Détection itérative des séquences pseudo-aléatoires

Mathieu Bouvier Des Noes

UNIVERSITÉ GRENOBLE ALPES

THÈSE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE ALPES

Spécialité : **Signal Image Parole Télécoms**

Arrêté ministériel : 7 août 2006

Présentée par
Mathieu BOUVIER DES NOES

Thèse dirigée par **Jean-Marc BROSSIER** et
codirigée par **Laurent ROS et Valentin SAVIN**
préparée au sein du
Commissariat à l'Energie Atomique (CEA) et du GIPSA-LAB
dans **l'école doctorale Electronique Electrotechnique Automatique et Traitement du Signal (EEATS)**

Détection itérative des séquences pseudo-aléatoires

Thèse soutenue publiquement le **15 Octobre 2015**,
devant le jury composé de :

Philippe CIBLAT
Professeur, Institut Mines-Télécom / Télécom ParisTech, Président du jury
Charly POULLIAT
Professeur, INP - ENSEEIHT Toulouse, Rapporteur
Sébastien HOUCKE
Professeur, Institut Mines-Télécom / Télécom Bretagne, Rapporteur
Pierre LOIDREAU
Ingénieur, DGA/IRMAR, Membre
Jean-Marc BROSSIER
Professeur, GIPSA-LAB, Membre
Laurent ROS
Maître de conférence, GIPSA-LAB, Membre
Valentin SAVIN
Ingénieur de recherche, CEA, Membre

UNIVERSITÉ DE GRENOBLE ALPES

ÉCOLE DOCTORALE E.E.A.T.S

Electronique Electrotechnique Automatique et Traitement du Signal

T H È S E

pour obtenir le titre de

docteur en sciences

de l'Université de Grenoble

Mention : Signal Image Parole Télécoms

Présentée et soutenue par

Mathieu BOUVIER DES NOES

Détection itérative des séquences pseudo-aléatoires

Thèse dirigée par Jean-Marc BROSSIER

préparée au Commissariat à l'Energie Atomique (CEA) et au GIPSA-LAB

soutenue le date de soutenance

Jury :

Rapporteurs :	Charly POULLIAT	-	Professeur, INP - ENSEEIHT Toulouse
	Sébastien HOUCKE	-	Professeur, Institut Mines-Télécom /
		-	Télécom Bretagne
Directeur :	Jean-Marc BROSSIER	-	Professeur, GIPSA-LAB
Encadrant :	Laurent ROS	-	Maître de conférence, GIPSA-LAB
Encadrant :	Valentin SAVIN	-	Ingénieur de recherche, CEA
Président :	Philippe CIBLAT	-	Professeur, Institut Mines-Télécom /
		-	Télécom ParisTech
Examinateur :	Pierre LOIDREAU	-	Ingénieur, DGA/IRMAR

Remerciements

Démarrer une thèse après 40 ans n'est pas habituel. Cela faisait longtemps que j'y songeais, sans oser passer le cap. C'est finalement le sujet qui m'a aidé à me lancer dans l'aventure. Ce dernier est dans la lignée de mes activités au CEA-LETI depuis mon arrivée au centre de Grenoble en 1998. Ce sujet amène à revisiter une problématique très connue, la détection des séquences pseudo-aléatoires, mais sous un angle nouveau. Cette thèse m'a fait découvrir le travail de recherche sur un longue durée, a enrichi pleinement mes connaissances et ouvert mon appétit à poursuivre cette expérience. Je tiens à remercier Dimitri Ktenas, Laurent Hérault et Roland Blanpain, qui m'ont autorisé à entreprendre cette thèse. Je remercie aussi Florian Pebay-Peyroula et Dominique Noguet de m'avoir permis de la terminer dans de très bonnes conditions.

Je remercie aussi Laurent Ros, Valentin Savin et Jean-Marc Brossier pour m'avoir soutenu, par leur encadrement, tout au long de ces 4 années.

Table des matières

Table des figures

Liste des tableaux

Table des sigles et acronymes

AWGN	*Additive White Gaussian Noise*
BCH	*Bose Chauduri Hocquenghem*
BER	*Bit Error Rate*
BP	*Belief Propagation*
BPSK	*Binary Phase Shift Keying*
CDMA	*Code Division Multiple Access*
CDMA2000	*Code Division Multiple Access 2000*
FER	*Frame Error Rate*
LDPC	*Low Density Parity Check*
LFSR	*Linear Feedback Shift Register*
MAP	*Maximum A Posteriori*
ML	*Maximum Likelihood*
MLSE	*Maximum Likelihood Sequence Estimation*
MS	*Min-Sum*
SPA	*Sum-Product Algorithm*
WCDMA	*Wideband CDMA*
3GPP	*3rd Generation Partnership Project*

Notations

Les m-séquences seront notées s et y, et une séquence de Gold z. La lettre x est réservée pour la variable d'un polynôme, e.g. $g(x)$. L'indice k d'une séquence $s(k)$ est toujours évalué modulo la période de la séquence N : $s(k \bmod N)$. Une séquence binaire à valeurs dans $0; 1$ est notée en minuscule (e.g. $s(k)$) alors que sa représentation BPSK à valeurs dans $-1; +1$ est notée en majuscule : $S(k) = (-1)^{s(k)}$.

Dans cette thèse, nous allons régulièrement employer une notation polynomiale, qui sera utile pour définir les équations de parité des codes cycliques. Elle est définie de la manière suivante : à chaque mot de code $c = (c_0, \cdots, c_{n-1})$ est associé le polynôme $c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$. Les calculs sur les polynômes sont effectués modulo $x^n + 1$ où n est la taille des mots du code cyclique.

Publications

Ce travail de thèse a donné lieu à 4 papiers :

- [P1] M. des Noes, V. Savin, L. Ros et J.M. Brossier, *Blind Identification of the Uplink Scrambling Code Index of a WCDMA Transmission and Application to Femtocell Networks*, IEEE International Conference on Communications (ICC), Budapest, Hongrie, Juin 2013.
- [P2] M. des Noes, V. Savin, L. Ros et J.M. Brossier, *Blind Identification of the Scrambling Code of a Reverse Link CDMA 2000 Transmission*, IEEE International Conference on Communications (ICC), Budapest, Hongrie, Juin 2013.
- [P3] M. des Noes, V. Savin, L. Ros et J.M. Brossier, *Improving the Decoding of M-Sequences by Exploiting their Decimation Property*, European Signal Processing Conference (Eusipco), Marrakech, Maroc, Septembre 2013.
- [P4] M. des Noes, V. Savin, L. Ros et J.M. Brossier, *Iterative decoding of Gold sequences*, IEEE International Conference on Communications (ICC), Londres, Juin 2015.

une présentation aux journées codage et cryptographie :

- [P5] M. des Noes, V. Savin, L. Ros et J.M. Brossier, 'Sécurité des communications par étalement de spectre', Les 7 Laux, France, 24 – 28 Mars 2014.

et 5 brevets :

- [P6] 'Méthode d'estimation aveugle d'un code d'embrouillage d'une liaison montante WCDMA', E.N. 1252338.
- [P7] 'Méthode d'estimation aveugle d'un code d'embrouillage d'une liaison montante CDMA 2000', E.N. 1252340.
- [P8] 'Méthode d'acquisition d'un signal GPS par décodage itératif', E.N. 1351423.
- [P9] 'Méthode d'acquisition d'une séquence de Gold par double décodage itératif', E.N. 1458115
- [P10] 'Méthode de décodage itératif de séquences LFSR à faible probabilité de fausse alarme', E.N. 1551116.

Introduction

Une séquence binaire pseudo-aléatoire est représentée par une suite de '0' et de '1' qui semble aléatoire, alors qu'elle est en réalité parfaitement déterministe. Ces séquences sont employées pour construire des séquences d'apprentissage qui facilitent les opérations de synchronisation et d'estimation de canal dans les systèmes de transmission. Elles constituent aussi un élément fondamental des systèmes de transmission par étalement de spectre [1]. Elles sont enfin utilisées dans le domaine de la cryptographie pour réaliser des chiffreurs par flot [2]. Ces séquences sont donc présentes dans les systèmes radiomobiles WCDMA (Wideband Code Division Multiple Access), CDMA2000 et LTE (Long Term Evolution) [3][4][5], les systèmes de localisation tels que le GPS (Global Positioning System) et Galileo [6], mais aussi le mécanisme de chiffrement du système Bluetooth [7].

Les séquences pseudo-aléatoires sont très majoritairement construites à partir de registres à décalage rebouclés, d'où leur nom en anglais : Linear Feedback Shift Register sequence (LFSR). Les séquences LFSR se génèrent très facilement et possèdent de bonnes propriétés stochastiques si les coefficients du rebouclage sont choisis correctement. Les séquences les plus utilisées sont les m-séquences, les séquences de Gold et les séquences de Kasami [8][9].

Dans les systèmes utilisant des séquences pseudo-aléatoires, le récepteur cherche toujours à se synchroniser avec les séquences reçues. Pour cela, la méthode conventionnelle consiste à générer la séquence et réaliser une corrélation avec le signal reçu. La synchronisation est dite acquise si la corrélation dépasse un seuil prédéfini. Cette méthode est très efficace si la séquence est relativement courte. Dans le cas d'une séquence longue (e.g. 2^{25} chips pour le système WCDMA), on ne peut plus appliquer directement cette méthode. Il faut mettre en œuvre des étapes intermédiaires afin de limiter le nombre de séquences à tester et d'estimer l'instant de synchronisation. Par exemple, le système WCDMA a défini deux phases intermédiaires (*primary and secondary synchronization*) pour limiter à 8 le nombre de séquences à tester [3] et déterminer le début de la séquence. Ces procédures augmentent la signalisation du système et aussi les calculs réalisés par le récepteur. Dans cette thèse, nous allons étudier une méthode de détection alternative, mettant en œuvre des techniques de décodage itératif. Elle permet entre autre de ne pas avoir à recourir à des étapes intermédiaires pour détecter les longues séquences.

La découverte des turbo-codes par Berrou et Glavieux [10] a engendré de nombreuses recherches sur les mécanismes de décodage itératif. La redécouverte des codes de Gallager [11] par Mac Kay et Neal [12] est intervenu dans ce contexte historique. Les codes de Gallager sont des codes linéaires en block $[n, k]$ [13] dont la matrice de parité $\mathbf{E}$ est creuse. Cela définit la famille des codes Low Density Parity Check (LDPC). Le décodeur a pour fonction de déduire, à partir des échantillons reçus, les bits $\mathbf{s} = (s_1 \cdots s_n)$ qui vérifient les $n-k$ équations de parité qui définissent le code. Cela se caractérise par la propriété suivante : $\mathbf{E}\mathbf{s}^T = 0$. Une équation de parité correspond à une ligne de la matrice $\mathbf{E}$. Un algorithme permettant de résoudre ce type de problème a été découvert plusieurs fois dans des domaines différents : décodage pro-

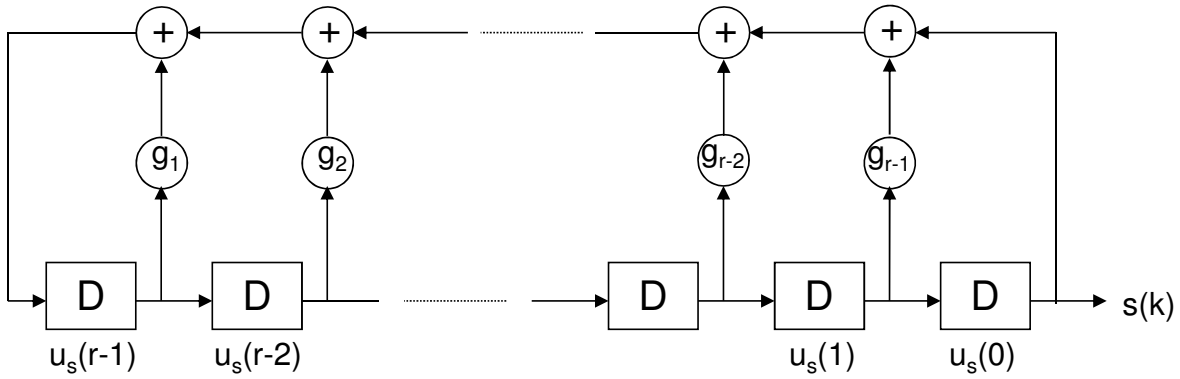

FIGURE 1 – Séquence LFSR (représentation de Fibonacci)

babiliste pour les codes LDPC [11] et propagation de croyance pour les réseaux bayesiens [14]. L'article tutoriel de Kschischang et al [15] présente un formalisme général qui s'adapte à tous les domaines utilisant des modèles sous la forme de graphes bipartites (e.g. chaînes de Markov, réseaux bayesiens). L'algorithme de propagation est utilisé dans des domaines aussi variés que l'intelligence artificielle [14], la physique statistique [16], la localisation [17], la résolution de problèmes de satisfiabilité [16] et le décodage itératif des turbo-codes [18] et des codes LDPC [12].

Le décodage des séquences LFSR a tout d'abord été étudié dans les années 60-70 avec des décodeurs à entrée dure [19][20]. Connaissant le polynôme caractéristique de la séquence $g(x) = \sum_{k=0}^{r} g_k x^k$, il s'agit de trouver l'état initial des registres. En particulier, l'algorithme de Massey [21] permet de déterminer l'état initial des registres et le polynôme caractéristique de la séquence $g(x)$, si on observe $2r$ bits consécutifs, non bruités, de la séquence, où r est le degré de $g(x)$. Ces algorithmes sont très sensibles à des erreurs sur les bits en entrée du décodeur et donc inappropriés si l'entrée est bruitée. La découverte des décodeurs itératifs à entrée souple a profondément changé la donne. Il est maintenant possible d'estimer l'état initial des registres à partir d'une observation bruitée. Les décodeurs de séquences LFSR à entrées souples ont d'abord été utilisés dans le domaine de la cryptographie pour réaliser des attaques sur les chiffreurs par flot [22][23]. Les performances de décodage dépendent très fortement du poids de Hamming des équations de parité [24]. Plus le poids est faible, meilleure est la probabilité de décoder correctement l'état initial de la séquence [24]. Malheureusement, trouver ces équations de parité n'est pas simple. Cela a logiquement donné lieu à de nombreux travaux pour améliorer la recherche des équations de parité de poids faible [25][26]. La technique de décodage des séquences LFSR est apparu plus tard dans le domaine des communications radio [27][28]. Elle a été étudiée pour décoder les séquences utilisées par des systèmes de transmission à étalement de spectre [29][30]. Ces systèmes amènent une difficulté supplémentaire : la séquence est modulée par les données émises. Il est donc nécessaire d'introduire des étapes de traitement préliminaires afin d'éliminer la modulation des données et ainsi observer la séquence à décoder. Au démarrage de cette thèse, la seule publication traitant de cet aspect du problème est celle de Kerr et Lodge [31].

Mis à part les références que nous avons citées, le décodage des séquences LFSR est un sujet

relativement peu traité dans la littérature. Les études menées dans le domaine de la cryptographie se sont focalisées principalement sur la recherche d'équations de parité de poids faible, qui est le point le plus problématique dans ce contexte. Dans le domaine des communications numériques, ces séquences ne sont pas employées pour leurs propriétés de correction d'erreurs. Par conséquent, il y a peu d'études sur les décodeurs adaptés, ainsi que l'analyse de leurs performances. Une des contributions de cette thèse est d'améliorer la compréhension des mécanismes de décodage, en se focalisant sur les performances des décodeurs itératifs par passage de messages. L'objectif est de comprendre comment le choix des équations de parité employées par le décodeur influe sur les performances (e.g. probabilité de détection correcte ou de fausse alarme). Nous allons tout d'abord établir un lien entre la théorie de la détection conventionnelle et le décodage des m-séquences. Nous allons ensuite détailler les propriétés des codes duaux des m-séquences et des séquences de Gold afin de déterminer le nombre d'équations de parité disponibles pour le décodeur. Nous nous intéressons aussi au choix de ces équations et à leur impact sur le graphe de décodage. L'identification de certaines structures topologiques (i.e. *absorbing set*) permet de comprendre l'origine des fausses alarmes et d'en tirer un algorithme de sélection des équations de parité qui minimise le taux de fausses alarmes. Nous montrons enfin par des applications concrètes l'utilité de ces techniques de décodage des séquences.

Nous allons maintenant présenter la structure du mémoire, qui reprend les sujets que nous venons d'identifier.

Organisation du mémoire et détail des contributions

Le **chapitre 1** présente l'état de l'art sur les séquences LFSR, et plus précisement les m-séquences et les séquences de Gold. Leurs principales propriétés mathématiques sont énoncées, elles seront en particulier exploitées dans les chapitres 3 et 4. Enfin, des exemples illustrant l'utilisation de ces séquences dans des systèmes opérationnels sont détaillés.

Le **chapitre 2** présente la théorie de la détection conventionnelle, puis établit un lien avec le décodage d'une séquence, si le générateur de la séquence est connu. Nous détaillons alors l'algorithme de décodage par passage de message et comment il est employé pour décoder des séquences LFSR. Cette opération, telle que présentée dans la littérature est en fait une problématique du type 'détecte et décode' [32]. Il s'agit de simultanément détecter la présence d'une séquence, et d'en décoder l'état initial. Dans le chapitre 2, nous établissons le lien entre la théorie classique de la détection et le décodage d'une séquence, si le générateur de la séquence est connu. Il s'agit d'un détecteur du type *Generalized Likelihood Ratio Test* (GLRT) dont la séquence représente le paramètre à estimer. Le décodage représente l'étape d'estimation de la séquence du détecteur. Le décodeur est implémenté avec un algorithme de décodage par passage de messages qui utilise une matrice de parité particulière. Elle concatène N_{eq} matrices de parité de référence $\mathbf{E}_a$, chacune étant générée à partir d'une équation de parité spécifique [29][30]. Cette dernière est caractérisée, en notation polynomiale, par un polynôme de référence $g_a(x)$ de degré r_a, tel que $g_a(0) = 1 : g_a(x) = 1 + \cdots + x^{r_a}$. Le poids de Hamming de l'équation de parité est noté t. C'est le nombre de coefficients non nuls du polynôme $g_a(x)$.

Le **chapitre 3** évalue le nombre et le degré minimal des équations de parité pour les m-séquences et les séquences de Gold. L'impact des équations de parité sur les performances de décodage est ensuite analysé. Un algorithme de sélection des équations de parité en est déduit.

Nous donnons des réponses aux problèmes suivants : combien de polynômes de référence $g_a(x)$ existe-t-il ? Quel est leur degré minimal ? Quels polynômes choisir pour avoir les meilleurs performances ? Comment expliquer les écarts de performance entre les configurations de polynômes choisies pour le décodage ? Pour cela, nous étudions les m-séquences et les séquences de Gold sous l'angle du codage canal. Par exemple, le nombre d'équations de parité de poids t est donné par le nombre d'éléments du code dual de la séquence ayant un poids t. Dans la suite du document, un 'élément du code dual' est un 'mot de code du code dual'. Cette formulation différente évite une redondance indigeste.

La construction d'une matrice de parité repose sur les polynômes de référence $g_a(x)$. Il est donc nécessaire de connaître le nombre de ces polynômes. On s'intéresse tout particulièrement à ceux ayant un poids de Hamming le plus faible possible. Ceci garantit de meilleures performances de décodage [24]. On souhaite donc connaître le nombre de polynômes de référence de degré r et ayant un poids t. Ceci permet d'une part de savoir s'ils existent en quantité suffisante, et, d'autre part, c'est particulièrement utile pour optimiser la procédure de recherche de ces polynômes. Les résultats de la littérature traitent complètement le cas des m-séquences, mais pas celui des séquences de Gold. Le nombre de polynômes de poids $t = 5$, lorsque le degré du polynôme r est impair, est évalué pour les séquences de Gold. Ce calcul est important, car Kasami avait déjà montré qu'il n'existe pas d'équations de parité de poids $t < 5$ lorsque r est impair [9]. La connaissance du nombre de polynômes est aussi utilisée par un modèle d'estimation du degré minimal des polynômes de référence. Cette estimation permet de fixer rapidement le nombre minimal de colonnes de la matrice de parité et par conséquent, la taille du vecteur d'observation qui doit alimenter le décodeur. Il est alors possible de déterminer si l'implémentation du décodage est faisable.

La recherche des équations de parité n'est pas abordé dans cette thèse. Ce sujet a été abondamment traité dans le domaine de la cryptographie pour implémenter des attaques sur les chiffreurs par flot [25][26]. Pour nos tests, nous avons trouvé ces équations par une recherche exhaustive.

Une fois les polynômes de référence trouvés, se pose le problème de leur sélection. Quels polynômes choisir pour avoir les meilleurs performances ? Il est vite apparu que cette sélection a un impact très fort sur le taux de fausses alarmes. Celles-ci apparaissent si le décodeur trouve une séquence alors qu'il n'y a que du bruit en entrée. Elles ont un impact décisif sur le temps d'acquisition d'un système et doivent être éliminées. Les premières simulations ont montré que le taux de fausse alarme est très sensible au choix des polynômes de référence. Il peut être très élevé (e.g. $P_{FA} = 0.2$) ou très faible (e.g. $P_{FA} = 10^{-6}$). Cette variabilité du taux de fausses alarmes est liée à la structure topologique du graphe de décodage, et en particulier à la présence, au nombre et à la taille des ensembles absorbants élémentaires (*absorbing set* en anglais)[33][34]. Nous allons donc étudier ce problème pour mettre en évidence l'effet de ces ensembles absorbants. Nous en déduisons un algorithme qui minimise les probabilités de fausse alarme et de détection erronée. Ceci améliore significativement le temps moyen d'acquisition

d'une séquence.

Le **chapitre 4** propose deux applications du décodage des séquences LFSR : la détection et le décodage du code d'embrouillage de la liaison montante des systèmes WCDMA et CDMA2000. Ceci met en évidence l'intérêt du décodage des séquences pseudo-aléatoires pour détecter la présence d'interféreurs puissants dans les réseaux mobiles de $3^{\text{ième}}$ génération. Les techniques de décodages, couplées aux propriétés mathématiques des m-séquences, sont exploitées pour mettre au point des algorithmes de détection et décodage des codes d'embrouillages des systèmes WCDMA et CDMA2000. Il est alors possible de mettre en œuvre des solutions pour limiter les effets néfastes de ces brouilleurs puissants, en particulier pour les réseaux femto cellulaires.

Les principales contributions

Dans cette thèse, nous avons contribué à améliorer les connaissances sur les sujets suivants :

- Algorithme d'estimation du code d'embrouillage du système WCDMA [P1].
- Algorithme d'estimation du code d'embrouillage du système CDMA2000 [P2].
- Identification du lien entre la théorie de la détection classique et le décodage des séquences LFSR [P3].
- Calcul du nombre d'équations de parité de poids $t = 5$ pour les séquences de Gold ayant un degré r impair [P4][P5].
- Études des ensembles absorbants dans un graphe de décodage incluant plusieurs matrices de parité de référence, et identification des cycles transverses qui détruisent les ensembles absorbants et induisent des fausses alarmes.
- Calcul du nombre de cycles de longueur 6 et 8 lorsque la matrice de parité emploie plusieurs polynômes de référence.
- Mise au point d'un algorithme de sélection des équations de parité qui minimise le nombre de cycles de longueur 6 et 8, et ainsi minimise la probabilité de fausse alarme.

Ce travail a donné lieu à 4 papiers, 5 brevets et une présentation lors des journées Codage et Cryptographie 2014.

Chapitre 1

Séquences pseudo-aléatoires

Sommaire

1.1 Définition

Une séquence binaire pseudo-aléatoire est une suite de '0' et de '1' qui semble aléatoire. Ces séquences peuvent avoir un intérêt pour des mécanismes de synchronisation, d'estimation de canal [3][4] ou bien encore pour des transmissions par étalement de spectre [1][35] ou des applications de cryptographie [2]. On recherche donc depuis longtemps des mécanismes permettant de les générer simplement. On souhaite aussi pouvoir informer un récepteur distant qu'une séquence particulière a été générée. C'est fondamental pour assurer une synchronisation entre les séquences générées par l'émetteur et le récepteur. Il faut donc pouvoir générer une longue séquence à partir d'une clé assez petite. Si le mécanisme de génération de la séquence est connue de l'émetteur et du récepteur, il suffit d'avertir le récepteur de la clé qui a été employée, et il pourra ainsi générer la séquence complète. On souhaite aussi pouvoir changer la séquence utilisée pour des raisons de sécurité.

Le terme "pseudo-aléatoire" indique que les séquences générées ont des propriétés proches de celles qui sont réellement aléatoires, mais qu'elles sont générées par des mécanismes déterministes (i.e. prévisibles). Généralement, on utilise pour cela des générateur dits LFSR (Linear Feedback Shift Register) constitués de r registres à décalages, qui sont illustrés par les figures

1.1 et 1.2. Le symbole 'D' modélise l'opérateur de délai. Ces générateurs ont l'intérêt d'être très simples à réaliser et permettent de générer des séquences ayant des propriétés particulièrement attractives. De plus, chaque séquence est spécifiée par le rebouclage des registres (coefficients g_k) et l'état initial de ces mêmes registres. Les coefficients g_k valent 1 si le bouclage est actif, 0 sinon. On a obligatoirement $g_r = g_0 = 1$. Si les autres coefficients sont connus, l'état initial des registres peut constituer la clé permettant au récepteur de générer la même séquence. Ces mécanismes sont donc très prisés des systèmes de communication et de chiffrement.

En plus de l'état initial des registres, une séquence LFSR s est spécifiée par son polynôme caractéristique $g_s(x)$ de degré r :

$$g_s(x) = \sum_{k=0}^{r} g_{s,k} x^k \tag{1.1}$$

Elle vérifie une équation de parité de la forme suivante ($n \geq 0$) :

$$\bigoplus_{k=0}^{r} g_{s,r-k} s(n+k) = 0$$

où $\bigoplus$ est la somme d'éléments binaires (modulo 2).

1.2 Représentations de Fibonacci et de Galois

Il existe deux représentations des séquences LFSR : Fibonacci (Figure 1.1) ou Galois (Figure 1.2). Elles permettent de générer au final la même séquence cyclique, mais elles peuvent être décalées. Par exemple, pour un état des registres identique au départ, la représentation de Fibonacci va générer la séquence $s(0), \ldots, s(N-1)$, et celle de Galois la séquence $s(i), \ldots, s(N-1), s(0), \ldots, s(i-1)$ où $i > 0$ est le décalage entre les 2 séquences.

On note $u_s(0), \cdots, u_s(r-1)$ l'état initial des registres de la séquence s. Soit $u_s(x) = u_s(0) + u_s(1)x + \cdots + u_s(r-1)x^{r-1}$ le polynôme construit à partir de cet état initial. Dans la représentation de Galois, le polynôme associé à la séquence s est le résultat de la division polynomiale de $u_s(x)$ par $g_s(x)$:

$$\frac{u_s(x)}{g_s(x)} = s(0) + s(1)x + s(2)x^2 + ... + s(N-1)x^{N-1}$$

Dans la représentation de Fibonacci, l'état initial des registres est donné par les r premiers éléments de la séquence s : $u_s(k) = s(k)$ pour $k = 0, \cdots r-1$. Cette propriété est utilisée pour générer la séquence s à partir du décodage de ces r premiers éléments.

On utilise l'une ou l'autre des représentations, en fonction des propriétés que l'on souhaite exploiter. Par exemple, celle de Galois permet de trouver l'état initial d'une séquence de Gold, alors que la représentation de Fibonacci est utilisée pour trouver l'état initial d'une m-séquence décimée (cf. section 4.2.2.3).

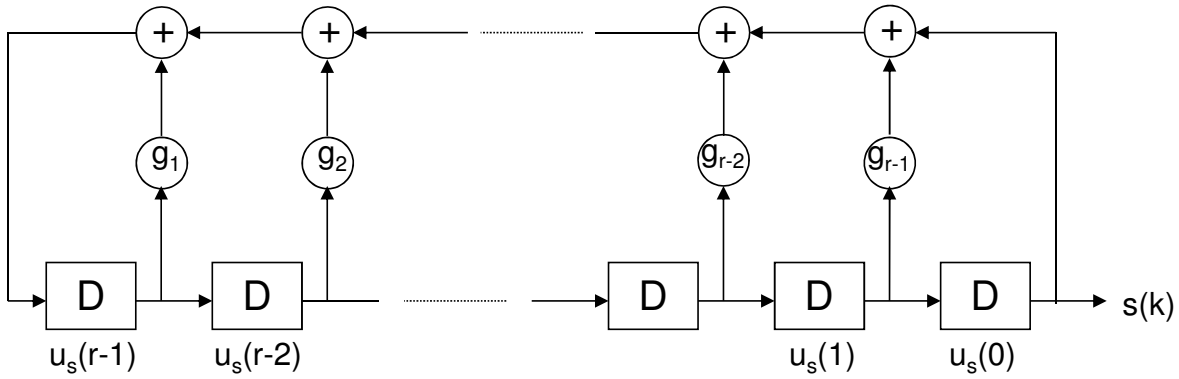

FIGURE 1.1 – Séquence LFSR (représentation de Fibonacci)

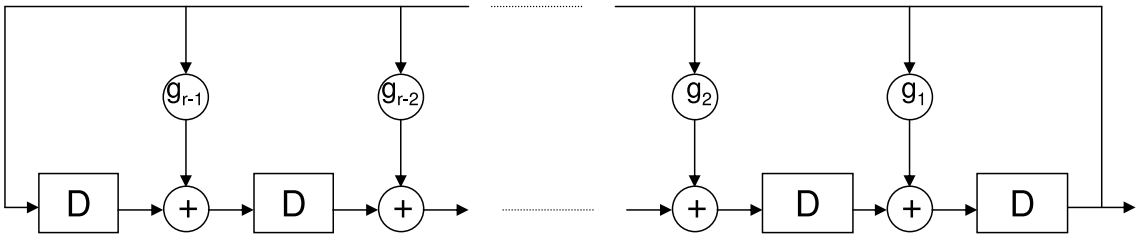

FIGURE 1.2 – Séquence LFSR (représentation de Galois)

1.3 Représentation avec la fonction Trace

Nous allons définir un certain nombre de notions qui nous serons utiles pour caractériser les m-séquences et les séquences de Gold [8] :

- On définit $K[x]$ le corps des polynômes dont les coefficients appartiennent à GF(2) (i.e. sont binaires).
- Un polynôme $g(x)$ est irréductible dans GF(2) s'il n'est pas divisible par un polynôme non constant : si $g(x)$ peut s'écrire sous la forme $g(x) = f(x)p(x)$, alors $f(x$ ou $g(x)$ est le polynôme constant.
- Soit le corps de Galois GF(2^r). Il est généré par une racine primitive $\alpha \in$ GF(2^r) tel que tout élément de $x \in$ GF(2^r) est une combinaison linéaire de $(1, \alpha, \alpha^2, \cdots, \alpha^{r-1})$: $x = \sum_{i=0}^{r-1} x_i \alpha^i$
- Le polynôme minimal de $\alpha \in$ GF(2^r) est le polynôme $p(x) \in K[x]$ de plus petit degré ayant α comme racine.
- Un polynôme $p(x)$ est primitif s'il est irréductible et le polynôme minimal d'une racine primitive α. Propriété : si α est la racine primitive de $p(x)$, tout polynôme $f(x)$ admettant α pour racine est un multiple de $p(x)$.

Soit α une racine primitive de GF(2^r). On définit la trace de GF(2^r) vers GF(2) par :

$$Tr(\alpha) = \alpha + \alpha^2 + \alpha^{2^2} + \cdots + \alpha^{2^r} \tag{1.2}$$

C'est une application linéaire qui sera utilisée pour expliquer certaines propriétés des m-séquences.

1.4 m-séquences

1.4.0.1 Définition

Une m-séquence est une séquence LFSR s dont le polynôme caractéristique $g(x) \in K[x]$ est primitif. Soit α la racine primitive de $g(x)$, la m-séquence s peut être générée de la manière suivante :

$$s(k) = Tr(\theta\alpha^k)$$

où $\theta \in \mathrm{GF}(2^r)$ détermine l'instant initial de la séquence.

1.4.0.2 Propriété

Une m-séquence est périodique de période N. C'est le plus petit entier tel que $s(k+N) = s(k)$. en utilisant la notion de trace, cette propriété est équivalente à $Tr(\theta\alpha^{k+N}) = Tr(\theta\alpha^k)$. la fonction $Tr(.)$ étant linéaire, la contrainte se traduit par : $Tr(\theta(\alpha^N - 1) = 0$. D'après [8], cette contrainte est vérifiée si et seulement si $\theta = 0$ ou $\alpha^N = 1$. Sachant que $\theta \neq 0$ car sinon la séquence $s(k)$ est nulle, N est donc le plus petite entier tel que : $\alpha^N = 1$. N est donc l'ordre de la racine primitive α : $N = ord(\alpha)$. Le polynôme primitif $g(x)$ associé à α est irréductible, donc l'ordre de α vaut $2^r - 1$ où r est le degré de $g(x)$ [8]. Par conséquent, la période de la m-séquence définie par le polynôme caractéristique $g(x)$ est $N = 2^r - 1$. C'est la longueur maximale atteignable parmi toutes les séquences LFSR de degré r [8].

Propriétés fondamentales des m-séquences [1] :

- Une m-séquence de longueur $N = 2^r - 1$, compte 2^{r-1} uns et $2^{r-1} - 1$ zéros.
- Si une fenêtre de taille r glisse le long de la séquence, chaque r-uplé apparaît une seule fois, excepté le r-uplé nul qui n'apparaît pas (sinon le reste de la séquence n'est constitué que de zéros).
- "run length" : Une m-séquence contient une seule sous-séquence de r '1' consécutifs.
- La fonction d'auto-corrélation cyclique prend 2 valeurs :

$$\theta(k) = \sum_{n=0}^{N-1} (-1)^{s(n) \oplus s(k+n)} = \begin{cases} N & k = 0 \\ -1 & k = 1, \cdots, N-1 \end{cases} \tag{1.3}$$

- Propriété d'addition : la somme de 2 versions décalées d'une même m-séquence donne une version décalée de cette même m-séquence. Soit t_1 un entier fixé, il existe un entier t_2 tel que $s(k) \oplus s(k+t_1) = s(k+t_2)$. Note : on rappelle que les indices d'une séquence sont toujours évalué modulo la longueur de la séquence.
- Propriété de décimation d'une m-séquence : la décimation d'un facteur 2 d'une m-séquence donne une version décalée de cette même m-séquence : il existe un entier t_3 tel que $s(2k) = s(k+t_3)$.
- Propriété de décimation entre les m-séquences : si s et y sont 2 m-séquences de même taille, elles sont liées par un facteur de décimation d qui est impair : $y(k) = s(dk)$.

