A} = \mathbf{G}$). De manière équivalente, on peut dire que la factorisation triangulaire a été transférée du décodeur vers l'opération de blanchissement du bruit.

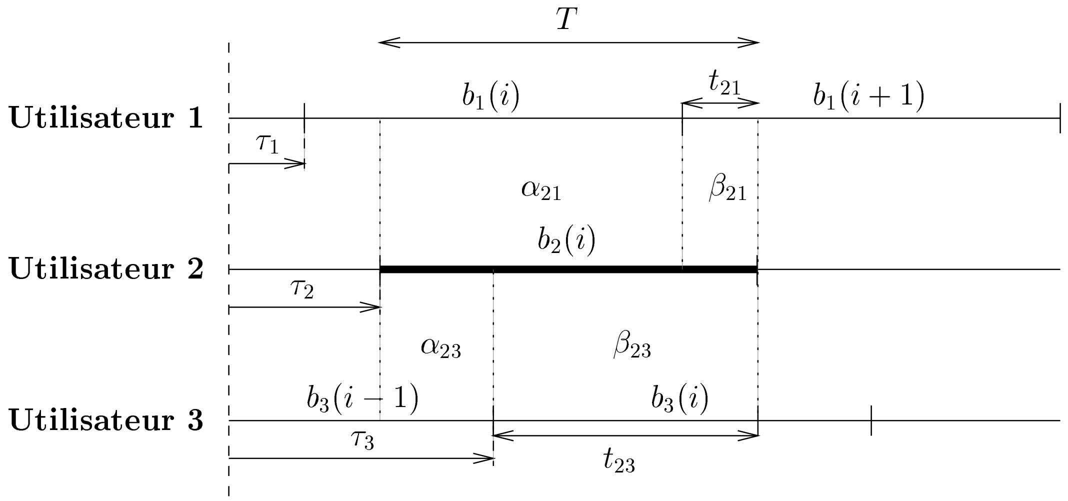


FIG. 4.4 – *Système d'accès multiple asynchrone avec 3 utilisateurs : interférences sur l'utilisateur 2.*

4.6 Décodage par sphères d'un système asynchrone

Considérons maintenant un système multi-utilisateurs asynchrone, où chaque utilisateur k a un retard τ_k avec $0 \leq \tau_1 \leq \tau_2 \leq \ldots \leq \tau_K < T$. La sortie du filtre adapté à l'utilisateur k à l'instant i est maintenant

$$\begin{aligned} y_k(i) &= \int_{-\infty}^{+\infty} s_k(t - iT - \tau_k) r_t \mathrm{d}t + n_k(i) \\ &= \sum_{\ell=1}^{K} \sum_{m=0}^{N-1} \omega_\ell b_\ell(m) \int_{-\infty}^{+\infty} s_k(t - iT - \tau_k) s_\ell(t - mT - \tau_\ell) \mathrm{d}t + n_k(i) \end{aligned} \quad (4.16)$$

Comme on peut le voir sur la figure 4.4, chaque symbole d'un utilisateur donné interfère avec un ou deux symboles des autres utilisateurs. Ces derniers symboles interfèrent également avec d'autres symboles et il est impossible de définir un réseau de dimension finie représentant exactement le système comme nous l'avons fait à la section 4.3. Pour résoudre ce problème, nous avons combiné le décodeur de réseau avec un système d'éliminations soustractives des interférences. La détection du symbole $b_k(i)$ prend en compte son désétalement complet, le désétalement partiel des symboles futurs des autres utilisateurs et les corrélations partielles avec les symboles passés de ces mêmes utilisateurs.

Le traitement conjoint des symboles $b_j(i)$ à l'instant i débute dès que la détection de tous les symboles $b_j(i-1)$, $j = 1 \ldots K$ est finie. La détection à l'instant i est effectuée en ordre croissant de k, c'est-à-dire que la détection de $b_k(i)$ utilise les symboles $b_1(i)$, $b_2(i)$, $\ldots$, $b_{k-1}(i)$ déjà détectés et les symboles précédents $b_{k+1}(i-1)$, $b_{k+2}(i-1)$, $\ldots$, $b_K(i-1)$. La procédure de détection pour un utilisateur k donné, à l'instant

i, dépend de trois vecteurs : les symboles passés $\mathbf{b_p}$, les symboles futurs $\mathbf{b_f}$ et le vecteur d'observation $\mathbf{y_f} = (y_{f1},\ldots,y_{fK})$. Les vecteurs de symboles sont :

$$\mathbf{b_p} = (b_{p1},\ldots,b_{pK}) = \Big(b_1(i)\ ,\ \ldots\ ,\ b_{k-1}(i)\ ,\ b_k(i-1)\ ,\ b_{k+1}(i-1)\ ,\ \ldots\ ,\ b_K(i-1) \Big)$$

$$\mathbf{b_f} = (b_{f1},\ldots,b_{fK}) = \Big(b_1(i+1)\ ,\ \ldots\ ,\ b_{k-1}(i+1)\ ,\ b_k(i)\ ,\ b_{k+1}(i)\ ,\ \ldots\ ,\ b_K(i) \Big)$$

Quand on décode le symbole $b_k(i)$, l'observation $y_{f\ell}$ associée à $b_{f\ell}$ est le résultat d'un désétalement partiel de durée $t_{k\ell}$, débutant avec le symbole $b_{f\ell}$ et s'achevant avec le symbole $b_k(i)$. On a donc :

$$\begin{array}{rcll} t_{k\ell} & = & \tau_k - \tau_\ell & \text{pour} \quad \ell < k \\ t_{k\ell} & = & T + \tau_k - \tau_\ell & \text{pour} \quad \ell \geq k \end{array}$$

Soit $\beta_{j\ell}$ l'intercorrélation entre les symboles b_{fj} et $b_{f\ell}$.

– Pour $\ell < k$:

$$\begin{array}{rcll} \beta_{j\ell} & = & \displaystyle\int_0^{\tau_k-\tau_\ell} s_j(t+\tau_\ell-\tau_j)s_\ell(t)\mathrm{d}t & \text{pour} \quad j \leq \ell \\ \beta_{j\ell} & = & \displaystyle\int_0^{\tau_k-\tau_j} s_j(t)s_\ell(t+\tau_j-\tau_\ell)\mathrm{d}t & \text{pour} \quad \ell < j < k \\ \beta_{j\ell} & = & \displaystyle\int_0^{\tau_k-\tau_\ell} s_j(t+\tau_\ell-\tau_j+T)s_\ell(t)\mathrm{d}t & \text{pour} \quad j \geq k \end{array}$$

– Pour $\ell \geq k$:

$$\begin{array}{rcll} \beta_{j\ell} & = & \displaystyle\int_0^{\tau_k-\tau_j} s_j(t)s_\ell(t+\tau_j-\tau_\ell+T)\mathrm{d}t & \text{pour} \quad j < k \\ \beta_{j\ell} & = & \displaystyle\int_0^{\tau_k-\tau_\ell+T} s_j(t+\tau_\ell-\tau_j)s_\ell(t)\mathrm{d}t & \text{pour} \quad k \leq j \leq \ell \\ \beta_{j\ell} & = & \displaystyle\int_0^{\tau_k-\tau_j+T} s_j(t)s_\ell(t+\tau_j-\tau_\ell)\mathrm{d}t & \text{pour} \quad \ell < j \end{array}$$

Soit $\alpha_{j\ell}$ l'intercorrélation entre le symbole $b_j(i)$ et le symbole précédemment détecté de l'utilisateur ℓ.

$$\begin{array}{rcll} \alpha_{j\ell} & = & \displaystyle\int_{-\infty}^{+\infty} s_j(t-\tau_j-T)s_\ell(t-\tau_\ell)\mathrm{d}t & \text{pour} \quad j < \ell \\ \alpha_{j\ell} & = & 0 & \text{pour} \quad j = \ell \\ \alpha_{j\ell} & = & \displaystyle\int_{-\infty}^{+\infty} s_j(t-\tau_j)s_\ell(t-\tau_\ell)\mathrm{d}t & \text{pour} \quad j > \ell \end{array}$$

A l'aide de (4.16) nous pouvons maintenant exprimer le vecteur d'observations $\mathbf{y_f} = (y_{f1},\ldots,y_{fK})$ associé à la détection de $b_k(i)$ comme

$$y_{fj} \;=\; \omega_j\beta_{jj}b_{fj} + \sum_{j<\ell<k}\omega_\ell\alpha_{j\ell}b_{p\ell} + \sum_{\ell\neq j}\omega_\ell\beta_{j\ell}b_{f\ell} + n_j \qquad \text{pour} \quad j < k$$

$$
\begin{aligned}
y_{fj} &= \omega_j \beta_{jj} b_{fj} + \sum_{\ell \neq j} \omega_\ell \alpha_{j\ell} b_{p\ell} + \sum_{\ell \neq j} \omega_\ell \beta_{j\ell} b_{f\ell} + n_j && \text{pour} \quad j = k \\
y_{fj} &= \omega_j \beta_{jj} b_{fj} + \sum_{\ell < k} \omega_\ell \alpha_{j\ell} b_{p\ell} + \sum_{\ell > j} \omega_\ell \alpha_{j\ell} b_{p\ell} + \sum_{\ell \neq j} \omega_\ell \beta_{j\ell} b_{f\ell} + n_j && \text{pour} \quad j > k
\end{aligned}
\tag{4.17}
$$

Les équations (4.17) peuvent être simplement écrites sous forme matricielle :

$$
\mathbf{y_f} = \mathbf{b_p} \mathbf{D}_\omega \mathbf{R_p} + \mathbf{b_f} \mathbf{D}_\omega \mathbf{R_f} + \mathbf{n} \tag{4.18}
$$

où $\mathbf{R_f} = [\beta_{ij}]$, $\mathbf{n}$ est un bruit additif gaussien de matrice de covariance $N_0 \mathbf{R_f}$ et

$$
\mathbf{R_p} = \begin{bmatrix}
0 & 0 & 0 & \cdots & 0 & 0 & \alpha_{1,k} & \alpha_{1,k+1} & & \cdots & \alpha_{1,K} \\
\alpha_{2,1} & 0 & 0 & \cdots & 0 & 0 & \alpha_{2,k} & \alpha_{2,k+1} & & \cdots & \alpha_{2,K} \\
\alpha_{3,1} & \alpha_{3,2} & 0 & \cdots & 0 & 0 & & & & & \\
\vdots & & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & & & \vdots \\
\alpha_{k-2,1} & \alpha_{k-2,2} & \cdots & \alpha_{k-2,k-3} & 0 & 0 & \alpha_{k-2,k} & \alpha_{k-2,k+1} & & \cdots & \alpha_{k-2,K} \\
\alpha_{k-1,1} & \alpha_{k-1,2} & \cdots & \alpha_{k-1,k-3} & \alpha_{k-1,k-2} & 0 & \alpha_{k-1,k} & \alpha_{k-1,k+1} & \cdots & & \alpha_{k-1,K} \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & & \cdots & & 0 & 0 & \alpha_{k+1,k} & 0 & 0 & \cdots & 0 \\
0 & & & & 0 & 0 & \alpha_{k+2,k} & \alpha_{k+2,k+1} & 0 & & 0 \\
\vdots & & & & \vdots & \vdots & \vdots & & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & \alpha_{K-1,k} & \alpha_{K-1,k+1} & \cdots & \alpha_{K,K-1} & 0
\end{bmatrix}
$$

Il existe K différentes paires de matrices $\mathbf{R_p}$ et $\mathbf{R_f}$, chacune étant utilisée pour la détection d'un utilisateur. Les symboles inclus dans $\mathbf{b_p}$ sont déjà détectés, on peut donc soustraire l'interférence passée $\mathbf{b_p} \mathbf{D}_\omega \mathbf{R_p}$ de l'observation $\mathbf{y_f}$ pour obtenir une nouvelle observation $\mathbf{z_f}$ que l'on fournit ensuite au décodeur de réseau,

$$
\mathbf{z_f} = \mathbf{y_f} - \mathbf{b_p} \mathbf{D}_\omega \mathbf{R_p} = \mathbf{b_f} \mathbf{D}_\omega \mathbf{R_f} + \mathbf{n}
$$

Nous aurions bien sûr pu retirer l'interférence du signal reçu et désétaler ensuite comme on fait habituellement dans les détecteurs par soustraction d'interférence mais nous avons préféré garder une écriture matricielle opérant directement sur le point reçu $\mathbf{y}$ au prix d'écritures plus complexes. Le vecteur $\mathbf{z_f}$ ci-dessus est un point de réseau corrompu par un bruit coloré. On peut donc appliquer les résultats de la section 4.5 pour détecter $b_k(i)$ en utilisant un décodeur par sphères dans l'espace réel K-dimensionnel. Il faut souligner que K étapes de décodage du réseau sont requises pour démoduler les symboles i des K utilisateurs si les utilisateurs ont tous des délais différents, alors qu'une seule étape de décodage suffisait pour décoder conjointement tous les utilisateurs dans le système synchrone. Si deux utilisateurs ou plus ont le même délai on peut évidemment les décoder en une seule étape de décodage.

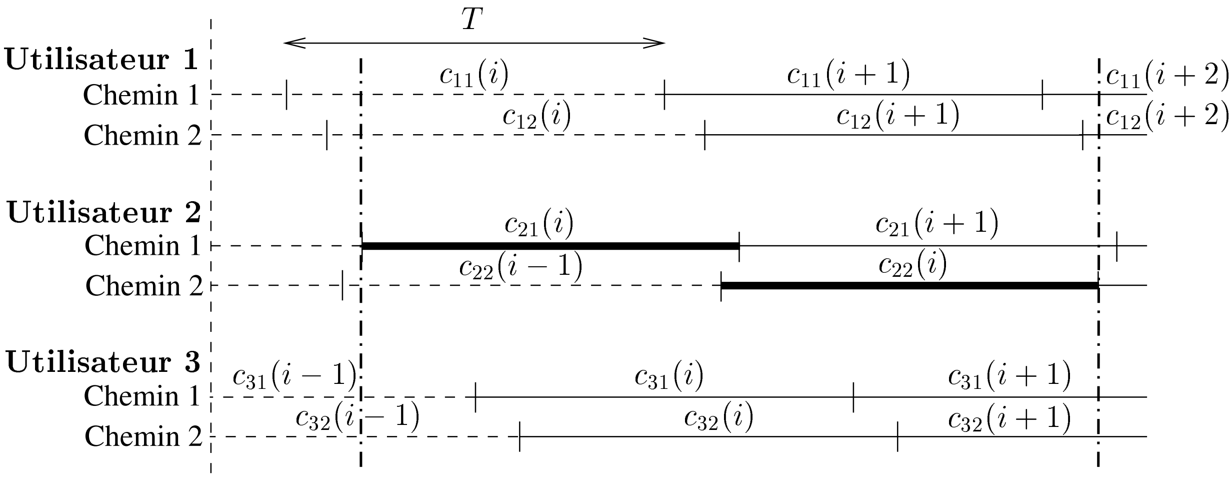


FIG. 4.5 – *Système d'accès multiple avec 3 utilisateurs sur un canal à 2 trajets : interférences sur l'utilisateur 2.*

4.7 Décodage par sphères sur canal multi-trajets

Considérons maintenant un système avec K utilisateurs asynchrones traversant chacun un canal multi-trajets différent. Chaque utilisateur k a un retard τ_k et le canal qu'il traverse est composé de P_k chemins, chaque chemin p ayant un retard ξ_{kp} et un coefficient $c_{kp}(i)$ à l'instant i, que nous supposons réel pour simplifier les notations. On a toujours la condition $0 \leq \tau_1 \leq \tau_2 \leq \ldots \leq \tau_K < T$ et on suppose également que la dispersion temporelle du canal est strictement inférieure au temps symbole ($0 \leq \xi_{kp} < T$, $\forall\ k = 1,\ldots,K$, $\forall\ p = 0,\ldots,P_k - 1$). On note τ_{kp} le retard total du chemin p de l'utilisateur k :

$$\tau_{kp} = \tau_k + \xi_{kp}$$

Le signal r_t reçu au temps t est la somme des signaux des K utilisateurs après passage dans le canal multi-trajets qui les affecte et d'un bruit blanc additif gaussien η_t.