Les deux dernières propriétés se démontrent en utilisant la fonction trace. La propriété d'addition s'écrit de la manière suivante : $s(k) \oplus s(k+t_1) = Tr(\theta(1+\alpha^{t_1})\alpha^k)$ Par conséquent, si $1+\alpha^{t_1} \neq 0$, ce qui est équivalent à $t_1 \equiv 0 \bmod N$, alors il existe t_2 tel que $1+\alpha^{t_1} = \alpha^{t_2}$. La relation de décimation se démontre en appliquant la propriété suivante : $Tr(x) = Tr(x^2)\ x \in \mathrm{GF}(2^r)$ [8].

Le polynôme $g_s(x)$ est primitif, et donc aussi minimal. Soit α la racine primitive de $g_s(x)$, tout polynôme $f(x)$ admettant α pour racine est un multiple de $g_s(x)$. Cette propriété sera employée pour définir les équations de parités satisfaites par une m-séquence.

1.5 Séquences de Gold

Une séquence de Gold est obtenue par l'addition binaire d'une "paire préférentielle" de m-séquences s et y de même taille [36] :

$$z(k) = s(k) \oplus y(k)$$

1.5.1 Définition d'une "paire préférentielle"

La notion de "paire préférentielle" est liée au facteur de décimation entre les séquences s et y : $y(k) = s(dk)$. Il vaut $d = 2^e + 1$, où l'entier e doit satisfaire la condition suivante [8] : $\mathrm{pgcd}(2^e+1, 2^r-1) = 1$. Cette condition est satisfaite si :

$$r \neq 0 \bmod(4) \Leftrightarrow r \text{ est impair ou } r = 2 \bmod(4)$$

et

$$\mathrm{pgcd(e,\ r)} = \begin{cases} 1 & \text{si } r \text{ est impair} \\ 2 & \text{si } r = 2 \bmod(4) \end{cases}$$

où $n \bmod(4)$ renvoie la valeur de n modulo 4. D'une manière analogue, le facteur de décimation q entre les séquences y et s ($s(k) = y(qk)$) est la solution de :

$$qd \equiv 1\, mod\ (2^r - 1) \tag{1.4}$$

Propriété : si r est impair, on a $d = 2^e + 1$ et $q = 2^e - 1$.

1.5.2 Intercorrélation d'une paire préférentielle

Soient s et y deux m-séquences préférentielles, leur intercorrélation est bornée par [36] :

$$|\theta(s,y)| = |\sum_{n=0}^{N-1} (-1)^{(s(n)\oplus y(k+n))}| < t$$

$$t = \begin{cases} 2^{(r+2)/2} + 1 & \text{si } r \text{ est pair} \\ 2^{(r+1)/2} + 1 & \text{si } r \text{ est impair} \end{cases}$$

Le niveau d'intercorrélation est inférieur à celui de 2 m-séquences. Ceci explique l'emploi des séquences de Gold dans un contexte multi-utilisateurs : les mécanismes de recherche de cellule des systèmes WCDMA et CDMA2000, ou bien encore les systèmes de positionnement GPS et Galileo.

1.5.3 Construction des séquences de Gold

Soit g_s et g_y les polynômes caractéristiques des deux m-séquences préférentielles s et y. En utilisant la notation de Galois de l'Eq. A.1, on peut obtenir la séquence de Gold z par :

$$z(x) = \frac{u_s(x)}{g_s(x)} + \frac{u_y(x)}{g_y(x)} = \frac{u_s(x)g_y(x) + u_y(x)g_s(x)}{g_s(x)g_y(x)} \tag{1.5}$$

On en déduit que la séquence LFSR définie par le polynôme $g_z(x) = g_s(x)g_y(x)$ va générer les séquences de Gold, chacune ayant une période $2^r - 1$. On en déduit aussi que dans la représentation de Galois, le polynôme représentant l'état initial de la séquence z est lié à ceux des séquences s et y par : $u_z(x) = u_s(x)g_y(x) + u_y(x)g_s(x)$

1.6 Exemples d'applications

1.6.1 Système WCDMA

Le système WCDMA de l'UMTS met en oeuvre une technique d'accès multiple de type CDMA [3]. Sur la liaison descendante, les canaux de données sont tout d'abord étalés par un code de Walsh-Hadamard C_{ch} qui permet de les rendre orthogonaux. Ces signaux étalés sont ensuite sommés chip à chip et le résultat est embrouillé par un code complexe Z_n. Ceci est illustré Figure 1.3. Ce dernier est construit à partir de séquences de Gold c_n :

$$Z_n(i) = C_n(i) + jC_n(i + 131072) \quad i = 0, 1, \ldots, 38399$$

où $C_n(i) = (-1)^{c_n}$ est la modulation BPSK de la séquence binaire c_n.

Les séquences de Gold c_n sont construites à partir de la paire de m-séquences préférentielle s et y :

$$c_n(k) = s(k + n) \oplus y(k)$$

L'indice de décalage n permet de générer une séquence de Gold qui est dédiée à une station de base ($n = 0, 1, 2, \ldots, 2^{18} - 2$).

Les polynômes caractéristiques des séquences s et y sont :

$$g_s(x) = 1 + x^7 + x^{18}$$
$$g_y(x) = 1 + x^5 + x^7 + x^{10} + x^{18}$$

A chaque début de trame (tous les 38400 chips), les registres des 2 m-séquences sont réinitialisés avec les valeurs suivantes :

$$s(0) = 1,\ \ s(1) = s(2) = \cdots = s(17) = 0$$
$$y(0) = y(1) = y(2) = \cdots = y(17) = 1$$

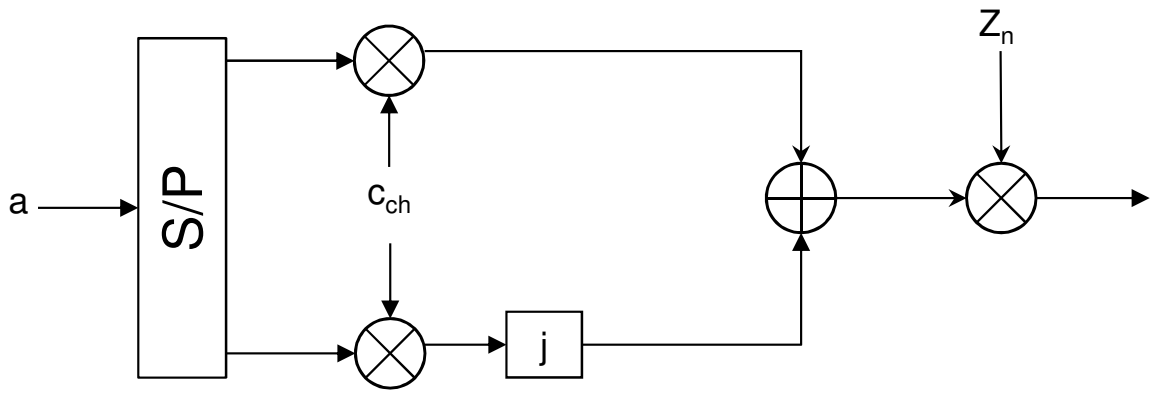

FIGURE 1.3 – Mécanisme d'étalement de la liaison descendante du système WCDMA

1.6.2 Global Positioning System (GPS)

Le système GPS utilise des séquences de Gold pour l'acquisition rapide du canal C/A (Coarse acquisition code). Le canal C/A émis par le $i^{\text{ième}}$ satellite est de la forme :

$$R_i(k) = d_i\left(\left\lfloor \frac{k}{L_s N} \right\rfloor\right)(-1)^{z_i(k)} \tag{1.6}$$

$z_i(k)$ est le code d'étalement alloué au satellite. Il est choisi parmi un ensemble de 32 séquences de Gold, ayant toutes une période $N = 1023$ chips, soit 1 ms. $d_i(l)$ est le $l^{\text{ième}}$ symbole BPSK du message de navigation. Chaque symbole dure 20 ms, soit $L_s = 20$ périodes de répétition du code d'étalement $z_i(k)$. Chaque satellite utilise une séquence spécifique ($i = 1, \ldots, 32$) :

$$z_i(k) = s(k) \oplus y(k + \tau_i)$$

Les deux m-séquences qui définissent les séquences de Gold du système ont les polynômes caractéristiques suivants [35] :

$$g_s(x) = x^{10} + x^3 + 1$$
$$g_y(x) = x^{10} + x^9 + x^8 + x^6 + x^3 + x^2 + 1$$

Les séquences $y(k+\tau_i)$ sont générées en utilisant la propriété d'addition des m-séquences. Les délais τ_i sont des entiers positifs choisis pour que les séquences puissent être générées avec une combinaison linéaire de 2 registres :

$$y(k + \tau_i) = u_{\beta_1}(k) \oplus u_{\beta_2}(k)$$

où $u_j(k)$ est l'état du $j^{\text{ième}}$ registre de la séquence y à l'instant k, dans la représentation de Fibonacci. L'origine de cette méthode de génération d'une séquence est expliquée dans la section A.3 de l'annexe. Les paramètres β_1 et β_2 indiquent les numéros des registres qui sont additionnés. Les valeurs de ces 2 paramètres sont donnés dans [35]. Les registres des 2 séquences sont initialisés à '1' au démarrage.

1.6.3 Cryptographie

Les mécanismes de chiffrement par flot ("stream cipher" en anglais) sont utilisés, par exemple, par Internet (RC4 pour les transactions sécurisés), et les systèmes GSM (A5/1, utilisé pour sécuriser la communication montante entre un terminal et une station de base) et Bluetooth (E0) [2]. Un signal de chiffrement est généré en continu puis additionné avec le message à transmettre. Ceci est illustré Figure 1.4.

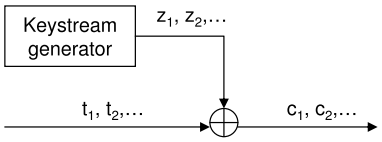

FIGURE 1.4 – Principe du chiffreur en ligne

La construction du signal de chiffrement z_1, z_2,.. met couramment en œuvre des séquences LFSR, qui sont combinées par une fonction booléenne non-linéaire f (Figure 1.5). Pour obtenir la séquence z_1, z_2,...ayant la plus longue période possible, on emploie couramment des m-séquences. Les polynômes des séquences sont connus et leur état initial définit la clé de chiffrement.

Il existe une littérature abondante sur une technique d'attaque de ce type de chiffreur : "Fast correlation attack" [22] [37] [38] [39]. Elle a été employée avec succès pour attaquer le mécanisme de chiffrement E0 du système Bluetooth [7]. Le principe est de considérer la séquence z comme une observation d'une des m-séquences au travers d'une canal binaire symétrique. Il est alors envisageable de retrouver l'état initial de la m-séquence en mettant en œuvre un décodeur approprié.

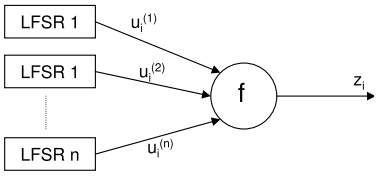

FIGURE 1.5 – Construction du chiffreur avec des séquences LFSR

Chapitre 2

Détection et décodage

Sommaire

2.1 Introduction

Cette thèse aborde les aspects pratiques de la détection et du décodage conjoints [32]. En effet, on cherche simultanément à détecter la présence d'une séquence LFSR (e.g. Gold) et aussi à en déterminer l'état initial. Le polynôme caractéristique est supposé connu, ce qui permet, une fois la séquence détectée et son état initial décodé, de générer cette séquence. Ceci permet ensuite de mettre en œuvre des traitements sur le signal reçu. On peut citer par exemple la détection multi-utilisateurs dans les systèmes CDMA, pour éliminer un interféreur puissant. La littérature classique sur le décodage des séquences pseudo-aléatoires [29][30][28] a laissé de côté la justification de la méthode de décodage employée. Nous allons combler ce manque en établissant le lien avec la théorie de la détection classique. Cette dernière sera tout d'abord expliquée, puis nous utiliserons le concept du Generalized Likelihood Ratio Test (GLRT) pour justifier la méthode de décodage des séquences. Enfin, nous allons détailler l'algorithme de décodage par passage de messages qui peut être implémenté en pratique.

2.2 Théorie de la détection classique

Dans cette section, les principes de la théorie de la détection sont synthétisés. Il s'agit du test du rapport de vraisemblance (Likelihood Ratio Test en anglais, LRT) et de sa version généralisée (GLRT) lorsque certains paramètres intervenant dans la détection sont inconnus.

2.2.1 Test du rapport de vraisemblance

Le test du rapport de vraisemblance est un test conventionnel pour prendre une décision parmi plusieurs hypothèses possible. Pour un test de détection de la présence ou de l'absence d'un signal $S(k) = (-1)^{s(k)}$, on définit les 2 hypothèses :

- Le signal $S(k)$ est présent, mais perturbé par un bruit additif $w(k)$:

$$H_1 : R(k) = S(k) + w(k)$$

- Le signal $S(k)$ est absent :

$$H_0 : R(k) = n(k)$$

La probabilité de fausse alarme est la probabilité de décider en faveur de l'hypothèse H_1 alors que le signal est absent. Elle est notée :

$$P_{FA} = P(H_1|H_0)$$

La probabilité de non-détection est la probabilité de décider en faveur de l'hypothèse H_0 alors que le signal est présent. Elle est notée :

$$P_{ND} = P(H_0|H_1)$$

La probabilité de détection correcte vaut : $P_{CD} = P(H_1|H_1) = 1 - P_{ND}$

L'idéal serait de pouvoir maximiser la probabilité de détection, tout en minimisant la probabilité de fausse alarme. Malheureusement, cela n'est pas possible, la probabilité de fausse alarme augmente si on accroît la probabilité de détection. Il faut donc trouver un compromis entre ces 2 critères de performance. Ceci est réalisé par les tests de Neyman-Pearson et bayesien.

2.2.1.1 Test de Neyman-Pearson

Le test de Neyman-Pearson (NP) permet de maximiser la probabilité de détection pour un niveau de fausse alarme prédéterminé $P_{FA} = \alpha$. On définit le vecteur d'observations $\mathbf{R} = (R(0), R(1), \cdots, R(N-1))^T$ et le ratio des densités de probabilité du signal reçu sous les hypothèse H_0 et H_1 : $L(\mathbf{R}) = \frac{p(\mathbf{R}|H_1)}{p(\mathbf{R}|H_0)}$

Le test de Neayman-Person est le suivant [40] :

- Si $L(\mathbf{R}) > \gamma \implies H_1$ est choisie.

– Si $L(\mathbf{R}) < \gamma \implies H_0$ est choisie.

Le seuil γ est choisi de tel façon que :

$$P_{FA} = \int_{\{R/L(\mathbf{R})>\gamma\}} p(R|H_0)dR = \alpha$$

2.2.1.2 Test bayesien

Dans certaines situations, il est possible de définir à l'avance la probabilité d'occurrence des hypothèses H_0 et H_1 (e.g. acquisition de signaux GPS, décodage de bits transmis par une source aléatoire...etc.). On peut alors définir un risque associé à une erreur lors de la prise de décision et chercher à le minimiser. C'est l'approche bayesienne de la théorie de la détection. Soit C la fonction de risque pour un test à 2 hypothèses :

$$C = C_{01}P(H_0|H_1)P(H_1) + C_{10}P(H_1|H_0)P(H_0)$$

où $P(H_0)$ et $P(H_1)$ sont les probabilité d'occurrence des hypothèses H_0 et H_1, et C_{01} et C_{10} les coûts associés à chaque erreur (probabilité de non détection et de fausse alarme). Ces derniers modélisent la pénalité induite par une erreur.

Le détecteur qui minimise le risque Bayesien décide de la manière suivante [40] :

– Si $L(\mathbf{R}) > \gamma \implies H_1$ est choisie.

– Si $L(\mathbf{R}) < \gamma \implies H_0$ est choisie.

où $\gamma = \frac{C_{10}p(H_0)}{C_{01}p(H_1)}$

On remarque que les tests NP et bayesien sont équivalents, seul le seuil de détection change. On appliquera donc un test de type LRT et on cherchera le seuil de détection γ qui garantit un niveau de fausse alarme $P_{FA} = \alpha$.

2.2.1.3 Application à la détection d'un signal

Nous allons illustrer le test bayesien en commençant par un cas simple : la détection cohérente. Elle ne correspond pas à un cas pratique car elle ne tient pas compte des rotations de phase introduites par le canal de propagation et de l'écart des fréquences porteuses de l'émetteur et du récepteur. Cela constitue néanmoins un bon exemple didactique pour comprendre la suite.

Exemple 1 : Détection cohérente

On cherche à détecter la séquence connue $\mathbf{S} = (S(0), S(1), \cdots, S(N-1))^T$ à partir du vecteur d'observation $\mathbf{R}$. Les hypothèses H_1 et H_0 s'écrivent de la manière suivante :

$$H_1 : \mathbf{R} = \mathbf{S} + \mathbf{n}$$
$$H_0 : \mathbf{R} = \mathbf{n}$$

$\mathbf{n}$ est un vecteur de bruit gaussien complexe i.i.d., de matrice de covariance $\sigma^2\mathbf{I}_N$. Les densités de probabilité du vecteur $\mathbf{R}$ en fonction des 2 hypothèses sont donc :

$$p(\mathbf{R}|H_1) = \frac{1}{\pi^N \sigma^{2N}} e^{-\frac{1}{\sigma^2}\|\mathbf{R}-\mathbf{S}\|^2)}$$
$$p(\mathbf{R}|H_0) = \frac{1}{\pi^N \sigma^{2N}} e^{-\frac{1}{\sigma^2}\mathbf{R}^H\mathbf{R}}$$

où $\| \mathbf{X} \|^2 = \sum_{j=0}^{N-1} |X_j|^2$ est la norme L_2 du vecteur $\mathbf{X}$.

En pratique, on calcule le rapport de vraisemblance logarithmique (log LRT) ($\Lambda(\mathbf{R}) = \ln(L(\mathbf{R}))$), ce qui simplifie les calculs. D'après [41], on trouve :

$$\Lambda(\mathbf{R}) = \frac{1}{\sigma^2}\left(2\Re(\mathbf{R}^H.\mathbf{S}) - \| \mathbf{S} \|^2\right) \tag{2.1}$$

$\Re(z)$ est la partie réelle du complexe z. Si chaque chip de la séquence S vaut ± 1, alors $\| \mathbf{S} \|^2 = N$. Le terme $\| \mathbf{S} \|^2$ étant constant, il peut être intégré dans le seuil γ. Il en est de même avec la variance du bruit qui est une constante de normalisation. Au final, le Log LRT devient :

$$T(\mathbf{R}) = \Re(\mathbf{R}^H.\mathbf{S}) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \tag{2.2}$$

On retrouve donc un résultat classique dans la littérature sur la détection des m-séquences : la détection par corrélation. Si le niveau de fausse alarme est fixé, il est possible d'en déduire le seuil de détection γ, ainsi que la probabilité de détection P_{CD} [41] :

$$\gamma = \sqrt{N\sigma^2}\mathrm{erf}^{-1}(1 - 2P_{FA})$$
$$P_{CD} = \frac{1}{2}\left[1 - erf\left(\frac{\gamma - N}{\sqrt{N\sigma^2}}\right)\right]$$

où $erf(x)$ est la fonction d'erreur :

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Exemple 2 : Détection non-cohérente

Dans un cas plus réaliste, le signal reçu est affecté d'un offset de phase θ qui est constant et inconnu du récepteur. Quel est le détecteur qui convient pour cette situation ? Il existe 2 approches :

- Bayesienne : le principe est de réaliser un LRT avec les densité de probabilité $p(\mathbf{R}|H_1, \theta)$ et $p(\mathbf{R}|H_0, \theta)$ moyennée par la densité de probabilité $p(\theta)$, si cette dernière est connue. On obtient alors :
$$L(\mathbf{R}) = \frac{p(\mathbf{R}|H_1)}{p(\mathbf{R}|H_0)} = \frac{E_\theta[p(\mathbf{R}|H_1, \theta)]}{E_\theta[p(\mathbf{R}|H_0, \theta)]}$$
Cette approche va être appliquée pour calculer le détecteur non-cohérent.
- Generalized Likelihood Ratio Test (GLRT) : il s'agit d'estimer le paramètre inconnu et d'appliquer le LRT avec cette estimation. Cette technique sera détaillée dans la section suivante.

On cherche à détecter la séquence connue $\mathbf{S} = (S(0), S(1), \cdots, S(N-1))^T$ à partir du vecteur d'observation $\mathbf{R} = (R(0), R(1), \cdots, R(N-1))^T$, sachant que le canal introduit une rotation de phase θ. Les hypothèses H_1 et H_0 s'écrivent de la manière suivante :

$$\begin{aligned} H_1 &: \mathbf{R} = \mathbf{S}e^{j\theta} + \mathbf{n} \\ H_0 &: \mathbf{R} = \mathbf{n} \end{aligned}$$

Les densités de probabilité correspondant aux 2 hypothèses sont donc :

$$\begin{aligned} p(\mathbf{R}|H_0) &= \tfrac{1}{\pi^N \sigma^{2N}} e^{-\frac{1}{\sigma^2}\mathbf{R}^H\mathbf{R}} \\ p(\mathbf{R}|H_1, \theta) &= \tfrac{1}{\pi^N \sigma^{2N}} e^{-\frac{1}{\sigma^2}(\mathbf{R}-\mathbf{S}e^{j\theta})^H(\mathbf{R}-\mathbf{S}e^{j\theta})} \end{aligned}$$

La moyenne de $p(r|H_1, \theta)$ par rapport à la la phase θ, de loi uniforme sur l'intervalle $[-\pi, +\pi]$, vaut :

$$E_\theta[p(\mathbf{R}|H_1, \theta)] = \frac{1}{\pi^N \sigma^{2N}} e^{-\frac{1}{\sigma^2}(\mathbf{R}^H\mathbf{R}+N)} \frac{1}{2\pi} \int_0^{2\pi} e^{\frac{2}{\sigma^2}|\mathbf{R}^H\mathbf{S}|\cos(\theta)} d\theta$$

Soit $I_0(z)$ la fonction de Bessel modifiée du premier ordre :

$$I_0(z) = \frac{1}{2\pi} \int_0^{2\pi} e^{z\cos(\theta)} d\theta$$

Le Log-LRT s'écrit finalement :

$$\ln\left(I_0\left(\frac{2}{\sigma^2}|\mathbf{R}^H\mathbf{S}|\right)\right) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \tag{2.3}$$

Il est intéressant de noter qu'il est possible de faire les approximations suivantes [41] :

$$\begin{aligned} &\ln(I_0(x)) \approx x \text{ pour } 10\log(x) > 10 \text{ dB} \\ &\ln(I_0(x)) \approx x^2/4 \text{ pour } 10\log(x) < 5 \text{ dB} \end{aligned}$$

Pour un SNR élevé, on retrouve le même détecteur que pour le cas cohérent. Pour un SNR faible, le détecteur optimal est bien approché par le détecteur quadratique, qui est très utilisé en pratique. Le test de log-LRT est alors :

$$T(\mathbf{R}) = |\mathbf{R}^H\mathbf{s}|^2 \underset{H_0}{\overset{H_1}{\gtrless}} \gamma$$

Le détecteur exploite l'énergie du signal sur la partie réelle et imaginaire, afin de compenser la rotation de phase θ.

2.2.2 Test du rapport de vraisemblance généralisé

Dans de nombreux cas, l'approche bayesienne est impossible à mettre en œuvre car les calculs de $E_\theta[p(R|H_1,\theta)]$ et $E_\theta[p(R|H_0,\theta)]$ sont impossibles à entreprendre analytiquement. On applique alors le test du rapport de vraisemblance généralisé (*Generalized Likelihood Ratio Test* en anglais, GLRT). Cette approche empirique consiste à estimer les paramètres inconnus ξ_1 et ξ_0 selon le critère du maximum de vraisemblance (MV) et à appliquer le LRT avec cette estimation [40] :

$$\begin{array}{l} L(\mathbf{R}) = \frac{p(\mathbf{R}|\hat{\xi}_1,H_1)}{p(\mathbf{R}|\hat{\xi}_0,H_0)} \\ \text{où } \hat{\xi}_i = \text{argmax}_\xi \; p(\mathbf{R}|\xi,H_i) \;\; i=0,1 \end{array} \tag{2.4}$$

Si le paramètre θ n'apparaît pas dans l'hypothèse H_0, alors le GLRT se simplifie :

$$\begin{array}{l} L(\mathbf{R}) = \max\limits_{\xi} L(\mathbf{R},\xi) \\ L(\mathbf{R},\xi) = \frac{p(\mathbf{R}|\xi,H_1)}{p(\mathbf{R}|H_0)} \end{array} \tag{2.5}$$

Nous allons maintenant montrer comment ce test du GLRT peut être appliqué pour détecter des séquences pseudo-aléatoires.

2.3 Détection du signal assistée par le codage canal

Les exemples présentés dans la section 2.2.1.3 ne correspondent pas exactement à la réalité d'un système de transmission. En pratique, un récepteur sait que la séquence reçue appartient à un ensemble de séquences possibles, mais il ne connaît pas celle qui est émise. Si la séquence est générée par un mécanisme de codage canal (e.g. BCH, code simplex ou code de Hamming), cela signifie que le récepteur connaît le mécanisme de codage mais pas le mot de code émis. Le récepteur doit donc détecter la présence ou l'absence d'un mot de code et le cas échéant trouver le mot de code. Dans le cas des séquences LFSR, le récepteur connaît donc le polynôme caractéristique de la séquence, mais pas son état initial. Finalement, le test d'hypothèses doit donc inclure la séquence à détecter parmi les paramètres inconnus.

L'estimation de la séquence revient à effectuer un décodage, qui peut s'implémenter suivant 2 critères :

- Maximum de Vraisemblance.
- Maximum a Posteriori (MAP) au niveau symbole.

2.3.1 Estimation suivant le critère du maximum de vraisemblance

Pour expliquer le mécanisme d'estimation de la séquence selon le critère MV, nous allons nous focaliser sur le cas d'une détection cohérente.

Les 2 hypothèses de détection sont :

$$H_1 : \mathbf{R} = \mathbf{S} + \mathbf{n}$$
$$H_0 : \mathbf{R} = \mathbf{n}$$

Le GLRT est défini par la relation suivante :

$$\begin{aligned} L(\mathbf{R}) &= \frac{p(\mathbf{R}|\hat{\mathbf{S}},H_1)}{p(\mathbf{R}|H_0)} \\ \hat{\mathbf{S}} &= \underset{\mathbf{S}}{\operatorname{argmax}}\, p(\mathbf{R}|\mathbf{S},H_1) \end{aligned} \tag{2.6}$$

où $\hat{\mathbf{S}}$ est la séquence estimée.

Si on estime la séquence la plus probable, cela revient à un décodage de type *Maximum Likelihood Sequence Estimation* (MLSE)[42]. On trouve alors :

$$\hat{\mathbf{S}} = \underset{\mathbf{S}}{\operatorname{argmax}}\, \Re(\mathbf{R}^H.\mathbf{S}) \tag{2.7}$$

Si le nombre de séquences possibles est limité (e.g 32 pour le GPS), il est possible d'implémenter le décodeur MLSE en testant toutes les séquences possibles et retenir celle qui donne la corrélation maximale. C'est par exemple, l'approche retenue par les récepteur GPS classiques [6]. Si jamais, le nombre de séquences est beaucoup plus élevé, il faudrait implémenter un décodeur de type Viterbi. La structure d'un codeur LFSR est très proche de celle d'un code convolutif. L'état des registres permet de générer le treillis qui est ensuite exploité par l'algorithme de Viterbi pour trouver la séquence la plus probable. Il est à noter que la séquence trouvée peut ne pas correspondre à l'une des séquences émises.

Pour une m-séquence de degré r, le nombre d'état du décodeur vaut $2^r - 1$, ce qui rend la complexité très rapidement prohibitive. On doit alors mettre en œuvre une version simplifiée de ce décodeur. Ceci est réalisable avec l'algorithme de décodage par passage de message de type Min-Sum (cf. section 2.4.3). s

2.3.2 Estimation suivant le critère MAP au niveau symbole

L'estimation des paramètres inconnus (cf. Eq. 2.7) peut aussi être envisagée avec un critère de type *Maximum A Posteriori* (MAP) au niveau symbole. On ne cherche plus à détecter la séquence émise, mais à décoder selon un critère MAP les chips constituants la séquence :

$$\begin{aligned} \hat{s}(i) &= \underset{s(i)}{\operatorname{argmax}}\, p(s(i)|\mathbf{R},H_1)\ i = 0,\cdots,N-1 \\ \hat{\theta} &= \underset{\theta}{\operatorname{argmax}}\, p(\theta|\mathbf{R},H_1) \end{aligned} \tag{2.8}$$

Le décodage MAP est moins complexe à mettre en œuvre pour le décodage des séquences LFSR (cf. section 2.4.3. C'est son principal atout pour les applications qui nous intéresse. Il faut noter que, comme pour le critère MV, la séquence trouvée peut ne pas correspondre à l'une des séquences émises.

2.3.2.1 Détection cohérente

Le GLRT consiste à estimer la séquence qui maximise le critère MAP au niveau symbole, puis à appliquer le test bayesien. Ceci revient à décoder les séquences LFSR suivant un critère MAP, ce qui est réalisable car elles peuvent être définies comme des mécanismes de codage canal. Le mot d'information utile est chargée initialement dans les registres, et le mot de code est la séquence générée à partir de cet état initial des registres. Il est alors possible d'estimer la séquence avec un algorithme de décodage par propagation de croyance. Cela nécessite de connaître les paramètres d'encodage. Par exemple, pour les séquences LFSR cela revient à connaître le polynôme caractéristique de la séquence. Ce mécanisme est détaillé dans la section 2.4.3 pour une m-séquence. Il repose sur la construction d'une matrice de parité, générée avec des équations de parité satisfaites par les séquences. L'étude des propriétés de ces équations de parité et leur sélection fera l'objet d'une étude spécifique dans le chapitre 3.

Si le décodeur trouve une séquence valide (i.e. toutes les équations de parité sont satisfaites), il peut s'agir de la séquence émise (détection correcte), ou bien d'une version décalée de cette séquence (détection erronée). Si le signal en entrée du décodeur n'est constitué que de bruit et qu'il détecte une séquence valide, il s'agit d'une fausse alarme.

Si les probabilités de détection erronée et de fausse alarme sont négligeables, l'étape de décodage fournit l'état initial des registres de la séquence émise, mais aussi l'information que le récepteur est synchronisé avec cette séquence. Dans ce cas, l'étape de vérification avec le LRT (Eq. 2.2) n'est pas nécessaire. Dans le cas contraire, cette étape devra être implémentée. Par conséquent, la complexité globale du récepteur dépendra des performances de l'algorithme de décodage itératif qui sera sélectionné pour implémenter le décodage MAP. Ceci nous a amené à réaliser une étude sur l'impact des équations de parité sur le décodeur MAP. Nous en avons déduit un algorithme qui minimise les fausses alarmes. Ceci est décrit dans la section 3.4.

2.3.2.2 Estimation conjointe de la séquence et des paramètres de synchronisation

Dans ce cas, le GLRT consiste à estimer conjointement la séquence qui maximise le critère MAP au niveau symbole ainsi que les autres paramètres de synchronisation inconnus. Il s'agit par exemple d'un offset de phase et de fréquence.

L'estimation peut être implémentée selon deux approches :

1. Processus itératif entre un bloc de synchronisation et un décodeur à sortie souple [43][44][45][46]. Les données souples en sortie de chaque itération du décodeur sont utilisées pour améliorer l'estimation des paramètres de synchronisation.
2. Estimation des paramètres inconnus "à l'intérieur" du décodeur. Il s'agit d'une extension aux décodeurs itératifs (turbo, BP) du 'Per-Survivor Processing' (PSP) qui a été originellement proposée pour un décodeur de type Viterbi [47]. Pour un décodeur par propagation de croyance, il s'agit d'inclure l'estimation des paramètres inconnus dans le graphe de décodage [48][49]. Pour un décodeur de Viterbi ou un turbo-décodeur, chaque état contient une estimation du paramètre qui dépend du chemin qui passe par cet état.

Au cours de cette thèse, nous avons mis en œuvre la première approche et validé son efficacité, mais nous n'avons pas approfondi cette voie. La deuxième approche n'a pas été évaluée.

2.4 Décodage d'une m-séquence par propagation de croyance

Dans cette section, nous allons illustrer le décodage des séquence LFSR avec le cas des m-séquences. Ceci simplifiera la description. Les principes utilisés pour construire le décodeur seront réutilisés dans le chapitre 3 pour les séquences de Gold.

Une m-séquence est connue dans le domaine du codage correcteur d'erreurs comme étant un mot d'un code "simplex". Ce dernier est un code linéaire cyclique $[N = 2^r - 1, k = r]$. Il est défini par le polynôme caractéristique $g(x)$ de la m-séquence de degré r [13] qui spécifie le polynôme de contrôle du code cyclique. Nous allons détailler cela dans la section 2.4.1. Le message à coder est constitué par l'état initial des registres de la séquence (r bits), et le mot de code est la séquence générée à partir de cet état initial ($2^r - 1$ bits). Le décodage d'une m-séquence repose donc sur les propriétés du code simplex. Nous allons tout d'abord rappeler les définitions d'un code cyclique, de son dual et d'un code simplex. Ceci sera ensuite exploité pour définir la matrice de parité qui est employée par le décodeur itératif.

2.4.1 Propriétés des m-séquences vues comme des codes correcteurs d'erreurs

2.4.1.1 Code cyclique

Un code linéaire C, de paramètres $[N, k]$ $(N > k)$, transforme un vecteur d'information $\mathbf{u} = (u_0, \cdots, u_{k-1})$ en un mot de code $\mathbf{c} = (c_0, \cdots, c_{N-1})$. Il est spécifié soit par sa matrice génératrice G ou sa matrice de parité E [13] :

$$\begin{aligned} \mathbf{c} &= \mathbf{u}\ \mathbf{G} \\ \mathbf{E}\mathbf{c}^T &= \mathbf{0} \end{aligned} \tag{2.9}$$

Les deux matrices dépendent du polynôme générateur du code. On le note $f(x)$, il est de degré $N - k$.

Rappel de notation : à chaque mot de code $c = (c_0, \cdots, c_{N-1})$ est associé le polynôme $c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{N-1}$.