$$r_t = S_t + \eta_t = \sum_{k=1}^{K} \sum_{i=0}^{N-1} \sum_{p=0}^{P_k-1} \omega_k c_{kp}(i) b_k(i) s_k(t - iT - \tau_{kp}) + \eta_t \tag{4.19}$$

Le filtre adapté à l'utilisateur k n'est plus un simple désétalement comme aux sections 4.5 et 4.6. Il faut maintenant faire le filtrage adapté de chaque chemin de l'utilisateur k. L'observation $y_{kj}(i)$ à l'instant i correspondant au chemin j de l'utilisateur k est

$$y_{kj}(i) = \int_{-\infty}^{+\infty} s_k(t - iT - \tau_{kj}) r_t \mathrm{d}t \tag{4.20}$$

Les observations sur les différents chemins d'un même utilisateur sont ensuite combinées en tenant compte du coefficient de chaque trajet. On appelle cette opération *Maximum Ratio Combining* (MRC). L'ensemble formé par les différents désétalements et la somme pondérée est appelé récepteur en râteau (*RAKE receiver*) [51] [52]. Il effectue en fait le filtrage adapté au signal de l'utilisateur k affecté par son canal. Sa sortie, qui tient compte des différents trajets et est donc moins sensible à un fort évanouissement sur l'un d'entre eux, bénéficie d'une diversité d'autant plus grande que le nombre de chemins dans le canal est grand. Cette observation s'exprime ainsi :

$$y_k(i) = \sum_{j=0}^{P_k-1} c_{kj}(i) y_{kj}(i) \tag{4.21}$$

En insérant l'expression (4.19) dans l'équation (4.20), on obtient la somme du signal utile, de l'interférence et du bruit.

$$y_{kj}(i) = \sum_{\ell=1}^{K} \sum_{m=0}^{N-1} \sum_{p=0}^{P_k-1} \omega_k c_{\ell p}(m) b_\ell(m) \int_{-\infty}^{+\infty} s_k(t-iT-\tau_{kj}) s_\ell(t-mT-\tau_{\ell p}) \mathrm{d}t + n_{kj}(i) \tag{4.22}$$

L'observation pour l'utilisateur k à l'instant i exprimée par (4.21) devient donc

$$y_k(i) = \sum_{j=0}^{P_k-1} \sum_{\ell=1}^{K} \sum_{m=0}^{N-1} \sum_{p=0}^{P_k-1} \omega_k c_{kj}(i) c_{\ell p}(m) b_\ell(m) \int_{-\infty}^{+\infty} s_k(t-iT-\tau_{kj}) s_\ell(t-mT-\tau_{\ell p}) \mathrm{d}t + n_k(i)$$

Comme on peut le voir sur la figure 4.5, chaque sortie du RAKE d'un utilisateur donné interfère avec 2 sorties (respectivement une sortie) déjà détectées et 2 sorties (respectivement une sortie) non encore détectées des autres utilisateurs (respectivement du même utilisateur). On peut procéder de la même manière qu'à la section 4.6. On détecte séparément chaque utilisateur à l'aide d'un détecteur par sphère associé à une élimination d'interférence. La figure 4.5 illustre la détection du symbole i de l'utilisateur 2 dans un système à 3 utilisateurs traversant des canaux avec 2 trajets. Une sortie de filtre adapté rassemble les interférences sur les 2 chemins. Les sorties concernées par la détection sont celles dont une partie se trouve entre les deux traits mixtes. On supprime l'interférence des symboles déjà détectés, qui sont représentés en traits interrompus et on considère toutes les autres sorties du RAKE concernées par cette détection comme les coordonnées d'un point d'un réseau de dimension $2K$ affecté par un bruit additif. La dimension du réseau est donc doublée par rapport au cas asynchrone avec un seul trajet. La variation des coefficients des canaux en fonction du temps est un autre facteur de complication. Il faut recalculer toutes les matrices dès que l'un d'entre eux change.

4.8 Performances analytiques

Nous calculons maintenant une borne analytique pour le gain du système en étudiant la structure de la constellation du réseau associé. Pour des raisons de simplicité, nous supposons que tous les utilisateurs sont synchrones et que le milieu d'accès multiple est un canal à bruit blanc additif gaussien idéal. D'après (4.3), la probabilité d'erreur par point P_e d'une constellation cubique S est approximativement

$$P_e \approx \frac{\tau(\Lambda)}{2}\mathrm{erfc}\left(\sqrt{\frac{3\zeta}{2^{\zeta+1}}\frac{E_b}{N_0}\gamma(\Lambda)}\right)$$

où $\tau(\Lambda)$ est la population de la première couche, erfc est la fonction d'erreur complémentaire, ζ est le nombre de bits par deux dimensions, E_b est l'énergie moyenne par bit sur fréquence porteuse. Le gain fondamental $\gamma(\Lambda)$ est donné par (4.4)

$$\gamma(\Lambda) = \frac{d^2_{\mathrm{Emin}}}{\mathrm{vol}(\Lambda)^{2/K}} \tag{4.23}$$

pour un réseau K-dimensionnel avec une distance euclidienne minimale d_{Emin} et un volume fondamental $\mathrm{vol}(\Lambda)$. Si $\mathbf{G}$ est la matrice génératrice de Λ,

$$\mathrm{vol}(\Lambda) = |\det(\mathbf{G})|$$

Le rapport énergétique $\gamma(\Lambda)$ représente le gain de Λ quand le réseau entier $\mathbb{Z}^K$ est pris comme référence.

$$\gamma(\mathbb{Z}^K) = 1$$

$\gamma(\Lambda)$ ne dépend que de la structure du réseau. Quand la constellation S n'a pas une forme cubique, le gain total $\gamma(S)$ est égal au produit du gain fondamental et du gain de forme $\gamma_s(S)$, dans lequel le dernier terme dépend du moment de second ordre de la constellation [25],

$$\gamma(S) = \gamma(\Lambda) \times \gamma_s(S)$$

Soit $\overline{\|\mathbf{b}\|^2}_{\mathrm{cube}}$ le moment de second ordre de la constellation entière S_{cube} obtenue par concaténation des symboles des K utilisateurs. Soit $\overline{\|\tilde{\mathbf{x}}\|^2}_S$ le moment de second ordre de la constellation S. Nous supposons que S et S_{cube} ont le même volume, c'est-à-dire que le volume fondamental du réseau est normalisé. L'équation (4.15) devient alors

$$\tilde{\mathbf{x}} = \mathbf{bG}/\sqrt[K]{|\det(\mathbf{G})|}$$

et donne

$$\overline{\|\tilde{\mathbf{x}}\|^2}_S = K\mathrm{E}\left[\tilde{x}_i^2\right] = \mathrm{E}\left[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\right] = \frac{1}{\sqrt[K/2]{|\det(\mathbf{G})|}}\mathrm{E}\left[\mathbf{bGG}^T\mathbf{b}^T\right] \tag{4.24}$$

$$\mathrm{E}\left[\mathbf{bGG}^T\mathbf{b}^T\right] = \mathrm{E}\left[\mathbf{b\Gamma b}^T\right] = \mathrm{E}\left[\sum_{i,j} b_i b_j \Gamma_{ij}\right] = \sum_i \Gamma_{ii}\mathrm{E}\left[{b_i}^2\right] = \frac{\mathrm{Trace}(\mathbf{\Gamma})}{K}\overline{\|\mathbf{b}\|^2}_{\mathrm{cube}} \tag{4.25}$$

En combinant les équations (4.24) et (4.25) on obtient la formule du gain de forme

$$\gamma_s(S) = \frac{\overline{\|\mathbf{b}\|^2}_{\mathrm{cube}}}{\overline{\|\tilde{\mathbf{x}}\|^2}_S} = \frac{K \cdot \sqrt[K/2]{\det(\mathbf{G})}}{\mathrm{Trace}(\mathbf{\Gamma})} \tag{4.26}$$

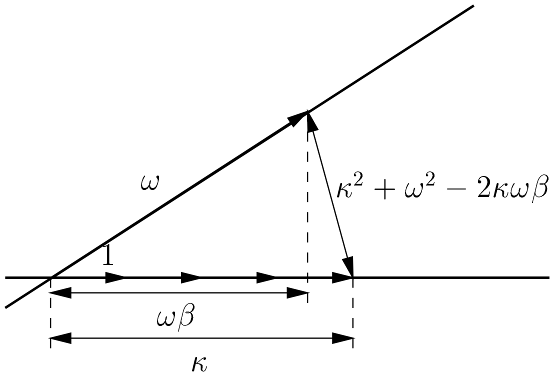


FIG. 4.6 – *Détermination géométrique de la distance minimale.*

Nous étudions maintenant le cas simple d'un système avec $K = 2$ utilisateurs synchrones. Nous supposons que l'utilisateur 1 a une amplitude unité et que l'utilisateur 2 a une amplitude $\omega \geq 1$. Le coefficient d'intercorrélation est noté $\beta \in [0,1]$. La matrice d'intercorrélations $\mathbf{R}$ et la matrice génératrice du réseau associé sont alors

$$\mathbf{R} = \begin{bmatrix} 1 & \beta \\ \beta & 1 \end{bmatrix} \quad \text{et} \quad \mathbf{G} = \mathbf{D}_\omega \mathbf{W} = \begin{bmatrix} 1 & 0 \\ \omega\beta & \omega\sqrt{1-\beta^2} \end{bmatrix} \tag{4.27}$$

Les performances du système CDMA sont comparées à celles d'un système de référence défini par une constellation S_o. Cette constellation de référence a une forme cubique et correspond au cas idéal de deux signatures orthogonales ($\beta = 0$, $d^2_{\mathrm{Emin}} = 1$). Nous avons

$$\begin{aligned} \gamma(\Lambda_o) &= \omega^{-1} \\ \gamma_s(S_o) &= \frac{2\omega}{1+\omega^2} \\ \gamma(S_o) &= \frac{2}{1+\omega^2} \end{aligned} \tag{4.28}$$

Finalement, le gain total $\gamma'(S)$ du système CDMA est défini comme le rapport de $\gamma(S)$ sur $\gamma(S_o)$,

$$\gamma'(S) = \frac{d^2_{\mathrm{Emin}}\,(1+\omega^2)}{\mathrm{Trace}(\mathbf{\Gamma})} \tag{4.29}$$

La distance quadratique minimimale du réseau, d^2_{Emin}, peut être déterminée géométriquement comme on peut le voir sur la figure 4.6. Dans la plupart des cas, la distance minimale est soit 1, soit la distance la plus courte séparant le vecteur de base de plus grand module ω d'un multiple du vecteur de plus petit module :

$$d^2 = \min\left(1,\kappa^2 + \omega^2 - 2\kappa\omega\beta\right)$$

où κ est l'entier le plus proche de $\omega\beta$. Il faut souligner que dans certains cas exceptionnels, par exemple pour des valeurs de β proches de 1.0, la distance d peut différer de la véritable distance minimale. Cette distance d peut également être obtenue par l'algorithme de réduction de base de Gauss [13, p. 23]. Elle est fournie par la première étape de cet algorithme. Dans les cas exceptionnels cités ci-dessus, la condition d'arrêt de l'algorithme n'est pas vérifiée et on peut envisager de poursuivre son exécution pour obtenir la vraie valeur de d^2_{Emin}. Cependant, dans tous les cas intéressants en pratique, $d^2_{\text{Emin}} = d^2$.

A partir de la définition (4.27) de $\mathbf{G}$, nous pouvons écrire

$$\text{Trace}(\mathbf{\Gamma}) = \text{Trace}(\mathbf{G}\mathbf{G}^T) = 1 + \omega^2 \tag{4.30}$$

Au moyen des équations (4.29) et(4.30), nous obtenons une expression simple du gain total du système d'accès multiple :

$$\gamma'(S) = d^2 \tag{4.31}$$

Par conséquent, tant que $d^2 = 1$, le système ne subit aucune dégradation des performances globales. En d'autres termes, la détection conjointe affiche une perte nulle pour des valeurs faibles et moyennes du coefficient de corrélation.

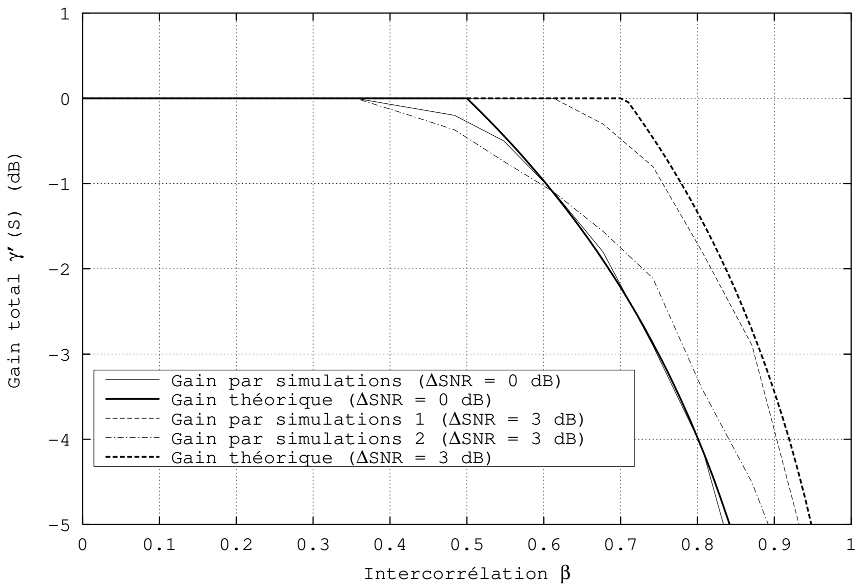


FIG. 4.7 – *Gain d'un système synchrone : 2 utilisateurs, modulation 16-PAM.*

Nous avons tracé le gain global théorique donné par l'équation (4.31) pour 2 utilisateurs avec une différence de rapports signal à bruit ΔSNR égale à 0 dB et 3 dB. Comme la théorie de l'information le prévoit, le gain est d'autant plus fort que la différence ΔSNR est élevée ! L'utilisateur le plus puissant a un effet négligeable sur le taux d'erreurs global. Le gain global est donc en grande partie lié à l'utilisateur le plus faible. Ce dernier est moins sensible aux variations d'intercorrélation puisque son taux d'erreur est plus fort. Cette définition du gain global donne donc de meilleurs résultats que celle qui consisterait à calculer le gain de chaque utilisateur et à moyenner sur tous les utilisateurs. Cependant l'étude du réseau ne permet pas d'accéder au gain de chaque utilisateur mais seulement à un gain global. Le gain théorique est comparé à celui calculé par simulations informatiques. La remarque sur la définition du gain global ne s'applique pas au cas $\Delta SNR = 0$ dB. Par contre, dans le cas $\Delta SNR = 3$ dB, deux courbes de gain par simulations sont tracées. Le gain par simulations 1 est le gain global tel qu'on l'a considéré dans le calcul théorique et c'est donc à cette courbe qui faut comparer la courbe du gain théorique. Cependant, pour donner une idée de la différence entre les deux définitions du gain global, nous avons tracé le gain par simulations 2, qui est la moyenne des gains de chaque utilisateur. Les simulations ont été effectuées avec une modulation d'amplitude en phase à 16 symboles (MAP-16 ou 16-PAM pour *Phase Amplitude Modulation*) pour que la constellation du réseau équivalent ait suffisamment de points.