Propriété : soit $f(x)$ le polynôme générateur du code C. Alors, chaque mot de code $c(x)$ est un multiple de $f(x)$, modulo $x^N - 1$.

Définition : Un code cyclique C satisfait la propriété suivante : si $(c_0, \cdots, c_{N-1}) \in C$ alors $(c_{N-1}, c_0, \cdots, c_{N-2}) \in C$.

Par conséquent, un décalage circulaire d'un bit du mot de code $c(x)$ est modélisé par le

polynôme $xc(x)$. Les codes de Hamming, BCH et simplex appartiennent à cette famille des codes cycliques [13].

2.4.1.2 Code dual

Définition : Soit C un code linéaire $[N, k]$, son dual $C^{\perp}$ est constitué par l'ensemble des vecteurs orthogonaux à tous les mots de code de C :

$$C^{\perp} = \{\mathbf{a} \,|\, \mathbf{a}\mathbf{c}^T = 0 \; \forall \; \mathbf{c} \in C\} \tag{2.10}$$

Lorsque $c \in C^{\perp}$, on dit que $c(x)$ est un polynôme de parité de C. Son poids est défini par le nombre d'éléments non nuls. C'est le poids dit de Hamming.

Propriétés du code dual :
- Le dual d'un code cyclique $[N, k]$ est un code cyclique $[N, N-k]$.
- Soit $f(x)$ le polynôme générateur du code C (de degré $N-k$). Le polynôme de contrôle de C est défini par : $h(x) = (x^N - 1)/f(x)$. Il est de degré k. Le polynôme générateur de $C^{\perp}$ est le réciproque du polynôme de contrôle : $f^{\perp}(x) = x^k h(x^{-1})$.

Nous allons maintenant mettre à profit ces définitions pour caractériser les propriétés du code simplex.

2.4.1.3 Code simplex

Définition : Un code simplex S, ayant un polynôme caractéristique $g(x)$ de degré r, est un code cyclique $[N = 2^r - 1, k = r]$ dont le polynôme de contrôle est $h(x) = g_{recip}(x) = x^r g(x^{-1})$, conformément aux notations de la figure 1.1.

Définition : Une m-séquence spécifiée par le polynôme caractéristique $g(x)$ est un mot du code simplex défini par son polynôme de contrôle $g_{recip}(x)$.

Propriété : la distance minimale d'un code simplex est $d_{min} = 2^{r-1}$.

En effet, chaque séquence contient 2^{r-1} uns (cf. sec. 1.4) et donc la distance par rapport au mot de code nul vaut 2^{r-1}, ce qui donne la distance minimale du code.

Définissons la matrice de parité $\mathbf{E}$, dont les lignes sont constituées par des mots de code de $S^{\perp}$. Chaque m-sequence $\mathbf{y} \in S$ doit vérifier $\mathbf{E}\mathbf{y}^T = 0$. Inversement, si E est de rang $N - r$, alors toute séquence $\mathbf{y}$ vérifiant $\mathbf{E}\mathbf{y}^T = 0$ est un mot du code simplex S.

Le code dual d'un code simplex, dont le polynôme de contrôle est $g_{recip}(x)$, est généré par $g(x)$. Par conséquent, $g(x)$ peut être employé pour construire facilement une matrice de parité. La

$i^{\text{ième}}$ ligne est constituée du mot de code $x^i g(x)$ $(i = 0, \cdots, N-r-1)$:

$$\mathbf{E} = \begin{bmatrix} g_r & \cdots & g_0 & 0 & \cdots & \cdots & 0 \\ 0 & g_r & \cdots & g_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & g_r & \cdots & g_0 & 0 \\ 0 & \cdots & \cdots & 0 & g_r & \cdots & g_0 \end{bmatrix} \tag{2.11}$$

Note : dans la littérature classique, les matrices de parité sont notées H. Étant donné le risque de confusion avec les hypothèses H_0 et H_1 définies à la section 2.2.1, nous avons préféré les noter E.

La matrice $\mathbf{E}$ définie par (2.11) est employée conventionnellement lors du décodage des m-sequences [29][28]. Elle ne contient cependant qu'un sous-ensemble des équations de parité qui peuvent être appliquées à une m-séquence. Il est connu dans la littérature que le dual d'un code simplex est le code de Hamming $H_r = [2^r - 1, 2^r - r - 1]$ généré par $g(x)$ [13]. Par conséquent, toutes les équations de parité d'une m-séquence sont déterminées par les éléments du code de Hamming H_r. Cette propriété est utile pour le décodage car on cherche des équations de parité de poids faible [24]. Si le polynôme $g(x)$ a un poids trop élevé, il faut chercher dans H_r les mots de poids faible afin d'améliorer les performances de décodage. Ceci constitue par exemple l'étape la plus importante et la plus complexe lors de la mise en place d'une attaque sur un chiffreur par flot [26][23]. En effet, trouver des mots de code de poids faible dans un code linéaire est un problème NP-complet [50].

Une fois la matrice de parité $\mathbf{E}$ construite, le décodeur applique un algorithme de décodage par passage de message sur le graphe de Tanner induit par $\mathbf{E}$. Le principe de décodage des m-séquences est donc de créer une matrice de parité creuse E et d'appliquer un algorithme de décodage par passage de messages sur le graphe de Tanner [28][15].

L'utilisation des techniques de décodage a déjà été mise en œuvre pour la synchronisation aveugle sur les trames reçues (*frame synchronizer*) [51][52]. Il existe cependant une singularité du décodage des séquences LFSR, les messages émis par les nœuds de contrôles sont fortement corrélés. Par conséquent, l'analyse théorique proposée dans [53], pour une détection basée sur les valeurs du syndrome, ne s'applique pas pour les séquences LFSR.

2.4.2 Décodage par propagation de croyance

On définit tout d'abord le graphe de Tanner correspondant au code. Il est constitué de deux types de variables : les nœuds de contrôle et les variables à proprement parler. On note V l'ensemble des variables à décoder et F l'ensemble des nœuds de contrôle. Un nœud de contrôle correspond à une équation de parité de la matrice E, c'est une ligne de E. Il est connecté aux variables qui interviennent dans l'équation de parité. Cela correspond aux '1' dans la ligne de la matrice E.

La Figure 2.1 présente une illustration d'un graphe de Tanner utilisé par l'algorithme de décodage par passage de messages. La séquence émise est notée $\mathbf{z}$ et $R(i) = (-1)^{z(i)} + w(i)$ est

l'observation de cette variable à l'entrée du décodeur ($i = 0, \cdots, N-1$). Ce modèle suppose que le canal de propagation n'a pas de mémoire. $w(i)$ est un bruit blanc additif gaussien de variance σ_0^2.

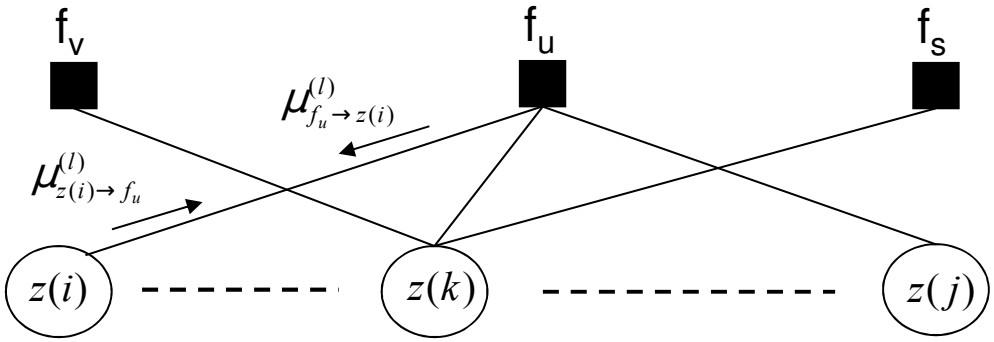

FIGURE 2.1 – Exemple d'un graphe de Tanner

Le décodeur MAP maximise la probabilité a posteriori au niveau symbole, sachant que le signal $\mathbf{R} = (R(0), \ldots, R(N-1))$ a été reçu :

$$\hat{z}(i) = \underset{z(i)}{\operatorname{argmax}}\, p(z(i)|\mathbf{R}) \; i = 0, \cdots, N-1$$

Le décodeur MLSE maximise la probabilité a priori de détecter la séquence entière :

$$\hat{\mathbf{z}} = \underset{\mathbf{z}}{\operatorname{argmax}}\, p(\mathbf{R}|\mathbf{z})$$

L'algorithme de décodage échange itérativement des 'messages' entre les variables et les nœuds de contrôle. En fonction du type de message échangé et si le graphe de Tanner est un arbre, on obtient un décodage de type MAP [15] ou MLSE [54]. S'il y a des cycles dans le graphe, on obtient une approximation du décodeur MAP ou MLSE. En pratique, si les cycles ont une longueur strictement supérieure à 4, le décodeur est performant.

Le message envoyé par la variable $z(i)$ au nœud f_u lors de l'itération l est défini par :

$$\mu^{(l)}_{z(i) \to f_u} = \Lambda(i)^{(l-1)} - \mu^{(l-1)}_{f_u \to z(i)} \tag{2.12}$$

$$\Lambda(i)^{(l-1)} = \mu_0(i) + \sum_{h \in B_i} \mu^{(l-1)}_{h \to z(i)} \tag{2.13}$$

B_i est l'ensemble des indices des nœuds connectés à la variable $z(i)$. $\mu_0(i)$ est le rapport de vraisemblance logarithmique (*Log Likelihood Ratio* en anglais, LLR) de la variable $z(i)$:

$$\mu_0(i) = \log\left(\frac{p(R(i)|z(i)=0)}{p(R(i)|z(i)=1)}\right) = 4\frac{R(i)}{\sigma_0^2} \tag{2.14}$$

L'algorithme *Sum-Product* (SPA) implémente d'une manière itérative le décodage MAP. Le message envoyé par le nœud f_u à la variable $z(i)$ à la $l^{\text{ième}}$ itération est défini de la manière

suivante [15] :

$$\mu^{(l)}_{f_u \to z(i)} = 2\text{atanh}\left(\prod_{j \in B_{u,i}} \tanh\left(\mu^{(l)}_{z(j)\to f_u}/2\right)\right) \tag{2.15}$$

$B_{u,i}$ est l'ensemble des indices des variables connectées au nœud f_u, à l'exception de $z(i)$.

L'algorithme *Min-Sum* (MS) implémente d'une manière itérative le décodage MLSE. Le message envoyé par le nœud f_u à la variable $z(i)$ est défini de la manière suivante [54] :

$$\mu^{(l)}_{f_u \to z(i)} = \left(\prod_{j \in B_{u,i}} \text{sgn}\left(\mu^{(l)}_{z(j)\to f_u}\right)\right) \min_{j \in B_{u,i}} \left(|\mu^{(l)}_{z(j)\to f_u}|\right) \tag{2.16}$$

l'algorithme MS est insensible à un facteur de multiplication commun à tous les LLR en entrée [54]. Ainsi, il n'est pas nécessaire d'estimer la variance du bruit σ_0^2, contrairement au SPA. D'un autre côté, l'algorithme MS est généralement moins performant en terme de BER et FER que le SPA [55][56]. Cependant, l'écart est très souvent relativement minime, ce qui justifie que l'algorithme MS soit utilisé en pratique.

Le code dual étant cyclique, si $c(x)$ est un de ses éléments, $xc(x), x^2c(x), \cdots, x^{N-1}c(x)$ sont aussi des éléments du dual. Il est donc suffisant de trouver les mots de code tels que $c(0) = 1$ pour en dériver $N-1$ autres. On peut donc construire des matrices de parité, telles que définies par (2.11), à partir d'un polynôme de parité initial de degré m : $c(x) = 1 + \cdots + x^m$. On nomme les polynômes $c(x)$, tels que $c(0) = 1$, *polynômes de référence*.

Pour le décodage des m-séquences, la probabilité de détection correcte est nettement améliorée lorsqu'on ajoute de la redondance au décodeur [30] [29]. Le principe est de concaténer plusieurs matrices de parité, chacune étant définie par un polynôme de référence $c_k(x)$ différent, pour former une matrice de parité globale :

$$E_{\text{eq}} = \begin{bmatrix} E_0 \\ E_1 \\ \vdots \\ E_{N_{eq}-1} \end{bmatrix} \tag{2.17}$$

où K est le nombre de polynômes de référence choisis pour le décodage.

Si E_{eq} est de rang $N-r$, alors son noyau définit le code simplex auquel appartient une m-séquence. Supposons que $\text{rg}(E_0) = N-r$, on peut se demander quel est l'apport des autres matrices de parité $E_1, \cdots, E_{K-1}$? Si on utilise un décodeur algébrique, la redondance n'apporte aucun bénéfice car ce qui importe c'est que la matrice de parité soit de rang $N-r$. Ce n'est cependant pas le cas avec un décodeur itératif par passage de message. En effet, les équations de parité supplémentaires vont modifier le graphe de Tanner du décodeur. Cela a pour effet de modifier les propriétés de certaines structures topologiques du graphe, nommées ensembles piégeants (*stopping sets*) ou ensembles absorbants (*absorbing sets*), et qui sont connues pour favoriser l'apparition d'un plancher d'erreurs lors d'un décodage itératif de type

SPA ou MS [33][34]. Le plancher est d'autant plus élevé que la taille de ces ensembles est petite. Si on ajoute de la redondance dans la matrice de parité, il est possible de réduire le nombre, voire d'éliminer ces ensembles de petite taille et ainsi d'améliorer les performances du décodeur [57][58][59]. Cette problématique sera détaillée et approfondie à la section 3.4.

Les polynômes de référence peuvent être trouvés par des méthodes exhaustives [26][23]. Si le polynôme caractéristique de la séquence est de poids faible (i.e. ≤ 5), il existe une méthode simple et efficace pour trouver des polynômes de parité [30]. Elle tire profit d'une propriété des polynômes caractéristiques des séquences LFSR. Si $g(x)$ est le polynôme caractéristique, alors $g(x^{2^n}) = g(x)^{2^n}$. Cette propriété assure que le polynôme $g_n(x) = g(x^{2^n})$ définit un mot du code dual (c'est un multiple de $g(x)$) et qu'il a le même poids que $g(x)$.

Dans la littérature sur le décodage des codes LDPC, il est connu que les performances sont meilleures lorsque le poids des équations de parité est faible [24]. On recherche donc les mots du code dual qui ont cette propriété. Dans le Chapitre 3, nous traiterons du nombre d'équations de poids $t = 3, 4$ ou 5 pour les m-séquences et les séquences de Gold.

Si l'algorithme a trouvé un mot de code (i.e. toutes les équations de parité sont satisfaites), le décodeur calcule la probabilité a posteriori $\Lambda(i)$ conformément à (2.13). Une décision dure est ensuite prise :

$$u_z(i) = \begin{cases} 0 \text{ si } \Lambda(i) \geq 0 \\ 1 \text{ sinon} \end{cases} \tag{2.18}$$

Si on considère la représentation de Fibonacci (Fig. 1.1), les r premiers bits correspondent à l'état initial des registres. L'opération de décodage est alors modélisée par une fonction $dec(.)$ qui produit un indicateur de convergence de l'algorithme I_c et un estimé de l'état initial de la séquence $\hat{u}_z$:

$$\{I_c, \hat{u}_z\} = dec(R(0), R(1), \cdots, R(N-1)) \tag{2.19}$$

I_c est défini par :

$$I_c = \begin{cases} 1 \text{ si toutes les équations de parité sont satisfaites} \\ 0 \text{ sinon} \end{cases} \tag{2.20}$$

2.4.3 Impact de l'hypothèse H_0 sur le décodage

Les hypothèses H_0 et H_1 induisent une modification du code perçu par le décodeur. En effet, l'hypothèse H_0 correspond à l'émission du mot 'nul' dans l'espace euclidien. Le décodeur doit donc détecter $N+1$ mots sachant que le mot nul ne possède pas les mêmes propriétés que les autres mots du code simplex. En particulier, la distance euclidienne entre 2 mots S_i et S_j du code simplex vaut :

$$d_{ij} = \sum_{k=0}^{N-1} \parallel (-1)^{s_i(k)} - (-1)^{s_j(k)} \parallel^2 = 2N - 2\theta(i-j)$$

où N est la longueur de la séquence et $\theta(k)$ est la fonction d'intercorrélation entre 2 m-séquences générées par le même polynôme caractéristique. D'après l'Eq. (1.3), on a $\theta(i-j) = -1$ si $i \neq j$, par on obtient :

$$d_{ij} = 2N + 2 \text{ si } i \neq j$$

D'un autre côté, la distance euclidienne entre chaque mot de code S_i et le mot nul vaut $d_{i0} = \sum_{k=0}^{N-1} \parallel (-1)^{s_i(k)} \parallel^2 = N$. On observe donc que la détection et le décodage conjoint introduit le mot de code nul, qui est plus proche de tous les mots du code simplex. Cela va donc modifier les performances du décodeur car les régions de décision optimales pour différentier les mots de code sont modifiées par l'introduction du mot nul. Merhav a identifié les nouvelles régions de décision optimales pour un canal discret sans mémoire [32]. Le cas du canal gaussien est toujours un sujet de recherche ouvert.

2.5 Conclusion

Dans ce chapitre, nous avons montré que le décodage d'une séquence LFSR correspond à un détecteur de type GLRT qui ne connaît pas la séquence émise, mais qui connaît sa méthode de construction (i.e. son polynôme caractéristique). Le décodeur implémente alors un GLRT qui utilise un décodeur pour estimer la séquence reçue. Il apparaît que le générateur LFSR produit les mots d'un code linéaire. Les séquences LFSR sont des mots de code qui peuvent être décodés avec un algorithme approprié. Le décodeur peut appliquer un critère MLSE ou MAP. Le décodeur MLSE exact est implémenté par un algorithme de Viterbi, qui s'avère impossible à réaliser pour les séquences employées habituellement. Le nombre d'états du décodeur est très élevé et la complexité devient prohibitive. On applique plutôt un décodage par passage de message qui permet d'obtenir une bonne approximation du décodeur MLSE ou MAP. L'approximation du décodeur MLSE est obtenue avec l'algorithme Min-Sum, alors que l'algorithme Sum-Product fournit celle du décodeur MAP. Dans ce chapitre, nous avons décrit le décodage d'une m-séquence. Cela a servi d'exemple pour expliquer le principe du décodeur. Dans le chapitre suivant, nous allons approfondir le cas des séquences de Gold.

Ce travail a donné lieu à la publication suivante :

- 'Improving the Decoding of M-Sequences by Exploiting their Decimation Property', European Signal Processing Conference (Eusipco), Marrakech, Maroc, Septembre 2013.

ode

Chapitre 3

Équations de parité

Sommaire

3.1 Introduction

Les équations de parité sont à la base du décodage par propagation de croyance des m-séquences et des séquences de Gold. Les performances de décodage dépendent fortement du poids de ces équations (i.e. du nombre de variables de chaque équation). Plus le poids est élevé plus les performances se dégradent [24]. Il est donc indispensable de trouver des équations de parité ayant un poids faible. Ce sujet de recherche a tout d'abord été initié dans le domaine de la cryptographie, pour les attaques sur les chiffreurs par flot [22][23]. Dans ce cadre, les travaux se sont focalisés sur la recherche d'équations de parité pour les m-séquences. Aucune solution algébrique permettant de trouver toutes les équations de parité d'un poids t donné n'a été pour l'instant trouvée. Les algorithmes proposés dans la littérature reposent sur des méthodes de recherche exhaustives, en essayant de limiter le nombre d'opérations [26].

Quant aux séquences de Gold, à l'exception notable du travail de Principe et al [29], la problématique de leur décodage itératif n'a pas été abordée jusqu'à présent dans le domaine des communications radio. Ce n'est cependant pas le cas dans le domaine du codage canal où l'étude des codes cycliques a été largement abordée et ceci dès les années 60. En particulier, Kasami a déterminé les poids des séquences de Gold, ainsi que le nombre d'équations de parités de poids $t = 3$ et 4 pour une séquence de Gold [9]. En effet, il y a des similitudes très fortes entre une séquence de Gold et un code BCH. Dans ce chapitre, nous allons tout d'abord donner le nombre d'équations de parité de poids t pour les m-séquences. Ces résultats sont tirés de la littérature. Nous allons ensuite réaliser la même opération pour les séquences de Gold. Nous nous appuierons sur les travaux de Kasami que nous avons étendus lorsque le degré des polynômes générateurs des 2 séquences s et y, formant la paire préférentielle, est impair.

Nous allons aussi étudier la valeur du plus petit degré possible parmi les équations de parité de poids t. Ce sujet est fondamental pour envisager une implémentation. Il est en effet utile pour minimiser le nombre d'opérations pour rechercher les équations de parité [25]. Il permet aussi de savoir rapidement si le décodage d'une séquence est réalisable. Ceci est illustré au chapitre 4 pour le décodage du code d'embrouillage du système WCDMA.

Dans une deuxième partie, nous allons proposer un algorithme pour sélectionner les équations de parité à utiliser lors du décodage. En effet, il est vite apparu que les probabilité de détection, mais surtout de fausse alarme étaient sensibles aux choix des équations de parité. Nous avons donc analysé la raison de cette sensibilité. Nous l'exposerons au travers d'une étude des ensembles absorbants (*Absorbing set*) générés par la matrice de parité du décodeur. Enfin, nous proposons un algorithme de sélection des équations, qui se fonde sur la minimisation du nombre de cycles de longueur 6 et 8 dans le graphe de Tanner.

3.2 Nombre d'équations de parité pour les m-séquences

Comme nous l'avons vu dans la section 2.4.1 du chapitre précédent, une m-séquence est un élément d'un code simplex. Son code dual est le code de Hamming $H_r = [N = 2^r-1, 2^r-r-1, 3]$ généré par $g(x)$, le polynôme caractéristique de la m-séquence [13]. Le terme '3' indique que ce code de Hamming possède une distance minimale $d_{min} = 3$.

Les équations de parité de poids t satisfaites par une m-séquence sont donc définies par les mots du code de Hamming H_r de poids t. Étant donné la m-séquence générée par $g(x)$, le code de Hamming est obtenu par tous les polynômes multiples de $g(x)$ et de degré inférieur ou égal à $2^r - 2$. Soit $h(x)$ le polynôme représentant un mot du code H_r. Si $h(x)$ est de la forme $h(x) = x^j(1+x^{i_1}+x^{i_2}+\cdots+x^{i_{t-1}})$, le mot de code est une version décalée de j bits du polynôme de référence $c(x) = 1+x^{i_1}+x^{i_2}+\cdots+x^{i_{t-1}}$. Il suffit donc de trouver les polynômes de référence de poids t pour trouver tous les autres.

Soit N_t^* le nombre de mots du code H_r ayant un poids t, et N_t le nombre de polynômes de référence multiples de $g(x)$. D'après [60], on a :

$$K_t^* = \frac{2^r - 1}{t} K_t \tag{3.1}$$

A partir de chaque polynôme de référence on obtient $2^r - 1$ autres polynômes de parité par un décalage circulaire à droite. On en obtient donc $(2^r - 1)N_t$. Cependant, si $j + i_u = 0 \bmod N$ pour $u = 1, \cdots, t-1$, on retrouve un polynôme de référence. On en trouve finalement t, le polynôme de départ plus les $t - 1$ autres. Par conséquence, le nombre de mots de code ayant un poids t est donné par l'Eq. (3.1).

De plus, en utilisant le polynôme énumérateur du code de Hamming, on obtient la relation de récurrence suivante [13, page 129] :

$$N_t = \frac{\begin{pmatrix} 2^r - 2 \\ t - 2 \end{pmatrix} - N_{t-1} - \left(\frac{t-1}{t-2}\right)(2^r - t + 1)\, N_{t-2}}{t - 1} \tag{3.2}$$

avec $N_1 = N_2 = 0$. A titre d'exemple, si $r = 10$ (système GPS), on trouve $N_3 = 511$ équations de parité de poids $t = 3$, ce qui est assez conséquent.

Lorsque le degré r est élevé, le terme $\begin{pmatrix} 2^r - 2 \\ t - 2 \end{pmatrix}$ devient prépondérant et on trouve l'approximation suivante :

$$N_t \approx \frac{2^{r(t-2)}}{(t-1)!} \tag{3.3}$$

Théorème 1 [61]
Soit le polynôme :

$$c(x) = x^{2(2^r-1)/3} + x^{(2^r-1)/3} + 1 \tag{3.4}$$

Si r est pair, $c(x)$ est un multiple de tous les polynômes primitifs de degré r.

Ceci signifie que l'équation de parité définie par $c(x)$ est valable pour toutes les m-séquences de degré r.

3.3 Équations de parité pour les séquences de Gold

Soit $z(k) = s(k) \oplus y(k)$ une séquence de Gold construite à partir de la paire de m-séquences préférentielle s et y (cf. section 1.5). Elle satisfait une équation de parité de poids t si :

$$\sum_{n=1}^{t} z(k + a_n) = \sum_{n=1}^{t} (s(k + a_n) \oplus y(k + a_n)) = 0 \ \forall k$$

Cela se traduit par :

$$\sum_{n=1}^{t} s(k + a_n) = \sum_{n=1}^{t} y(k + a_n) = 0$$

En effet, si ce n'est pas le cas, du fait de la propriété d'addition des m-séquences (cf. section 1.4), on doit avoir une relation de la forme $s(k+\tau_1) = y(k+\tau_2)$, ce qui est impossible par construction des séquences s et y (polynômes primitifs différents). Par conséquent, une équation de parité satisfaite par la séquence de Gold z est obligatoirement aussi satisfaite par les 2 m-séquences formant la paire préférentielle.

Soit α et β les racines primitives des polynômes générateurs $g_s(x)$ et $g_y(x)$. Les séquences s et y sont obtenues par la trace associée à leur racine primitive [8] :

$$s(k) = Tr(\theta_1 \alpha^k)$$
$$y(k) = Tr(\theta_2 \beta^k)$$

θ_1 et θ_2 spécifient l'état initial des registres des deux séquences. On a donc :

$$\sum_{n=1}^{t} s(k+a_n) = Tr\left(\theta_1 \alpha^k \sum_{n=1}^{t} (\alpha^{a_n})\right) = Tr(\theta_1 \alpha^k h(\alpha)) = 0$$

et de même

$$Tr(\theta_2 \beta^k h(\beta)) = 0$$

Par conséquent α et β sont des racines du polynôme $c(x) = \sum_{n=1}^{t} x^{a_n}$, représentant l'équation de parité :

$$c(\alpha) = c(\beta) = 0$$

Comme $\beta = \alpha^d$, le polynôme $c(x)$ doit satisfaire :

$$c(\alpha) = c(\alpha^d) = 0$$

Les polynômes $g_s(x)$ et $g_y(x)$ étant minimaux, $c(x)$ doit être un multiple de $g_s(x)$ et $g_y(x)$, et donc de $g_s(x)g_y(x) = \text{ppcm}(g_s(x), g_y(x))$. ppcm(a(x), b(x) est le plus petit commun multiple des polynômes $a(x)$ et $b(x)$.

Par conséquent, le code dual d'une séquence de Gold est un code cyclique C dont le polynôme générateur possède 2 racines : α et α^d où α est la racine primitive du polynôme caractéristique de la séquence s et d est le coefficient de décimation de la paire préférentielle. A titre d'exemple, si $d = 3$, le code dual est le code BCH capable de corriger 2 erreurs [13]. Le nombre d'équations de parité de poids t, noté N_t^*, est déterminé par le nombre de mots de code de C ayant un poids t. Parmi ces N_t^* équations, N_t correspondent à des polynômes de référence. Nous allons déterminer ce nombre en réutilisant les travaux de Pless et de Kasami [62][9].

3.3.1 Nombre d'équations de parité

Soit A_j le nombre de séquences de Gold de poids j. Ce nombre est évalué pour une paire de m-séquences préférentielles données. L'égalité de Pless relie les A_j et les N_j^* de la manière suivante [62] :

$$\sum_{j=0}^{N} (N-j)^t A_j = \sum_{j=0}^{t} \gamma_j N_j^* \tag{3.5}$$

où $N = 2^r - 1$ et

$$\gamma_j = \sum_{k=0}^{t} k! S(t,k) 2^{2r-k} \left(\begin{array}{c} N-j \\ N-k \end{array} \right) \tag{3.6}$$

$S(t,k)$ est le nombre de Stirling du deuxième type [63, §24.1.4] :

$$S(t,k) = \frac{1}{k!} \sum_{i=1}^{k} (-1)^{k-i} \left(\begin{array}{c} k \\ i \end{array} \right) i^t$$

Si on connaît les valeurs des A_j, il est alors possible de trouver N_t^* en résolvant itérativement le système linéaire formé par les $t-1$ premières égalités :

$$\gamma_t N_t^* = \sum_{j=0}^{N} (N-j)^t A_j - \sum_{j=0}^{t-1} \gamma_j N_j^* \tag{3.7}$$

Au final le nombre d'équations ayant un terme constant est obtenu en appliquant l'Eq. (3.1).

On notera que $A_0 = N_0^* = 1$. De plus $N_1^* = 0$ car sinon cela signifie que la séquence de Gold est nulle. De la même manière, $N_2^* = 0$ car sinon, cela signifie que chaque m-séquence est égale à une version décalée de l'autre, ce qui n'est pas possible. Les valeurs des A_j ont été calculées par Kasami dans son article sur les propriétés d'inter-corrélation des paires de m-séquences [9].

Soit $d = 2^e+1$ le facteur de décimation de la paire préférentielle (cf. section 1.5). Soit $\mathrm{pgcd}(a,b)$ le plus grand commun diviseur des entier a et b. Si $\mathrm{pgcd}(r,e) = \mathrm{pgcd}(r,2e) = c$, alors [9] :

$$\begin{array}{ll} A_0 = 1 & \\ A_{j_1} = (2^{r-c-1} + 2^{(r-c)/2-1})N & j_1 = 2^{r-1} - 2^{(r+c)/2-1} \\ A_{j_2} = (2^{r-c-1} - 2^{(r-c)/2-1})N & j_2 = 2^{r-1} + 2^{(r+c)/2-1} \\ A_{j_3} = (2^r - 2^{r-c} + 1)N & j_3 = 2^{r-1} \\ A_j = 0 & \text{pour les autres j} \end{array} \tag{3.8}$$

En pratique :
- Si $r = 2 \bmod(4)$: $c = 2$
- Si r est impair : $c = 1$

3.3.2 cas $t = 3$ et 4

Si on applique directement l'égalité de Pless (Eq. 3.5), on retrouve les résultats obtenus par Kasami pour $t = 3$ et $t = 4$ [9] :
- Si r est impair : $N_3 = N_4 = 0$
- Si r est pair et $r = 2 \bmod(4)$:

$$\begin{array}{l} N_3 = 1 \\ N_4 = (2^r - 4)/3 \end{array} \tag{3.9}$$

D'après le Théorème 1, lorsque r est pair, l'unique équation de parité de poids $t = 3$ est définie par :

$$c(x) = x^{2(2^r-1)/3} + x^{(2^r-1)/3} + 1$$

En effet, $c(x)$ est un multiple de $g_s(x)$ et $g_y(x)$ et donc un multiple du produit car les 2 polynômes sont minimaux. Il est remarquable qu'il n'y a pas d'équations de parité de poids $t < 5$ lorsque r est impair. Ceci a un impact important sur les performances de décodage qui sont d'autant meilleures que le poids t est faible.

3.3.3 Cas $t = 5$

Nous allons maintenant aborder le cas $t = 5$ lorsque r est impair. Il faut résoudre l'Eq. (3.7) pour $t = 5$:

$$N_5^* = \frac{1}{\gamma_5}\left(\sum_{j=0}^{N}(N-j)^5 A_j - \gamma_0\right)$$

où on a tenu compte que $N_0^* = 1$ et $N_1^* = N_2^* = N_3^* = N_4^* = 0$. Il faut donc calculer les 3 termes : γ_5, γ_0 et $\sum_{j=0}^{N}(N-j)^5 A_j$

γ_0 et γ_5 se calculent en appliquant directement l'Eq.3.6 :

$$\gamma_5 = 5!2^{2r-5}$$

$$\gamma_0 = 2^{2r-5}(N^5 + 10N^4 + 15N^3 - 10N^2)$$

Le troisième terme vaut :

$$\begin{aligned}\sum_{j=0}^{N}(N-j)^5 A_j = \quad & a^5(2^{2r}-1) + 5a^4(2^{2r-1} - 2^{r-1}) + 10a^3(2^{r-2} - 2^{2r-2}) \\ & +10a^2(2^{3r+c-3} - 2^{2r+c-3}) + 5a(2^{4r+c-4} - 2^{3r+c-4}) \\ & +N^5 + 2^{4r+2c-5} - 2^{3r+2c-5}\end{aligned} \tag{3.10}$$

où $a = 2^{r-1} - 1$.

En développant les termes en a^j et $N^j = (2^r - 1)^j$, on obtient :

$$N_5^* = (2^{5r-5} - 11\,2^{4r-5} + 26\,2^{3r-5} - 16\,2^{2r-5})/(5!2^{2r-5})$$

soit

$$N_5 = \frac{2^{3r} - 11\,2^{2r} + 26\,2^r - 16}{24(2^r - 1)} = \frac{2^{2r} - 10\,2^r + 16}{24}$$

et finalement :

$$N_5 = \frac{2(2^{r-3} - 1)(2^{r-1} - 1)}{3} \tag{3.11}$$

Les tables 3.1 et 3.2 donnent le nombre d'équations trouvées pour $t = 4$ et 5. On observe que ce nombre est particulièrement élevé, ce qui signifie que ce ne sera pas un facteur limitant pour construire un décodeur.

r	6	10	18
N_4	20	340	87380

TABLE 3.1 – Nombre d'équations de parité de poids $t = 4$ (r pair).

r	7	9	11
N_5	630	10710	173910

TABLE 3.2 – Nombre d'équations de parité de poids $t = 5$ (r impair).

3.3.4 Degré minimal

Connaître le nombre d'équations de parité est utile pour savoir si de telles équations existent. Cependant, d'un point de vue pratique, il est fondamental de connaître, parmi l'ensemble des équations de parité d'un poids t, celles qui ont le degré le plus faible. En effet, le décodeur itératif va uniquement considérer un nombre restreint d'équations de parité et sélectionner celles qui ont le degré le plus faible. Quelle raison à cela ? Supposons que le décodeur considère un vecteur d'observations de $M < N$ chips. Si le degré de l'équation de parité vaut m, alors le décodeur ne peut exploiter que $M - m$ équations de parité, obtenues par décalage cyclique à droite. Il est dont important que m soit le plus petit possible, car les performances de décodage dépendent de $M - m$. Elles s'améliorent si $M - m$ augmente. De plus, connaître la valeur du degré minimal permet de minimiser le nombre de calculs lors de la recherche des équations de parité [25].

Nous allons présenter une méthode d'estimation de ce degré minimal. Elle est détaillée dans [60]. Elle a été proposée pour le cas des m-séquences, mais il s'avère que le raisonnement et les résultats s'appliquent directement aussi au cas des séquences de Gold.

Soit $M_1 = max(i_1, \cdots, i_{t-1})$, où $(i_1, \cdots, i_{t-1})$ est le $(t-1)$-uplet des coefficients d'une équation de parité de poids t : $c(x) = 1 + x^{i_1} + \cdots + x^{i_{t-1}}$. On a $M_1 = i_{t-1}$ en considérant que les coefficients sont classés dans l'ordre croissant.

Soit $M_2 = max(i_1, \cdots, i_{t-1})$, où $(i_1, \cdots, i_{t-1})$ est n'importe lequel des $(t-1)$-uplets prenant ses valeurs entre 1 et $N-1$. Il en existe $N_{upplets} = \begin{pmatrix} N-1 \\ t-1 \end{pmatrix}$ au total. Si les $(t-1)$-uplets sont ordonnés par ordre croissant, alors $M_2 = i_{t-1}$.

La distribution des $(t-1)$-uplets $(i_1, \cdots, i_{t-1})$ des équations de parité montre un caractère aléatoire et semble être uniforme. La méthode d'estimation repose sur l'hypothèse que les variables M_1 et M_2 suivent la même distribution. Si on sélectionne aléatoirement un $(t-1)$-uplets, la probabilité qu'il corresponde à une équation de parité vaut alors $N_t/N_{upplets}$. Le nombre moyen d'équations de parité de degré inférieur ou égal à m est donc :

$$J_m = \begin{pmatrix} m \\ t-1 \end{pmatrix} \frac{N_t}{N_{upplets}} \tag{3.12}$$

La validité du modèle est illustré par les figures 3.1, 3.2 et 3.3 qui comparent la mesure de J_m obtenue par simulation, avec le modèle de l'Eq. 3.12 pour $r = 10$, 11 et 14.

Le degré minimal m_0 est obtenu pour $J_{m_0} = 1$. Il satisfait donc l'équation :

$$N_t \begin{pmatrix} m_0 \\ t-1 \end{pmatrix} = \begin{pmatrix} N-1 \\ t-1 \end{pmatrix} \tag{3.13}$$

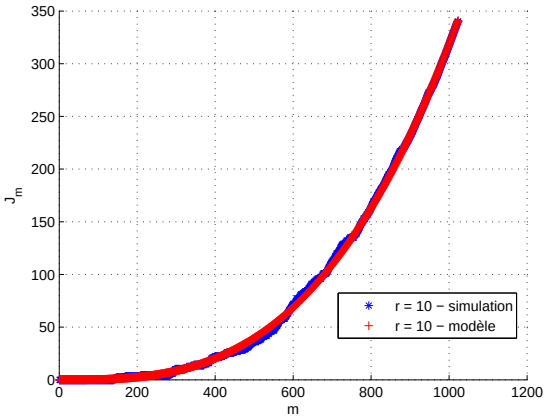

FIGURE 3.1 – J_m en fonction de m pour $r = 10$

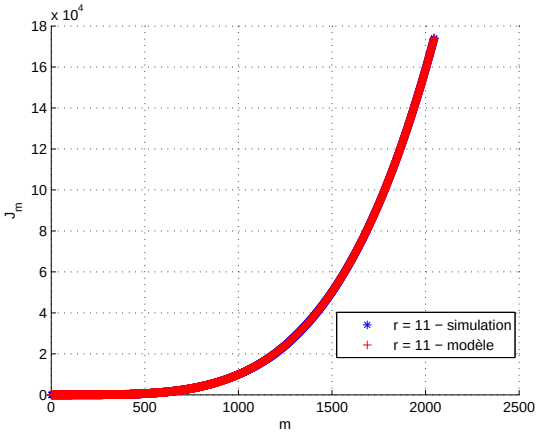

FIGURE 3.2 – J_m en fonction de m pour $r = 11$

Il est possible de trouver une approximation de cette estimation pour m_0 et r grands, ce qui est le cas en pratique. Soit les approximations suivantes :

$$\begin{aligned} &m_0(m_0-1)\cdots(m_0-t+2) \approx m_0^{t-1} \\ &(N-1)(N-2)\cdots(N-t+1) \approx (N-1)^{t-1} \approx N^{t-1} \end{aligned} \tag{3.14}$$

On obtient alors :

$$m_0 \approx \frac{N}{(N_t)^{1/(t-1)}} \tag{3.15}$$

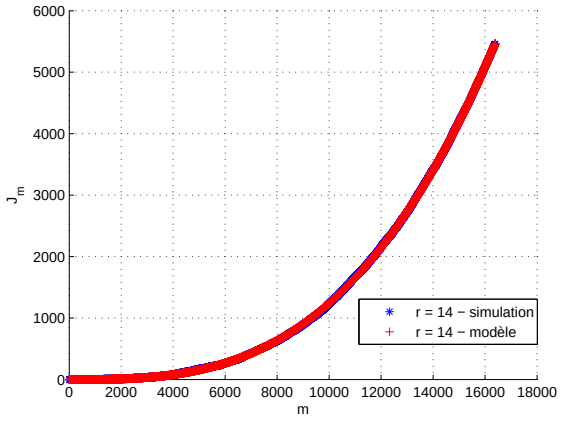

FIGURE 3.3 – J_m en fonction de m pour $r = 14$

Pour des m-séquences, en appliquant l'approximation de l'Eq. 3.3, on trouve :

$$m_0(t) \approx 2^{r/(t-1)}((t-1)!)^{1/(t-1)} \tag{3.16}$$

Cela indique que le degré minimal diminue lorsque le poids augmente. Ceci explique probablement pourquoi les équations de poids $t = 4$ ou 5 ont été choisies pour les attaques sur les chiffreurs par flot au détriment des équations de poids $t = 3$ [23][26]. En effet, dans ces articles le degré du polynôme générateur de la séquence vaut $r = 40$, ce qui donne : $m_0(3) \approx 1048576$, $m_0(4) \approx 10297$ et $m_0(5) \approx 1024$. Les besoins en mémoire sont nettement moindres pour $t = 4$ ou 5 que pour $t = 3$.

Pour des séquences de Gold, en utilisant les Eq. (3.9) et (3.11), on trouve :

$$\begin{array}{ll} m_0(4) \approx 1.44\,2^{2r/3} & r \text{ pair} \\ m_0(5) \approx 2.21\,2^{r/2} & r \text{ impair} \end{array} \tag{3.17}$$

Le degré minimal est nettement plus élevé que pour les m-séquences. On passe d'un ordre de grandeur de $2^{r/4}$ pour les m-séquences de poids $t = 5$ à $2^{r/2}$ pour les séquences de Gold de même poids.

Le Tableau 3.3 répertorie les résultats obtenus par une recherche exhaustive ('simulation'), par la méthode d'estimation décrite au-dessus, ainsi que l'approximation proposée. Les résultats montrent que la méthode d'estimation est relativement précise, ainsi que l'approximation. On en déduit par exemple qu'il est possible de détecter le canal CPICH utilisé par la liaison descendante du système WCDMA[3], avec les équations de parité de poids $t = 4$ que l'on aura trouvé avec une recherche appropriée [26]. En effet, dans ce cas, on a $r = 18$ et $m_0 = 5907$ ce qui est inférieur à la durée d'une trame (38400 chips).

Les résultats présentés dans le Tableau 3.3 sont cependant issus d'une mesure avec une seul point. Une seule séquence de Gold a été testée pour un degré de polynôme r donné. Pour remédier à ce problème, toutes les séquences de Gold d'un degré $r = 6, 7, 9, 10, 11, 13, 18$ et 19 ont été générées. Il a fallu pour cela trouver toutes les paires de m-séquences préférentielles

TABLE 3.3 – Degré minimal de l'équation de parité de poids $t = 4$ ou 5 (pour une séquence de Gold).

r	7	9	10	11	13	14	18	25
t	5	5	4	5	5	4	4	5
Simulation	24	48	111	114	219	1103	9822	-
Estimation	27	52	148	102	202	930	-	-
Approximation	25	50	146	100	200	930	5907	12821

pour chaque degré r. Les figures 3.4 et 3.5 montrent la distribution des degrés minimaux des équations de parité pour les séquences de Gold de degré $r = 11$ et 13. Il apparaît que la valeur moyenne du degré minimal est proche de la valeur prédite par l'Eq. (3.15). Cependant, il existe une disparité entre les différentes séquences de Gold. La variance du degré minimal augmente avec la longueur de la séquence, comme l'illustre le Tableau 3.4. Cette variance d'estimation a un impact important lorsque r est grand. L'erreur d'estimation est élevée. Par exemple, pour $r = 18$, le degré minimal moyen prévu par le modèle vaut $m_{0,estimation} = 5907$. Dans la pratique, si on considère la séquence de Gold générée par les séquences dont les polynômes générateurs sont 1000201 et 1002241 en représentation octale, on trouve $m_{0,simulation} = 9822$. On en déduit que le modèle de prédiction de m_0 est relativement précis pour estimer la valeur moyenne du degré minimal. Il ne prend cependant pas en compte la variance qui s'avère importante. Il reste néanmoins utile car il donne une bonne estimation de l'ordre de grandeur de m_0. Cela permet de savoir rapidement s'il sera possible d'implémenter un décodeur avec ces équations de parités.

r	10	11	13
σ_{m_0}	27	25	58

TABLE 3.4 – Variance de m_0.

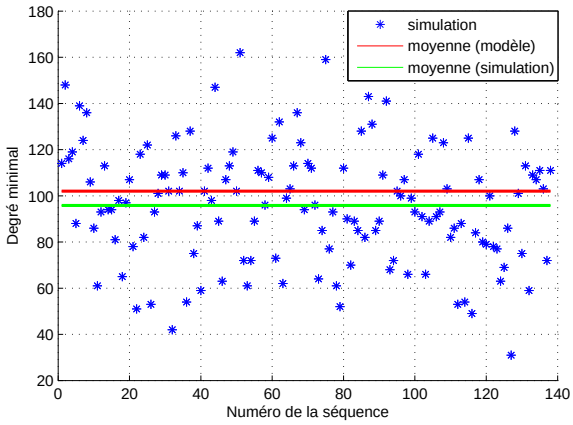

FIGURE 3.4 – Degré minimal des équations de parité pour les séquences de Gold de degré $r = 11$

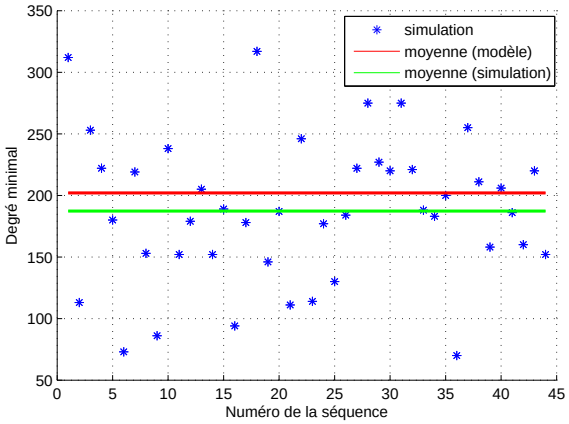

FIGURE 3.5 – Degré minimal des équations de parité pour les séquences de Gold de degré $r = 13$

3.4 Sélection des équations de parité pour les m-séquences

Dans le chapitre précédent, nous avons évalué le nombre d'équations de parité de poids t et nous avons vu que cela ne constituerait pas un facteur limitatif pour construire un décodeur. Il y aura suffisamment d'équations. Il reste donc à trouver ces équations, et ensuite à sélectionner celles qui seront utilisées pour construire le décodeur. La problématique de trouver les équations de parité a été largement abordée dans le domaine de la cryptographie pour attaquer les chiffreur par flot [25] [23] [26]. Nous avons décidé dans cette thèse de ne pas aborder ce sujet. Pour nos simulations, nous avons trouvé les équations de parité par une recherche exhaustive. Nous nous sommes concentrés sur la sélection des équations de parité, de manière à obtenir une probabilité de fausse alarme la plus faible possible.

Une équation de parité est définie par une relation de la forme suivante : $y(k) \oplus y(k+i_a) \oplus \cdots \oplus y(k+r_a) = 0$. On lui associe le polynôme de parité $g_a(x) = 1 + x^{i_a} + \cdots + x^{r_a}$. Le degré de l'équation est r_a et son poids vaut t. Il correspond au nombre de coefficients non nuls de $g_a(x)$. Il est connu dans la littérature [24] que les performances de décodage décroissent lorsque t augmente. On cherche donc à implémenter le décodeur des m-séquences ou des séquences de Gold avec t le plus faible possible. Nous avons montré dans la section précédente qu'il y a beaucoup d'équations de parité de poids $t = 3$ pour les m-séquences. Pour les séquences de Gold, cela dépend de la parité du degré des polynômes générateurs (cf. paragraphe 1.5).

Lorsque $t = 3$, les premiers résultats de simulation ont mis en évidence l'impact significatif du choix des polynômes de parité sur la probabilité de fausse alarme P_{FA}. Celle-ci peut par exemple varier de 0.1 à 10^{-4} lorsqu'on considère un décodage avec $K = 5$ polynômes de référence. Si $t > 3$, quelque soit la combinaison des polynômes choisis, nous n'avons jamais observé de fausse alarme : $P_{FA} \approx 0$. Nous avons donc concentré notre étude au cas $t = 3$ qui correspond au décodage des m-séquences. En effet, pour les séquences de Gold, il n'existe qu'un seul polynôme de parité de poids $t = 3$, et encore, uniquement si r est pair (cf. section

3.3).

Le problème que l'on cherche à résoudre peut être formulé de la manière suivante : étant donné un ensemble de N_t polynômes de référence de poids t, comment sélectionner les K polynômes qui vont minimiser la probabilité de fausse alarme P_{FA} ? Il est connu dans la littérature sur les codes LDPC que minimiser le nombre de cycles courts dans le graphe de Tanner est un bon critère de conception de ces codes. Cette approche est suivie dans de nombreuses contributions pour optimiser les performances de décodage ou bien concevoir de nouveaux codes [64]. Dans notre contexte, cela correspond à maximiser la probabilité de détection correcte P_{CD}.

La justification de notre démarche est la suivants : si le graphe de Tanner déduit de E est un arbre (i.e. il n'a pas de cycle), l'algorithme SPA est équivalent au décodeur MAP [15] et l'algorithme MS est équivalent au MLSE [54]. Le décodage est donc optimal. Par conséquent, tendre vers une configuration sans cycle est une bonne stratégie de conception. Ceci est réalisé en minimisant le nombre de cycles courts. En suivant cette piste, notre approche a consisté à tout d'abord éliminer les configurations donnant lieu à des cycles de longueur 4, puis de choisir celles qui minimisent le nombre de cycles de longueur 6 (I_6) et 8 (I_8). Cela repose donc sur la matrice de parité qui est choisie pour le décodage.

Le décodage des m-séquences a été détaillé au paragraphe 2.4.3. Une matrice de parité élémentaire E_a correspondant au polynôme de référence $g_a(x) = \sum_{k=0}^{r_a} g_{a,k} x^k$ est définie par :

$$\mathbf{E}_a = \begin{bmatrix} g_{a,r_a} & \cdots & \cdots & g_{a,0} & 0 & \cdots & \cdots & 0 \\ 0 & g_{a,r_a} & \cdots & \cdots & g_{a,0} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & g_{a,r_a} & \cdots & \cdots & g_{a,0} \\ g_{a,0} & \cdots & \cdots & \cdots & 0 & g_{a,r_a} & \cdots & g_{a,1} \\ \vdots & \ddots & \cdots & \cdots & 0 & \ddots & \ddots & \vdots \\ g_{a,r_a-1} & \cdots & g_{a,0} & 0 & \cdots & \cdots & 0 & g_{a,r_a} \end{bmatrix} \tag{3.18}$$

où $g_{a,0} = g_{a,r} = 1$

La matrice E_a est Toeplitz, et circulante si le récepteur observe la séquence sur toute sa longueur N. Si K polynômes $g_{a_0}(x), \ldots, g_{a_{K-1}}(x)$ sont sélectionnés, le décodeur concatène les K matrices élémentaires pour former une matrice de parité globale $E = [E_{a_0}^T \ldots E_{a_{K-1}}^T]^T$. On note aussi que lorsque la matrice de parité est circulante, les variables participent toutes au même nombre d'équations de parité et ont donc la même probabilité d'être corrigées. Ceci améliore la probabilité de décodage, comme cela avait été relevé dans [59]. Il faut néanmoins s'assurer que la matrice E soit de rang $N - r$, afin que le mot trouvé appartienne au code. Il suffit pour cela qu'une des matrices E_{a_i} soit de rang $N - r$. Cette propriété est satisfaite si le polynôme $p_i(x) = pgcd(g_{a_i}(x), x^N + 1)$ est de degré r [65].

Le décodeur applique un algorithme de décodage par passage de messages sur le graphe de Tanner défini par la matrice de parité globale E. Les premiers résultats de simulation ont été surprenant et contre-intuitifs. La probabilité d'erreur de détection $P_e = 1 - P_{CD} = P_{ND} + P_{WD}$

augmente lorsque I_6 diminue, alors qu'on s'attendait à l'effet inverse. D'un autre côté, il est apparu que P_{FA} se réduisait lorsque I_6 diminuait. Nous avons donc mené une étude pour comprendre l'origine de ce phénomène. Il peut être expliqué par l'existence d'ensembles absorbants (*absorbing sets*) dans le graphe de Tanner du décodeur [58]. Nous allons donc tout d'abord expliquer comment la présence de certains ensembles absorbants réduit P_{FA} mais augmente P_e. Nous allons aussi montrer par un exemple comment des cycles "transverses" de longueur 6 peuvent détruire ces ensembles absorbants et ainsi augmenter P_{FA}. Un cycle transverse fait intervenir des équations de parité appartenant à plusieurs matrices élémentaires E_{a_i}. Nous en déduirons un algorithme de sélection de la configuration des polynômes de référence $g_{a_0}(x), \ldots, g_{a_{K-1}}(x)$ minimisant le nombre de cycles transverses de longueur 6 et 8. Il apparaît en effet prioritaire de minimiser P_{FA}, même si cela s'accompagne d'une légère augmentation de P_e. Cela minimise le temps d'acquisition, comme nous le montrerons à la section 3.5.

I_6 et I_8 sont évalués en appliquant la méthode d'énumération proposées par Halford et Chugg pour compter les cycles courts dans un graphe de Tanner [66]. La structure Toeplitz et la propriété de circularité des matrices E_a sont mises à profit pour obtenir une expression analytique de I_6 et I_8. Elles dépendent du nombre d'équations K, de la longueur de la séquence N, du poids des équations t et des polynômes de référence sélectionnés. Ces expressions sont ensuite utilisées pour trouver la combinaison de polynômes minimisant les nombres de cycles de longueur 6 et 8.

3.4.1 Ensembles absorbants

3.4.1.1 Définition

Les ensembles absorbants ont été introduits pour expliquer les planchers d'erreurs qui apparaissent lors du décodage de certains codes LDPC [34][67]. Ce sont des structures topologiques particulières du graphe bipartite, qui expliquent la non-convergence de certains algorithmes de décodage. Nous allons donc réutiliser les notations définies à la section 2.4.3. On note V l'ensemble des variables à décoder et F l'ensemble des nœuds de contrôle.

Définition : Ensemble absorbant [58]
Pour un sous-ensemble D de V, on définit $O(D)$ (resp. $E(D)$) l'ensemble des nœuds de contrôle connectant un nombre impair (resp. pair) de variables de D. Un ensemble absorbant de paramètre (a, b) est un sous-ensemble D de V tel que :

1. D contient a variables et $O(D)$ contient b nœuds de contrôle.
2. chaque variable de D a strictement plus de voisins dans $E(D)$ que dans $O(D)$.

Définition : Ensemble absorbant complet [58]
Un (a, b) ensemble absorbant est complet si toutes les variables dans $V \setminus D$ ont strictement plus de voisins dans $F \setminus O(D)$ que dans $O(D)$. En d'autres termes, cela signifie que les variables n'appartenant pas à l'ensemble absorbants doivent avoir moins de connexions à $O(D)$ qu'au reste des nœuds de contrôle de F à l'exclusion de $O(D)$.

Un ensemble absorbant complet est un cas particulier des ensembles piégeant (*trapping set*), avec la particularité d'être un point fixe de l'algorithme de *Bit Flipping* [11]. Cela signifie que si les variables d'un ensemble absorbant complet D sont toutes erronées et que les autres variables sont correctes, elles ne pourront pas être corrigées par cet algorithmes de décodage. Cette propriété rend les ensembles absorbants complets responsables des planchers d'erreur lors du décodage des codes LDPC [67][57].

Habituellement, on désire éliminer ces ensembles pour supprimer les planchers d'erreurs. Dans notre contexte, nous allons prendre le contrepied et chercher au contraire à s'assurer de leur présence. En effet, dans une situation où le décodeur n'est alimenté que par du bruit, nous souhaitons que le décodeur ne converge pas vers un mot de code. Sinon, nous obtenons une fausse alarme. On désire au contraire que le bruit 'active' un ou plusieurs ensembles absorbants complets afin de garantir la non-convergence et ainsi éliminer les fausses alarmes. Ceci est obtenu si les équations de parité de $E(D)$ sont satisfaites et celles de $O(D)$ ne le sont pas.

Si le nombre d'ensembles est suffisamment important et qu'ils ont une petite taille, la probabilité d'en activer un sera élevée. D'un autre côté, lorsque le décodeur est alimenté avec une séquence et du bruit, la présence de ces ensembles va dégrader la probabilité de détection. On retrouve le compromis usuel entre la probabilité de détection et de fausse alarme [41]. La différence avec les systèmes de détection conventionnels est que le compromis repose sur le nombre et la taille des ensembles absorbants complets du graphe de Tanner.

Nous allons maintenant proposer une méthode de construction des plus petits ensembles absorbants pour K polynômes de référence. Cela va permettre d'expliquer pourquoi certaines configuration de polynômes détruisent les ensembles absorbants complets et induisent un taux de fausse alarme plus élevé.

3.4.1.2 Méthode de construction

La Figure 3.6 montre un cycle de longueur 6 passant par les variables $y(k)$, $y(k+i_a)$ et $y(k+i_a-r_a)$. Chaque équation de parité $F_a(k)$ correspond à la $k^{\text{ième}}$ ligne de la matrice E_a : $y(k) \oplus y(k+i_a) \oplus y(k+r_a) = 0$. D'après la définition, les variables $y(k), y(k+i_a)$, et $y(k-r_a+i_a)$ forment un ensemble absorbant complet. Deux autres cycles de longueur 6 sont obtenus en remplaçant k par $k-i_a$ et k par $k+r_a-i_a$. Au final, chaque matrice de parité E_a contient N ensembles absorbants complets de paramètre $(3,3)$.

Propriété : Une matrice E_a contient des cycles de longueur 8 si et seulement s'il existe deux entiers p et q tels que $1 \leq p,q \leq 4$ et $pi_a = qr_a$.

Cette propriété se démontre en énumérant les configurations théoriques correspondant aux cycles de longueur 8. La démonstration n'a pas été inclue dans ce mémoire.

Si $K > 1$, cette propriété n'est plus vraie. La Figure 3.7 montre les cycles de longueur 8 contenant la variables $y(k)$ et $y(k+i_a)$, et mettant en jeu les nœuds de 2 matrices E_a et E_b lorsque $K = 2$.

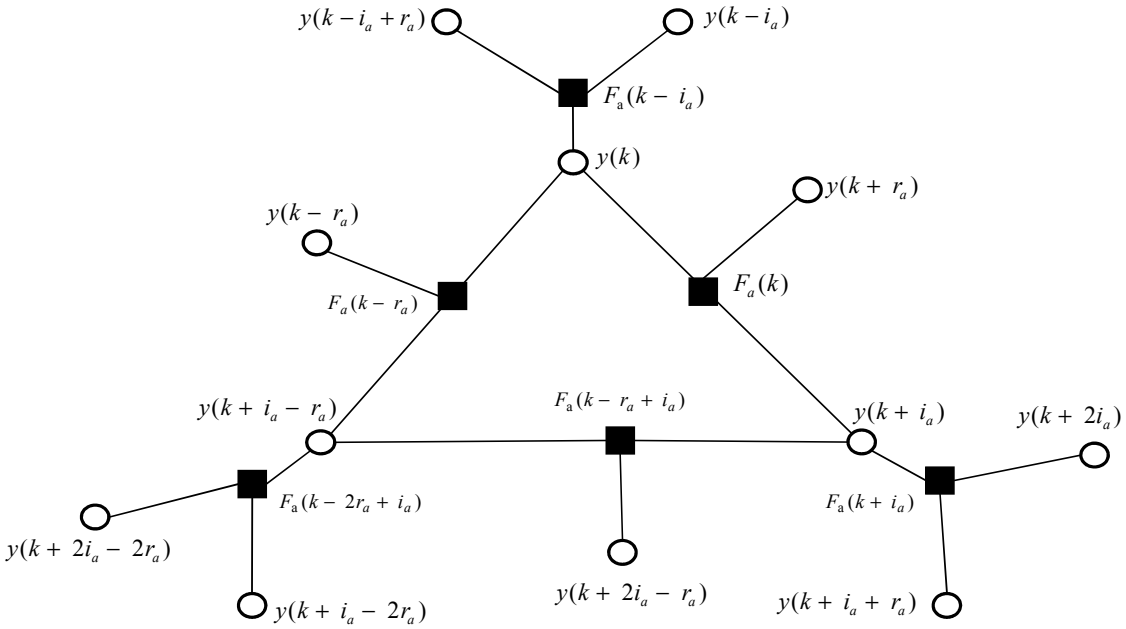

FIGURE 3.6 – Cycle de longueur 6

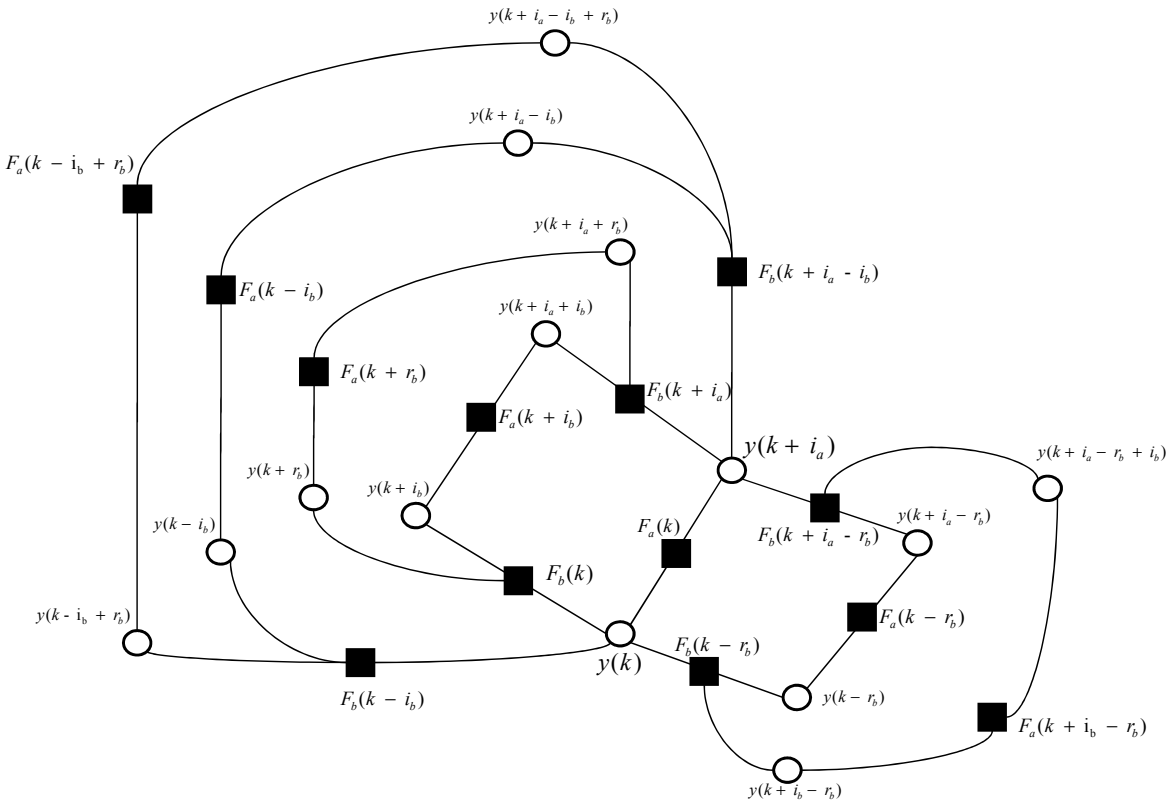

FIGURE 3.7 – Cycle de longueur 8 contenant $y(k)$ et $y(k+i_a)$

La méthode de construction du plus petit ensemble absorbant $D_K(k)$ pour K polynômes de référence utilise celui qui a été trouvé pour $K-1$ polynômes ($D_{K-1}(k)$). L'indice k indique que l'ensemble absorbant contient la variable $y(k)$. Les autres ensembles du graphe sont obtenus par un simple décalage cyclique.

On distingue 2 cas : K est pair ou impair. Si K est pair ($K = 2Q$), alors chaque variable de l'ensemble absorbant $D_K(k)$ soit être reliée à $3Q + 1$ nœuds de contrôle de $E(D_K(k))$. Ceci est dû au fait que chaque variable de $D_K(k)$ doit est connecté à *strictement* plus de nœuds de $E(D_K(k))$ que de $O(D_K(k))$. Si K est impair ($K = 2Q + 1$), chaque variable doit être relié à $3Q + 2$ nœuds de contrôle de $E(D_K(k))$.

On note $D_K(k) = \{y(k), y(k + \alpha_1), \cdots, y(k + \alpha_{a_K-1})\}$ les variables de l'ensemble absorbant contenant a_K éléments, dont la variable $y(k)$. Si on ajoute une matrice de parité de référence ayant pour polynôme $g_c(x) = 1 + x^{i_c} + x^{r_c}$, la matrice E_c contient des cycles de longueur 6 comme celui montré à la Figure 3.6. Les variables de ce cycle sont $\{y(k), y(k + \beta_c), y(k + \gamma_c)\}$ où les paramètres β_c et γ_c ne dépendent que de i_c et r_c.

Supposons que K soit impair : $K = 2Q + 1$. Pour que l'ensemble $D_{2(Q+1)}(k)$ soit absorbant, il faut ajouter 2 connexions dans $E(D_{2Q+2}(k))$ à chaque variable de $D_{2Q+1}(k)$. Ceci est réalisé par duplication et translation de $D_{2Q+1}(k)$:

$$D_{2(Q+1)}(k) = D_{2Q+1}(k) \cup D_{2Q+1}(k + \beta_c) \cup D_{2Q+1}(k + \gamma_c) \tag{3.19}$$

Nous devons vérifier que $D_{2(Q+1)}(k)$ est bien un ensemble absorbant. Nous allons pour cela compter le nombre de connexions de $y(k)$ dans $E(D_{2(Q+1)}(k))$. Sachant que $y(k) \in D_{2Q+1}(k)$, $y(k)$ possède $3Q + 2$ connexions avec $E(D_{2Q+1}(k))$. De plus, $y(k)$ appartient au cycle $\{y(k), y(y + \beta_c), y(k + \gamma_c)\}$, de longueur 6, ce qui ajoute 2 connexions à $E(D_{2(Q+1)}(k))$. Par conséquent, $y(k)$ possède $3Q + 2 + 2 = 3(Q + 1) + 1$ connexions avec $E(D_{2(Q+1)}(k))$ ce qui est strictement supérieur au nombre de connexions avec $O(D_{2(Q+1)}(k))$. L'ensemble $D_{2(Q+1)}(k)$ est donc absorbant.

Supposons que K soit pair : $K = 2Q$. Pour que l'ensemble $D_{2Q+1}(k)$ soit absorbant, il faut ajouter une seule connexion à chaque variable de $D_{2Q}(k)$. Ceci est réalisé par duplication et translation de $D_{2Q}(k)$:

$$D_{2Q+1}(k) = D_{2Q}(k) \cup D_{2Q}(k + i_c) \tag{3.20}$$

La translation par i_c peut être remplacée par tous les indices des variables connectées à $y(k)$ par l'intermédiaire des nœuds de contrôle de la matrice E_c, soit : $-i_c$, r_c, $-r_c$, $r_c - i_c$ et $i_c - r_c$.

Sachant que $y(k) \in D_{2Q}(k)$, $y(k)$ possède $3Q + 1$ connexions avec $E(D_{2Q}(k))$. De plus, $y(k)$ est connecté avec $y(k + i_c)$ par l'intermédiaire du nœud de contrôle $F_c(k)$. Par conséquent, $y(k)$ possède $3Q + 2$ connexions avec $E(D_{2Q+1}(k))$ ce qui est strictement supérieur au nombre de connexions avec $O(D_{2Q+1}(k))$. L'ensemble $D_{2Q+1}(k)$ est donc absorbant.

On trouve une relation de récurrence sur la taille des ensembles absorbants construits selon la méthode que nous avons exposée : $a_{2Q} = 3a_{2Q-1}$ et $a_{2Q+1} = 2a_{2Q}$. Sachant que $a_1 = 3$, on

obtient finalement :

$$a_{2Q} = 2^{Q-1}3^{Q+1} \tag{3.21}$$
$$a_{2Q+1} = 2^{Q}3^{Q+1} \tag{3.22}$$
$$\tag{3.23}$$

Le nombre d'ensembles absorbants contenant la variable $y(k)$ vaut :

$$3a_{2Q-1} = 2^{Q-1}3^{Q+1} = a_{2Q} \text{ si } K = 2Q \tag{3.24}$$
$$6a_{2Q} = 2^{Q}3^{Q+2} = 3a_{2Q+1} \text{ si } K = 2Q+1 \tag{3.25}$$
$$\tag{3.26}$$

Cette méthode de construction des ensembles absorbants suppose que chaque nœud de contrôle reliant 2 variables de l'ensemble n'est pas connecté à une troisième de ce même ensemble. En effet, si cela arrive, ce nœud n'appartient plus à $E(D_K)$ mais à $O(D_K)$. Par conséquent, les variables reliées à ce nœud auront plus de connexions avec $O(D_K)$ qu'avec $E(D_K)$ et l'ensemble ne sera plus absorbant. Nous allons montrer maintenant que cette situation est susceptible d'apparaître s'il y a des cycles transverses dans le graphe. Un cycle transverse fait intervenir des équations de parité appartenant à plusieurs matrices élémentaires E_{a_i}.