4.9 Résultats de simulations

Nous avons appliqué l'algorithme de décodage par sphères à la détection conjointe de 4 utilisateurs dans un système d'accès multiple par étalement de spectre à séquence directe. Les signatures utilisées sont des séquences de Gold de période 7. Le premier utilisateur a une puissance fixe. Les utilisateurs 2, 3 et 4 ont des puissances de transmission égales et nous faisons varier leur rapport signal à bruit de manière à observer les effets de l'éblouissement sur le premier utilisateur. Les résultats sont comparés à ceux d'un détecteur hybride SIC-PIC, identique à celui présenté dans la section 2.7, avec éliminations dures sur 3 itérations.

Nous avons d'abord considéré une transmission synchrone avec une modulation BPSK sur un canal gaussien. La figure 4.8 montre les performances ML du décodeur par sphères. Il est très résistant à l'éblouissement comparé au détecteur SIC-PIC. Les performances des différents utilisateurs sont similaires contrairement à celles du SIC-PIC, qui dépendent des valeurs des intercorrélations. Pour l'utilisateur 4, nous observons un gain de 5.5 dB pour le décodeur par sphères sur le détecteur SIC-PIC. Sur la figure 4.9, avec une modulation d'amplitude en phase (16-PAM), le décodeur par sphères maintient ses performances avec une complexité similaire tandis que le détecteur SIC-PIC ne fonctionne plus. Un détecteur ML exhaustif aurait à calculer $16^4 = 65536$ métriques pour détecter chaque point !

Considérons maintenant un système d'accès multiple asynchrone. Les retards des utilisateurs 1, 2, 3 et 4 sont respectivement 0, 2, 4 et 6 périodes chip. Sur la figure 4.10, on note une dégradation due à la combinaison entre le décodage par sphères et l'élimination d'interférences. En effet, notre détecteur n'est plus optimal selon le critère de maximum de vraisemblance. La perte est de 5.5 dB pour l'utilisateur 2 quand on le compare à l'algorithme ML de Verdú. La perte en rapport signal à bruit est cependant encore

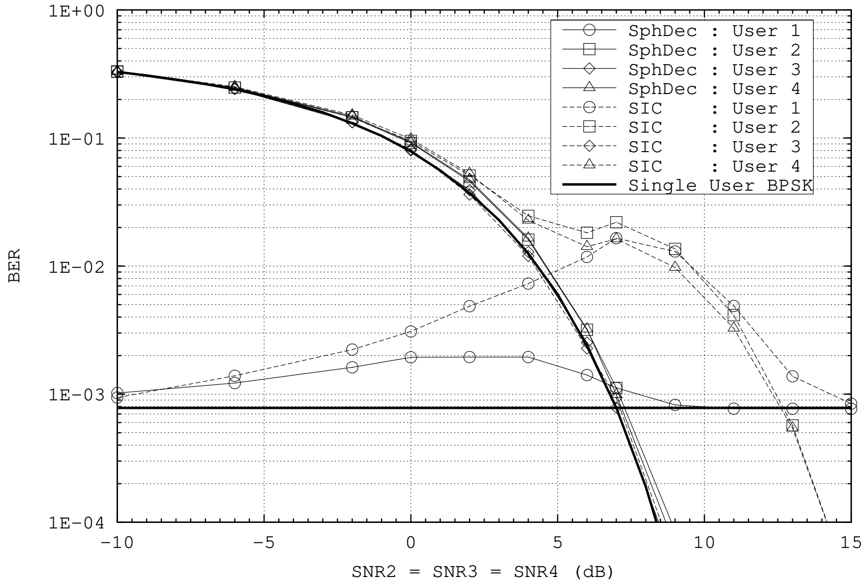


FIG. 4.8 – *Système synchrone: 4 utilisateurs, modulation BPSK, SNR1 = 7 dB, 3 itérations pour le détecteur SIC-PIC à éliminations dures.*

plus importante pour l'éliminateur SIC-PIC. Le phénomène est encore plus marquant avec une modulation 4-PAM comme illustré sur la figure 4.11. Notre détecteur affiche une perte de 7 dB (utilisateur 2) par rapport aux performances de la 4-PAM mono-utilisateur tandis que le détecteur SIC-PIC pur ne fonctionne toujours pas.

Il n'est pas surprenant d'après leur sructure que les détecteurs de type SIC et PIC ne supportent pas l'emploi de modulations de grande taille. Pour vérifier les bonnes performances du décodage par sphères, nous devons le comparer au détecteur MMSE à retour de décision, qui est un des meilleurs détecteurs multi-utilisateurs existants et qui fonctionne de surcroît avec les grandes modulations. La figure 4.12 présente cette comparaison pour un système de 7 utilisateurs synchrones, étalés par des séquences de Gold de longueur 7, et met en évidence l'intérêt de l'emploi du décodeur par sphères pour une augmentation de complexité qui reste raisonnable. La figure 4.13 présente le même système, qui emploie maintenant une modulation 4-PAM et dont les utilisateurs ont pour retards respectifs 0, 1, 2, 3, 4, 5 et 6 périodes chip. Les performances du décodage par sphères combiné à l'élimination d'interférence restent acceptables face à celles du MMSE à retour de décision. Les performances moyennes des deux détecteurs sont équivalentes mais les utilisateurs décodés par sphères ont des performances plus homogènes les unes par rapport aux autres.

Les tables 4.1 et 4.2 permettent de quantifier le gain en nombre d'opérations obtenu en effectuant le décodage ML d'un système multi-utilisateurs synchrone par le décodage par sphères au lieu de l'effectuer par l'algorithme de Viterbi. Cette comparaison est établie pour une modulation 16-PAM. La complexité du décodage par sphères a été déterminée par simulation. Le rayon de recherche a été choisi égal à la borne supérieure de

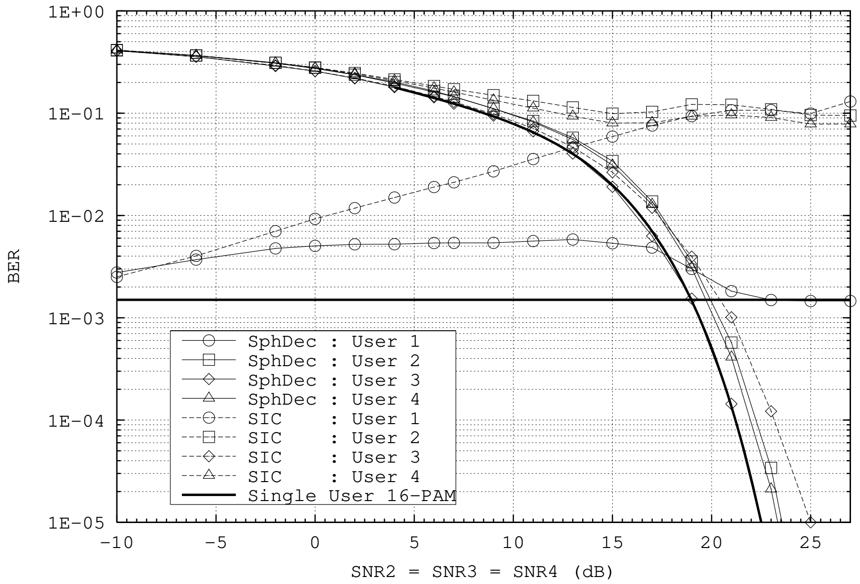


FIG. 4.9 – *Système synchrone: 4 utilisateurs, modulation 16-PAM, SNR1 = 19 dB, 3 itérations pour le détecteur SIC-PIC à éliminations dures.*

K	Additions	Produits	Divisions	Racines	Total	Total $\times K$
4	111	68	14	14	208	832
7	480	332	49	49	910	6371

TAB. 4.1 – *Complexité du décodage ML par sphères (sans LLL) pour un rayon de recherche égal à la borne supérieure de Rogers avec une modulation 16-PAM.*

K	Additions	Produits	Total	Total $\times K$
4	10^5	10^5	2.10^5	8.10^5
7	6.10^8	6.10^8	12.10^8	8.10^9

TAB. 4.2 – *Complexité du décodage ML par algorithme de Viterbi pour une modulation 16-PAM.*

Rogers [14] et aucune réduction de base par l'algorithme LLL [13][49] n'a été entreprise. Une telle réduction permet de réduire encore la complexité du décodage par sphères.

4.10 Conclusions

Nous avons proposé dans ce chapitre une nouvelle technique de détection conjointe basée sur le décodage d'un réseau de points (empilement de sphères) par un algorithme de décodage par sphères. Cet algorithme est optimal pour les systèmes synchrones et conserve des performances excellentes quand les utilisateurs sont asynchrones. L'algorithme peut décoder conjointement jusqu'à 32 utilisateurs, la dimension 32 étant la

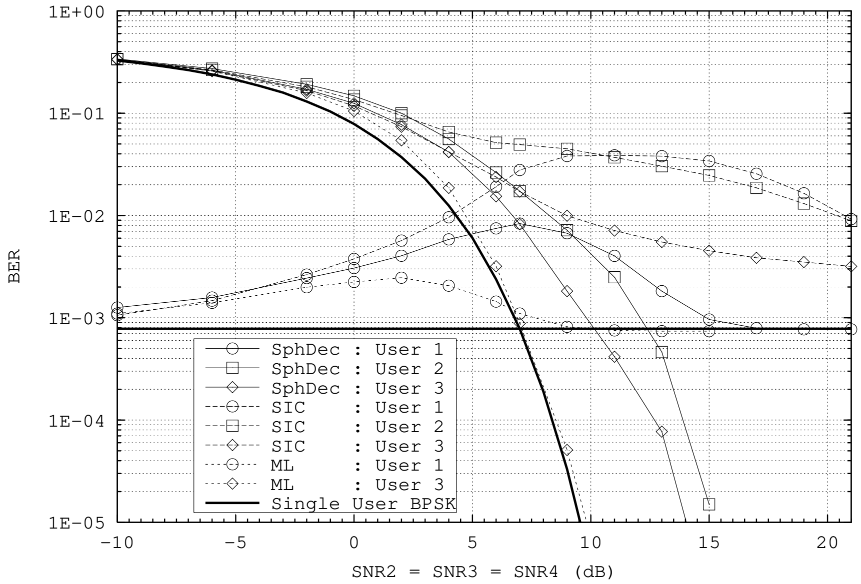


FIG. 4.10 – *Système asynchrone: 4 utilisateurs, modulation BPSK, SNR1 = 7 dB, 3 itérations pour le détecteur SIC-PIC à éliminations dures.*

limite pratique pour la complexité du décodage par sphères. De plus la complexité de la détection ne dépend pas de la taille de la modulation et de grandes modulations PAM ou QAM peuvent être utilisées. Nous avons également fait une étude analytique du gain, dans laquelle les performances sont calculées à partir des paramètres du réseau.

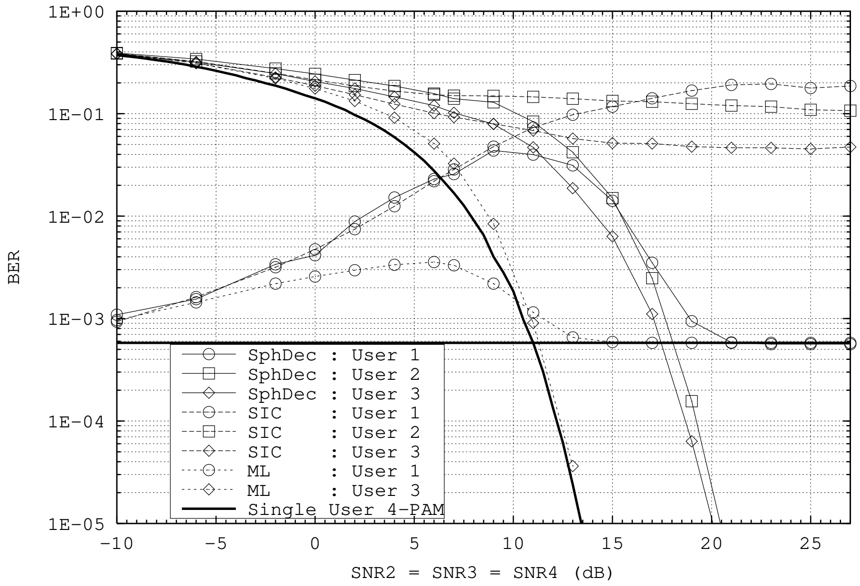


FIG. 4.11 – *Système asynchrone : 4 utilisateurs, modulation 4-PAM, SNR1 = 11 dB, 3 itérations pour le détecteur SIC-PIC à éliminations dures.*

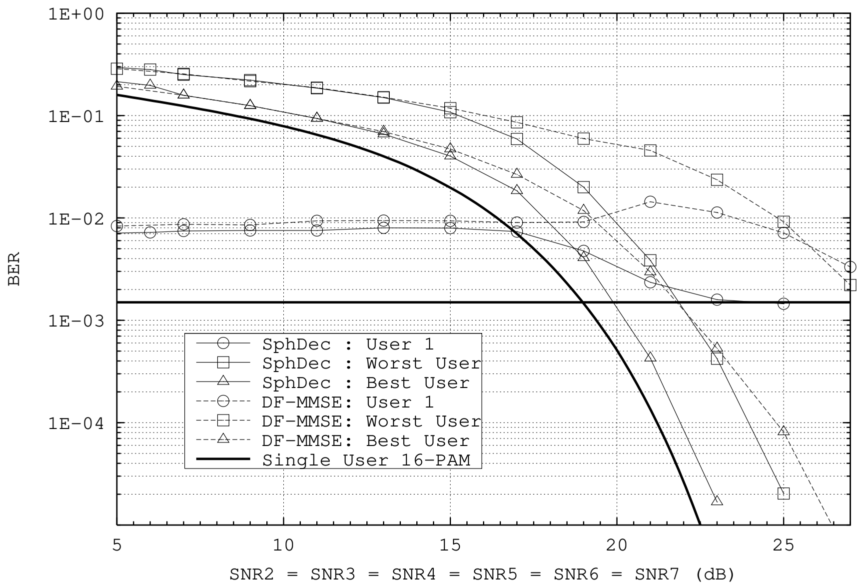


FIG. 4.12 – *Système synchrone : 7 utilisateurs, modulation 16-PAM, SNR1 = 19 dB.*

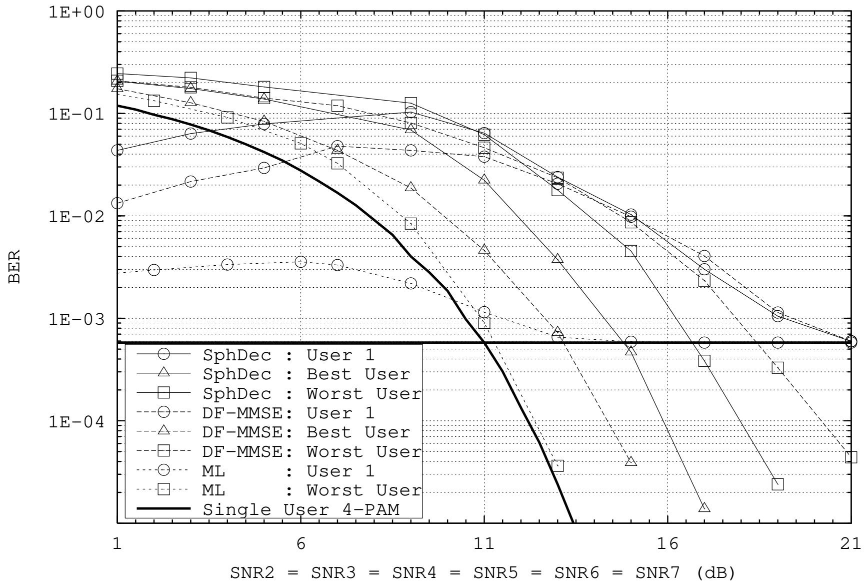


FIG. 4.13 – *Système asynchrone : 7 utilisateurs, modulation 4-PAM, SNR1 = 11 dB.*

Chapitre 5

Système pratique avec une modulation MAQ-16

5.1 Introduction

Les systèmes CDMA proposés jusqu'ici utilisent généralement une modulation de phase binaire (BPSK) ou une modulation de phase en quadrature (4-PAM ou QPSK pour *Quadrature Phase Shift Keying*) et des séquences pseudo-aléatoires binaires. Nous proposons dans ce chapitre l'étude d'un système CDMA employant une modulation MAQ-16 à forte efficacité spectrale. La constellation MAQ-16 contient 16 symboles complexes. Nous avons vu au chapitre précédent que les techniques classiques de détection multi-utilisateurs s'avéraient dans ce cas, soit, comme l'algorithme de Viterbi, trop complexes, soit, comme le SIC, inefficaces. Le décodage par sphères, dont les performances restent bonnes et la complexité raisonnable pour les fortes efficacités spectrales, est le candidat idéal pour la détection de symboles MAQ-16. D'autre part, les symboles de la modulation étant complexes, il est bon de savoir si l'emploi de séquences complexes ne pourrait pas diminuer les intercorrélations entre les utilisateurs et améliorer ainsi les performances de la détection conjointe.