Supposons que D_K est un ensemble absorbant. Par construction, D_{K+1} est construit par duplication et translation de D_K. Les liens entre les différents ensembles translatés sont réalisés par les nœuds de contrôle de la matrice de parité qui est ajouté. Par exemple, si $K = 2Q$, les variables des ensembles $D_{2Q}(k)$ et $D_{2Q}(k+i_c)$ de l'ensemble absorbant $D_{2Q+1}(k) = D_{2Q}(k) \cup D_{2Q}(k+i_c)$ sont reliées par des nœuds de contrôle de type $F_c(k+\alpha+i_c)$, où α est un indice d'une variable appartenant à $D_{2Q}(k)$. Supposons que le nœud $F_c(k+i_c)$ qui relie $y(k)$ à $y(k+i_c)$ connecte 3 variables de $D_{2Q+1}(k)$, c'est à dire qu'il n'est plus dans $E(D_{2Q+1}(k))$. Il est donc relié à une autre variable que nous nommerons $y(k+\alpha)$ par l'intermédiaire du nœud $F_c(k+i_c)$. Nous supposerons aussi que $y(k+\alpha) \in D_{2Q}(k)$. Si $y(k+\alpha) \in D_{2Q}(k+i_c)$, le raisonnement qui va suivre est aussi valable en remplaçant k par $k+i_c$.

Si la variable $y(k+\alpha)$ appartient à $D_{2Q}(k)$, il existe déjà un chemin entre $y(k)$ et $y(k+\alpha)$ dans le sous-graphe induit par les variables de $D_{2Q}(k)$ et les nœuds de contrôle reliant les variables $D_{2Q}(k)$. Cette propriété se démontre facilement par récurrence. Ce chemin ne contient pas de nœud de type $F_c(k+...)$. Par conséquent, si $F_c(k+i_c)$ connecte $y(k)$ et $y(k+\alpha)$, cela crée un cycle dans le graphe. Ce cycle contient des nœuds appartenant à des matrices élémentaires différentes. Ces cycles sont identifiés sous l'appellation de 'cycles transverses'.

3.4.1.3 Exemples

Si $K = 2$, le plus petit ensemble absorbant est illustré à la Figure 3.8. Il est de paramètre $(9, 18)$. Les traits rouges identifient le cycle de longueur 6 de départ $D_1(k) = \{y(k), y(k+i_a), y(k+i_a-r_a)\}$ et ses 2 versions décalées $D_1(k+r_b)$ et $D_1(k+r_b-i_b)$. Les traits noirs

montrent les arêtes du graphe reliant les variables appartenant à des ensembles différents. Chaque variable est connectée à 4 nœuds de degré 2 et 2 de degré 1. Nous avons validé par simulation qu'il n'existe pas d'ensemble de plus petite taille. On observe aussi que l'ensemble est une composition de cycles de longueur 6 et 8, ce qui est cohérent avec les prédiction de [68].

Pour $K = 3$, on procède selon la méthode précédente. L'ensemble absorbant le plus petit est construit à partir d'un ensemble $(9, 18)$ identifié pour $K = 2$. Chaque variable appartenant à cet ensemble initial peut être connecté à 6 autres variables par l'intermédiaire de 3 nœuds de contrôle de la matrice E_c. Afin d'obtenir l'ensemble absorbant le plus petit, il est suffisant de connecter chaque variable à un seul nœud de degré pair supplémentaire. Supposons que $(y(k+\alpha_0), y(k+\alpha_1), \cdots, y(k+\alpha_8))$ forme l'ensemble absorbant initial (de paramètre $(9, 18)$), alors, l'ensemble des variables $(y(k+\alpha_0), y(k+\alpha_1), \cdots, y(k+\alpha_8), y(k+\alpha_0+i_c), y(k+\alpha_1+i_c), \cdots, y(k+\alpha_8+i_c)$ constitue un ensemble absorbant de paramètres $(18, 72)$. C'est l'ensemble de plus petite taille pour $K = 3$. Il existe 6 autres ensembles obtenus en échangeant i_c par $-i_c$, r_c, $-r_c$, $r_c - i_c$ ou $i_c - r_c$. Il n'est pas possible de représenter graphiquement la structure d'un tel ensemble.

Cet ensemble $(18, 72)$ vérifie la condition (1) de la définition d'un ensemble absorbant. Nous allons voir que la vérification de la condition (2) est directement liée à la présence de cycles transverses de longueur 6. Ceci est mis en évidence par l'intermédiaire d'un exemple. La Figure 3.10 montre un exemple d'un cycle transverse. Il a été obtenu pour la m-séquence ayant le polynôme caractéristique $g(x) = 1+x^2+x^3+x^8+x^{10}$. Les polynômes de référence donnant lieu à ces cycles transverses sont $g_a(x) = 1+x^{34}+x^{65}$, $g_b(x) = 1+x^{37}+x^{76}$ et $g_c(x) = 1+x^{72}+x^{77}$.

La Figure 3.9 montre les connexions de la variable $y(k + r_b - i_b + r_c)$ - au centre - dans un ensemble D_3 de paramètre $(18, 72)$ obtenu par le procédé décrit précédemment. Cet ensemble vérifie la condition (1) de la définition d'un ensemble absorbant. Quand il n'y a pas de cycle transverse, chaque variable est connectée à 5 nœuds appartenant $E(D_3)$ et 4 à $O(D_3)$. D_3 est donc un ensemble absorbant, le condition (2) étant également satisfaite. Si un cycle transverse apparaît, comme par exemple celui de la Figure 3.10, alors de nouvelles connexions naissent au sein du graphe. Dans l'exemple de la Figure 3.9, cela induit une nouvelle connexion entre le nœud $F_c(k + r_b - i_b)$ et la variable $y(k + i_a + r_c)$ car $i_a + r_c = i_c + r_b - i_b$. Or cette dernière appartient aussi à l'ensemble D_3. Par conséquent, le nœud $F_c(k + r_b - i_b)$ est connecté 3 fois dans D_3, il appartient donc à $O(D_3)$. Au final, les variables $y(k+r_b-i_b+r_c)$, $y(k+r_b-i_b)$ et $y(k + i_a + r_c)$ ont plus de voisins dans $O(D_3)$ que dans $E(D_3)$, et par conséquent, l'ensemble D_3 n'est plus absorbant.

3.4.1.4 Influence des ensembles absorbants sur la probabilité de fausse alarme

Si la taille et le nombre des ensembles absorbants diminuent, le décodeur améliore sa capacité à corriger des erreurs et converger vers un mot de code. Cela augmente la probabilité de détection correcte P_{CD}, mais aussi celle de mauvaise détection erronée P_{WD}. Nous avons aussi observé que la présence d'ensembles absorbants complets permet de diminuer la probabilité de fausse

alarme P_{FA}. En effet, si le décodeur n'est alimenté que par du bruit, la probabilité qu'un ensemble absorbant de petite taille se bloque est élevée. Pour un ensemble absorbant D_K, ceci est obtenu si les équations de parité de $E(D_K)$ sont satisfaites et celles de $O(D_K)$ ne le sont pas. Cette hypothèse a été validée par une étude des équations de parité au cours du décodage. Nous avons employé la même technique que Richardson pour l'étude des ensembles piégeant [33]. Un vecteur de N chips binaires est généré aléatoirement, puis décodé par avec l'algorithme de *Bit Flipping* [11]. Le décodeur utilise une seule matrice de parité $\mathbf{E}_a$ construite avec un polynôme de référence $g_a(x) = 1 + x^{i_a} + x^{r_a}$ de poids $t = 3$, conformément à l'Eq. (3.18). Les valeurs des paramètres i_a et r_a dépendent de la m-séquence que l'on souhaite décoder, ils sont indiqués dans les tableaux 3.5 et 3.6 pour les m-séquences ayant les polynômes caractéristiques 2415 et 4445 en notation octale.

Nous avons observé un blocage systématique du décodeur au bout de 50 itérations. A la dernière itération, nous avons identifié les ensembles absorbants de longueur 6 correspondant au polynôme $g_a(x)$ (cf. figure 3.6), et plus particulièrement ceux qui sont bloqués. Nous avons mesuré la probabilité qu'un ensemble absorbant bloqué à la $50^{\text{ième}}$ et dernière itération le soit aussi depuis la $Q^{\text{ième}}$ itérations. Elle est notée $P_{\text{blocage}}(Q)$. Les tableaux 3.5 et 3.6 montrent les résultats obtenus pour les m-séquences 2415 et 4445, et des polynômes de référence $g_a(x)$ différents. On observe que tous les ensembles absorbants $(3, 3)$ bloqués à la troisième itération le reste jusqu'au bout.

Le blocage des ensembles absorbants empêche le décodeur itératif de converger vers un mot de code. Il ne réalise donc pas de fausse alarme. Inversement, si le décodeur ne possède pas d'ensemble absorbant de petite taille, cela améliore sa capacité à converger vers un mot de code valide, ce qui engendre une fausse alarme.

TABLE 3.5 – Probabilité de blocage pour les ensembles absorbants de la m-séquence 2415.

$i_a = 25, r_a = 49$		$i_a = 34, r_a = 65$		$i_a = 37, r_a = 76$	
Q	$P_{\text{blocage}}(Q)$	Q	$P_{\text{blocage}}(Q)$	Q	$P_{\text{blocage}}(Q)$
0	0.012	0	0.01	0	0.015
1	0.686	1	0.66	1	0.685
2	1	2	1	2	1
3	1	3	1	3	1

TABLE 3.6 – Probabilité de blocage pour les ensembles absorbants de la m-séquence 4445.

$i_a = 4, r_a = 49$		$i_a = 22, r_a = 73$		$i_a = 56, r_a = 93$	
Q	$P_{\text{blocage}}(Q)$	Q	$P_{\text{blocage}}(Q)$	Q	$P_{\text{blocage}}(Q)$
0	0.0	0	0.0	0	0.002
1	0.452	1	0.484	1	0.492
2	0.999	2	1	2	0.999
3	1	3	1	3	1

On retrouve finalement la problématique du compromis entre probabilité de détection et de fausse alarme, mais présenté sous une forme nouvelle (cf. section 2.2.1.3). C'est la taille et le

nombre d'ensembles absorbants qui va déterminer le point de fonctionnement entre les deux probabilités.

Nous avons montré par un exemple concret comment les cycles de longueur 6 transverses peuvent détruire les ensembles absorbants. Il est très probable que le même effet puisse être obtenu avec des cycles de longueur 8 transverses, sur les ensembles absorbants plus grands. Nous en avons tiré une hypothèse générale pour proposer un algorithme de sélection des polynômes de parité. Si on souhaite limiter au maximum le nombre de fausses alarmes, il est nécessaire de choisir les polynômes qui ne détruisent pas les ensembles absorbants. Un moyen d'y parvenir est de sélectionner les polynômes qui génèrent le moins de cycles transverses de longueur 6 et 8. Il faut donc tout d'abord s'assurer que le graphe ne contient pas de cycle de longueur 4, puis évaluer le nombre de cycles de longueur 6 et 8 : I_6 et I_8.

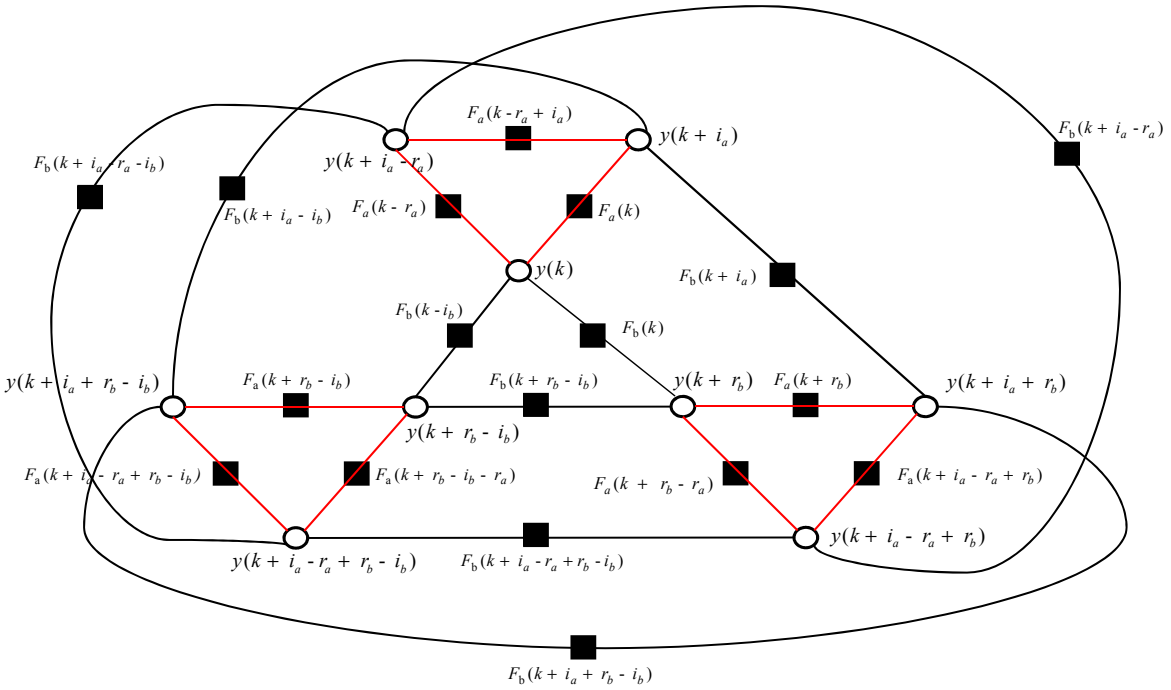

FIGURE 3.8 – Ensemble absorbant complet pour $K = 2$ polyômes de parité

3.4.2 Détection des cycles de longueur 4

Un cycle de longueur 4 existe si 2 variables appartiennent simultanément à 2 équations de parité. Dans ce cas, il est possible de dessiner un carré dans la matrice de parité. Supposons qu'un tel cycle existe entre deux matrices E_a et E_b, avec $r_b \geq r_a$. En prenant en compte la structure Toeplitz de chaque matrice, il existe un cycle de longueur 4 si et seulement si une des conditions suivantes est satisfaite :

$$r_b - i_b = i_a; \quad r_b - i_b = r_a; \quad r_b - i_b = r_a - i_a; \quad r_a - i_a = i_b \tag{3.27}$$

Cette propriété est vérifiable en testant les différentes configurations possibles.

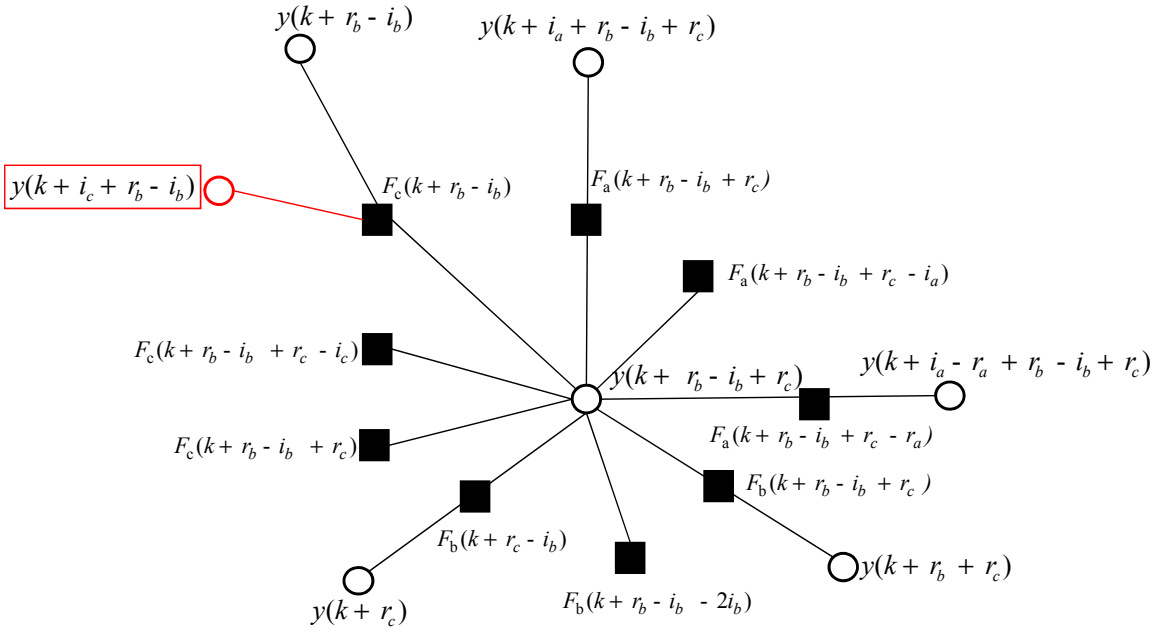

FIGURE 3.9 – destruction d'un ensemble absorbant par un cycle de longueur 6

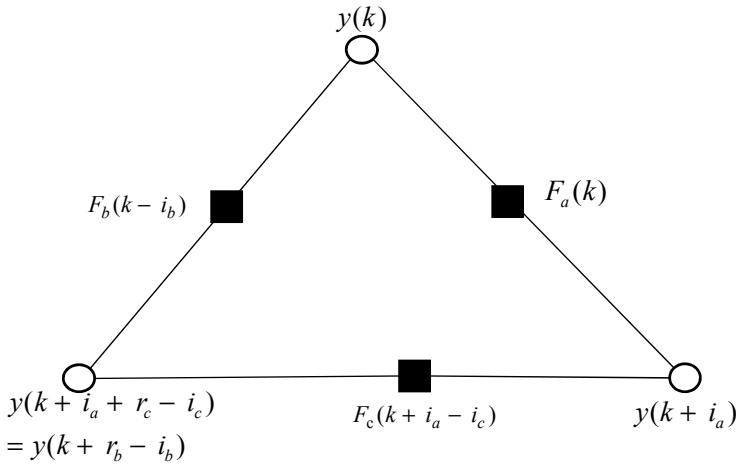

FIGURE 3.10 – Cycle de longueur 6 transverse

Note : si $a=b$, la seule solution est $r_a=2i_a$.

3.4.3 Détermination du nombre de cycles de longueur 6 et 8

Nous supposons tout d'abord que les polynômes de parité sont sélectionnés afin qu'il n'y ait pas de cycle de longueur 4 dans la matrice de parité. Il suffit pour cela de vérifier pour chaque couple de polynôme qu'aucune des conditions de l'Eq. 3.27 n'est satisfaite.

Les nombres de cycles de longueur 6 (I_6) et 8 (I_8) sont calculés analytiquement grâce à la

méthode d'énumération proposée par Halford et Chugg [66]. Ils dépendent du nombre de polynômes K, de leur poids t, de la longueur de la séquence N et de la sélection de polynômes $g_{a_0}(x), \ldots, g_{a_{K-1}}(x)$.

Définissons tout d'abord quelques opérations matricielles :

$$\begin{array}{ll} Y = X - \lambda = [y(i,j)] & \text{où } y(i,j) = x(i,j) - \lambda \\ Y = max(X, 0) = [y(i,j)] & \text{où } y(i,j) = max(x(i,j), 0) \\ U = X \circ Y = [u(i,j)] & \text{où } u(i,j) = x(i,j)y(i,j) \\ Z(X) = X - X \circ I & \end{array} \tag{3.28}$$

I est la matrice identité ayant les mêmes dimensions que celles de la matrice X. $Z(X)$ est l'opérateur qui annule les éléments diagonaux d'une matrice X.

On définit aussi les matrices suivantes : $A = EE^T$, $B = E^TE$, $\tilde{A} = A \circ I$, $\tilde{B} = B \circ I$, $\tilde{B}_m = max(\tilde{B} - 1, 0)$ et $\tilde{A}_m = max(\tilde{A} - 1, 0)$, où E la matrice de parité globale définie par l'Eq. 2.17.

En appliquant la méthode détaillée dans [66], on obtient :

$$\begin{array}{l} I_6 = \dfrac{1}{6} tr(L_6) \\ L_6 = Z(A)A^2 - E\tilde{B}_m BE^T - Z(A\tilde{A}_m)A - EZ(B\tilde{B}_m)E^T + \tilde{A}_m E\tilde{B}_m E^T \end{array} \tag{3.29}$$

où $tr(A)$ est la trace de la matrice A.

Afin de calculer $tr(L_6)$, nous supposerons que $E = [E_{a_0}^T \ldots E_{a_{K-1}}^T]^T$ est composée de K matrices élémentaires E_a circulantes, chacune ayant t variables par ligne (cf. Eq. 3.18). Le polynôme de référence associé à la matrice E_a est $g_a(x) = \sum_{k=0}^{r_a} g_{a,k} x^k$. La propriété de circularité des matrices E_a repose sur l'hypothèse que le décodeur observe la séquence sur toute sa longueur $M = N$. Comme nous le verrons, cela simplifie grandement les calculs de I_6 et I_8. Nous montrerons aussi par simulation que l'algorithme de sélection des polynômes est aussi efficace lorsque cette hypothèse n'est pas satisfaite ($M < N$).

L'évaluation de $tr(L_6)$ est détaillée dans l'Annexe C.1. Nous obtenons finalement :

$$I_6 = \frac{1}{6}\left(tr\left(A^3\right) - KNt\left(K^2t^2 + 3Kt(t-1) + t^2 - 3t + 2\right)\right) \tag{3.30}$$

En employant la même méthode, nous avons aussi évalué le nombre de cycles de longueur 8 (I_8). Ceci est détaillé dans l'Annexe C.2. Au final, nous obtenons :

$$\begin{array}{rl} I_8 = & \dfrac{1}{8}(tr(A^4) - 4(Kt + t - 2)tr(A^3) + 3K^3t^3 + 2K^2t^2(5t - 7) \\ & +2Kt(5t^2 - 16t + 11) + 3t^3 - 14t^2 + 22t - 11) \end{array} \tag{3.31}$$

Les équations 3.30 et 3.31 ont été validées grâce à la libraire en C++ mise en ligne par Halford [66]. Celle-ci implémente les Eq. (3.29) et (C.8). Cette librairie n'est malheureusement plus disponible en téléchargement.

Ces équations dépendent du calcul de $tr(A^3)$ et $tr(A^4)$. La matrice A étant de grande taille (typiquement plus grande que 1000×1000), il est nécessaire d'optimiser ces calculs. En effet, l'algorithme de sélection des polynômes de parité fait un usage intensif de ces formules, il est nécessaire de les simplifier.

Soit la matrice F_{ab} définie par $F_{ab} = E_a E_b^T$. La matrice A est structurée de la manière suivante :

$$\mathbf{A} = \begin{bmatrix} F_{00} & F_{01} & F_{02} & \cdots & \cdots & \cdots & F_{0K-1} \\ F_{10} & F_{11} & F_{12} & \cdots & \cdots & \cdots & F_{1K-1} \\ \vdots & & \ddots & & & & \vdots \\ \vdots & & & \ddots & & & \vdots \\ \vdots & & & & \ddots & & \vdots \\ \vdots & & & & & \ddots & \vdots \\ F_{K-10} & F_{K-11} & F_{K-12} & \cdots & \cdots & \cdots & F_{K-1N-1} \end{bmatrix} \tag{3.32}$$

On en déduit que :

$$tr\left(A^3\right) = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \sum_{k=0}^{K-1} tr\left(F_{ji} F_{kj} F_{ik}\right) \tag{3.33}$$

Soit $I_{ijk} = tr\left(F_{ij} F_{kj} F_{ik}\right)$. En supposant que $i \geq j \geq k$, on a :

$$I_{ijk} = \sum_{s=0}^{N-1} \sum_{h=0}^{N-1} \sum_{l=0}^{N-1} F_{ki}(l,s) F_{kj}(l,h) F_{ji}(h,s) \tag{3.34}$$

Comme cela est détaillé dans l'Annexe C.3, la propriété de circularité des matrices F_{ab} peut être exploitée pour calculer les matrices F_{ij}, F_{kj} et F_{ik} :

$$I_{ijk} = \sum_{s=0}^{N-1} \sum_{h=0}^{N-1} \sum_{l=0}^{N-1} F_{ki}(0, \phi(s-l)) F_{kj}(0, \phi(h-l)) F_{ji}(0, \phi(s-h)) \tag{3.35}$$

où $\phi(u)$ est défini par :

$$\phi(u) = \begin{cases} u & \text{si } 0 \leq u \leq N-1 \\ N+u & \text{si } -N < u < 0 \end{cases} \tag{3.36}$$

On définit maintenant Ω_{ji}, l'ensemble des indices u vérifiant $F_{ji}(0,u) \neq 0$. Il est explicité par l'Eq. (C.12) l'Annexe C.3. Le coefficient $F_{ji}(0, \phi(s-h))$ est strictement positif si et seulement si $\phi(s-h) \in \Omega_{ji}$. On en déduit que le terme $F_{ki}(0, \phi(s-l)) F_{kj}(0, \phi(h-l)) F_{ji}(0, \phi(s-h))$ est strictement positif si et seulement si :

$$\alpha = \phi(s-h) \in \Omega_{ji} \text{ and } \beta = \phi(h-l) \in \Omega_{kj} \text{ and } \gamma = \phi(s-l) \in \Omega_{ki} \tag{3.37}$$

De plus, on peut montrer que : $\gamma = (\alpha + \beta)\ mod\ N$.

Finalement, l'Eq. 3.35 devient :

$$I_{ijk} = N \sum_{\alpha \in \Omega_{ji}} \sum_{\beta \in \Omega_{kj}} F_{ki}(0, (\alpha + \beta)\ mod\ N) F_{kj}(0, \beta) F_{ji}(0, \alpha) \tag{3.38}$$

Cette équation est employée pour calculer $tr\left(A^3\right)$ dans l'Eq. 3.33 et ensuite I_6 avec l'Eq. 3.30.

En utilisant la même méthode, le calcul de $tr(A^4)$ peut aussi être simplifié.

3.4.4 Algorithmes de sélection des polynômes de parité

Nous avons vu à la section 3.4.1 que les cycles transverses ont pour effet de détruire les ensembles absorbants de petite taille, ce qui favorise l'apparition de fausses alarmes. Comme nous le verrons dans la section 3.5, les fausses alarmes ont un effet extrêmement nocif sur le temps d'acquisition d'une m-séquence. Il est donc fondamental de les limiter au maximum, quitte à accepter une légère dégradation de la probabilité de détection. L'idée directrice de l'algorithme est donc de sélectionner les polynômes de parité $(g_0(x), \cdots, g_{K-1}(x))$ qui minimisent le nombre de cycles de longueur 6 et 8. Par ce biais, les cycles transverses seront aussi limités au maximum.

Il existe une borne inférieure sur le nombre de cycles. Par exemple, chaque matrice E_a contient δ_t cycles de longueur 6 par variable :

$$\delta_t = 3 \binom{t}{3} \tag{3.39}$$

Il y a en effet 3 cycles de longueur 6 par variable lorsque $t = 3$ (cf. section 3.4.1). Pour $t > 3$, on a $\binom{t}{3}$ triplets de coefficients du polynôme $g_a(x)$, chacun donnant naissance à 3 cycles de longueur 6. Chaque cycle contient 3 variables, il y a donc $N\delta_t/3$ cycles de longueur 6 par matrice E_a. I_6 est donc borné inférieurement :

$$I_6 \geq NK\delta_t/3 \tag{3.40}$$

Si une combinaison de polynômes de parité atteint cette borne, elle minimise le nombre de cycle de longueur 6. Dans ce cas, les seuls cycles de longueur 6 sont ceux qui apparaissent à l'intérieur d'une seule matrice E_a. Il n'y a pas de cycle transverse de longueur 6.

Les résultats de simulation ont montré que ces configurations sont nombreuses lorsque $t = 3$. Lorsque $t > 4$, l'atteinte de la borne inférieure dépend du nombre de polynômes K. Si $K > 4$, ces configurations n'existent pas toujours. Dans ce cas, l'algorithme sélectionne les configurations qui minimisent l'Eq. (3.30).

Lorsque $t = 3$, il y a 36 cycles de longueur 8 passant par $y(k)$, soit 9 par variable. Pour un ensemble de K polynômes, on a $\binom{K}{2}$ paires, chacune donnant naissance à $9N$ cycles de

longueur 8. Le nombre minimal de cycles de longueur 8 vaut donc :

$$I_{8,min} = 9N \binom{K}{2} \binom{t}{3} = \frac{3NK(K-1)t(t-1)(t-2)}{4} \tag{3.41}$$

Les polynômes de parité de poids t sont choisis parmi un ensemble Υ_t, déterminé au préalable par une recherche exhaustive.

L'algorithme de minimisation de la probabilité de fausse alarme P_{FA} (Algorithme 1) cherche la combinaison de polynômes qui minimise le nombre de cycles de longueur 6 et 8. Lorsque les bornes inférieures (Eq. (3.40) et Eq. (3.41)) sont atteintes, il n'y a pas de cycles transverses de longueur 6 et 8. Cela garantit l'existence d'ensembles absorbants complets qui vont bloquer la convergence de l'algorithme de décodage par passage de messages lorsqu'il est alimenté par du bruit. L'ensemble des configurations de polynômes $(g_0(x), \cdots, g_{K-1}(x))$ minimisant I_6 et I_8 est appelé Λ_K. Le nombre de combinaisons testées est noté N_s. Il correspond aux K-uplets de polynômes de Λ_K, ou bien à un nombre fixé arbitrairement pour prendre en considération le cas où I_6 n'atteint pas la borne inférieure.

Il est aussi possible de créer un algorithme de maximisation de la probabilité de détection P_{CD}. Ce dernier chercherait au contraire à éliminer les ensembles absorbants et donc à maximiser le nombre de cycles transverses. L'inconvénient de cet algorithme est qu'il maximise aussi la probabilité de mauvaise détection P_{WD}. Lorsque le RSB est très bas, le décodeur va trouver des mots de code erronés alors qu'il serait préférable qu'il ne converge pas. On se retrouve dans la même problématique que pour la fausse alarme. Ceci sera illustré dans l'exemple applicatif de la section 3.5.

Algorithm 1 Minimisation de P_{FA} pour $t = 3$

```
q = 0 et min = +∞
while q < N_s do
    Sélectionner K polynômes de parité distincts g_0(x), ··· , g_{K-1}(x) ∈ Υ_t
    calculer I_6 selon l'Eq. 3.30
    if I_6 == NK then
        Calculer I_8 selon l'Eq. 3.31
        if (I_8 < min) ou (I_8 = 9NK(K − 1)/2) then
            Sauvegarder la configuration (g_0(x), ··· , g_{K-1}(x)) ∈ Λ_K et la valeur de I_8.
            min ← I_8
        end if
    end if
end while
```

3.5 Application

Nous allons dans cette section étudier l'influence du poids des équations de parité sur le temps d'acquisition d'une séquence de Gold. Cela correspond au temps mis par le récepteur

pour se synchroniser avec la séquence émise $\mathbf{z}$. C'est le paramètre qui est finalement le plus important dans un système de synchronisation. Dans le contexte de cette thèse, 'synchronisation' signifie que le récepteur a trouvé l'état initial de la séquence de Gold. Les polynômes générateurs de la paire de m-séquences préférentielles $g_s(x)$ et $g_y(x)$ sont supposés connus du récepteur. Le temps d'acquisition dépend des probabilités de détection correcte, de fausse alarme et de détection erronée, et ces dernières dépendent du poids des équations de parité.

Nous allons supposer que le récepteur observe à l'instant k la séquence émise, selon le modèle suivant :

$$R(k) = (-1)^{z(k)} + n(k) \tag{3.42}$$

Le début de la séquence est fixé par convention à l'instant $k = 0$.

Cela correspond par exemple au modèle d'un récepteur GPS après multiplication différentielle au niveau chip [69], ou bien encore au modèle d'observation développé à la section 4.2 pour la détection du code d'embrouillage de la liaison montante du système WCDMA.

Nous allons appliquer deux stratégies pour la recherche de synchronisation :

- recherche directe : il s'agit de détecter directement l'état initial de la séquence de Gold $\mathbf{z}$.
- recherche en série : dans de nombreux systèmes à étalement de spectre (e.g. WCDMA, GPS, Galileo), l'état initial d'une des deux m-séquences $\mathbf{s}$ et $\mathbf{y}$ utilisées pour générer la séquence de Gold est connu à un instant donné (e.g $k = 0$). Supposons que la séquence connue est $\mathbf{y}$. La recherche en série exploite cette information additionnelle pour se ramener à une détection séquentielle de la m-séquence $\mathbf{s}$. L'intérêt de cette approche est de pouvoir utiliser des équations de parité ayant un poids $t = 3$, alors que $t \geq 4$ pour les séquences de Gold. Ce poids faible assure une bonne probabilité de détection à un RSB très faible. Son inconvénient est sa plus grande sensibilité aux fausses alarmes. L'algorithme de sélection des équations de parité que nous avons proposé dans la section précédente va alors permettre de minimiser les fausses alarmes et améliorer le temps d'acquisition.

Ces deux méthodes de synchronisation sont analogues aux recherches en série et en parallèle des systèmes à étalement de spectre fonctionnant par corrélation [70][71].

3.5.1 Recherche directe

3.5.1.1 Description

Le modèle de l'Eq. 3.42 représente une observation bruitée de la séquence de Gold $\mathbf{z}$. Il est alors possible de trouver son état initial à l'aide d'un algorithme de décodage par passage de messages. Ce dernier est implémenté conformément à la description de la section 2.4.3. Il nécessite de trouver des équations de parité satisfaites par la séquence $\mathbf{z}$. Ceci est réalisable car les polynômes générateurs des séquences $\mathbf{s}$ et $\mathbf{y}$ sont connus (cf. section 3.3). Si le degré des polynômes r est pair, alors le poids minimal des équations de parité vaut $t = 4$. Si r est impair, le poids minimal vaut $t = 5$.

Supposons que le récepteur démarre à l'instant q. Il va recueillir M chips $(R(q), \cdots, R(q+M-1))$ qui serviront de signal d'entrée au décodeur. Si celui-ci fait une erreur de détection, le récepteur va perdre du temps pour se rendre compte de son erreur. Le temps d'acquisition est alors pénalisé. Si le décodeur ne détecte pas de séquence, l'opération est réitérée au chip suivant.