5.2 Le décodage par sphères complexe

Pour pouvoir appliquer le décodage par sphères à un système de K utilisateurs modulés par une modulation MAQ-16, il faut réécrire, dans le cas de symboles et de séquences complexes, la représentation en réseau d'un système synchrone, que nous avons déjà présentée avec des symboles réels et des séquences binaires à la section 4.3. Les notations non précisées sont les mêmes que dans cette section.

Soit $b_k(i)$ le symbole complexe de l'utilisateur k émis à l'instant i dont la partie réelle $b_k^R(i)$ et la partie imaginaire $b_k^I(i)$ appartiennent à l'alphabet entier $\mathcal{A}$ de cardinalité $|\mathcal{A}|$. Chaque utilisateur k transmet un bloc de N symboles avec une amplitude du signal ω_k. Les symboles sont étalés par une signature complexe $s_k(t) = s_k^R(t)+j.s_k^I(t)$ avec $j = \sqrt{-1}$, de durée la période symbole T. Les K symboles complexes transmis à l'instant i sont placés dans un vecteur ligne de valeurs réelles $\mathbf{b}(i)$ défini comme

$$\mathbf{b}(i) = \Big(\; b_1^R(i) \; , \; b_1^I(i) \; , \; \ldots \; , \; b_K^R(i) \; , \; b_K^I(i) \; \Big)$$

Le signal modulé correspondant est, en fonction du temps t:

$$S_t = \sum_{i=0}^{N-1} \sum_{k=1}^{K} \omega_k b_k(i) s_k(t - iT)$$

Nous supposons que le canal est un canal idéal à bruit blanc additif gaussien. Soit $r_t = S_t + \eta_t$ le signal reçu au temps t et η_t un bruit gaussien complexe de moyenne nulle et dont les composantes ont une variance N_0.
Soit le vecteur ligne

$$\mathbf{y}(i) = \Big(\; y_1^R(i) \; , \; y_1^I(i) \; , \; \ldots \; , \; y_K^R(i) \; , \; y_K^I(i) \; \Big)$$

tel que $y_k(i) = y_k^R(i) + j.y_k^I(i)$ soit la sortie complexe à l'instant i du filtre adapté à l'utilisateur k:

$$\begin{aligned}
y_k(i) &\triangleq \int_{-\infty}^{+\infty} {s_k}^*(t - iT) r_t dt + n_k(i) \\
&= \sum_{\ell=1}^{K} \omega_\ell b_\ell(i) \int_0^T s_\ell(t) {s_k}^*(t) dt + n_k(i) \\
&= \sum_{\ell=1}^{K} \omega_\ell b_\ell(i) R_{\ell k} + n_k(i)
\end{aligned} \tag{5.1}$$

$$\text{avec} \quad R_{\ell k} = \int_0^T s_\ell(t) {s_k}^*(t) dt = R_{\ell k}^R + j.R_{\ell k}^I \quad \text{pour} \quad k,\, \ell = 1, \ldots, K$$

Si on décompose les éléments complexes de (5.1) en parties réelles et parties imaginaires, on obtient

$$\begin{aligned}
\Big[y_k^R(i) + j.y_k^I(i)\Big] &= \sum_{\ell=1}^{K} \omega_\ell \Big[b_\ell^R(i) + j.b_\ell^I(i)\Big]\Big[R_{\ell k}^R + j.R_{\ell k}^I\Big] + \Big[n_k^R(i) + j.n_k^I(i)\Big] \\
&= \sum_{\ell=1}^{K} \omega_\ell \Big[b_\ell^R(i).R_{\ell k}^R - b_\ell^I(i).R_{\ell k}^I\Big] + j.\sum_{\ell=1}^{K} \omega_\ell \Big[b_\ell^R(i).R_{\ell k}^I + b_\ell^I(i).R_{\ell k}^R\Big] \\
&\quad + \Big[n_k^R(i) + j.n_k^I(i)\Big]
\end{aligned} \tag{5.2}$$

Soient $\mathbf{D}_{\omega,2} = \text{diag}(\omega_1,\omega_1,\ldots,\omega_K,\omega_K)$ et $\mathbf{R}$ la matrice de taille $2K \times 2K$ telle que

$$\mathbf{R} = \begin{bmatrix}
R_{11}^R & R_{11}^I & \cdots & R_{1K}^R & R_{1K}^I \\
-R_{11}^I & R_{11}^R & \cdots & -R_{1K}^I & R_{1K}^R \\
\vdots & \vdots & & \vdots & \vdots \\
R_{K1}^R & R_{K1}^I & \cdots & R_{KK}^R & R_{KK}^I \\
-R_{K1}^I & R_{K1}^R & \cdots & -R_{KK}^I & R_{KK}^R
\end{bmatrix}$$

L'équation (5.2) peut alors se mettre sous forme matricielle

$$\mathbf{y}(i) = \mathbf{b}(i)\mathbf{M} + \mathbf{n}(i) \tag{5.3}$$

où la matrice $\mathbf{M}$ de taille $2K \times 2K$ est définie par $\mathbf{M} = \mathbf{D}_{\omega,2}\mathbf{R}$ et où le vecteur de bruit $\mathbf{n}(i) = (n_1^R(i), n_1^I(i), \ldots, n_K^R(i), n_K^I(i))$ a pour matrice de covariance $N_0\mathbf{R}$.

On voit que l'on peut modéliser un système CDMA complexe avec K utilisateurs par un réseau de dimension $2K$. On peut donc facilement généraliser au cas complexe les sections 4.5, 4.6 et 4.7 en remplaçant chaque élément v_k de vecteur ligne par la paire $(v_k^R\ v_k^I)$ et chaque élément $V_{\ell k}$ de matrice par la matrice de taille 2×2 ci-dessous :

$$\begin{bmatrix} V_{\ell k}^R & V_{\ell k}^I \\ -V_{\ell k}^I & V_{\ell k}^R \end{bmatrix}$$

On peut donc, au prix d'une dimension de réseau double, c'est-à-dire au prix d'une complexité du décodage par sphères 64 fois plus importante, bénéficier d'une efficacité spectrale double.

5.3 Choix des séquences d'étalement

Les séquences les plus communément utilisées en CDMA sont des séquences binaires prenant les valeurs ± 1. Cependant la présence de symboles complexes dans le signal suggère l'emploi de séquences également complexes pour étaler ces symboles. Dans [37], les auteurs avaient déjà montré quel intérêt on aurait à utiliser des séquences complexes mais s'étaient limités à une modulation BPSK étalée par une séquence complexe. Nous allons nous intéresser maintenant à l'emploi avec des modulations en quadrature de séquences complexes et plus particulièrement de séquences polyphases, c'est-à-dire de séquences $\{\hat{a}_n\}$ dont les symboles sont des racines complexes de l'unité :

$$\hat{a}_n = e^{j\frac{2\pi}{p}a_n} \tag{5.4}$$

où p est généralement un nombre premier et $\{a_n\}$ une séquence p-aire, c'est-à-dire $a_n \in \{0,1,2,\ldots,p-1\}$.

Une propriété importante d'un ensemble de séquences est sa corrélation maximale non-triviale $R_{\max}$, qui est la valeur maximale parmi toutes les autocorrélations, exceptées celles en phase, et parmi toutes les intercorrélations possibles. La borne inférieure de Sidelnikov [63], qui ne s'applique qu'aux séquences polyphases, donne à penser qu'un gain substantiel peut être espéré de l'emploi de séquences complexes. Pour un ensemble de λ^u séquences de longueur λ, avec $\lambda \gg u$ et $u \geq 1$ entier, les bornes de Sidelnikov admettent une bonne approximation [21] :

$$R_{\max}^2 \geq \begin{cases} \lambda\Big(2u+1-\frac{1}{1.3.5\ldots(2u-1)}\Big), & \text{pour des séquences binaires} \\ \lambda\Big(u+1-\frac{1}{1.2.3\ldots 2u}\Big), & \text{pour des séquences non binaires} \end{cases} \tag{5.5}$$

Les approximations (5.5) permettent d'espérer un gain d'environ 3dB sur la valeur de l'interférence si on remplace les séquences binaires par des séquences complexes. En effet, pour des ensembles de taille équivalente à la longueur des séquences ($u \approx 1$), qui s'appliquent donc bien au CDMA, on obtient :

$$R_{\max} \geq \begin{cases} \sqrt{2\lambda}, & \text{pour des séquences binaires} \\ \sqrt{\lambda}, & \text{pour des séquences non binaires} \end{cases} \tag{5.6}$$

Le résultat encourageant de la borne (5.6) ainsi que l'étude analytique de [37] nous incitent à parcourir le large éventail des séquences polyphases connues pour choisir celles qui conviendraient le mieux à notre système. Nous pourrons ainsi les comparer aux séquences de Gold, qui sont les séquences pseudo-aléatoires binaires les mieux adaptées aux systèmes CDMA asynchrones. Quels doivent être les critères de choix d'une séquence pseudo-aléatoire qu'on désire appliquer à un système d'accès multiple ? Tout d'abord, la taille M de l'ensemble des séquences, qui donne le nombre d'utilisateurs qui peuvent simultanément transmettre sur le canal, doit être suffisamment grande. En général on désire avoir $M \approx \lambda$. L'intercorrélation entre les séquences doit être suffisamment faible pour que l'interférence entre les utilisateurs ne compromette pas la détection. Une faible fonction d'autocorrélation hors phase peut être aussi souhaitable pour faciliter la synchronisation du récepteur sur la séquence pseudo-aléatoire d'un utilisateur donné. Nous verrons dans la suite, que la connaissance de la corrélation maximale non triviale $R_{\max}$ d'un ensemble de séquences ne suffit pas pour apprécier la qualité de celui-ci mais qu'il faut s'intéresser à la distribution ou à la moyenne des intercorrélations, qui sont beaucoup plus instructives.

Nous allons étudier deux types d'intercorrélations, les intercorrélations périodiques et les intercorrélations apériodiques. L'intercorrélation périodique $R^P_{i,\ell}(\tau)$ entre les séquences $\{\hat{a}^i_n\}$ et $\{\hat{a}^\ell_n\}$ pour un décalage τ est :

$$R^P_{i,\ell}(\tau) = \sum_{n=0}^{\lambda-1} \hat{a}^i_{n+\tau}[\hat{a}^\ell_n]^*$$

Les différentes valeurs prises par cette fonction pour un ensemble de séquences donné sont souvent connues mais cette corrélation périodique ne tient pas compte des valeurs des symboles étalés, qui modifient le résultat de la corrélation. On voit sur la figure 5.1,

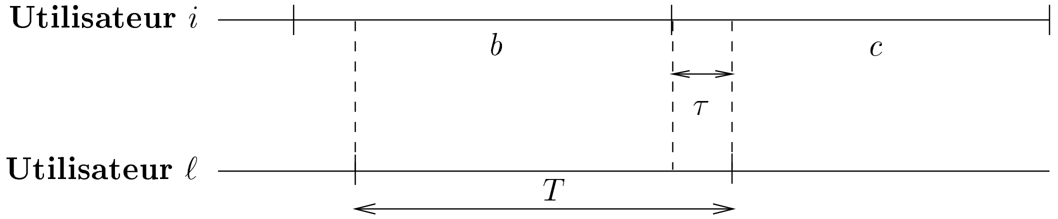


FIG. 5.1 – *Interférences de l'utilisateur i sur l'utilisateur ℓ dans un système d'accès multiple asynchrone.*

que l'expression de la corrélation apériodique $R^A_{i,\ell}(\tau,b,c)$ tenant compte des valeurs des symboles complexes b et c de l'utilisateur i, qui interfère sur l'utilisateur ℓ, est :

$$R^A_{i,\ell}(\tau,b,c) = b.\sum_{n=0}^{\lambda-\tau-1} \hat{a}^i_{n+\tau}[\hat{a}^\ell_n]^* + c.\sum_{n=\lambda-\tau}^{\lambda-1} \hat{a}^i_{n-(\lambda-\tau)}[\hat{a}^\ell_n]^*$$

On peut ensuite calculer la moyenne $\overline{R^A}$ de $|R^A_{i,\ell}(\tau,b,c)|$ sur toutes les séquences de l'ensemble, tous les décalages τ (sauf pour $i=\ell$ et $\tau=0$) et tous les couples (b,c) pour avoir une idée de l'interférence moyenne entre deux utilisateurs d'un système CDMA utilisant cet ensemble de séquences. C'est surtout ce critère qui va nous être utile pour choisir un bon ensemble [53].

Nous avons sélectionné cinq familles de séquences parmi les nombreuses familles présentées dans [21] et laissé de côté les ensembles de séquences aux bonnes propriétés de corrélation mais de taille trop faible comme les séquences quadriphases à fonctions de Bent, les séquences de Frank, de Milewski ... Cette sélection n'est évidemment pas exhaustive.

5.3.1 Séquences binaires de Gold

L'ensemble des séquences de Gold [28][29] de longueur $\lambda = 2^m - 1$ est de taille $\lambda + 2$. Il est obtenu par la sommation modulo 2 de deux m-séquences préférées de longueur $\lambda = 2^m - 1$, qui présentent de bonnes propriétés d'intercorrélation. λ séquences de Gold différentes sont obtenues en décalant une séquence par rapport à l'autre et en faisant leur somme. Les deux séquences restantes sont les deux m-séquences elles-mêmes. Une m-séquence de longueur maximale $\lambda = 2^m - 1$ est générée par une récursion dont le polynôme caractéristique [41] est un polynôme primitif de degré m sur $\mathbb{Z}_2[x]$. Supposons que deux séquences préférées sont générées par les polynômes primitifs $h_1(x)$ et $h_2(x)$, les séquences de Gold construites à partir de ces deux séquences peuvent alors être générées par une récursion linéaire de polynôme caractéristique $h(x) = h_1(x)h_2(x)$. Ces séquences ne sont bien évidemment pas de longueur maximale.

Les valeurs prises par les intercorrélations périodiques sont [21][41][52] :

$$R^P_{i,\ell}(\tau) = \begin{cases} -1, \quad -1+2^{\frac{m+1}{2}}, \quad -1-2^{\frac{m+1}{2}}, & \text{pour } m \text{ impair} \\ -1, \quad -1+2^{\frac{m+2}{2}}, \quad -1-2^{\frac{m+2}{2}}, & \text{pour } m \equiv 2 \bmod 4 \end{cases}$$

Il n'existe pas de séquences préférées pour $m \equiv 0 \bmod 4$ [62] [42].

La figure 5.2 donne la distribution des corrélations périodiques normalisées par la longueur de la séquence pour des séquences de Gold de longueur 127. La ligne en traits interrompus figure la moyenne qui vaut $\overline{R^P} = 8.387/127 = 0.0660$. La moyenne des corrélations apériodiques avec une modulation QPSK est $\overline{R^A} = 9.15/127 = 0.0720$.

5.3.2 Séquences binaires de Gold généralisées

Les séquences de Gold n'existent pas pour $m \equiv 0$ mod 4, c'est-à-dire notamment pour les longueurs 15 et 255. Cependant, pour ces valeurs de m, il existe des ensembles de séquences de longueur $\lambda = 2^m - 1$ dont la corrélation maximale $R_{\max} = 1 + 2^{(m+2)/2}$ est la même que celle des ensembles de séquences de Gold. Pour cette raison, on les appelle ensembles de séquences de Gold généralisées.

Soit $f = 1 + 2^{(m+2)/2}$, on peut montrer que si $m \equiv 0$ mod 4, pgcd$(f,\lambda) = 3$. On prend une m-séquence de période λ. Par décimation d'ordre f de cette séquence, on peut obtenir 3 séquences différentes de période $\lambda/3$. Chacune de ces 3 séquences peut être sommée modulo 2, décalée de $\lambda/3$ manières différentes, avec la m-séquence d'origine. On obtient ainsi λ séquences, qui, avec la m-séquence elle-même, forment les $\lambda+1$ séquences de l'ensemble des séquences de Gold généralisées.

Les valeurs prises par les intercorrélations périodiques sont [21]

$$-1,\ -1-2^{m/2},\ -1+2^{m/2},\ -1-2^{(m+2)/2},\ -1+2^{(m+2)/2}$$

5.3.3 Séquences quadriphases de type I sur $\mathbb{Z}_4$

Le premier ensemble de séquences complexes testé est l'ensemble des séquences quadriphases de type I sur $\mathbb{Z}_4$ [65][5]. Elles ont la même période que les séquences de Gold et forment un ensemble de la même taille que celui des séquences de Gold. Elles se construisent simplement à l'aide d'une récurrence sur $\mathbb{Z}_4$ mais contrairement aux m-séquences, qui parcourent au cours d'une période tous les états possibles du registre, les séquences quadriphases de type I parcourent $\lambda + 2 = 2^m + 1$ ensembles d'états différents selon l'initialisation. Tandis qu'une récurrence sur $\mathbb{Z}_2$ donnait une seule m-séquence, cette récurrence sur $\mathbb{Z}_4$ donne $\lambda + 2$ séquences. Ces séquences quadriphases sont en cela proches des séquences de Gold, qui ne sont pas, elles non plus, de longueur maximale. Comme pour les m-séquences sur $\mathbb{Z}_2[x]$, il faut choisir un polynôme primitif sur $\mathbb{Z}_4[x]$ (cf. table dans [21, pp. 418-420]) comme polynôme caractéristique de la récurrence. On applique finalement la transformation (5.4) à la sortie de la récurrence pour obtenir la séquence quadriphase d'éléments $\{1, -1, j, -j\}$.

On sait calculer théoriquement la distribution des $R^P_{i,\ell}(\tau)$ [5] :

Si $m = 2t + 1$ alors

$$R^P_{i,\ell}(\tau) = \begin{cases} 2^m - 1, & 2^m + 1 \text{ fois} \\ -1, & 2^{2m} - 2 \text{ fois} \\ -1 + 2^t + j.2^t, & (2^{m-2} + 2^{t-1})(2^{2m} - 2) \text{ fois} \\ -1 + 2^t - j.2^t, & (2^{m-2} + 2^{t-1})(2^{2m} - 2) \text{ fois} \\ -1 - 2^t + j.2^t, & (2^{m-2} - 2^{t-1})(2^{2m} - 2) \text{ fois} \\ -1 - 2^t - j.2^t, & (2^{m-2} - 2^{t-1})(2^{2m} - 2) \text{ fois} \end{cases}$$

Si $m = 2t$ alors

$$R^P_{i,\ell}(\tau) = \begin{cases} 2^m - 1, & 2^m + 1 \text{ fois} \\ -1, & 2^{2m} - 2 \text{ fois} \\ -1 + 2^t, & (2^{m-2} + 2^{t-1})(2^{2m} - 2) \text{ fois} \\ -1 - 2^t, & (2^{m-2} - 2^{t-1})(2^{2m} - 2) \text{ fois} \\ -1 + j.2^t, & 2^{m-2}(2^{2m} - 2) \text{ fois} \\ -1 - j.2^t, & 2^{m-2}(2^{2m} - 2) \text{ fois} \end{cases}$$

La figure 5.3 donne la distribution des corrélations périodiques normalisées par la longueur de la séquence pour des séquences quadriphases de type I de longueur 127. Ces séquences ont une très bonne corrélation maximale mais toutes les valeurs d'intercorrélations sont concentrées près de cette valeur maximale et la moyenne des corrélations périodiques normalisées est donc élevée. Elle vaut $\overline{R^P} = 11.175/127 = 0.0880$ et la moyenne des corrélations apériodiques avec une modulation QPSK vaut $\overline{R^A} = 10.56/127 = 0.0831$. Ces séquences sont donc bien moins adaptées aux systèmes CDMA que ne le laissait croire la valeur $R_{\max}$.

5.3.4 Séquences polyphases de Lüke de type III

Les séquences de Lüke de type III sont des séquences polyphases de période $\lambda = p-1$, où p est un nombre entier supérieur ou égal à 5. Elles forment un ensemble de $M = p$ séquences, dont les éléments sont les racines p-ièmes de l'unité. Soit α un élément primitif de $GF(p)$, c'est-à-dire un élément de $GF(p) - \{0\}$ tel que $\alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{p-2}$ soient tous distincts modulo p, la première séquence est

$$\hat{a}^0_n = e^{j\frac{2\pi}{p}\alpha^n}$$

et les $p - 1$ autres sont

$$\hat{a}^{i+1}_n = e^{j\frac{2\pi}{p}(\alpha^n + \alpha^{2n+i})} \quad \text{pour } i \leq p - 2$$

Ces séquences ont des intercorrélations périodiques non triviales telles que

$$|R^P_{i,\ell}(\tau)| \leq \sqrt{p} + 1 \quad \text{pour } i \neq \ell \text{ et } \tau \neq 0$$

La figure 5.4 donne la distribution des corrélations périodiques normalisées par la longueur de la séquence pour des séquences de Lüke de type III de longueur 126 et on constate malheureusement que toutes les valeurs sont concentrées autour de la valeur maximale. La moyenne des corrélations périodiques normalisées vaut $\overline{R^P} = 11.13/126 = 0.0883$ et celle des corrélations apériodiques avec une modulation QPSK vaut $\overline{R^A} = 10.96/126 = 0.0870$.

5.3.5 Séquences polyphases de Trachtenberg

Les séquences de Trachtenberg [67] sont des séquences polyphases de période $\lambda = p^m - 1$, dont l'ensemble a une taille $M = \lambda + 2$, où p est un entier premier supérieur ou égal à 3 et m un entier impair.

Soit Tr la trace de α dans $GF(p^m)$ relativement au sous-corps $GF(p)$ [41] :

$$Tr(\alpha) = \alpha + \alpha^p + \alpha^{p^2} + \alpha^{p^3} + \ldots + \alpha^{p^{m-1}}$$

On choisit α élément primitif de $GF(p^m)$. On construit les M séquences de Trachtenberg en utilisant la transformation (5.4) et les séquences p-aires telles que :

$$a_n^i = \begin{cases} Tr(\alpha^n + \alpha^{dn+i}), & \text{pour } 0 \leq i \leq p^m - 2 \\ Tr(\alpha^n), & \text{pour } i = p^m - 1 \\ Tr(\alpha^{dn}), & \text{pour } i = p^m \end{cases}$$

avec $d = (p^{2k} + 1)/2$ ou $d = p^{2k} - p^k + 1$, k étant un entier tel que $d \not\equiv p^j \bmod \lambda$ avec j entier.

Ces séquences ont des intercorrélations périodiques non triviales réelles égales à -1, $-1+p^{(m+e)/2}$ ou $-1-p^{(m+e)/2}$ avec $e = \text{pgcd}(m,k)$. La figure 5.5 donne la distribution des corrélations périodiques normalisées par la longueur de la séquence pour des séquences de Trachtenberg de longueur 124. Les phases des éléments de ces séquences prennent donc 5 valeurs ($p = 5$). La valeur de corrélation maximale est très élevée mais de nombreuses corrélations valent -1, ce qui explique la faible moyenne des corrélations périodiques normalisées : $\overline{R^P} = 5.73/124 = 0.0462$. Celle des corrélations apériodiques avec une modulation QPSK vaut $\overline{R^A} = 7.90/124 = 0.0637$.

5.3.6 Séquences polyphases de Helleseth

Les séquences de Helleseth [30] sont des séquences polyphases de période $p^m - 1$, où p est un nombre premier supérieur ou égal à 3 et m un entier pair tel $p^{m/2} \not\equiv 2 \bmod 3$. La taille de leur ensemble est $\lambda + 2$. Elles s'obtiennent exactement de la même façon que les séquences de Trachtenberg en prenant $d = 2p^{m/2} - 1$.

Les intercorrélations périodiques non triviales de ces séquences ont les valeurs réelles suivantes : -1, $-1+p^{m/2}$, $-1-p^{m/2}$ et $-1+2p^{m/2}$. La figure 5.6 donne la distribution des corrélations périodiques normalisées par la longueur de la séquence pour des séquences de Helleseth de longueur 120. Les phases des éléments de ces séquences prennent donc 11 valeurs ($p = 11$). La moyenne des corrélations périodiques normalisées vaut $\overline{R^P} = 8.12/120 = 0.0677$ et celle des corrélations apériodiques avec une modulation QPSK vaut $\overline{R^A} = 9.05/120 = 0.0754$.

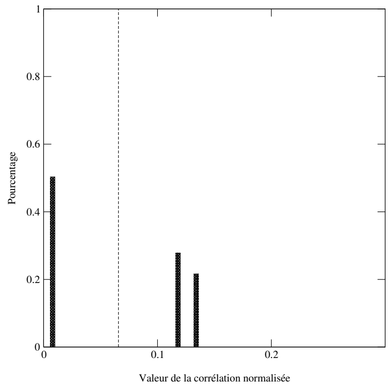


FIG. 5.2 – *Distribution des corrélations périodiques normalisées des séquences de Gold de longueur 127.*

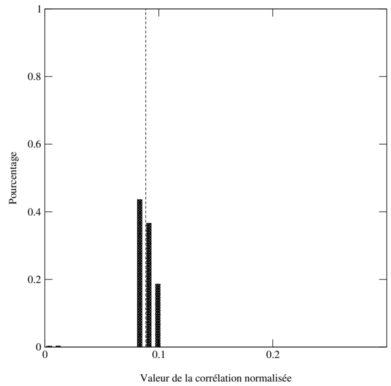


FIG. 5.4 – *Distribution des corrélations périodiques normalisées des séquences de Lüke de type III de longueur 126.*

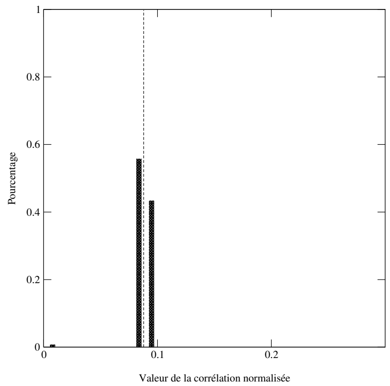


FIG. 5.3 – *Distribution des corrélations périodiques normalisées des séquences de type I de longueur 127.*

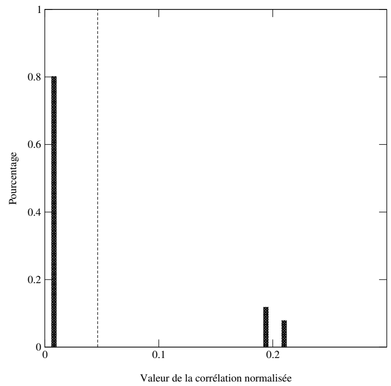


FIG. 5.5 – *Distribution des corrélations périodiques normalisées des séquences de Trachtenberg de longueur 124.*

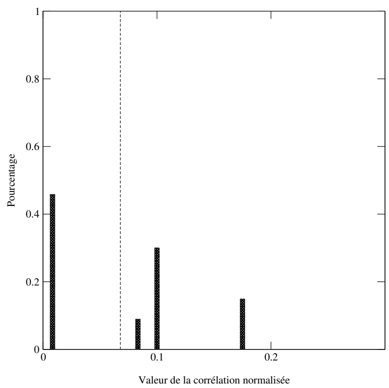


FIG. 5.6 – *Distribution des corrélations périodiques normalisées des séquences de Helleseth de longueur 120.*

Le tableau 5.1 résume les résultats obtenus dans cette section. Les séquences, pour des longueurs proches de 125, sont classées par ordre d'intercorrélations apériodiques moyennes $\overline{R^A}$ croissantes. Les séquences du haut du tableau, à savoir les séquences de Trachtenberg et les séquences de Gold, sont très intéressantes pour le CDMA, les autres beaucoup moins. On voit clairement sur ce tableau que la valeur de $|R_{\max}|/\lambda$ n'est pas un critère pour le choix d'un ensemble de séquences dans un système CDMA.

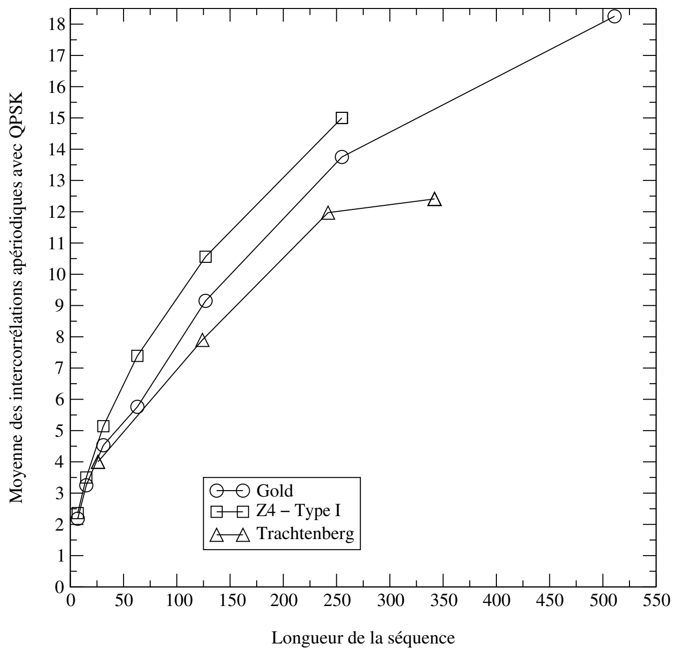


FIG. 5.7 – *Moyenne des corrélations apériodiques non triviales pour des symboles QPSK en fonction de la longueur pour différents ensembles de séquences.*

La figure 5.7 représente la moyenne des corrélations apériodiques non triviales pour des symboles QPSK correspondant à différents types de séquences en fonction de leur longueur. Les points correspondant aux séquences de Gold de longueur 15 et 255 ont été calculés sur des séquences de Gold généralisées, les véritables séquences de Gold n'existant pas pour ces longueurs. On constate que quelle que soit la longueur des séquences, les séquences de Trachtenberg sont celles qui présentent les propriétés les plus favorables au CDMA.