L'opération de décodage est modélisée par une fonction $dec_z(.)$ qui produit un indicateur de décodage et l'état initial de la séquence $\mathbf{z}$ qui a été trouvé par le décodeur :

$$\{I_c, \hat{U}_z\} = dec_z(R(q), R(q+1), \cdots, R(q+M-1)) \tag{3.43}$$

I_c est la fonction d'indication du décodeur (cf. Eq. 2.20). Son résultat vaut 1 si le décodeur a trouvé un mot de code valide, et un 0 sinon. $\hat{U}_z$ représente l'état initial de la séquence $\mathbf{z}$. C'est un vecteur de $2r$ éléments binaires.

La méthode de recherche directe est décrite par l'algorithme 2.

Algorithm 2 Recherche directe

$q = q_0$
while $I_c = 0$ and $q - q_0 < L_{max}$ **do**
  $\{I_c, \hat{U}_z\} = dec_z(R(q), R(q+1), \cdots, R(q+M-1))$
  $q \leftarrow q + 1$
end while
Return État initial décodé : $\hat{U}_z$

q_0 représente l'instant de démarrage de l'algorithme. C'est une variable aléatoire, distribuée uniformément sur la durée de la séquence. L_{max} est le nombre maximal de tentatives au delà duquel le processus de détection est arrêté.

3.5.1.2 Temps d'acquisition

A chaque instant q, le récepteur lance une acquisition et prend une décision :
- le récepteur a décodé avec succès la séquence $\mathbf{z}$.
- le récepteur s'est trompé et va réessayer à l'instant suivant (i.e. $q+1$).

Ce modèle ne considère pas de fausse alarme car il suppose que la séquence $\mathbf{z}$ est toujours présente.

La valeur moyenne du temps d'acquisition est la suivante [72] :

$$E[T_{acq}] = \left(M + \frac{P_{ND} + \kappa_p P_{WD}}{P_{CD}} \right) T_c \tag{3.44}$$

T_c est la durée d'un chip, et $\kappa_p T_c$ est le temps de pénalité lorsque le récepteur réalise une détection erronée. Cela correspond au temps que va mettre le récepteur pour détecter qu'il a fait une erreur. C'est généralement le mécanisme d'asservissement de synchronisation

(temps/fréquence) qui permet de détecter une erreur éventuelle. P_{CD} est la probabilité de détection correcte, P_{WD} celle de détection erronée et P_{ND} la probabilité de non-détection.

On observe que les erreurs de détection vont pénaliser fortement le temps d'acquisition.

3.5.2 Recherche en série

3.5.2.1 Description

A chaque instant q, le récepteur suppose qu'il est synchronisé avec le début de la séquence $\mathbf{y}$ (i.e l'hypothèse H_1 de la section 2.2.1 est satisfaite). Il recueil M chips du signal $(R(q), \cdots, R(q+M-1))$ et les multiplie chip à chip par la séquence $\mathbf{y}$:

$$V_q(k) = R(q+k)(-1)^{y(k)} \qquad k = 0, \ldots, M-1 \tag{3.45}$$

Si l'hypothèse H_1 est valide (i.e. $q = 0$), cette opération élimine $y(k)$ de l'Eq. (3.42) :

$$V_q(k) = (-1)^{y(k)\oplus s(k)\oplus y(k)} + n(k)(-1)^{y(k)} = (-1)^{s(k)} + n'(k) \tag{3.46}$$

Dans ce cas, $\mathbf{V}_q = (V_q(0)\, V_q(1) \cdots V_q(M-1))$ est une observation bruitée de la modulation BPSK de la m-séquence $\mathbf{s}$. Il est alors possible de trouver son état initial à l'aide d'un algorithme de décodage par passage de messages. Nous avons montré à la section 3.2 que le décodeur peut utiliser de nombreuses équations de parité de poids $t = 3$, ce qui permet d'optimiser la probabilité de détection correcte. Il est implémenté conformément à la description de la section 2.4.3.

Si l'hypothèse H_1 n'est pas valide (i.e. $q \neq 0$), la multiplication avec la séquence $y(k)$ embrouille le signal et on ne peut observer la séquence $\mathbf{s}$:

$$V_q(k) = (-1)^{y(k)\oplus s(k+q)\oplus y(k+q)} + n(k)(-1)^{y(k)} \tag{3.47}$$

Dans ce cas, le décodeur ne devrait pas détecter un mot de code valide en essayant de décoder la séquence $\mathbf{s}$. Si c'est le cas, le détecteur produit une fausse alarme.

L'opération de décodage est modélisée par une fonction $dec_s(.)$ qui produit un indicateur de décodage et l'état initial de la séquence qui a été trouvé par le décodeur :

$$\{I_c, \hat{U}_s\} = dec_s(V_q(0), V_q(1), \cdots, V_q(M-1)) \tag{3.48}$$

$\hat{U}_s$ représente l'état initial de la séquence $\mathbf{s}$. C'est un vecteur de r éléments binaires.

La méthode de recherche en série est décrite par l'algorithme 3.

3.5.2.2 Temps d'acquisition

A chaque instant q, le récepteur lance une acquisition et prend une décision :

Algorithm 3 Recherche en série

$q = q_0$
while $I_c = 0$ and $q - q_0 < L_{max}$ **do**
  Calcule $V_q(0), V_q(1), \cdots, V_q(M-1)$ d'après l'Eq. (4.19)
  $\{I_c, \hat{U}_s\} = dec_s(V_q(0), V_q(1), \cdots, V_q(M-1))$
  $q \leftarrow q + 1$
end while
Return État initial décodé : $\hat{U}_x$

- le récepteur est synchronisé avec la séquence **s**.
- le récepteur n'est pas synchronisé avec la séquence **s**.

Il est possible de définir un diagramme d'état du récepteur en fonction de la décision prise. Ce diagramme est ensuite utilisé pour calculer le temps moyen d'acquisition, en appliquant la méthodologie décrite dans la référence [70]. Celle-ci est présentée et détaillée dans l'Annexe D.

La valeur moyenne du temps d'acquisition dépend des probabilités de détection et de fausse alarme :

$$E[T_{acq}] = \left(\frac{1 + \kappa_p P_{WD} + (1 + \kappa_p P_{FA}) \left(\frac{N-1}{2}\right) (2 - P_{CD})}{P_{CD}} \right) T_c \tag{3.49}$$

T_c est la durée d'un chip, et $\kappa_p T_c$ est le temps de pénalité lorsque le récepteur détecte une fausse alarme. P_{FA} est la probabilité de fausse alarme.

3.5.3 Performance

Nous allons dans cette section évaluer les performances des méthodes de recherche directe et en série, puis les comparer. Pour cela nous utilisons une séquence de Gold de degré impair ($r = 11$) :

$$g_s(x) = 1 + x^2 + x^5 + x^8 + x^{11} \tag{3.50}$$

$$g_y(x) = 1 + x^2 + x^{11} \tag{3.51}$$

Elle sera identifiée par sa notation en octal $(4445, 4005)$. Elle n'est pas utilisée par des systèmes civils connus.

Le polynôme générateur de la séquence de Gold est $g_z(x) = g_s(x)g_y(x) = 1 + x^4 + x^5 + x^7 + x^8 + x^{10} + x^{16} + x^{17} + x^{22}$. Le poids du polynôme est trop élevé pour être employé directement pour construire la matrice de parité du décodeur. Le degré r étant impair, on pourra utiliser des équations de parité de poids $t = 5$.

Les équations de parité que nous avons utilisées pour le décodage sont listées dans les tableaux E.1 et E.2 dans l'Annexe E.

Lorsque $t > 3$, les simulations ont montré une absence de fausse alarme pour la recherche séquentielle ($P_{FA} \approx 0$) ou de détection erronée pour la recherche directe ($P_{WD} \approx 0$). Ceci

est dû à la présence de nombreux ensembles absorbants complets qui piègent le décodeur et l'empêchent de converger lorsque la séquence est absente du signal d'entrée (i.e. fausse alarme) ou que la variance du bruit est trop importante (i.e. détection erronée). Les simulations relatives à la probabilité de détection mesurent la probabilité d'erreur de détection : $P_e = 1 - P_{CD} = P_{ND} + P_{WD}$. Cela correspond au Frame Error Rate (FER) qui est mesuré pour quantifier les performances des codes correcteurs d'erreurs.

La figure 3.11 montre la probabilité de détection obtenue en fonction du RSB, pour une recherche directe avec la séquence de Gold $(4005, 4445)$. Les configurations d'équations de parité utilisées sont indiquées dans le tableau 3.7. Les équations de parité sont notées $c_l(x)$ et sont identifiées de la manière suivantes : $c_l(x) = 1 + x^k + x^j + x^i + x^m$ pour $t = 5$. Les valeurs des paramètres k, j, i et m sont listées dans le tableau E.1 de l'Annexe E.

On observe que P_{CD} s'améliore lorsque la combinaison d'équations de parité choisie augmente le nombre de cycles de longueur 6. Le nombre d'ensembles absorbants de petite taille diminue lorsque I_6 augmente, ce qui améliore les performances (cf. section 3.4.1). En effet, la probabilité que le décodeur soit piégé diminue, il est donc moins sujet à un blocage qui l'empêche de converger.

TABLE 3.7 – Configuration des équations de parité pour le calcul de P_{CD} et P_{WD} de la séquence $(4005, 4445)$.

Référence	configuration	I_6
a	$(c_1, c_2, c_3, c_4, c_6)$	405306
b	$(c_1, c_2, c_3, c_5, c_9)$	239499
c	$(c_1, c_2, c_3, c_7, c_8)$	176042

La figure 3.12 montre le même type de comportement pour le décodage de la m-séquence (4445). Les configurations d'équations de parité utilisées sont indiquées dans le tableau 3.8. Elles sont de la forme $c_l(x) = 1 + x^i + x^m$ et les valeurs des paramètres m et i sont listées dans le tableau E.2 de l'Annexe E. Lorsque le nombre de cycles de longueur 6 diminue de 38893 à 10235, la performance se dégrade de 2 dB pour une probabilité d'erreur de détection cible de $P_e = 10^{-2}$. D'un autre côté, la figure 3.13 montre l'évolution de la probabilité de mauvais décodage P_{WD} en fonction du RSB pour plusieurs configurations d'équations de parité. On observe que P_{WD} augmente fortement lorsque le RSB devient très faible et que le nombre de cycles de longueur 6 est élevé. Dans cette situation, le décodeur a tendance à trouver un mot de code, mais se trompe systématiquement. Le récepteur va alors perdre du temps à détecter l'erreur de décodage. On retrouve le même effet néfaste qu'une fausse alarme. Ceci pénalise le temps d'acquisition. Il est donc nécessaire de choisir la combinaison d'équations de parité qui va réaliser le bon compromis entre P_{CD}, P_{WD} et P_{FA}. L'algorithme de sélection des équations de parité que nous avons mis au point minimise P_{WD} et P_{FA} au détriment de P_{CD}.

La figure 3.14 montre la probabilité de fausse alarme obtenue en fonction de l'inverse de la variance du bruit exprimé en dB, $-10log(\sigma^2)$, pour une recherche en série de la m-séquence 4445. Les configurations d'équations de parité utilisées sont indiquées dans le tableau 3.8. On observe que P_{FA} s'améliore très significativement lorsque la combinaison d'équations de parité

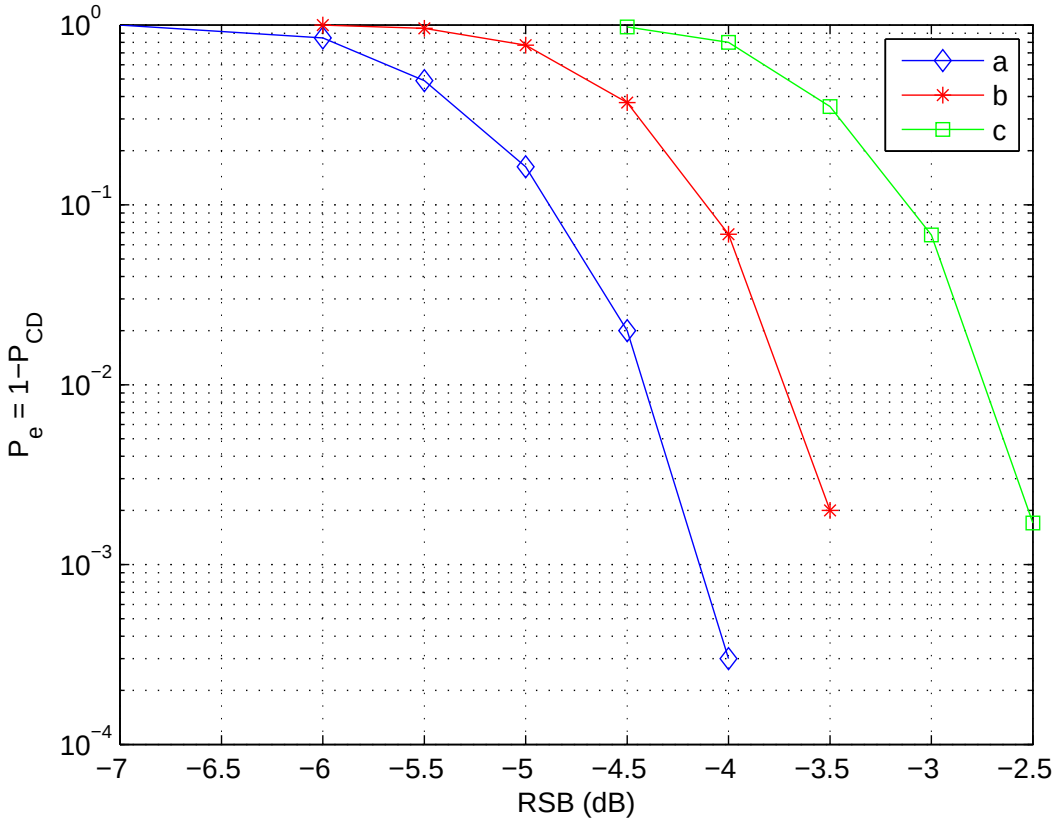

FIGURE 3.11 – Probabilité d'erreur de détection ($P_e = 1 - P_{CD}$)- séquence de Gold $(4005, 4445)$, recherche directe

TABLE 3.8 – Configuration des équations de parité pour le calcul de P_{CD} et P_{FA} de la séquence 4445.

Référence	configuration	I_6	I_8
a	$(c_1, c_2, c_3, c_4, c_6)$	38893	589536
b	$(c_1, c_2, c_3, c_4, c_9)$	24564	442152
c	$(c_1, c_2, c_3, c_4, c_{11})$	20470	350037
d	$(c_1, c_2, c_3, c_5, c_7)$	16376	419635
e	$(c_1, c_2, c_3, c_5, c_8)$	10235	450340
f	$(c_1, c_2, c_3, c_8, c_{10})$	10235	337755
g	$(c_1, c_2, c_3, c_{12}, c_{13})$	10235	196512

minimise le nombre de cycles de longueur 6. Pour un même nombre de cycles de longueur 6, la combinaison qui réduit ceux de longueur 8 apporte aussi un gain appréciable. Dans ce cas, on maximise le nombre d'ensembles absorbants et donc la possibilité que le décodeur reste piégé s'il est alimenté uniquement avec du bruit. La figure 3.15 montre le niveau de fausse alarme lorsque la longueur du vecteur d'observation de la séquence est tronquée de

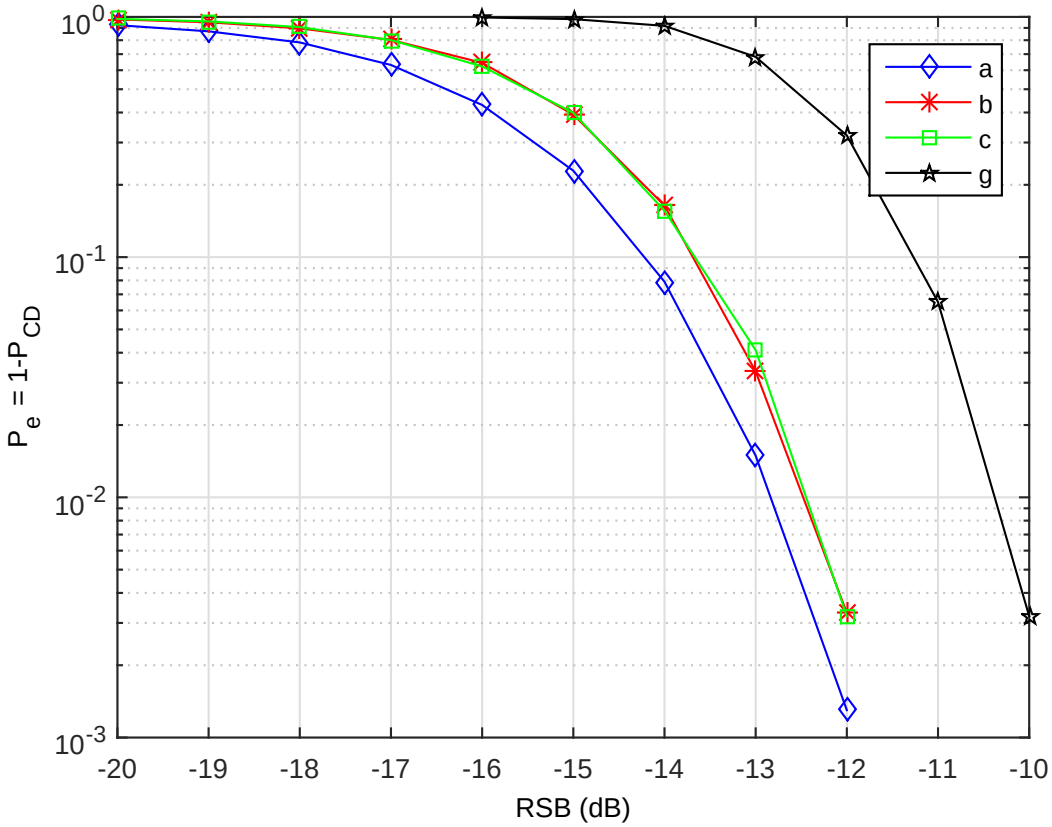

FIGURE 3.12 – Probabilité d'erreur de détection ($P_e = 1 - P_{CD}$)- m-séquence 4445 - recherche en série

$M = 2047$ à $M = 1023$ chips. Il apparaît clairement que cela influe peu sur le niveau de fausse alarme. Si la séquence observée est tronquée, la propriété de circularité de chaque matrice de parité élémentaire $\mathbf{E}_a$ n'est plus valide. Néanmoins l'utilisation de l'algorithme de sélection des équations de parité reste efficace pour limiter le nombre de cycles transverses.

Ces résultats corroborent l'analyse que nous avons faite à la section 3.4.1 sur l'impact des ensembles absorbants de petite taille.

La figure 3.16 montre le temps moyen d'acquisition en fonction du RSB pour la recherche directe de la séquence de Gold $(4005, 4445)$. La configuration ayant le plus de cycles de longueur 6 assure une acquisition de la séquence en $M = N = 2047$ chips pour un RSB supérieur à -6.5 dB. Si le nombre de cycle diminue, l'acquisition en $M = N = 2047$ chips est obtenue pour un RSB de -4.5 dB, soit une dégradation de 2 dB. Cet écart correspond à l'écart observé à la figure 3.11 pour la mesure de P_{CD}.

La figure 3.17 montre le temps moyen d'acquisition en fonction du RSB pour la recherche en série de la séquence 4445. Du fait de la pénalité induite par une fausse alarme ($\kappa_p = 2047$), le temps moyen d'acquisition est très sensible à P_{FA}. Ceci est particulièrement visible pour la

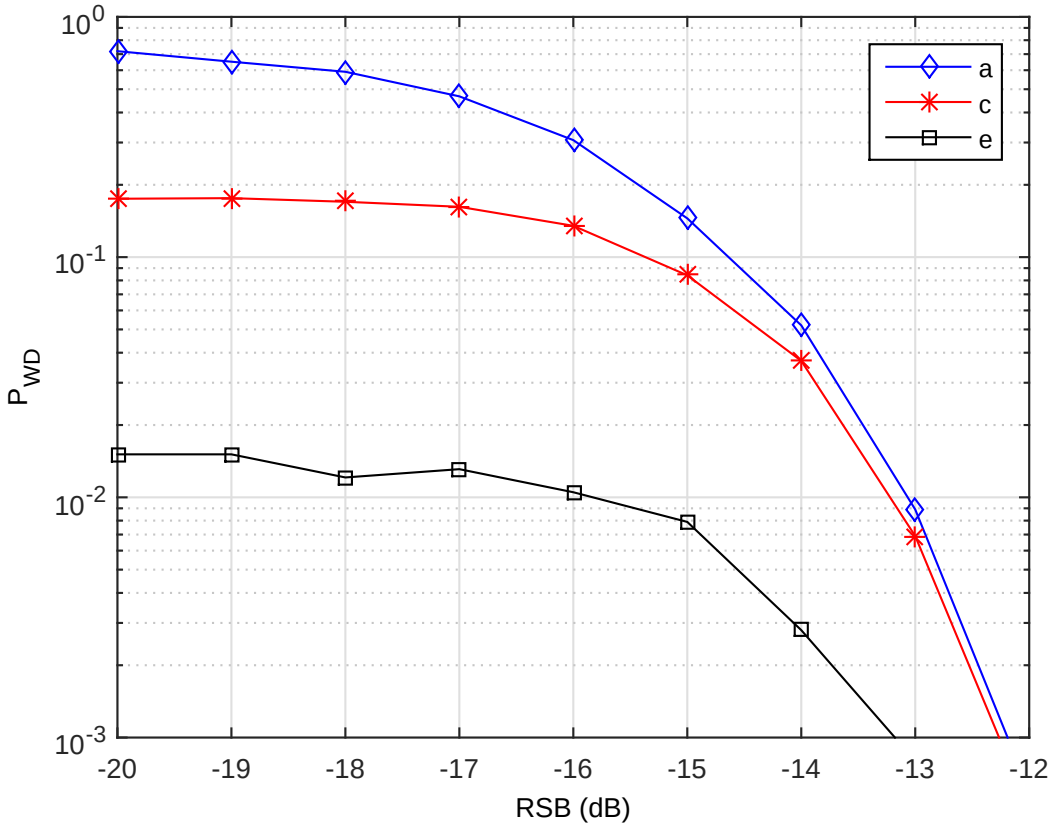

FIGURE 3.13 – Probabilité de mauvaise détection (P_{WD})- m-séquence 4445 - recherche en série

configuration 'e' qui induit une probabilité de fausse alarme de $1.6\,10^{-2}$ et un temps moyen d'acquisition de l'ordre de 18 fois la longueur de la séquence. Il est donc fondamental de minimiser cette variable, quitte à dégrader légèrement la probabilité de détection correcte. Lorsque la probabilité de détection vaut 1 et qu'il n'y a aucune fausse alarme, le temps d'acquisition moyen vaut $T_{acq} = M + N/2$. Dans les simulations, nous avons $M = N = 2047$ ce qui explique pourquoi T_{acq}/N tend vers 1.5 avec la configuration 'g'. Le temps d'acquisition converge vers une valeur acceptable à un RSB beaucoup plus bas que pour la recherche directe. En effet, le temps d'acquisition obtenu avec la configuration 'g' pour la recherche en série se stabilise à 1.5 longueur de séquence pour un RSB de -10 dB, alors que pour la recherche directe, la configuration 'a' se stabilise à 1 longueur de séquence pour un RSB supérieur à -6 dB. On observe donc un gain de l'ordre de 4 dB en faveur de la recherche en série. Ceci est néanmoins obtenu au prix d'une charge de calcul accrue. Bien que cet aspect ne soit pas abordé dans cette thèse, réaliser un décodage au rythme chip pour la recherche en série est très probablement complexe à implémenter.

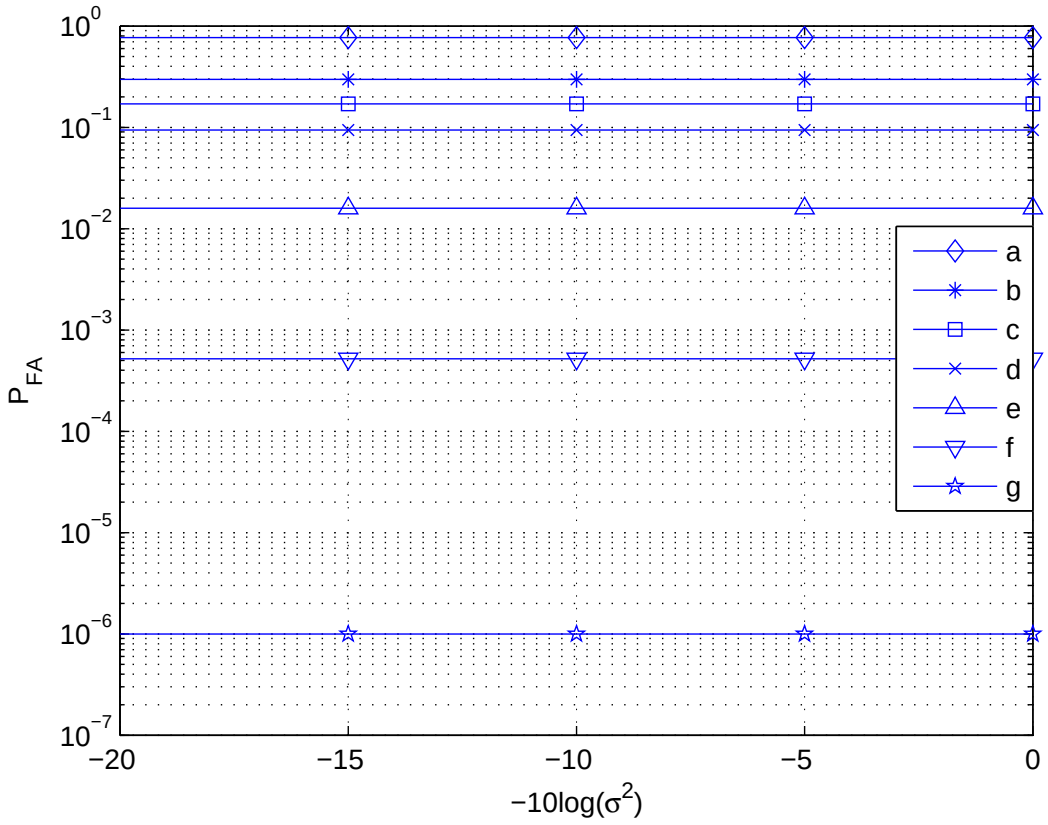

FIGURE 3.14 – Probabilité de fausse alarme - m-séquence 4445

3.6 Conclusion

Les équations de parité sont un constituant indispensable du décodeur. Elles appartiennent au code dual de la séquence que l'on cherche à décoder. Le code dual d'un m-séquence est le code de Hamming construit avec le polynôme caractéristique de la séquence. Le nombre de mots de poids t du code de Hamming est connu, par conséquent le nombre d'équations de parité de poids t d'une m-séquence est lui aussi connu. Pour les séquences de Gold, Kasami a dénombré le nombre d'équations pour $t = 3$ et 4. Nous avons calculé le nombre d'équations de parité de poids $t = 5$ lorsque le degré du polynôme caractéristique r est impair. Ce calcul est important car il n'y a pas d'équations de parité de poids $t < 5$ lorsque r est impair. Le nombre d'équations de parité est aussi utilisé pour estimer le degré minimal des équations d'un poids t donné. Cette information peut être exploitée par l'algorithme de recherche des équations pour minimiser le nombre de calculs. Cela permet aussi de déterminer rapidement la taille du vecteur d'observation nécessaire pour le décodeur. Nous avons montré que le modèle de prédiction estime correctement la valeur moyenne du degré minimal de l'ensemble des séquences de Gold. Nous avons néanmoins mis en évidence une grande variabilité du degré minimal des séquences autour de cette valeur moyenne. Il reste donc un point d'interrogation sur l'applicabilité de cette méthode d'estimation.

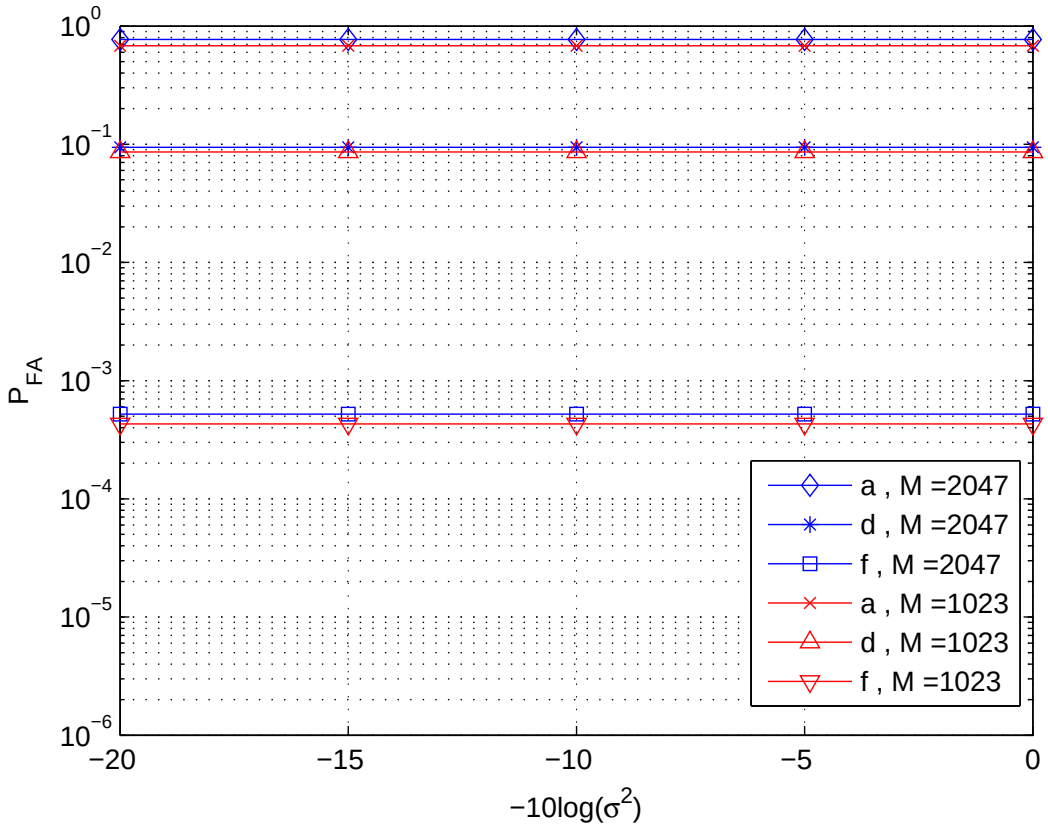

FIGURE 3.15 – Probabilité de fausse alarme pour une séquence tronquée- m-séquence 4445

Nous avons ensuite identifié les ensembles absorbants complets de plus petite taille lorsque le décodeur emploie $1, 2$ ou 3 polynômes de parité. Nous avons montré par un exemple que des cycles 'transverses' peuvent détruire ces ensembles absorbants, ce qui génère des fausses alarmes. Nous en avons fait une hypothèse générale pour proposer un algorithme de sélection des polynômes de parité. Ce dernier minimise le nombre de cycles transverses de longueur 6 et 8, ce qui minimise la probabilité de fausse alarme lorsque le poids des équations de parité vaut $t = 3$. Ce travail a nécessité de calculer le nombre de cycles de longueur 6 et 8 pour les matrices de parité concaténant plusieurs matrices de référence. Ce calcul a été mené en utilisant le modèle proposé par Chugg et Halford pour énumérer le nombre de cycles courts dans un graphe de décodage. Les résultats de simulation corroborent notre hypothèse sur l'impact des cycles transverses et l'algorithme permet de sélectionner les équations de parité qui minimisent la probabilité de fausse alarme. Le temps d'acquisition d'une séquence de Gold est ainsi très notablement réduit. Il reste néanmoins que notre hypothèse n'est pas démontrée formellement.

Ce travail a donné lieu à une publication [P4] et un brevet [P10]. Un article de journal est en cours de soumission. Il rassemble les travaux sur l'impact des ensembles absorbants et des cycles transverses sur la probabilité de fausse alarme. Il présente aussi la méthode de calcul des cycles de longueur 6 et 8, ainsi que l'algorithme de sélection des équations de parité.