Séquences	λ	$\|R_{\max}\|/\lambda$	$\overline{R^P}$	$\overline{R^A}$
Trachtenberg	124	0.210	0.0462	0.0637
Gold	127	0.134	0.0660	0.0720
Helleseth	120	0.175	0.0677	0.0754
$\mathbb{Z}_4$ - Type I	127	0.095	0.0880	0.0831
Lüke - Type III	126	0.099	0.0883	0.0870

TAB. 5.1 – *Comparaison des propriétés de corrélation de différents ensembles de séquences.*

5.4 Résultats de simulations

De manière à montrer l'avantage que l'on peut trouver à utiliser des séquences de Trachtenberg, nous avons simulé un système utilisant une modulation MAQ-16. Tous les utilisateurs ont un même rapport signal sur bruit égal à 10 dB. Nous comparons les performances de ce système avec des séquences de Trachtenberg de longueur 124 et des séquences de Gold de longueur 127. Comme les intercorrélations entre les séquences peuvent fortement varier selon les séquences que l'on choisit pour le système et les retards relatifs des utilisateurs, nous avons simulé de nombreux systèmes en tirant aléatoirement les séquences du système parmi l'ensemble des séquences de Trachtenberg ou de Gold selon le cas et les retards des utilisateurs parmi toutes les combinaisons des retards possibles. Bien sûr il eut été illusoire de penser que nous aurions le temps de tester exhaustivement toutes les combinaisons et il faut donc relativiser le résultat en considérant que l'échantillon testé est assez réduit. Cependant les valeurs numériques regroupées dans le tableau 5.2 montrent des résultats encourageants pour l'utilisation des séquences de Trachtenberg. Pour $K = 8$ utilisateurs celles-ci ne présentent aucun avantage par rapport aux séquences de Gold. Par contre, pour $K = 15$ utilisateurs, quand les valeurs des intercorrélations deviennent critiques, les performances des séquences de Gold sont en moyenne 10 fois moins bonnes que celles des séquences de Trachtenberg. Ces dernières sont encore très proches des performances mono-utilisateur de la MAQ-16 à 10 dB, à savoir $1.74.10^{-3}$. Le nombre d'utilisateurs peut paraître faible mais chaque symbole MAQ-16 transmet 4 bits et l'efficacité spectrale totale est proche de 1/2 (60/124). Le système est donc équivalent à un système de 30 utilisateurs modulés par une QPSK.

Séquences	λ	K	BER	Nombre de tirages
Gold	127	8	$2.07.10^{-3}$	93
Trachtenberg	124	8	$2.12.10^{-3}$	93
Gold	127	15	$3.53.10^{-2}$	15
Trachtenberg	124	15	$3.23.10^{-3}$	15

TAB. 5.2 – *Comparaison des performances du décodage par sphères pour des séquences de Gold et des séquences de Trachtenberg.*

Malheureusement la simulation d'un système avec 60 utilisateurs, une modulation QPSK et un étalement par séquence de Gold de longueur 127, utilisant un détecteur hybride SIC-PIC donne des résultats meilleurs. En effet, si tous les utilisateurs ont un même rapport signal sur bruit égal à 10 dB, on obtient une probabilité d'erreur par bit moyenne sur 29 tirages de $1.47.10^{-3}$, c'est-à-dire que pour une efficacité spectrale double, proche de 1, et une complexité moindre, le système avec modulation QPSK et détecteur SIC-PIC a de meilleures performances à même rapport signal sur bruit que le système avec modulation MAQ-16 et décodage par sphères ! Il faudrait augmenter l'efficacité spectrale de ce dernier système pour voir s'il fonctionne mieux pour de très fortes efficacités spectrales. La complexité du décodage par sphères est malheureusement un facteur limitant, c'est pourquoi nous allons comparer les performances des deux systèmes pour un facteur d'étalement égal à 31. Avec des séquences d'étalement de

Gold de longueur 31, le système avec modulation MAQ-16 et décodage par sphères a un taux d'erreur par bit moyen égal à $8.94.10^{-3}$ sur 4 tirages avec 10 utilisateurs à 10 dB. Le système avec 20 utilisateurs modulés par une QPSK et détectés par un détecteur SIC-PIC présente un taux d'erreur moyen de $6.14.10^{-3}$ sur 101 tirages.

On voit donc que même dans les cas de très fortes efficacités spectrales supérieures à 1, les systèmes constitués de modulations QPSK et d'un récepteur SIC-PIC ont de meilleures performances que les systèmes plus complexes employant des modulations MAQ-16 et un récepteur associant le décodeur par sphères et une élimination dure des interférences. Remarquons que les systèmes du premier type sont plus éloignés des performances de la modulation QPSK mono-utilisateur que ceux du deuxième type ne le sont des performances mono-utilisateur de la MAQ-16. Cependant la différence entre les performances mono-utilisateur de la QPSK et de la MAQ-16 suffit à combler le handicap de la QPSK.

5.5 Conclusions

L'emploi de séquences complexes pour l'étalement permet d'améliorer les performances des systèmes DS-CDMA incluant des modulations avec des symboles complexes. Par contre l'emploi de modulations MAQ-16, rendu possible grâce au décodage par sphères, s'avère inutile puisque l'utilisation de modulations QPSK détectées par un SIC-PIC permet d'atteindre des performances meilleures à même rapport signal à bruit et même efficacité spectrale totale. Cela revient en fait à remplacer chaque utilisateur employant une MAQ-16 par 2 utilisateurs employant une QPSK. Pour que l'emploi de la modulation MAQ-16 ait un intérêt dans les systèmes multi-utilisateurs, il faudrait pouvoir décoder des réseaux de très grande dimension avec le détecteur par sphères et atteindre ainsi des efficacités spectrales totales supérieures à 2, valeur au dessus de laquelle le détecteur SIC-PIC ne peut plus fonctionner.

Chapitre 6

Conclusions et perspectives

La maximisation du nombre d'utilisateurs est un des enjeux essentiels dans les sytèmes CDMA. Le détection multi-utilisateurs contribue largement à améliorer l'efficacité spectrale totale des systèmes. Nous avons proposé dans ce document deux méthodes différentes de détection multi-utilisateurs autorisant un gain en efficacité spectrale autrement que par l'augmentation du nombre des utilisateurs quand les séquences pseudo-aléatoires et le facteur d'étalement sont fixés.

Dans la première méthode, présentée au chapitre 3, nous avons supprimé les signatures pseudo-aléatoires qui permettent habituellement de séparer les utilisateurs. Pour compenser leur absence, nous avons employé des codes rendus indépendants par l'entrelacement de leur sortie. Le décodage conjoint itératif utilisé est capable d'éliminer complètement l'interférence entre les utilisateurs et ce d'autant mieux que les codes sont bien choisis et la taille de l'entrelaceur élevée. On obtient alors les performances du système mono-utilisateur employant le même code pour des systèmes d'efficacité spectrale totale égale à 1/2 et ces performances sont quasiment atteintes, pour peu que les puissances des utilisateurs soient légèrement différentes, dans les systèmes d'efficacité spectrale totale égale à 1.

Nous avons constaté que les codes concaténés, les plus puissants pour les systèmes mono-utilisateur, se révélaient inefficaces pour supprimer l'interférence entre les utilisateurs et surtout quand ceux-ci ont des amplitudes proches. Une étude plus poussée sur ces faibles performances des codes concaténés lorsqu'ils sont inclus dans notre système multi-utilisateurs doit être encore menée. Il serait bon de vérifier si la longueur du code n'est pas un facteur limitant pour les performances du système. Une telle étude pourrait être menée sur un code de Hamming ou un code convolutif fermé régulièrement employés dans un système avec entrelacement infini. Une mise en évidence de leurs mauvaises performances permettrait de conclure qu'un code trop court, tout comme un entrelaceur trop court, nuit aux bonnes performances du système. Il serait alors possible d'imaginer que les turbo-codes, que nous avons choisis courts pour garantir de bonnes performances à faible rapport signal à bruit, sont en fait trop courts. On pourrait alors choisir d'utiliser des turbo-codes très longs et de les décoder en itérant tout d'abord sur les codes supérieurs des différents utilisateurs puis sur les codes inférieurs.

La deuxième méthode proposée dans ce document permet, grâce à un détecteur conjoint fondé sur un décodeur universel de réseaux de points, d'augmenter l'efficacité spectrale totale en augmentant la taille des modulations utilisées. Malheureusement les simulations ont montré que pour un même rapport signal à bruit et une même efficacité spectrale totale, un système employant des modulations QPSK et un détecteur conjoint de type SIC-PIC atteignait les mêmes performances qu'un système utilisant des modulations MAQ-16 et un décodeur par sphères. Il faudrait augmenter le nombre d'utilisateurs jusqu'à ce que le détecteur SIC-PIC avec la modulation QPSK ne fonctionne plus du tout en espérant que le décodeur par sphères, pour la même efficacité spectrale totale, fonctionne encore. Cependant la complexité du décodeur au-delà de la dimension 40 rend cette entreprise difficile à réaliser.

Nous envisageons d'appliquer un décodage par sphères à sorties souples qui donnerait une valeur de la fiabilité du point décodé et permettrait ainsi d'améliorer la qualité de l'élimination d'interférences, qui serait alors non plus dure mais souple, dans les systèmes asynchrones. Ce nouveau détecteur apporterait peut-être un gain suffisant pour que l'emploi de la modulation MAQ-16 gagne en intérêt vis-à-vis de l'emploi de la modulation QPSK.

Enfin, il serait intéressant de diminuer la complexité du décodage par sphères, qui est actuellement de l'ordre de $\mathcal{O}(K^6)$, ou d'utiliser un décodage de réseaux de points différent pour obtenir une complexité de l'ordre de $\mathcal{O}(K^i)$ avec $i < 6$.

Annexe A

Algorithme Aller-Retour ou Forward-Backward

A.1 Description générale

L'algorithme Aller-Retour [1] estime les probabilités a posteriori des états et des transitions d'une source de Markov observée à travers un canal discret, bruité et sans mémoire. Nous avons ici brièvement reporté les équations principales caractérisant les différentes étapes de l'algorithme.

Soit $0 \leq i < N$ représentant le temps et $\{X_i\}$ une séquence de sorties d'une source de Markov, notée X_0^{N-1}.

S_i est l'état de la source de Markov à l'instant i.

L'algorithme utilise les informations suivantes :

- $p_i(m' \mid m) = \Pr\{S_i = m' \mid S_{i-1} = m\}$ est la probabilité de transition de l'état m à l'état m'.
- $q_i(X \mid m,m') = \Pr\{X_i = X \mid S_{i-1} = m; S_i = m'\}$ est la probabilité que la sortie vaille X sachant la transition.
- $p(Y_i \mid X_i)$ est la probabilité a posteriori sur la sortie du canal Y_i sachant l'entrée X_i.

L'algorithme Aller-Retour traite la séquence reçue Y_0^{N-1} pour estimer les APPs des transitions :

$$\Pr\{S_{i-1} = m \ ; \ S_i = m' \mid Y_0^{N-1}\} = \frac{\Pr\{S_{i-1} = m \ ; \ S_i = m' \ ; \ Y_0^{N-1}\}}{\Pr\{Y_0^{N-1}\}}$$

Il est plus simple de calculer la probabilité conjointe

$$\sigma_i(m,m') = \Pr\{S_{i-1} = m \;;\; S_i = m' \;;\; Y_0^{N-1}\}$$

et de normaliser ensuite $\sigma_i(m,m')$ pour trouver la probabilité a posteriori.

On définit

- $\alpha_i(m) = \Pr\{S_i = m \;;\; Y_0^i\}$, la probabilité conjointe d'être dans l'état m à l'instant i et d'avoir une observation *passée* Y_0^i de l'instant 0 à l'instant i.
- $\beta_i(m) = \Pr\{Y_{i+1}^{N-1} \mid S_i = m\}$, la probabilité d'avoir une observation *future* Y_{i+1}^{N-1} de l'instant $i+1$ à l'instant $N-1$ sachant que l'état à l'instant i est l'état m.
- $\gamma_i(m,m') = \Pr\{S_i = m' \;;\; Y_i \mid S_{i-1} = m\}$, la métrique utilisée par l'algorithme, qui est la probabilité d'arriver à l'état m' et d'observer Y_i à l'instant i sachant que l'état à l'instant précédent était l'état m.

Puisque la source est markovienne et que le treillis débute et s'achève dans l'état 0, les APPs peuvent se calculer en quatre étapes :

- **Initialisation**

$$\begin{array}{c} \alpha_0(0) = 1 \quad \text{et} \quad \alpha_0(m) = 0 \quad \forall m \neq 0 \\ \beta_{N-1}(0) = 1 \quad \text{et} \quad \beta_{N-1}(m) = 0 \quad \forall m \neq 0 \end{array}$$

- **Procédure *Aller***

$$\forall i = 0, 1, \ldots, N-1, \quad \alpha_i(m') = \sum_m \gamma_i(m,m')\alpha_{i-1}(m)$$

- **Procédure *Retour***

$$\forall i = N-2, \ldots, 1, 0, \quad \beta_i(m) = \sum_{m'} \gamma_{i+1}(m,m')\beta_{i+1}(m')$$

- **Calcul des APPs**

$$\sigma_i(m,m') = \alpha_{i-1}(m) \times \gamma_i(m,m') \times \beta_i(m')$$

La métrique de l'algorithme Aller-Retour peut s'exprimer comme fonction des entrées :

$$\gamma_i(m,m') = \sum_X p_i(m' \mid m) \times q_i(X \mid m \;,\; m') \times p(Y_i \mid X_i)$$

A.2 Application aux codes convolutifs

On note X_0^{N-1} la séquence représentant la sortie d'un code convolutif de taux k/n, X_i la sortie à l'instant i et Y_i l'observation correspondante.

$$\begin{array}{rcl} X_i &=& (x_{i1} \;,\; x_{i2} \;,\; \ldots,\; x_{in}) \\ Y_i &=& (y_{i1} \;,\; y_{i2} \;,\; \ldots,\; y_{in}) \end{array}$$

Les observations sur les différents bits sont indépendantes, donc

$$p(Y_i \mid X_i) = \prod_{u=1}^{n} p(y_{iu} \mid x_{iu})$$

Pour calculer les $\mathrm{APP}(x_{i\ell} = 0)$, on somme les $\sigma_i(m,m')$ correspondant à toutes les transitions sur lesquelles $x_{i\ell} = 0$:

$$\mathrm{APP}(x_{i\ell} = 0) \;=\; \sum_{m,m' \,\mid\, x_{i\ell}=0} \sigma_i(m,m') = \sum_{m,m' \,\mid\, x_{i\ell}=0} \alpha_{i-1}(m) \times \gamma_i(m,m') \times \beta_i(m') \quad \text{(A.1)}$$

Comme une transition donnée ne comporte qu'une unique valeur de X_i, $q_i(X \mid m,m')$ n'a qu'une seule valeur non nulle égale à 1. La probabilité a priori $p_i(m' \mid m)$ est égale à 0.5^k car on ne dispose d'aucune information a priori sur les k bits d'information. Finalement, la probabilité a posteriori peut être réécrite comme

$$\mathrm{APP}(x_{i\ell} = 0) \;=\; 0.5^k \sum_{m,m' \,\mid\, x_{i\ell}=0} \alpha_{i-1}(m) \times \prod_{u=1}^{n} p(y_{iu} \mid x_{iu}) \times \beta_i(m') \quad \text{(A.2)}$$

Si le code convolutif est non systématique, les APPs des bits d'information ont une expression similaire à (A.1) ou (A.2) et la sommation dépend maintenant de la valeur de l'entrée du codeur.

Annexe B

Algorithme Max-Log-MAP

L'algorithme Max-Log-MAP [20] est une approximation de l'algorithme Aller-Retour présenté dans l'annexe A. Pour utiliser cette approximation, il faut travailler dans le domaine logarithmique. C'est pourquoi on introduit la notion de rapport de vraisemblance logarithmique (LLR pour *Log-Likelihood Ratio*) défini comme suit pour une variable aléatoire X pouvant prendre deux valeurs 0 ou 1 :

$$\mathrm{LLR}(X) = \ln\left(\frac{\Pr(X=1)}{\Pr(X=0)}\right)$$

On s'intéresse au cas particulier du décodage des codes convolutifs. On note toujours X_0^{N-1} la séquence représentant la sortie du code de taux k/n, X_i la sortie à l'instant i et Y_i l'observation associée. La LLR du bit codé $x_{i\ell}$ s'écrit d'après l'équation (A.1)

$$\mathrm{LLR}(x_{i\ell}) = \ln\left(\frac{\displaystyle\sum_{m,m' \ | \ x_{i\ell}=1} \alpha_{i-1}(m) \times \gamma_i(m,m') \times \beta_i(m')}{\displaystyle\sum_{m,m' \ | \ x_{i\ell}=0} \alpha_{i-1}(m) \times \gamma_i(m,m') \times \beta_i(m')}\right) \tag{B.1}$$

Les procédures Aller et Retour normalisées s'écrivent :

– **Procédure *Aller***

$$\forall i = 0,\, 1, \ldots,\, N-1, \quad \alpha_i(m') = \frac{\displaystyle\sum_m \gamma_i(m,m')\alpha_{i-1}(m)}{\displaystyle\sum_{m'}\sum_m \gamma_i(m,m')\alpha_{i-1}(m)} \tag{B.2}$$

– **Procédure *Retour***

$$\forall i = N-2, \ldots,\, 1,\, 0, \quad \beta_i(m) = \frac{\displaystyle\sum_{m'} \gamma_{i+1}(m,m')\beta_{i+1}(m')}{\displaystyle\sum_{m}\sum_{m'} \gamma_{i+1}(m,m')\beta_{i+1}(m')} \tag{B.3}$$

Pour simplifier les calculs, on doit s'intéresser aux logarithmes des différentes variables et les procédures (B.2) et (B.3) deviennent :

$$\begin{aligned} \bar{\alpha}_i(m') &= \ln\left(\alpha_i(m')\right) \\ &= \ln\left(\sum_m \gamma_i(m,m')\alpha_{i-1}(m)\right) - \ln\left(\sum_{m'}\sum_m \gamma_i(m,m')\alpha_{i-1}(m)\right) \end{aligned} \tag{B.4}$$

$$\begin{aligned} \bar{\beta}_i(m) &= \ln\left(\beta_i(m)\right) \\ &= \ln\left(\sum_{m'} \gamma_{i+1}(m,m')\beta_{i+1}(m')\right) - \ln\left(\sum_{m}\sum_{m'} \gamma_{i+1}(m,m')\beta_{i+1}(m')\right) \end{aligned} \tag{B.5}$$

On peut maintenant utiliser l'approximation qui est à la base de la simplification :

$$\ln(e^{\delta_1} + \ldots + e^{\delta_n}) \approx \max_{i\in\{1\ldots n\}} \delta_i \tag{B.6}$$

En appliquant l'approximation (B.6) aux équations (B.4) et (B.5), on obtient

$$\begin{aligned} \bar{\alpha}_i(m') &= \ln\left(\sum_m e^{\bar{\gamma}_i(m,m')+\bar{\alpha}_{i-1}(m)}\right) - \ln\left(\sum_{m'}\sum_m e^{\bar{\gamma}_i(m,m')+\bar{\alpha}_{i-1}(m)}\right) \\ &\approx \max_m\left(\bar{\gamma}_i(m,m') + \bar{\alpha}_{i-1}(m)\right) - \max_{m',m}\left(\bar{\gamma}_i(m,m') + \bar{\alpha}_{i-1}(m)\right) \end{aligned}$$

$$\begin{aligned} \bar{\beta}_i(m) &= \ln\left(\sum_{m'} e^{\bar{\gamma}_{i+1}(m,m')+\bar{\beta}_{i+1}(m')}\right) - \ln\left(\sum_{m}\sum_{m'} e^{\bar{\gamma}_{i+1}(m,m')+\bar{\beta}_{i+1}(m')}\right) \\ &\approx \max_{m'}\left(\bar{\gamma}_{i+1}(m,m') + \bar{\beta}_{i+1}(m')\right) - \max_{m,m'}\left(\bar{\gamma}_{i+1}(m,m') + \bar{\beta}_{i+1}(m')\right) \end{aligned}$$

avec

$$\bar{\gamma}_i(m,m') = \ln\left(\gamma_i(m,m')\right)$$

De la même manière on peut trouver une approximation de la LLR (B.1)

$$\begin{aligned} \mathrm{LLR}(x_{i\ell}) \approx & \max_{m,m' \mid x_{i\ell}=0}\left(\bar{\alpha}_{i-1}(m) + \bar{\gamma}_i(m,m') + \bar{\beta}_i(m')\right) \\ & - \max_{m,m' \mid x_{i\ell}=1}\left(\bar{\alpha}_{i-1}(m) + \bar{\gamma}_i(m,m') + \bar{\beta}_i(m')\right) \end{aligned}$$

Cet algorithme est appliqué dans la section 3.3.3 à un système multi-utilisateurs. Dans ce cas, le calcul de la métrique $\bar{\gamma}_i(m,m')$ fait également appel à l'approximation (B.6).

Annexe C

Algorithme Log-MAP

L'algorithme Max-Log-MAP, présenté dans l'annexe B est sous-optimal et donne donc de moins bons résultats que l'algorithme Aller-Retour classique présenté dans l'annexe A. Pour retrouver les performances de ce dernier dans le domaine logarithmique, on doit calculer $\ln\left(e^{\delta_1}+\ldots+e^{\delta_n}\right)$ de manière exacte au lieu d'utiliser l'approximation $\max_{i\in\{1\ldots n\}}\delta_i$ de l'égalité (B.6). L'utilisation du logarithme Jacobien permet de trouver une solution à ce problème [20] :

$$\ln\left(e^{\delta_1}+e^{\delta_2}\right)=\max\left(\delta_1,\delta_2\right)+\ln\left(1+e^{-|\delta_1-\delta_2|}\right) \tag{C.1}$$

On nomme fonction de correction la fonction $f_c(x)$ telle que :

$$f_c(x)=\ln\left(1+e^{-x}\right),\quad \forall x\in\mathbb{R}^+$$

On peut démontrer par récurrence que l'on sait calculer exactement $\ln\left(e^{\delta_1}+\ldots+e^{\delta_n}\right)$. On initialise la récurrence avec l'équation (C.1). On suppose ensuite que l'on a déjà calculé

$$\delta=\ln\left(e^{\delta_1}+\ldots+e^{\delta_{n-1}}\right)$$

On peut alors écrire

$$\begin{aligned}\ln\left(e^{\delta_1}+\ldots+e^{\delta_n}\right)&=\ln\left(e^{\delta}+e^{\delta_n}\right)\\&=\max\left(\delta,\delta_n\right)+f_c\Big(|\delta-\delta_n|\Big)\end{aligned}$$

Il faut donc procéder par recherches successives du maximum entre deux termes et corriger à chaque fois le maximum grâce à la fonction de correction f_c. L'algorithme Aller-Retour est ainsi préservé dans le domaine logarithmique mais le calcul de la fonction f_c nous fait perdre le gain en simplicité acquis grâce à l'algorithme Max-Log-MAP. Pour simplifier le calcul, on peut se contenter de calculer par avance un faible nombre de

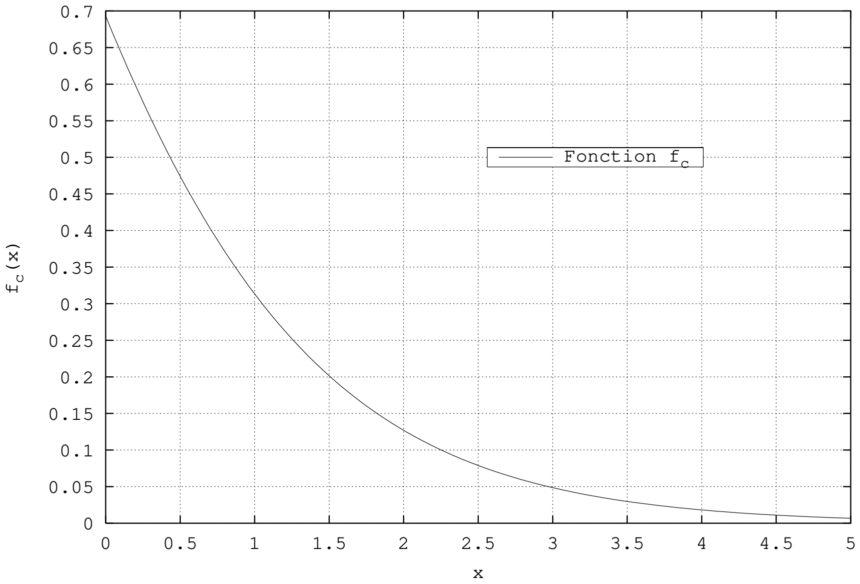


FIG. C.1 – *Fonction de correction f_c.*

valeurs de la fonction f_c [57][58]. Cette approximation ne fait pas énormément chuter les performances. On peut voir sur la figure C.1 la courbe de la fonction de correction, dont les valeurs sont comprises entre 0 et ln(2).

Cet algorithme est appliqué à un système multi-utilisateurs dans la section 3.3.3.

Annexe D

Calcul des polynômes énumérateurs de poids conditionnels

Nous désirons calculer les polynômes énumérateurs de poids conditionnels d'un code convolutif non systématique en nous inspirant de la méthode présentée dans [16] et [2]. Nous nous intéressons à une portion de treillis longue de w branches ou $W = 2w$ bits codés, avec w impair, pour un code convolutif non systématique de taux 1/2. Aucun état de départ ni d'arrivée n'est spécifié, la portion de treillis étant choisie arbitrairement au milieu d'un treillis de taille infinie.

Le diagramme d'états d'un code convolutif décrit les transitions entre les états du code, chaque transition étant étiquetée par la valeur du bit d'information i et les valeurs des 2 bits codés d_1 et d_2 (i/d_1d_2). La figure D.1 présente le schéma du codeur du code (7,5) non systématique et la figure D.2 le diagramme d'états correspondant.

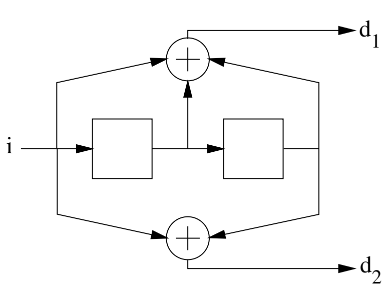


FIG. D.1 – *Schéma du codeur (7,5) non systématique*

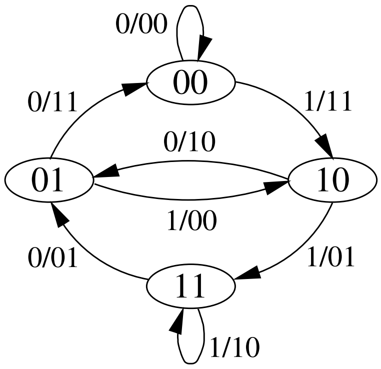


FIG. D.2 – *Diagramme d'états d'un code (7,5) non systématique*

On construit la matrice de transfert du code **T** en plaçant dans chaque élément

T_{jk} une valeur correspondant à la transition de l'état j vers l'état k. Cette valeur est nulle si la transition n'existe pas. Si la transition existe, on met une valeur $LI^iD_1{}^{d_1}D_2{}^{d_2}$ dépendant de la valeur i du bit d'information et des valeurs d_1 et d_2 des bits codés. Pour le code (7,5) non systématique, on obtient la matrice de transfert suivante :

$$\mathbf{T} = \begin{bmatrix} L & 0 & LID_1D_2 & 0 \\ LD_1D_2 & 0 & LI & 0 \\ 0 & LD_1 & 0 & LID_2 \\ 0 & LD_2 & 0 & LID_1 \end{bmatrix}$$

Intéressons-nous maintenant aux 4 matrices de transfert obtenues en fixant la valeur d'un des deux bits codés. On note $\mathbf{T}_{d_1=1}$ la matrice de transfert telle que le premier bit codé ait la valeur '1'. De même on définit $\mathbf{T}_{d_1=0}$, $\mathbf{T}_{d_2=1}$ et $\mathbf{T}_{d_2=0}$. Pour construire chacune de ces 4 matrices, on supprime dans $\mathbf{T}$ les transitions ne vérifiant pas la condition. On obtient ainsi pour le code (7,5) :

$$\mathbf{T}_{d_1=1} = \begin{bmatrix} 0 & 0 & LID_1D_2 & 0 \\ LD_1D_2 & 0 & 0 & 0 \\ 0 & LD_1 & 0 & 0 \\ 0 & 0 & 0 & LID_1 \end{bmatrix} \quad ; \quad \mathbf{T}_{d_1=0} = \begin{bmatrix} L & 0 & 0 & 0 \\ 0 & 0 & LI & 0 \\ 0 & 0 & 0 & LID_2 \\ 0 & LD_2 & 0 & 0 \end{bmatrix}$$

$$\mathbf{T}_{d_2=1} = \begin{bmatrix} 0 & 0 & LID_1D_2 & 0 \\ LD_1D_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & LID_2 \\ 0 & LD_2 & 0 & 0 \end{bmatrix} \quad ; \quad \mathbf{T}_{d_2=0} = \begin{bmatrix} L & 0 & 0 & 0 \\ 0 & 0 & LI & 0 \\ 0 & LD_1 & 0 & 0 \\ 0 & 0 & 0 & LID_1 \end{bmatrix}$$

De même on construit $\mathbf{T}_{i=1}$ et $\mathbf{T}_{i=0}$ les 2 matrices de transfert telles que le bit d'information i ait une valeur donnée. Pour le code (7,5) non systématique, on obtient :

$$\mathbf{T}_{i=1} = \begin{bmatrix} 0 & 0 & LID_1D_2 & 0 \\ 0 & 0 & LI & 0 \\ 0 & 0 & 0 & LID_2 \\ 0 & 0 & 0 & LID_1 \end{bmatrix} \quad ; \quad \mathbf{T}_{i=0} = \begin{bmatrix} L & 0 & 0 & 0 \\ LD_1D_2 & 0 & 0 & 0 \\ 0 & LD_1 & 0 & 0 \\ 0 & LD_2 & 0 & 0 \end{bmatrix}$$

Nous nous intéressons à une fenêtre de w branches du code. Le coefficient de la ligne j et de la colonne k de la matrice $\mathbf{T}^w$ obtenue en élevant la matrice $\mathbf{T}$ à la puissance w est de la forme

$$\mathcal{A}_{jk}(L,I,D_1,D_2) = L^w \sum_{i=0}^{w} \sum_{d_1=0}^{w} \sum_{d_2=0}^{w} I^i D_1{}^{d_1} D_2{}^{d_2} A_{jk}(i,d_1,d_2)$$

En posant $L = 1$, $I = 1$ et $D_1 = D_2$, on obtient

$$\mathcal{A}_{jk}(D) = \sum_{d=0}^{2w} D^d A_{jk}(d)$$

où $A_{jk}(d)$ est le nombre de mots de code de longueur w, de poids d, partant de l'état j et arrivant à l'état k. Le nombre total de mots de code de longueur w, de poids d, partant d'un état quelconque et arrivant à un état quelconque est

$$A(d) = \sum_j \sum_k A_{jk}(d)$$

Pour obtenir $A_{d_1=1}(d)$, le nombre de mots de code de longueur w tels que le premier bit de la branche au milieu de la fenêtre soit égal à '1', il ne faut plus considérer la matrice $\mathbf{T}^w$ mais la matrice $\mathbf{T}^{(w-1)/2}\mathbf{T}_{d_1=1}\mathbf{T}^{(w-1)/2}$. On procède de même pour obtenir $A_{d_1=0}(d)$, $A_{d_2=1}(d)$, $A_{d_2=0}(d)$, $A_{i=1}(d)$ et $A_{i=0}(d)$. Ces distributions de poids conditionnelles sont employées dans la section 3.6 pour déterminer les performances d'un système multi-utilisateurs avec décodage itératif quand la longueur de l'entrelaceur est infinie.