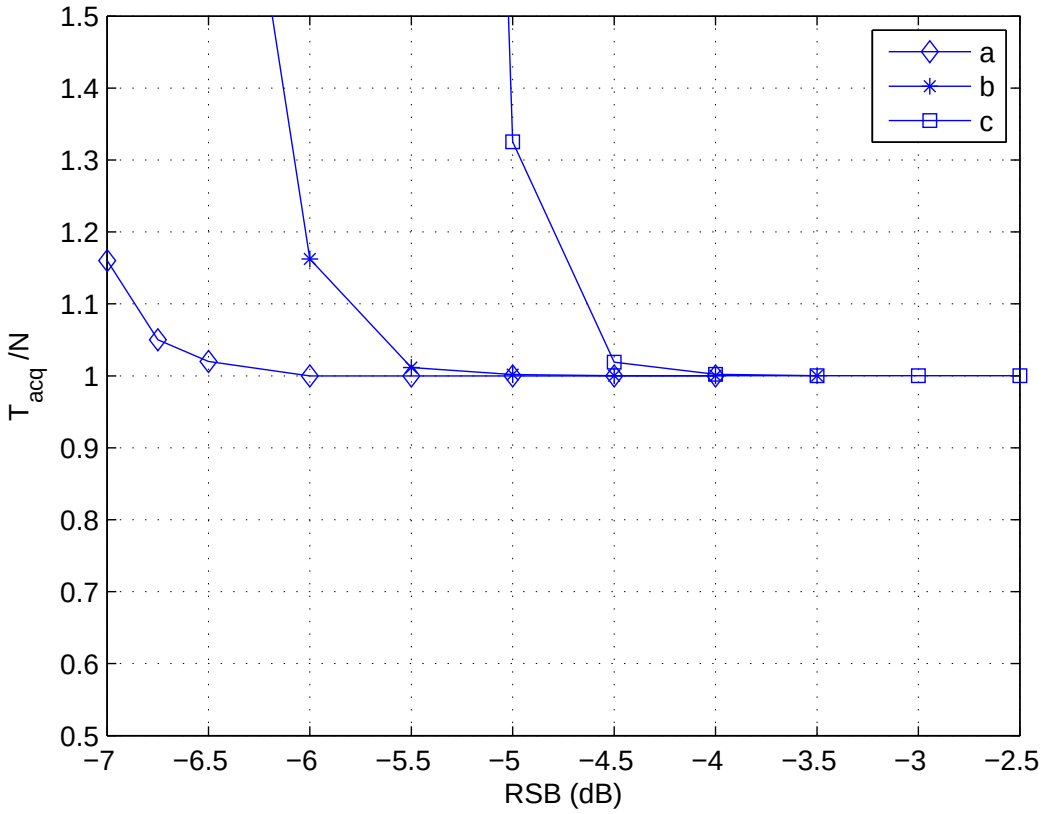

FIGURE 3.16 – Temps moyen d'acquisition - recherche directe - séquence $(4005, 4445)$

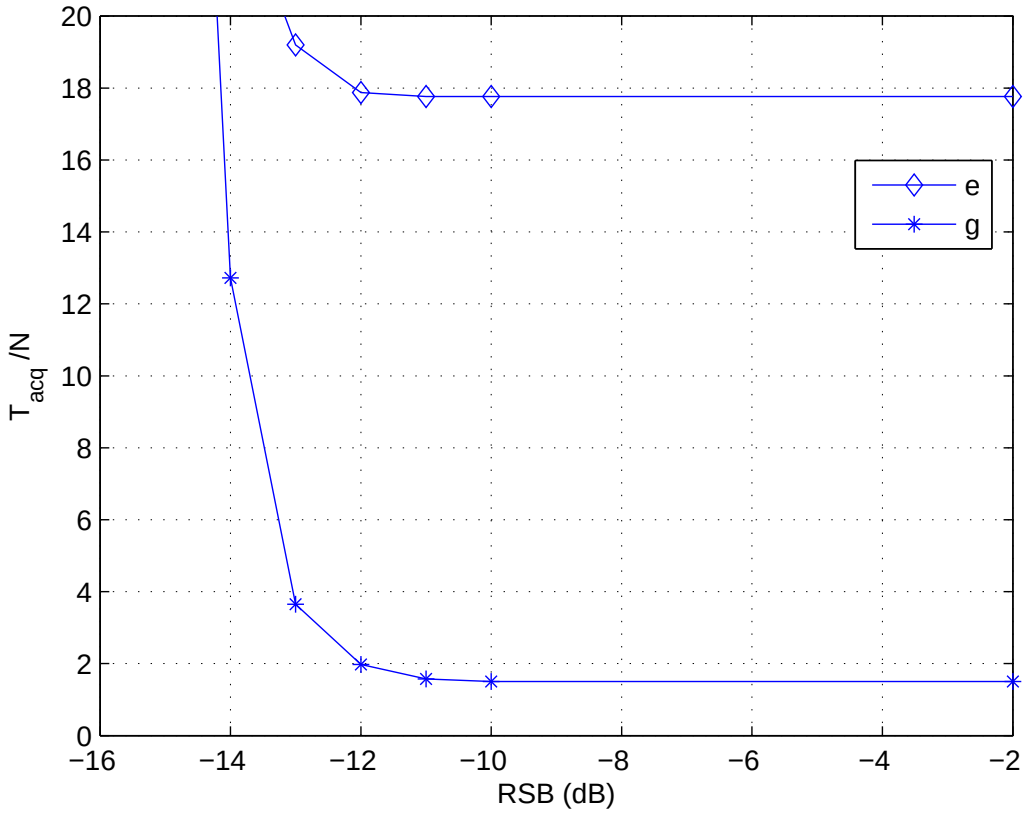

FIGURE 3.17 – Temps moyen d'acquisition - recherche en série - séquence 4445

Chapitre 4

Détection des codes d'embrouillages

Sommaire

4.1 Introduction

Dans ce chapitre, nous allons illustrer l'intérêt du décodage des séquences pseudo-aléatoires par deux applications : la détection aveugle des codes d'embrouillage des liaisons montantes des systèmes WCDMA et CDMA2000. Le 3GPP a standardisé le système WCDMA (Wideband Code-Division Mulitple Access) pour son mode duplex de l'UMTS (Universal Mobile Telecommunication System)[73], alors que le système CDMA2000 est la version américaine de la 3G, standardisé par le 3GPP2 [4]. Ces technologies sont en exploitation depuis plus de 10 ans et sont donc très matures. Elles utilisent une technique d'accès multiple de type CDMA, en mode asynchrone. Les premières versions de ces standards n'avaient cependant pas anticipé l'apparition des femto-cellules [74]. Cette technologie a été bâtie en utilisant l'interface radio pré-existante, ce qui pose des problèmes d'interférence. Le développement des femto-cellules permet de répondre à la demande croissante de capacité des réseaux mobiles. Une femto-cellule est couverte par une station de base (*Base Station*, BS) de type *femto*, qui est conçue spécialement pour un environnement intérieur (e.g. maison, bureau) et qui ne requiert pas de déploiement coordonné. Une station de base de type *macro* assure la couverture globale d'une large zone, alors que les *femto* BS offrent une meilleure qualité de service aux

équipements qui leurs sont connectées. Une femto BS peut être configurée pour fonctionner en mode *ouvert* ou *fermé* par rapport aux équipements qui rentrent dans sa zone de couverture [75]. En mode *ouvert*, un terminal visiteur peut basculer de la BS macro à la BS femto. Cela induit une complexité additionnelle pour router les paquets de données de la BS femto vers le réseau de l'opérateur. Il se pose aussi un problème de sécurité pour garantir la confidentialité et l'intégrité des données entre la BS femto et le réseau de l'opérateur. En mode *fermé*, un terminal visiteur reste connecté à la BS macro et ne peut basculer vers la BS femto. Bien que cela simplifie l'architecture du réseau, cela peut générer un niveau d'interférence inacceptable à la BS femto. Cette situation apparaît par exemple si le téléphone visiteur transmet avec une forte puissance car il est éloigné de la BS macro. Cette configuration est illustrée Figure 4.1. Le mécanisme de contrôle de puissance assure que la puissance reçue à la BS macro atteint un niveau requis constant. Le niveau reçu à la BS femto peut donc être très largement supérieur à celui d'un utilisateur connecté à la BS femto. C'est l'effet proche-loin bien connu des systèmes CDMA [76]. Dans certaines configurations, cela peut bloquer toutes les communications de la femto-cellule et créer un trou dans la zone de couverture [77]. Afin de résoudre ce problème, il est nécessaire de mettre en œuvre des techniques d'annulation des interférences. Ce sujet a été largement étudié depuis 25 ans pour les systèmes CDMA [78]. Toutes les techniques proposées exploitent la connaissance des codes d'embrouillage des utilisateurs. Ces derniers sont caractérisés par un identifiant alloué à chaque utilisateur par la BS macro. **Cet identifiant n'est pas connu de la BS femto si elle fonctionne en mode fermé car il n'y a aucun lien de signalisation entre les BS macro et femto. Cette dernière doit donc estimer les paramètres permettant de générer le code d'embrouillage d'un utilisateur.**

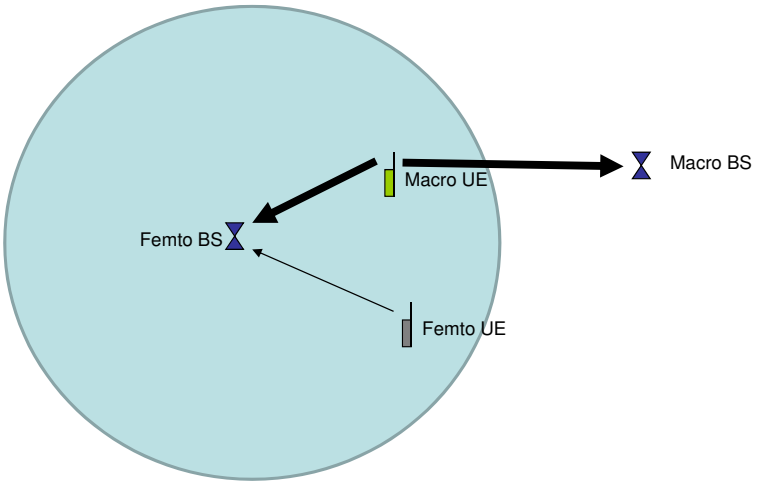

FIGURE 4.1 – Interférence générée par une équipement macro au niveau de la BS femto

L'article de Kerr and Lodge [31] est le seul qui aborde l'estimation du code d'embrouillage. Ils exploitent les caractéristiques de transmission (mise en trame, méthode d'étalement, multiplexage des canaux) de chaque système [3][4] pour définir une étape de pré-traitement qui permet d'éliminer les données modulées. Le signal qui en résulte ne dépend que du code d'embrouillage. Connaissant les polynômes caractéristiques des m-séquences qui sont utilisées pour générer le code d'embrouillage, il est alors possible de mettre en œuvre des techniques de

décodage pour estimer l'état des registres de ces m-séquences, et ainsi de faire l'identification de ce code d'embrouillage. L'article de Kerr et Lodge s'intéresse au décodage du code d'embrouillage du système CDMA2000, et plus spécifiquement des services de données. Dans ce chapitre, nous allons détailler de nouvelles méthodes d'estimation du code d'embrouillage pour les systèmes WCDMA et CDMA2000, puis évaluer leur performance. Dans le cas spécifique du système CDMA2000, nous nous sommes focalisés sur les canaux de services de voix qui utilisent une modulation orthogonale et qui n'ont pas été traités par Kerr et Lodge.

Nous avons aussi utilisé un décodeur itératif par passage de message (cf. section 2.4.3) qui est différent et surtout plus performant que de celui de Kerr et Lodge.

4.2 Système WCDMA

Dans le système WCDMA, l'identifiant du code d'embrouillage prend la forme d'un index encodé sur 24 bits, qui est alloué à l'utilisateur par l'intermédiaire d'un message de signalisation dédié [3].

4.2.1 Génération du signal de la voie montante

Dans cette section, les méthodes de modulation étalement et multiplexage du système WCDMA sont tout d'abord décrites. Ensuite, la construction du code d'embrouillage est détaillée [3]. Ceci est indispensable pour comprendre l'algorithme d'estimation du code d'embrouillage.

4.2.1.1 Étalement et multiplexage

Les bits des canaux de données et de contrôle sont transposés sur une modulation BPSK (binary phase-shift keying) afin que le bit 0 soit représenté par la valeur $+1$ et le bit 1 par -1. Ces 2 canaux sont ensuite multiplexés en phase et en quadrature (I/Q). Dans le standard, ces canaux sont nommés DPDCH et DPCCH, pour dedicated physical data channel et dedicated physical control channel. Ceci est illustré par la Figure 4.2. Les canaux sont respectivement étalés par les séquences C_d and C_c. Le facteur d'étalement du canal DPCCH est fixe : $SF_{\mathrm{DPCCH}} = 256$. Celui du canal DPDCH ($SF_{\mathrm{DPDCH}}$) est variable, il dépend du type de service. La séquence C_c utilisée pour étaler le DPCCH est la séquence 'tout à 1'. La séquence C_d est extraite de la matrice de Walsh-Hadamard de taille SF_{DPDCH}. On sélectionne la colonne $SF_{\mathrm{DPDCH}}/4$ de cette matrice [3]. En analysant les séquences C_d, on observe qu'elles sont constituées d'une répétition du motif '$+1, +1, -1, -1$'. Chaque canal étalé est ensuite pondéré par un coefficient réel β qui permet de régler la puissance émise, et les canaux sont ensuite multiplexés en phase et en quadrature. Le signal complexe obtenu est enfin multiplié chip à chip par le code d'embrouillage C_n. L'index n indique que le code d'embrouillage est spécifique à chaque utilisateur.

A l'instant kT_c, le signal émis est modélisé de la manière suivante :

$$J(k) = C_n(k)(\beta_d X_{\text{DPDCH}}(k) + j\beta_c X_{\text{DPCCH}}(k)) \tag{4.1}$$

X_{DPDCH} et X_{DPCCH} représentent les symboles a_{DPDCH} et a_{DPCCH} des canaux DPDCH et DPCCH, étalés par les séquences C_d et C_c ($k = 0, ..., 38399$) :

$$\begin{aligned} X_{\text{DPDCH}}(k) &= a_{\text{DPDCH}} \left(\left\lfloor \frac{k}{SF_{\text{DPDCH}}} \right\rfloor \right) C_d(k \bmod SF_{\text{DPDCH}}) \\ X_{\text{DPCCH}}(k) &= a_{\text{DPCCH}} \left(\left\lfloor \frac{k}{SF_{\text{DPCCH}}} \right\rfloor \right) C_c(k \bmod SF_{\text{DPCCH}}) \end{aligned} \tag{4.2}$$

L'instant $k = 0$ corresponds au début de la trame.

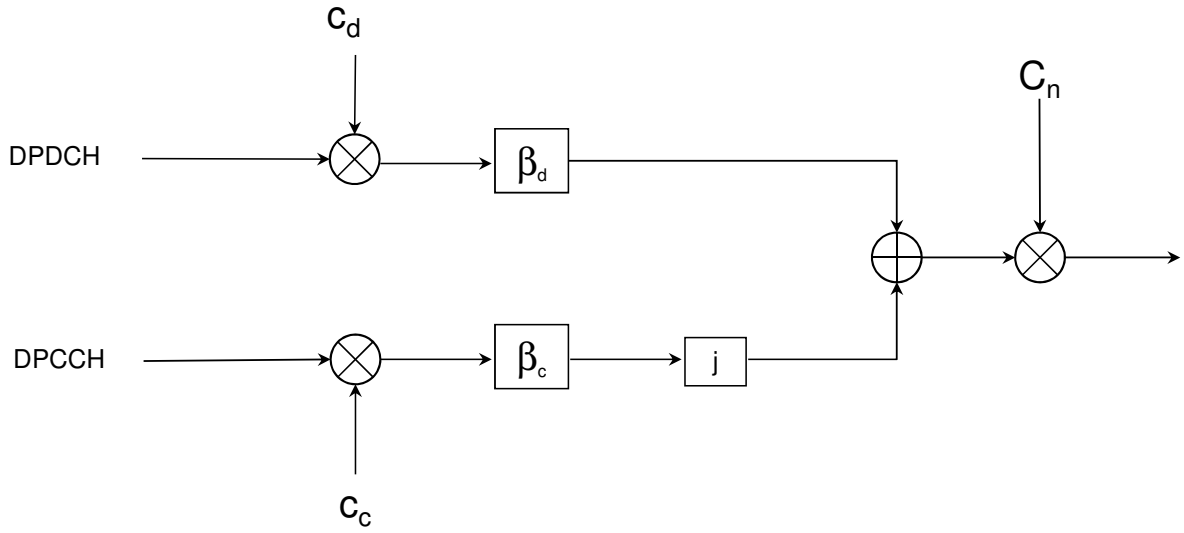

FIGURE 4.2 – Etalement et multiplexage I/Q

4.2.1.2 Génération du code d'embrouillage

Le code d'embrouillage complexe est généré d'après la formule suivante ($k = 0, 1, \ldots, 38399$) [3] :

$$C_n(k) = Z_n(k) \left(1 + j(-1)^k Z_\tau \left(2\lfloor k/2 \rfloor\right)\right) \tag{4.3}$$

Z_n et Z_τ sont définis par :

$$\begin{aligned} Z_n(k) &= (-1)^{(s_n(k) \oplus y(k))} \\ Z_\tau(k) &= Z_n((k + \tau) \bmod (2^{25} - 1)) \end{aligned} \tag{4.4}$$

$\tau = 16777232$ est un délai constant, défini par la standard.

C_n est donc construit à partir de deux m-sequences, s_n and y, qui sont définies par leurs polynômes caractéristiques :

$$\begin{aligned} g_{s_n}(x) &= x^{25} + x^{22} + 1 \\ g_y(x) &= x^{25} + x^{24} + x^{23} + x^{22} + 1 \end{aligned} \tag{4.5}$$

Le code d'embrouillage dépend de l'index n. Soit $n_0, \cdots, n_{23}$ les 24 bits de la représentation binaire de la valeur de n, où n_0 est le bit de poids faible. Les registres des séquences s_n et y sont réinitialisés à chaque début de trame (i.e. tous les 38400 chips) avec les valeurs suivantes :

$$\begin{aligned} &s_n(0) = n_0, s_n(1) = n_1, \ldots, s_n(23) = n_{23}, s_n(24) = 1 \\ &y(0) = y(1) = y(2) = \cdots = y(23) = y(24) = 1 \end{aligned} \tag{4.6}$$

Le vecteur $(n_0, \cdots, n_{23})$ est alloué par la station de base macro dans un message de signalisation dédié. Ainsi, sous l'hypothèse que la synchronisation temporelle avec le début de la trame est acquise, la séquence y est connue, contrairement à la séquence s.

D'après la propriété d'addition des m-séquences (cf. section 1.4), on a :

$$\begin{aligned} s_n(i+\tau) &= s_n(i+4) \oplus s_n(i+7) \oplus s_n(i+18) \\ y(i+\tau) &= y(i+4) \oplus y(i+6) \oplus y(i+17) \end{aligned} \tag{4.7}$$

Cette propriété est utilisée pour générer la séquence Z_τ de l'Eq. 4.4.

4.2.2 Estimation du code d'embrouillage

L'algorithme d'estimation du code d'embrouillage consiste à estimer le vecteur $(n_0, \cdots, n_{23})$. Il est découpé en 4 étapes :

- Étape 1 : Un pré-traitement qui permet d'observer une séquence LFSR directement reliée aux séquences s_n et y ;
- Étape 2 : Élimination de l'influence de la séquence y, ce qui procure une observation d'une version décalée de la séquence s_n.
- Étape 3 : Estimation de l'état initial de la séquence résultante de l'étape 2 à l'aide d'un algorithme de décodage par passage de messages ;
- Étape 4 : Détermination de l'état initial de la séquence s_n : $(n_0, \cdots, n_{23})$.

Afin de faciliter la compréhension de l'algorithme, nous allons restreindre la description à un environnement mono-utilisateur, canal AWGN et un signal échantillonné au rythme chip. La robustesse à un canal multi-trajets et un environnement multi-utilisateurs seront discutés dans les sections 4.2.3.2 et 4.2.3.3.

4.2.2.1 Pré-traitement

Le canal de propagation est supposé avoir un seul trajet (AWGN). Par conséquent, le signal émis (cf. Eq. 4.35) ne subit qu'une rotation de phase. Au récepteur, il est modélisé par :

$$T(k) = e^{j\theta(k)} S_n(k)(\beta_d X_{\mathrm{DPDCH}}(k) + j\beta_c X_{\mathrm{DPCCH}}(k)) + n(k) \tag{4.8}$$

$\theta(k) = \theta_0 + 2\pi k \Delta F T_c$ est la rotation de phase introduite par le canal (θ_0), et l'effet cumulatif du décalage de fréquence porteuse entre l'émetteur et le récepteur ΔF. Le bruit additionnel $n(k)$ représente le résultat en bande de base du bruit thermique du récepteur plus toutes

les sources d'interférences. Il est modélisé par un bruit blanc gaussien complexe circulaire de variance σ_0^2.

La première opération consiste à appliquer un traitement différentiel entre deux échantillons successifs :

$$\begin{aligned} U(k) &= T(k+1)T(k)^* \\ &= e^{j(\theta(k+1)-\theta(k))}C_n(k+1)C_n(k)^*\{\beta_d^2 X_{\mathrm{DPDCH}}(k+1)X_{\mathrm{DPDCH}}(k) \\ &\quad +\beta_c^2 X_{\mathrm{DPCCH}}(k+1)X_{\mathrm{DPCCH}}(k) + j\beta_c\beta_d(X_{\mathrm{DPCCH}}(k+1)X_{\mathrm{DPDCH}}(k) \\ &\quad -X_{\mathrm{DPDCH}}(k+1)X_{\mathrm{DPCCH}}(k))\} + I(k) \end{aligned} \tag{4.9}$$

où $I(k)$ contient tout les produits croisés des termes de bruit.

La rotation de phase résultante $\theta(k+1) - \theta(k)$ peut être enlevée car elle est négligeable. En effet, $\theta(k+1) - \theta(k) = 2\pi\Delta F T_c$, où $T_c \approx 260$ ns est la durée d'un chip. D'après les spécifications de l'UMTS [79][80], l'oscillateur du récepteur doit avoir une précision inférieure à 0.15 ppm par rapport à la fréquence porteuse de l'émetteur f_0. Pour $f_0 = 2$ GHz, cela donne $\Delta F = 300$ Hz et $\Delta F T_c = 7.8\ 10^{-5}$. Par conséquent, le terme $e^{j(\theta(k+1)-\theta(k))}$ peut être négligé, et ce premier pré-traitement permet ainsi de se débarrasser de l'influence de la phase du canal.

On définit la séquence $\tilde{Z}(k) = Z_n(2k+1)Z_n(2k)Z_\tau(2k)$. D'après l'Eq. 4.3, le code d'embrouillage vérifie la relation suivante :

$$C_n(2k+1)C_n(2k)^* = -2j\tilde{Z}(k) \tag{4.10}$$

De plus, étant donné les propriétés des séquences d'étalement C_d et C_c, les signaux étalés $X_{\mathrm{DPDCH}}(k)$ et $X_{\mathrm{DPCCH}}(k)$ satisfont les relations suivantes :

$$\begin{aligned} X_{\mathrm{DPDCH}}(2k+1)X_{\mathrm{DPDCH}}(2k) &= X_{\mathrm{DPCCH}}(2k+1)X_{\mathrm{DPCCH}}(2k) = 1 \\ X_{\mathrm{DPCCH}}(2k+1)X_{\mathrm{DPDCH}}(2k) &= X_{\mathrm{DPDCH}}(2k+1)X_{\mathrm{DPCCH}}(2k) \end{aligned} \tag{4.11}$$

Remarque : ces relations ne sont valides que si $k = 0$ correspond au début de la trame. Si le récepteur est décalé d'un nombre impair de chips, ces propriétés ne sont plus satisfaites.

Définissons la variable $V(k)$, qui correspond aussi au $2^{\text{ième}}$ pré-traitement (décimation et prise de partie imaginaire) du signal reçu, appliqué après le traitement différentiel :

$$V(k) = -\Im\left(U(2k)\right) \tag{4.12}$$

où $\Im(z)$ est la partie imaginaire de la variable complexe z.

En appliquant les équation 4.10 et 4.11, $V(k)$ devient :

$$V(k) = 2(\beta_d^2 + \beta_c^2)\tilde{Z}(k) + w(k) \tag{4.13}$$

La variable $w(k)$ contient tous les termes de bruit.

D'après la définition de la séquence Z_τ (Eq. 4.4), il apparaît que $\tilde{Z}$ est la modulation BPSK d'une séquence binaire $\tilde{z}$:

$$\tilde{z}(k) = \tilde{s}(k) \oplus \tilde{y}(k) \tag{4.14}$$

où $\tilde{s}(k)$ et $\tilde{y}(k)$ sont définies par :

$$\begin{array}{l} \tilde{s}(k) = s_n(2k) \oplus s_n(2k+1) \oplus s_n(2k+\tau) \\ \tilde{y}(k) = y(2k) \oplus y(2k+1) \oplus y(2k+\tau) \end{array} \tag{4.15}$$

En intégrant les résultats de l'Eq. 4.7, les séquences $\tilde{s}(k)$ et $\tilde{y}(k)$ deviennent :

$$\tilde{s}(k) = s_n(2k) \oplus s_n(2k+1) \oplus s_n(2k+4) \oplus s_n(2k+7) \oplus s_n(2k+18) \tag{4.16}$$

$$\tilde{y}(k) = y(2k) \oplus y(2k+1) \oplus y(2k+4) \oplus y(2k+6) \oplus y(2k+17) \tag{4.17}$$

Les propriétés d'addition et de décimation des m-séquences (cf. section 1.4) permettent de conclure que $\tilde{s}$ (resp. $\tilde{y}$) est une version décalée de la m-sequence s_n (resp. y). Ces deux séquences forment une 'paire préférentielle', ce qui signifie que $\tilde{z}$ est une séquence de Gold [36]. On en déduit finalement que $V(k)$ représente l'observation de la séquence de Gold $\tilde{z}$ corrompue par du bruit additif.

De plus, comme nous l'avons vu à la section 2.4.3, il est possible de décoder une séquence pseudo-aléatoire avec un algorithme de décodage par passage de messages. En appliquant ces techniques, il est possible de retrouver l'état initial des registres de la séquence $\tilde{z}$, puis des séquences s_n et y. Le polynôme caractéristique de la séquence z_n est :

$$g_z(x) = g_{s_n}(x) g_y(x) = x^{50} + x^{49} + x^{48} + x^{46} + x^{45} + x^{44} + x^{24} + x^{23} + 1 \tag{4.18}$$

Son poids est élevé (il vaut 9), ce qui est néfaste pour les performances de décodage. Nous avons donc proposé une nouvelle stratégie de détection, qui exploite la connaissance à priori de l'état initial de la séquence y au début de chaque trame. Cette stratégie implémente une recherche en série de la séquence s_n. L'intérêt est que le polynôme $g_{s_n}(x)$ a un poids $t = 3$, ce qui assure une bonne performance de détection.

4.2.2.2 Recherche en série

La procédure de recherche en série combine deux opérations : synchronisation avec le début de la trame et estimation de l'état initial des registres.

Nous allons utiliser les hypothèses de synchronisation H_0 et H_1 définies à la section 2.2.1, que nous rappelons ici :

- H_0 : le récepteur n'est pas synchronisé avec le début de la trame. Ceci correspond à une absence de signal, tel que cela a été défini à la section 2.2.1.
- H_1 : le récepteur est synchronisé avec le début de la trame. On observe le signal à détecter.

A chaque instant qT_c, le récepteur suppose que l'hypothèse H_1 est valide. Il va donc appliquer la procédure d'estimation que nous allons détailler maintenant. Étant donné que l'état initial des registres de la séquence y est connu au début de la trame (tout à '1'), le récepteur peut générer la séquence y puis $\tilde{y}$ d'après l'Eq. 4.17. Les M échantillons du vecteur $(V(q), \cdots, V(q+M-1))$

obtenus par l'intermédiaire de l'Eq. 4.13 sont multipliés chip à chip par la modulation BPSK de $\tilde{y}$:

$$R_q(k) = V(k+q)(-1)^{\tilde{y}(k)} \qquad k = 0, \ldots, M-1 \tag{4.19}$$

Si l'hypothèse H_1 est satisfaite, cette opération élimine le terme $\tilde{y}(k)$ de l'Eq. 4.14 :

$$R_q(k) = 2(\beta_d^2 + \beta_c^2)(-1)^{\tilde{s}(k)} + w_q(k) \tag{4.20}$$

où $w_q(k) = (-1)^{\tilde{y}(k)} w(k+q)$ est le terme de bruit embrouillé par la séquence $\tilde{y}$. Par conséquent, $R_q(k)$ est une observation bruitée de la séquence $\tilde{s}$, modulée en BPSK.

Nous avons établi dans la section précédente que $\tilde{s}$ est une version décalée de la séquence s_n. Elle peut donc être décodée avec un algorithme de décodage par passage de messages, tel que décrit à la section 2.4.3. La matrice de parité du décodeur est construite à partir du polynôme caractéristique de la séquence s_n. Elle est constituée par la concaténation de K matrices de parité élémentaires, de la forme :

$$E = \begin{bmatrix} g_r & \cdots & g_0 & 0 & \cdots & \cdots & 0 \\ 0 & g_r & \cdots & g_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & g_r & \cdots & g_0 & 0 \\ 0 & \cdots & \cdots & 0 & g_r & \cdots & g_0 \end{bmatrix} \tag{4.21}$$

Les matrices élémentaires ne sont pas circulantes car le récepteur n'observe pas la séquence sur toute sa longueur. Chaque matrice est générée par le polynôme $g_l(x) = g_{s_n}(x^{2^l})$. On exploite en effet le propriété suivante : $g(x^{2^l}) = g(x)^{2^l}$, qui est valable dans GF(2).

Le vecteur $(R_q(0), \cdots, R_q(M-1))$ alimente le décodeur de type Min-Sum [54] et ce dernier indique s'il a trouvé un mot de code valide. Si la réponse est positive il fournit aussi l'état initial des registres qui a permis de générer ce mot de code :

$$\{I_c, \hat{A}_{\tilde{s}}\} = dec(R_q(0), R_q(1), \cdots, R_q(M-1)) \tag{4.22}$$

où Ic est la fonction d'indication de l'appartenance au code, définie par l'Eq. 2.20.

Si le décodeur ne trouve pas un mot de code, cela signifie que l'hypothèse H_1 n'est pas valide ou que le décodeur a fait une erreur. Dans les 2 cas, la procédure d'estimation est recommencée à l'instant $q+1$ suivant. Si le décodeur trouve un mot de code, l'état initial de la séquence s_n est calculé par une simple multiplication matricielle. La méthode de calcul de cette matrice sera détaillée à la section suivante.

4.2.2.3 Détermination de l'état initial de la séquence s_n

Le décodeur fournit les r chips qui représentent l'état initial de la séquence $\tilde{s}$. On les modélise sous la forme du vecteur colonne $U_{\tilde{s}}$. L'objectif est donc de remonter à l'état initial de la séquence s_n : $U_{s_n} = (n_0, n_1, \cdots, n_{24})^T$, sachant que $n_{24} = 1$ (cf. Eq. 4.6). Cette tâche

est réalisée en deux étapes. La première consiste à déterminer l'état initial d'une séquence intermédiaire s_{decim}, qui par une décimation d'un facteur 2 donne la séquence $\tilde{s}$:

$$\tilde{s}(k) = s_{decim}(2k) \tag{4.23}$$

D'après l'Eq. 4.16, s_{decim} est reliée à s_n par la formule suivante :

$$s_{decim}(k) = s_n(k) \oplus s_n(k+1) \oplus s_n(k+4) \oplus s_n(k+7) \oplus s_n(k+18) \tag{4.24}$$

Il existe une matrice de transposition B_s entre les vecteurs $U_{s_{decim}}$ et $U_{\tilde{s}}$ [81] :

$$U_{s_{decim}} = B_s U_{\tilde{s}} \tag{4.25}$$

B_s dépend uniquement du polynôme caractéristique de la sequence s. La méthode de construction est détaillée dans la section A.4 de l'Annexe.

La deuxième étape consiste à calculer U_{s_n} à partir de $U_{s_{decim}}$. On exploite pour cela la propriété d'addition des m-séquences et l'Eq. 4.24. Soit $U_s(k)$ l'état des registres de la séquence s à l'instant k. Il existe une matrice de transition G_s entre les instants k et $k+1$ (cf. Eq. A.3 dans la section A.2 de l'Annexe) :

$$U_s(k+1) = G_s U_s(k) \tag{4.26}$$

Ainsi, nous avons :

$$U_{s_n} = (I_r + G_s + G_s^4 + G_s^7 + G_s^{18})^{-1} U_{s_{decim}} \tag{4.27}$$

où I_r est la matrice identité de taille $r \times r$.

Ces deux étapes sont finalement combinées dans une seule multiplication matricielle :

$$U_{s_n} = T_s U_{\tilde{s}} \tag{4.28}$$

où

$$T_s = (I_r + G_s + G_s^4 + G_s^7 + G_s^{18})^{-1} B_s \tag{4.29}$$

Il est important de noter que les calculs de l'Eq. 4.28 doivent être réalisés dans GF(2). De plus, la matrice T_s doit être calculée une seule fois, puis mémorisée. Cela représente $25^2 = 625$ valeurs binaires à conserver.

Si le décodeur détecte un mot de code valide, il est important de vérifier si c'est une fausse alarme ou non. La taux de fausse alarme qui a été mesuré par simulation est trop élevé pour assurer un temps d'acquisition raisonnable. Une étape de vérification a donc été implémentée pour pallier ce problème.

4.2.2.4 Etage de vérification

Soit $\tilde{s}_{estim}$ la séquence générée à partir de l'état initial $U_{\tilde{s}}$ qui a été estimé. L'étage de vérification calcule une variable de décision fondée sur la corrélation entre le signal reçu $R_q(k)$ et la séquence estimée $\tilde{s}_{estim}$:

$$D = \sum_{k=0}^{L-1} R_q(k)(-1)^{\tilde{s}_{estim}(k)} \tag{4.30}$$

où L est la longueur d'intégration cohérente, exprimée en chips. La variable D est ensuite comparée à un seuil de détection γ. Si $D \geq \gamma$, la détection est déclarée correcte, sinon elle est rejetée comme étant une fausse alarme. γ est fixé pour un niveau de fausses alarmes pré-défini : $P_{FA,verif} = P(D \geq \gamma | H_0)$.

Sous l'hypothèse H_0, si les variables $R_q(k)$ sont supposées indépendantes, identiquement distribuées, gaussiennes de moyenne nulle et de variance σ^2 et non-corrélées avec la séquence estimée $(-1)^{\tilde{s}_{estim}(k)}$, alors $P_{FA,verif}$ est défini par la formule suivante [41] :

$$P_{FA,verif} = \frac{1}{2}\left(1 - erf\left(\frac{\gamma}{\sigma\sqrt{2L}}\right)\right) \tag{4.31}$$

où $erf(x)$ est la fonction erreur :

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Le seuil normalisé $\gamma_{norm} = \dfrac{\gamma}{\sigma\sqrt{2L}}$ peut alors être évalué en fonction du niveau de fausse alarme $P_{FA,verif}$ désiré. La calcul de γ_{norm} suppose que la variance du bruit (thermique et interférences) est estimé par le récepteur. Ceci est atteint avec de bonnes performances car le RSB du signal en entrée du module de vérification est largement négatif.

Les performances de la procédure globale (i.e., décodage et vérification) sont caractérisées par les probabilités de détection correcte P_{CD}, de mauvaise détection P_{WD}, de non-détection P_{ND}, et de fausse alarme P_{FA}. Si on note $\hat{U}_{s_n}$ l'état initial de la séquence s_n estimé par l'algorithme, ces probabilités sont définies par :

$$\begin{aligned}
&P_{CD} = P(I_c = 1 \text{ et } \hat{U}_{s_n} = U_{s_n} \text{ et } D \geq \gamma | H_1) \\
&P_{WD} = P(I_c = 1 \text{ et } \hat{U}_{s_n} \neq U_{s_n} \text{ et } D \geq \gamma | H_1) \\
&P_{FA} = P(I_c = 1 \text{ et } D \geq \gamma | H_0) \\
&P_{ND} = P(I_c = 0 \text{ ou } I_c = 1 \text{ et } D \leq \gamma | H_1) \\
&P_e = 1 - P_{CD}
\end{aligned} \tag{4.32}$$

4.2.2.5 Synthèse de l'algorithme

Le pseudo-code de l'algorithme est présenté ci-dessous. Il doit être implémenté au rythme chip, ce qui requiert un décodeur fonctionnant à un rythme très élevé.