Bibliographie

[1] L.R. Bahl, J. Cocke, F. Jelinek, J. Raviv : "Optimal decoding of linear codes for minimizing symbol error rate", *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284-287, Mars 1974.

[2] S. Benedetto, G. Montorsi : "Unveiling turbo-codes : some results on parallel concatenated coding schemes", *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 409-428, Mars 1996.

[3] S. Benedetto, G. Montorsi : "Design of parallel concatenated convolutional codes", *IEEE Transactions on Communications*, vol. 44, no. 5, pp. 591-600, Mai 1996.

[4] C. Berrou, A. Glavieux, P. Thitimajshima : "Near Shannon limit error-correcting coding and decoding : turbo-codes", *Proceedings of ICC'93*, Genève, pp. 1064-1070, Mai 1993.

[5] S. Botzas, R. Hammons, P.V. Kumar : "4-phase sequences with near-optimum correlation properties", *IEEE Transactions on Information Theory*, vol. 38, no. 3, pp. 1101-1113, Mai 1992.

[6] J. Boutros, E. Viterbo, C. Rastello and J.C. Belfiore : "Good lattice constellations for both Rayleigh fading and Gaussian channels", *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 502-518, Mars 1996.

[7] J. Boutros : "Réseaux de points pour les canaux à évanouissements", Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications, 1996.

[8] J. Boutros, O. Pothier, G. Zémor : "Generalized low density (Tanner) codes", *Proceedings of ICC'99*, Vancouver, Canada, Juin 1999.

[9] J. Boutros, S. Vialle : "Performance Limits of Concatenated Codes with Iterative Decoding", soumis à *ISIT'2000*, Sorrento, Italie, Septembre 1999.

[10] G. Caire : "Adaptive linear receivers for DS/CDMA. Part I : steady-state performance analysis", soumis à *IEEE Transactions on Communications*, Janvier 1999.

[11] G. Caire : "Adaptive linear receivers for DS/CDMA. Part II : improved non-data aided schemes", soumis à *IEEE Transactions on Communications*, Janvier 1999.

[12] N. Chayat, S. Shamai (Shitz) : "Convergence properties of iterative soft onion pee-

ling", *Proceedings of ITW'99*, Afrique du Sud, pp. 9-11, Juin 1999.

[13] H. Cohen : *Computational algebraic number theory*, Springer Verlag 1993.

[14] J.H. Conway, N.J. Sloane : *Sphere packings, lattices and groups*, 3rd ed., 1998, Springer-Verlag, New York.

[15] O. Damen, J.C. Belfiore : "Joint Multiuser Decoding", *Proceedings of ICT'98*, Porto Carras, Grèce, vol. 2, pp. 62-66, Juin 1998.

[16] D. Divsalar, S. Dolinar, R.J. McEliece, F. Pollara : "Transfer function bounds on the performance of turbo codes", *TDA Progress Report 42-122*, JPL, Août 1995.

[17] L. Duan, B. Rimoldi, R. Urbanke : "Approaching the AWGN channel capacity without active shaping", *Proceedings of ISIT'97*, Ulm, Allemagne, p. 374, Juillet 1997.

[18] A. Duel-Hallen : "Decorrelating decision-feedback multi-user detector for synchronous code-division multiple access channel", *IEEE Transactions on Communications*, vol. 41, no. 2, pp. 285-290, Février 1993.

[19] A. Duel-Hallen : "A family of multiuser decision-feedback detectors for asynchronous code-division multiple-access channels", *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 421-434, Fév./Mars/Avr. 1995.

[20] J.A. Erfanian, S. Pasupathy, G. Gulak : "Reduced complexity symbol detectors with parallel structures for ISI channels", *IEEE Transactions on Communications*, pp. 1661-1671, Fév./Mars/Avr. 1994.

[21] P. Fan, M. Darnell : *Sequence design for communications applications*, Wiley, 1996.

[22] U. Fawer, B. Aazhang : "A multiuser receiver for code division multiple access communications over multipath channels", *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 1556-1565, Fév./Mars/Avr. 1995.

[23] U. Fincke, M. Pohst : "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis", *Mathematics of computation*, vol. 44, pp. 463-471, Avril 1985.

[24] G.D. Jr. Forney : "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference", *IEEE Transactions on Information Theory*, vol. 18, pp. 363-378, Mai 1992.

[25] G.D. Jr. Forney : "Coset codes I : introduction and geometrical classification", *IEEE Transactions on Information Theory*, vol. 34, pp. 1123-1151, Septembre 1988.

[26] R.G. Gallager : *Low-density parity-check codes*, MIT Press, 1963.

[27] R.G. Gallager : *Information theory and reliable communication*, Wiley, 1968, p. 74.

[28] R. Gold : "Optimal binary sequences for spread spectrum multiplexing", *IEEE Transactions on Information Theory*, vol. 13, pp. 619-621, 1967.

[29] R. Gold : "Maximal recursive sequences with 3-valued recursive crosscorrelation functions", *IEEE Transactions on Information Theory*, vol. 14, pp. 154-156, 1968.

[30] T. Helleseth : "Some results about the cross-correlation function between two maximal linear sequences", *Discrete Math.*, vol. 16, pp. 209-232, 1976.

[31] J. Hadamard : "Résolution d'une question relative aux déterminants", *Bulletin Scientifique des Mathématiques*, vol. 17, no. 2, pp. 240-248, 1893.

[32] N. Ibrahim : "Codage et décodage de canal pour un système de communication à accès multiple", Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications, 1999.

[33] T. Kasami : *Combinatorial mathematics and its applications*, "Weight distribution of Bose-Chaudhuri-Hocquenghem codes", Chapel Hill, 1969. Réimprimé dans *Key papers in the development of Coding Theory*, E.R. Berlekamp, IEEE Press, 1974.

[34] A. Klein, G.K. Kaleh, P.W. Baier : "Zero forcing and minimum mean-square-error equalization for multiuser detection in code-division multiple-access channels", *IEEE Transactions on Vehicular Technology*, vol. 45, no. 2, pp. 276-287, Mai 1996.

[35] R. Kohno, H. Imai, M. Hatori, S. Pasupathy : "An adaptative canceller of cochannel interference for spread-spectrum multiple-access communication networks in a power line", *Journal on Selected Areas in Communications*, vol. 8, no. 4, pp. 691-699, Mai 1990.

[36] R. Kohno : "Pseudo-noise sequences and interference cancellation techniques for spread spectrum systems-spread spectrum theory and techniques in Japan", *IEICE Transactions*, vol. E.74, pp. 1083-1092, Mai 1991.

[37] A.W. Lam, F.M. Özlütürk : "Performance bounds for DS/SSMA communications with complex signature sequences", *IEEE Transactions on Communications*, vol. 40, no. 10, pp. 1607-1614, Octobre 1992.

[38] R. Lupas, S. Verdú : "Linear multiuser detectors for synchronous code-division multiple-access channels", *IEEE Transactions on Information Theory*, vol. 35, no. 1, pp. 123-136, Janvier 1989.

[39] R. Lupas, S. Verdú : "Near-far resistance of multiuser detectors in asynchronous channels", *IEEE Transactions on Communications*, vol. 38, no. 4, pp. 496-508, Avril 1990.

[40] U. Madhow, M.L. Honig : "MMSE interference suppression for direct-sequence spread spectrum CDMA", *IEEE Transactions on Communications*, vol. 42, no. 12, pp. 3178-3188, Décembre 1994.

[41] R.J. McEliece : *Finite fields for computer scientists and engineers*, Kluwer, 1987.

[42] G. McGuire, A.R. Calderbank : "Proof of a conjecture of Sarwate and Pursley regarding pairs of binary m-sequences", *IEEE Transactions on Information Theory*,

vol. 41, no. 4, pp. 1153-1155, Juillet 1995.

[43] M. Moher : "An iterative multiuser decoder for near-capacity communications", *IEEE Transactions on Communications*, vol. 46, pp. 870-880, Juillet 1998.

[44] S. Moshavi : "Multi-user detection for DS-CDMA communications", *IEEE Communications Magazine*, pp. 124-136, Octobre 1996.

[45] L.B. Nelson, H.V. Poor : "Iterative multiuser receivers for CDMA channels : an EM-based approach", *IEEE Transactions on Communications*, vol. 44, no. 12, pp. 1700-1710, Décembre 1996.

[46] E. Papproth : "Conception et traitement de signal pour les systèmes radiomobiles AMRC", Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications, 1996.

[47] P. Patel, J. Holtzman : "Performance comparison of a DS/CDMA system using a successive interference cancellation (IC) scheme and a parallel IC scheme under fading", *Proceedings of ICC'94*, New Orleans, USA, pp. 510-514, Mai 1994.

[48] P. Patel, J. Holtzman : "Analysis of a simple successive interference cancellation scheme in a DS/CDMA system", *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 5, pp. 796-807, Juin 1994.

[49] M. Pohst, H. Zassenhaus : *Algorithmic Algebraic Number Theory*, Cambridge University Press, 1989.

[50] O. Pothier : "Design and decoding of compound codes : turbo and low density codes", Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications, 1999.

[51] R. Price, P.E. Jr. Green : "A communication technique for multipath channels", *Proc. IRE*, vol. 46, pp. 555-570, Mars 1958.

[52] J.G. Proakis : *Digital communications, third edition*, McGraw-Hill, 1995.

[53] M.B. Pursley : "Performance evaluation for phase-coded spread-spectrum multiple-access communication - part I : system analysis", *IEEE Transactions on Communications*, vol. 25, no. 8, pp. 795-799, Août 1977.

[54] M.C. Reed, C.B. Schlegel, P.D. Alexander, J.A. Asenstorfer : "Iterative multiuser detection for CDMA with FEC: near-single-user performance", *IEEE Transactions on Communications*, vol. 46, pp. 1693-1699, Décembre 1998.

[55] T. Richardson, R. Urbanke : "The capacity of low-density parity check codes under message-passing decoding", *Bell Labs Report*, Novembre 1998.

[56] B. Rimoldi, R. Urbanke : "A rate-splitting approach to the Gaussian multiple-access channel", *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 364-375, Mars 1996.

[57] P. Robertson, E. Villebrun, P. Hoeher : “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain”, *Proceedings of ICC'95*, pp. 1009-1013, Juin 1995.

[58] P. Robertson, P. Hoeher, E. Villebrun : “Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding”, *European Transactions on Telecommunications*, vol. 8, no. 2, pp. 119-125, Mars/Avril 1997.

[59] Y. Sanada and Q. Wang : “A co-channel interference cancellation technique using orthogonal convolutional codes”, *IEEE Transactions on Communications*, vol. 44, no. 5, pp. 549-556, Mai 1996.

[60] Y. Sanada and Q. Wang : “A co-channel interference cancellation technique using orthogonal convolutional codes on multipath Rayleigh fading channel”, *IEEE Transactions on Vehicular Technology*, vol. 46, no. 1, pp. 114-128, Février 1997.

[61] H. Sari, F. Vanhaverbeke, M. Mœneclaey : “Some novel concepts in multiplexing and multiple access”, présenté au *Second International Workshop on Multicarrier Spread Spectrum (MCSS'99)*, Oberpfaffenhofen, Allemagne, Septembre 1999.

[62] D.V. Sarwate, M.B. Pursley : “Crosscorrelation properties of pseudo-random and related sequences”, *Proceedings of the IEEE*, vol. 68, no. 5, pp. 593-619, Mai 1980.

[63] V.M. Sidelnikov : “On mutual correlation of sequences”, *Soviet Math Doklady*, pp. 197-201, no. 12, 1971.

[64] M.K. Simon, J.K. Omura, R.A. Scholtz, B.K. Levitt : *Spread spectrum communications*, 3 vol., Computer Science Press, 1985.

[65] P. Solé : *Coding theory and applications*, *Lecture notes in computer science*, vol. 388, Springer-Verlag, 1989.

[66] F. Tarköy : “Iterative multi-user decoding for asynchronous users”, *Proceedings of ISIT'97*, Ulm, Germany, p. 30, Juillet 1997.

[67] H.M. Trachtenberg : *On the cross-correlation functions of maximal recurring sequences*, PhD thesis, University of Southern California, 1970.

[68] G. Ungerboeck : “Channel coding with multilevel/phase signals”, *IEEE Transactions on Information Theory*, pp. 55-67, Janvier 1982.

[69] R. Vallet : “Applications de l'identification des modèles de Markov cachés aux communications numériques”, Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications, 1991.

[70] M.K. Varanasi and B. Aazhang : “Multistage detection in asynchronous code division multiple access communications”, *IEEE Transactions on Communications*, vol. 38, no. 4, pp. 509-519, Avril 1990.

[71] M.K. Varanasi and B. Aazhang : “Near-optimum detection in synchronous code

division multiple access systems", *IEEE Transactions on Communications*, vol. 39, no. 5, pp. 725-736, Mai 1991.

[72] S. Verdú : "Minimum probability of error for asynchronous Gaussian multiple-access channels", *IEEE Transactions on Information Theory*, pp. 85-96, Janvier 1986.

[73] S. Verdú : *Multiuser detection*, Cambridge, 1998.

[74] A.J. Viterbi : "Error bound for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Transactions on Information Theory*, vol. 13, pp. 260-269, Avril 1967.

[75] A.J. Viterbi : "Very low rate convolutional codes for maximum theoritical performance of spread-spectrum multiple-access channels", *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 4, pp. 641-649, Mai 1990.

[76] A.J. Viterbi : *CDMA : principles of spread spectrum*, Addison-Wesley, 1995.

[77] E. Viterbo and E. Biglieri : "A universal lattice decoder", *Proceedings of 14ème Colloque GRETSI*, Juan-les-Pins, pp. 611-614, Septembre 1993.

[78] E. Viterbo and J. Boutros : "A universal lattice code decoder for fading channels", *IEEE Transactions on Information Theory*, vol. 45, pp. 1639-1642, Juillet 1999.

[79] Z. Xie, R. Short, C.K. Rushforth : "A family of suboptimum detectors for coherent multiuser communications", *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 4, pp. 683-690, Mai 1990.

[80] K. Yao : "Error probability of asynchronous spread spectrum multiple access communication systems", *IEEE Transactions on Communications*, vol. 25, no. 8, pp. 803-809, Août 1977.

Liste des publications

(Disponibles à l'adresse http://www.comelec.enst.fr/publications)

[81] L. Brunel, J. Boutros : "Iterative joint decoding for multiuser detection", présenté au *Mediterranean Workshop on Coding and Information Integrity*, Ein Boqeq, Israël, Octobre 1997.

[82] L. Brunel, J. Boutros : "Euclidean space lattice decoding for joint detection in CDMA systems", *Proceedings of ITW'99*, Afrique du Sud, p. 129, Juin 1999.

[83] L. Brunel, J. Boutros : "Code division multiple access based on independent codes and turbo decoding", *Annales des Télécommunications*, tome 54, no. 7-8, pp. 401-410, Juillet/Août 1999.

[84] L. Brunel, J. Boutros : "Lattice decoding for joint detection in direct sequence CDMA systems", soumis à *IEEE Transactions on Information Theory*, Mars 1999.

[85] O. Pothier, L. Brunel, J. Boutros : "A low complexity FEC scheme based on the intersection of interleaved block codes", *Proceedings of VTC'99*, Houston, Mai 1999.