Algorithm 4 Algorithme de détection du code d'embrouillage

$q = 0$
Décision = ÉCHEC
while Décision == ÉCHEC **do**
  Calcule $R_q(0), R_q(1), \cdots, R_q(M-1)$ d'après l'Eq. 4.19
  $\{I_c, \hat{U}_{\tilde{s}}\} = dec(R_q(0), R_q(1), \cdots, R_q(M-1))$
  if $I_c = 1$ **then**
    Détermine l'état initial de la séquence s_n : $\hat{U}_{s_n} = T_s \hat{U}_{\tilde{s}}$
    Calcule la variable D d'après l'Eq. 4.30
    if $D \geq \gamma$ **then**
      Décision = SUCCÈS
    end if
  end if
  $q \leftarrow q + 1$
end while

4.2.3 Evaluation des performances

Les performances sont mesurées par les probabilités d'erreur de détection $P_e = 1 - P_{CD}$, de fausse alarme P_{FA}, et de détection erronée P_{WD}, et le temps moyen d'acquisition T_{acq}. Ce dernier est finalement le paramètre le plus important pour un dispositif opérationnel. Il mesure le temps moyen pour détecter correctement le code d'embrouillage sachant que l'instant de démarrage est choisi aléatoirement par rapport au début de la trame.

Ces paramètres ont été mesurés avec l'environnement de simulation suivant :

- L'émetteur envoie un signal correspondand au service de voix de référence (12.2 kbps), tel que spécifié dans la norme [82] : $SF_{\mathrm{DPDCH}} = 64$, $SF_{\mathrm{DPCCH}} = 256$, $\beta_c = 1$ and $\beta_d = 11/15$.
- Lorsqu'on mesure P_e et P_{WD}, le récepteur est synchronisé avec le début de la trame (i.e. l'hypothèse H_1 est satisfaite). Lorsqu'on mesure P_{FA}, le récepteur est alimenté avec du bruit gaussien (i.e. l'hypothèse H_0 est satisfaite).
- Le décodeur implémente un algorithme de décodage de type Min-Sum (MS) [54]. Il s'arrête lorsqu'il trouve un mot de code ou bien lorsque le nombre maximal d'itérations N_{iter} est atteint.
- Une acquisition est validée si le code d'embrouillage est correctement détecté dans l'intervalle défini par le début de la trame t_0 et $t_0 + W$, où W est la fenêtre d'observation qui prend en compte l'étalement maximal du canal. Il a été fixé à $W = 5$ chips pour supporter le canal 'Case 3' qui est le plus étalé (cf. Tableau 4.1).

Sous l'hypothèse H_1, le rapport signal à bruit (RSB) à l'entrée du récepteur est défini par RSB $= 2(\beta_c^2 + \beta_d^2)/\sigma^2$, où σ^2 est la variance du bruit blanc gaussien.

4.2.3.1 Probabilités de détection et de fausse alarme

La configuration optimale des paramètres M, N_{iter}, et K a été obtenue au moyen de simulations sur un canal de type AWGN (i.e. mono-trajet). Durant cette phase, l'étage de vérification a été omis. En effet, son but est de rejeter des fausses alarmes tout en minimisant la dégradation de la probabilité de détection correcte. Par conséquent, la probabilité d'erreur de détection P_e est supposée être insensible à la présence ou non de l'étage de vérification. Les paramètres de configuration sélectionnés sont ceux qui minimisent P_e.

La Figure 4.3 montre P_e en fonction du RSB pour plusieurs valeurs du nombre maximum d'itérations de l'algorithme de décodage (N_{iter}). Il n'est pas nécessaire d'excéder $N_{iter} = 20$ itérations. Cette valeur a été retenue pour l'ensemble des simulations.

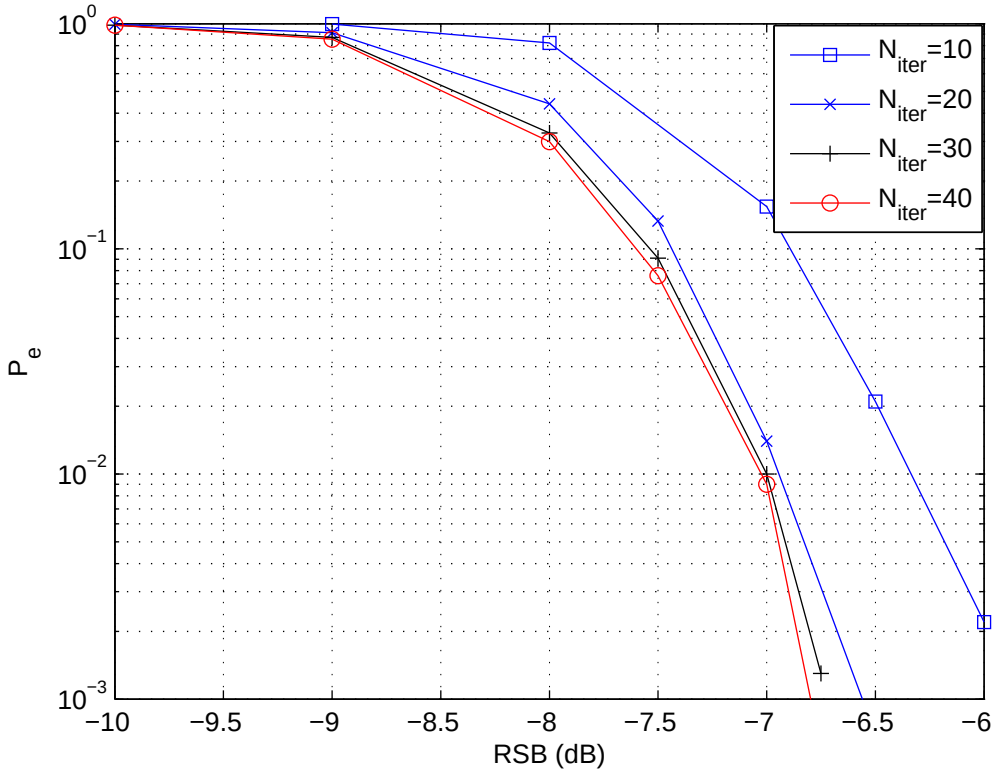

FIGURE 4.3 – Influence du nombre d'itérations N_{iter}. L'étage de validation est omis. $M = 4000$ et $K = 7$

La Figure 4.4 montre P_e en fonction du RSB pour plusieurs valeurs de K. Pour une valeur cible $P_e = 10^{-2}$, un gain de 7 dB est obtenu avec $K = 7$ matrices élémentaires par rapport à la configuration avec une seule matrice. Les performances obtenues avec l'algorithme Sum-producte (SP) [15] sont aussi montrées à titre de comparaison. Le gain est assez minime et ne vaut pas l'augmentation de complexité qui est induite par l'algorithme.

La Figure 4.5 montre la sensibilité de la détection aux nombre de variables M utilisées par le

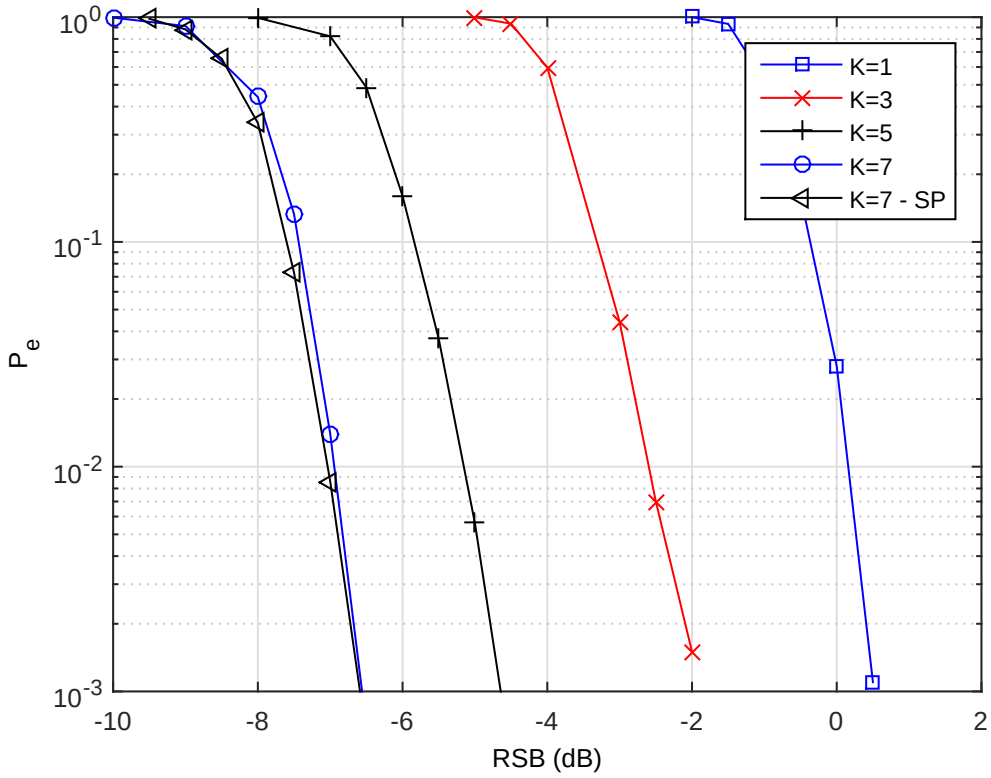

FIGURE 4.4 – Influence du nombre d'équations de parité K. L'étage de validation est omis. $N_{iter} = 20$ et $M = 4000$

décodeur. Il n'est pas nécessaire d'utiliser plus de 4000 chips. Un gain de 0.75 dB est obtenu en doublant la taille de $M = 2000$ à $M = 4000$ chips. Ce gain est relativement faible, ce qui laisse de la marge pour trouver un bon compromis complexité/performance en vue d'une réalisation matérielle. En effet, la complexité du décodeur dépend fortement du nombre de variables et de leur connectivité. Si on peut diviser par 2 le nombre de variables, le gain en ressources matérielles est important.

La Figure 4.6 montre la sensibilité de P_{FA} au nombre d'itérations N_{iter} lorsque l'étage de vérification est omis (RSB $= -5$ dB). Lorsque le nombre d'équations de parité est élevé, le décodeur possède moins d'ensembles absorbants (cf. section 3.4.1) et corrige plus d'erreurs lorsque le nombre d'itérations croît. Ceci augmente la probabilité de fausse alarme. Elle atteint un niveau inacceptable $P_{FA} = 0.03$ lorsque $N_{iter} = 20$. Heureusement, l'étage de vérification va fournir une solution efficace pour réduire ce taux de fausses alarmes.

La Figure 4.7 montre P_{FA} en fonction du seuil de détection normalisé γ_{norm} lorsque $-10log(\sigma^2) = -10$ et -5 dB. Il a aussi été observé que P_{FA} était indépendant du RSB. En effet, l'algorithme de décodage MS est insensible à une multiplication des LLRs en entrée par un facteur commun. Cela signifie qu'il produira le même résultat qu'on l'alimente avec les vecteurs Y ou $\zeta^2 Y$. Lors de l'évaluation de P_{FA}, le vecteur d'entrée est constitué d'échantillons

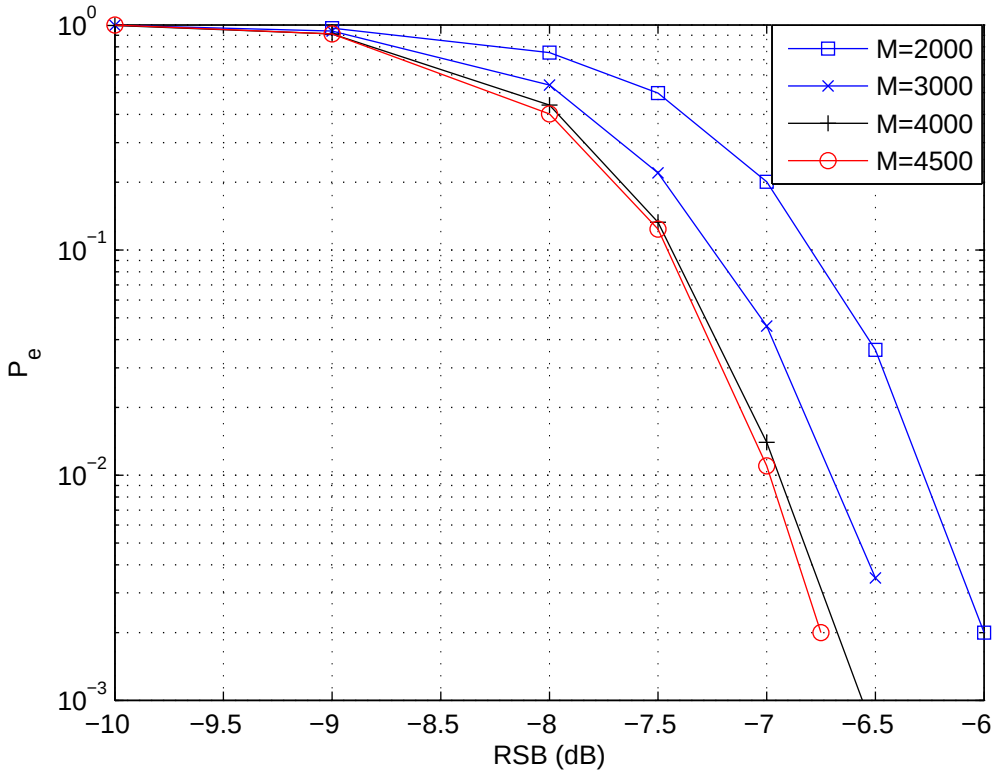

FIGURE 4.5 – Influence du numbre de variables M. L'étage de validation est omis. $N_{iter} = 20$ et $K = 7$

de bruit blanc gaussien de variance σ^2. Par conséquent, P_{FA} reste inchangé quand on applique un facteur multiplicatif ζ/σ pour modifier le RSB.

Le taux de fausses alarmes cible doit être inférieur à 10^{-4}. Cela correspond à un seuil normalisé $\gamma_{norm} \geq 4$. Si le mot de code trouvé par le décodeur était indépendant des variables d'entrées $R_q(k)$, P_{FA} devrait suivre la courbe théorique donnée par l'Eq. 4.31. C'est la courbe avec la légende 'i.i.d.' dans la Figure 4.7. On observe cependant une grande différence. Ceci s'explique de la manière suivante. Si le graphe de Tanner du décodeur est un arbre, l'algorithme MS implémente un décodage de type MLSE [54]. Il trouve donc le mot de code ayant la corrélation la plus élevée avec le signal d'entrée. L'hypothèse d'indépendance qui a été faite pour trouver l'Eq. 4.31 n'est donc pas valable. Ceci explique l'écart observé.

Une fois que γ_{norm} est choisi, l'impact de la longueur d'intégration cohérente L sur la probabilité de détection raté de l'ensemble (décodeur + étage de vérification) doit être évalué. Ceci est montré à la Figure 4.8. Les performances obtenues lorsque l'étage de vérification est omis servent de référence. Pour $\gamma_{norm} = 4.25$, la longueur d'intégration doit être supérieure ou égale à $L = 1024$. Cela assure une dégradation minimale de P_e. Si $L = 512$, une dégradation de 2.5 dB est observée pour $P_e = 10^{-2}$, par rapport à la configuration de référence. Cela valide aussi notre hypothèse de départ. L'étage de validation a peu d'influence sur P_e si γ et L sont sélec-

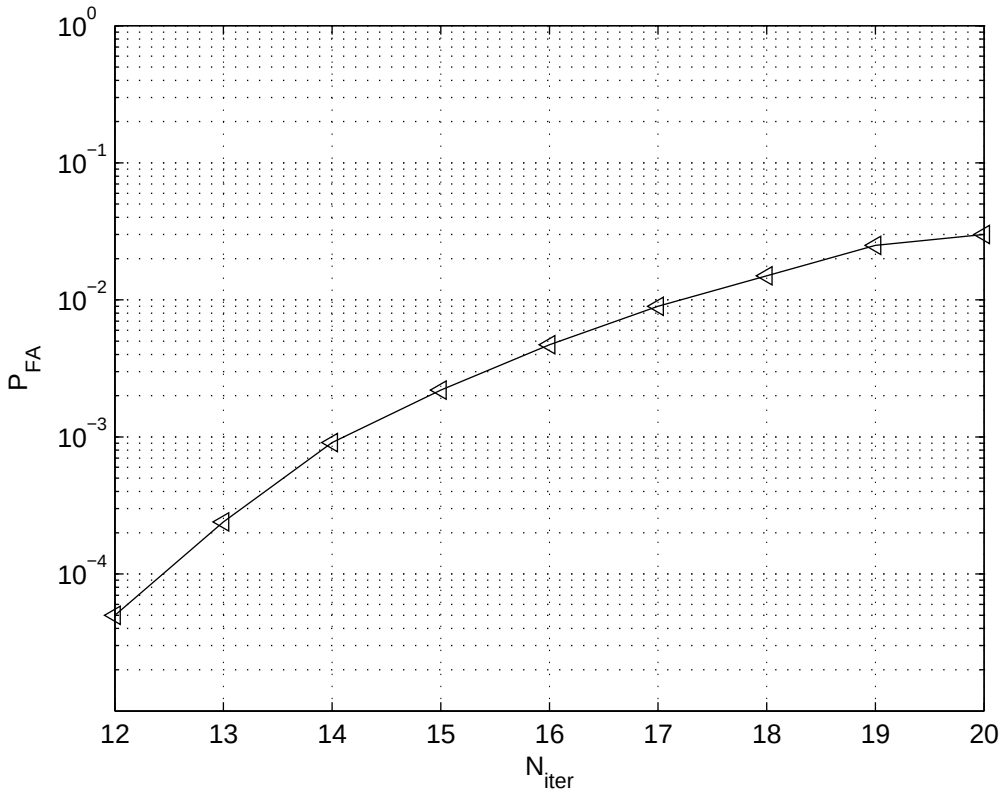

FIGURE 4.6 – Probabilité de fausse alarme : influence du nombre d'itérations. $-10log(\sigma^2) = -5$ dB

tionnés correctement. Les performances obtenues avec le décodeur proposé par Kerr et Lodge [83] sont aussi affichées. Notre algorithme offre une amélioration de 5 dB pour $P_e = 10^{-2}$. Ceci est dû au choix du décodeur. Ils ont implémenté un décodeur qui utilise les bits les plus fiables pour corriger par un mécanisme de ré-encodage les bits les moins fiables. Le décodeur utilise pour cela une matrice de parité mise sous une forme systématique (voir [83] pour plus de détails). S'il y a beaucoup d'erreurs parmi les bits les plus fiables, le décodeur n'arrivera pas à corriger les $M - r$ bits les moins fiables. Cette situation apparaît souvent lorsque le décodeur travaille à une RSB négatif, ce qui explique la limitation des performances. Avec notre approche, une BS femto peut détecter un utilisateur macro ayant un RSB = -6 dB. Cela correspond à un interféreur puissant dans le contexte du réseau WCDMA.

Un autre résultat a aussi été obtenu avec ces simulations : l'étage de vérification élimine les détections erronées ($P_{WD} = 0$). Si un mot de code erroné est décodé, la corrélation avec le signal d'entrée ne dépasse pas le seuil fixé.

Les valeurs des paramètres $\gamma_{norm} = 4.25$ et $L = 1024$ seront choisies pour toutes les autres simulations.

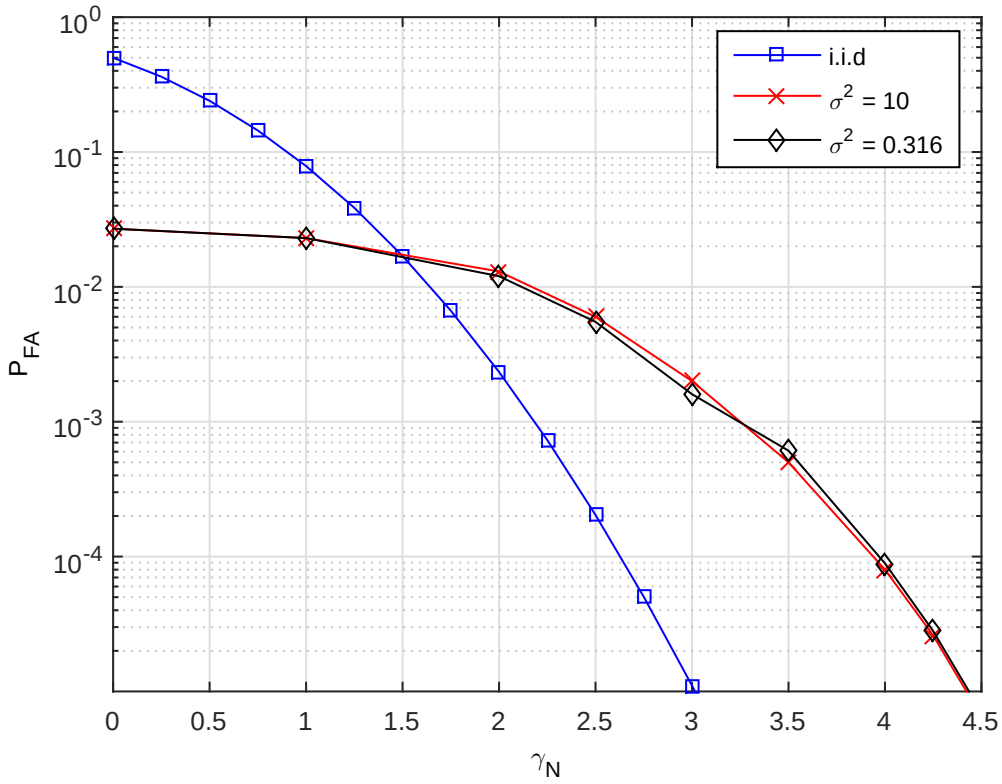

FIGURE 4.7 – Probabilité de fausse alarme en sortie de l'étage de vérification.

4.2.3.2 Robustesse à des trajets multiples

La robustesse à un canal multi-trajets a été évaluée en appliquant le canal 'case 3' défini par le 3GPP pour les tests de conformité des stations de base [80]. Son profil puissance-délai est donné dans le Tableau 4.1. L'atténuation moyenne est donnée par rapport au trajet le plus puissant. Afin de ne tenir compte que de l'impact des trajets multiples, l'énergie de la réponse impulsionnelle du canal est normalisée à 1. Cela rend le RSB constant à l'entrée du récepteur. La Figure 4.9 montre P_e en fonction du RSB. Les performances sont dégradées par rapport à un canal mono-trajet car le RSB par trajet diminue. Or le détecteur se synchronisant sur un trajet, il est sensible au RSB par trajet et non au RSB global. On observe aussi un légère dégradation par rapport au scénario sans étage de vérification. Même si la dégradation par rapport à la configuration mono-trajet est significative, l'algorithme fonctionne toujours, il est robuste aux trajets multiples.

4.2.3.3 Robustesse à un environnement multi-utilisateurs

Considérons une configuration où deux utiliateurs sont reçus avec un délai de ν chips. Supposons aussi que le récepteur est synchronisé avec le début de la trame du premier uti-

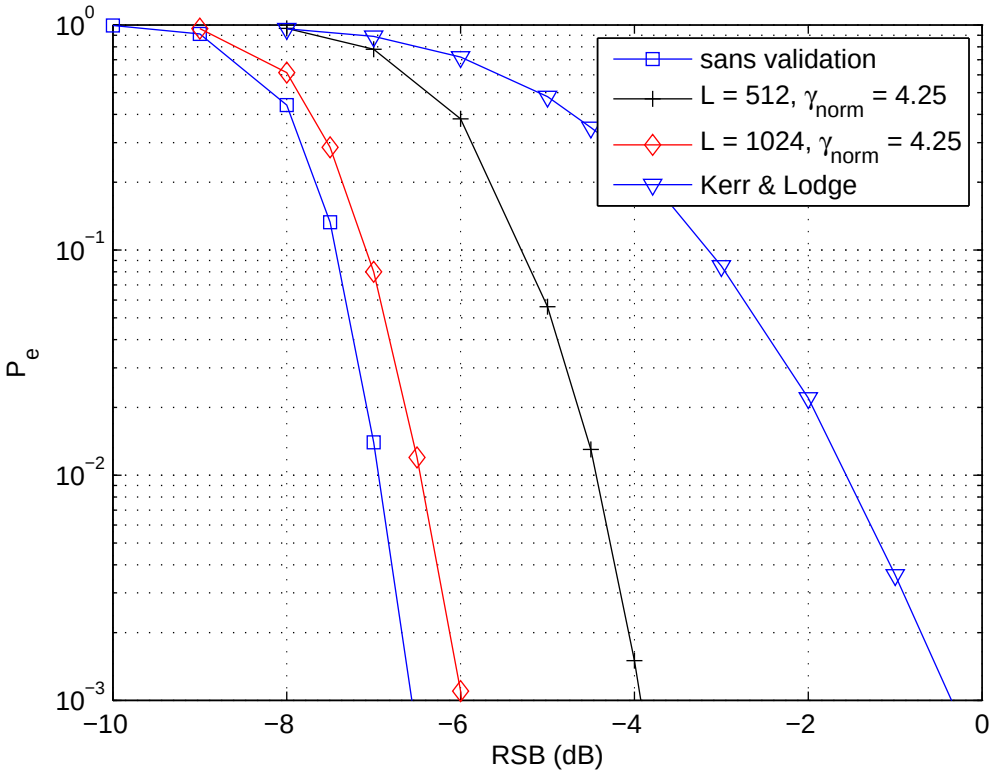

FIGURE 4.8 – Influence de γ_{norm} et L sur P_e.

TABLE 4.1 – Modèle de canal 'Case 3'

Délai (chip)	Atténuation moyenne (dB)
0	0
1	-3
2	-6
3	-9

lisateur. Si If $\nu \geq 1$, la multiplication par $(-1)^{\tilde{y}(k)}$ dans l'Eq. 4.19 va embrouiller le signal du deuxième émetteur. Par conséquent, le signal retardé de ν chips est vu comme une source additionnelle de bruit par le décodeur. Cela peut dégrader les performances de détection car le RSB diminue, mais l'algorithme est toujours fonctionnel. D'un autre côté, si $\nu = 0$, le décodeur peut éventuellement raté la détection des deux transmissions. Cela dépend du ratio de puissance entre les 2 émetteurs. Ceci est illustré par la Figure 4.10. La probabilité de détection ratée pour le premier utilisateur est affichée en fonction du ratio de puissance entre les 2 utilisateurs $\rho = 10log(P_{interf}/P_u)$. La puissance du premier utilisateur est notée P_u et celle du deuxième P_{interf}. Le niveau de bruit thermique est fixé de telle manière que le RSB vaille -6 dB dans une configuration mono-utilisateur. La borne mono-utilisateur est aussi affichée pour montrer la performance asymptotique lorsque $\rho \to -\infty$. La dégradation est régulière, mais

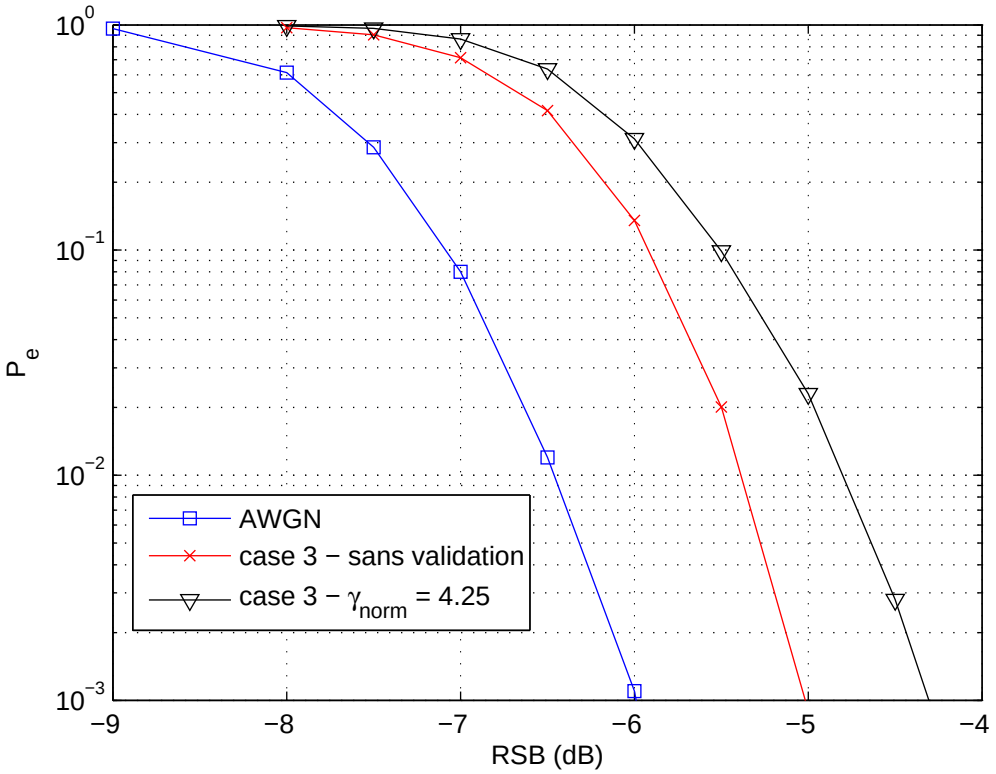

FIGURE 4.9 – Canal à trajets multiples (case 3) - $W = 5$.

l'algorithme n'arrive plus à détecter l'utilisateur d'intérêt si ρ est proche de 0 dB. Heureusement, le standard UMTS a défini une procédure de synchronisation qui limite la probabilité que deux utilisateurs soient reçus d'une manière synchrone au niveau d'une station de base. Elle est détaillée dans [82]. L'algorithme de détection va donc très probablement opérer dans des configurations d'utilisateurs asynchrones.

4.2.3.4 Temps moyen d'acquisition

Le temps d'acquisition mesure le temps nécessaire pour détecter correctement l'état des registres du code d'embrouillage, sachant qu'on démarre à un instant aléatoire. Les résultats de simulation ont montré que P_{WD} est négligeable. Par conséquent, le temps moyen d'acquisition a la même expression que celui obtenu pour une recherche séquentielle conventionnelle [70] :

$$T_{acq} = \left(\frac{1 + \lambda(2 - P_{CD})}{P_{CD}} + M \right) T_c \tag{4.33}$$

où $\lambda = (1 + \Delta P_{FA}) \left(\frac{N-1}{2} \right)$, et $N = 38400$ est la longueur de la trame (en chips).

Δ est la pénalité de fausse alarme. Cela correspond aux nombres de chips nécessaires pour détecter une mauvaise synchronisation et relancer la procédure d'acquisition. Nous avons sup-

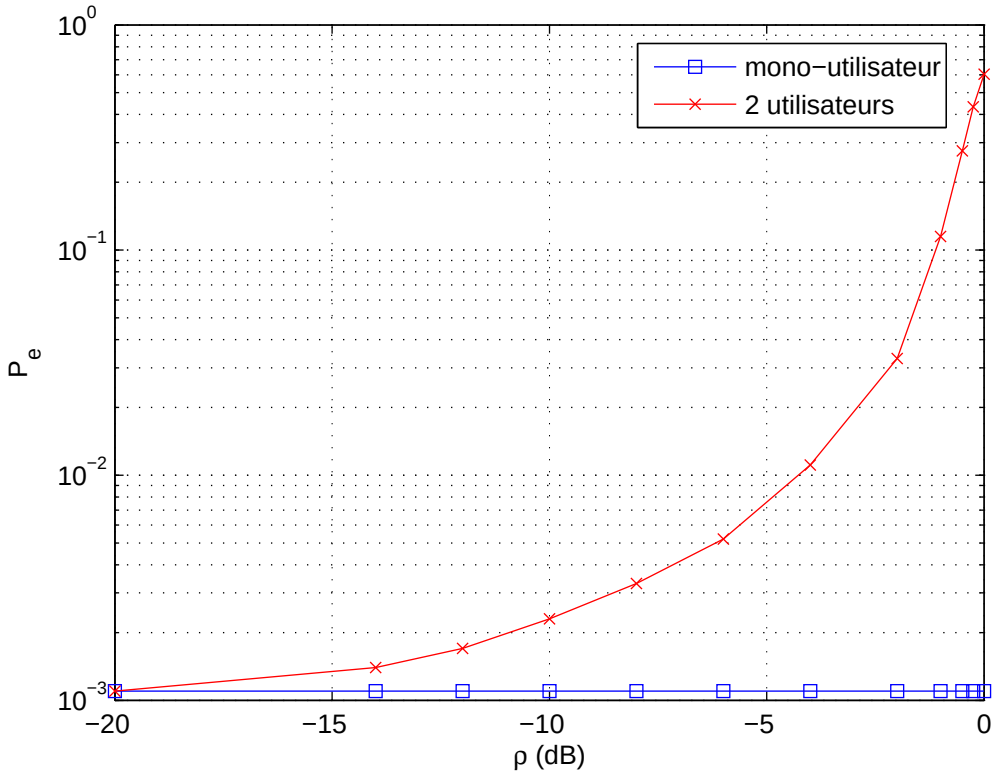

FIGURE 4.10 – P_e pour une réception synchrone de 2 utilisateurs.

posé que cette pénalité valait une trame : $\Delta = 38400$. La Figure 4.11 montre T_{acq} en fonction du RSB pour les canaux AWGN et 'case 3'. Le résultat est normalisé par la durée d'une trame de 10 ms ($T_{acq}/0.01$) pour faciliter l'interprétation. On observe que le code d'embrouillage peut être détecté en une trame pour $RSB > -5$ dB.

D'après la norme [82], un utilisateur émettant le service de voix de référence (12.2 kbps) doit être reçu correctement à la BS macro à laquelle il est attaché, pour un RSB plus grand que -17 dB (canal AWGN). Ainsi, s'il est reçu à la BS femto avec un RSB de -5 dB, sa puissance est 12 dB plus grande qu'un utilisateur femto. C'est donc un interféreur particulièrement fort. Heureusement, l'index de son code d'embrouillage peut être détecté avec l'algorithme que nous avons proposé. Cela offre l'opportunité d'atténuer l'impact de cet interféreur en implémentant des algorithmes de suppression des interférences. Les trous de couverture qui apparaissent autour d'une BS femto peuvent donc être traités pour les faire disparaître.

4.2.4 Conclusions

Nous avons proposé un algorithme de détection de l'index du code d'embrouillage utilisé pour la liaison montante d'une transmission WCDMA. Il exploite la procédure de modulation, étalement et multiplexage du standard WCDMA, il est donc dédié à ce système. Les

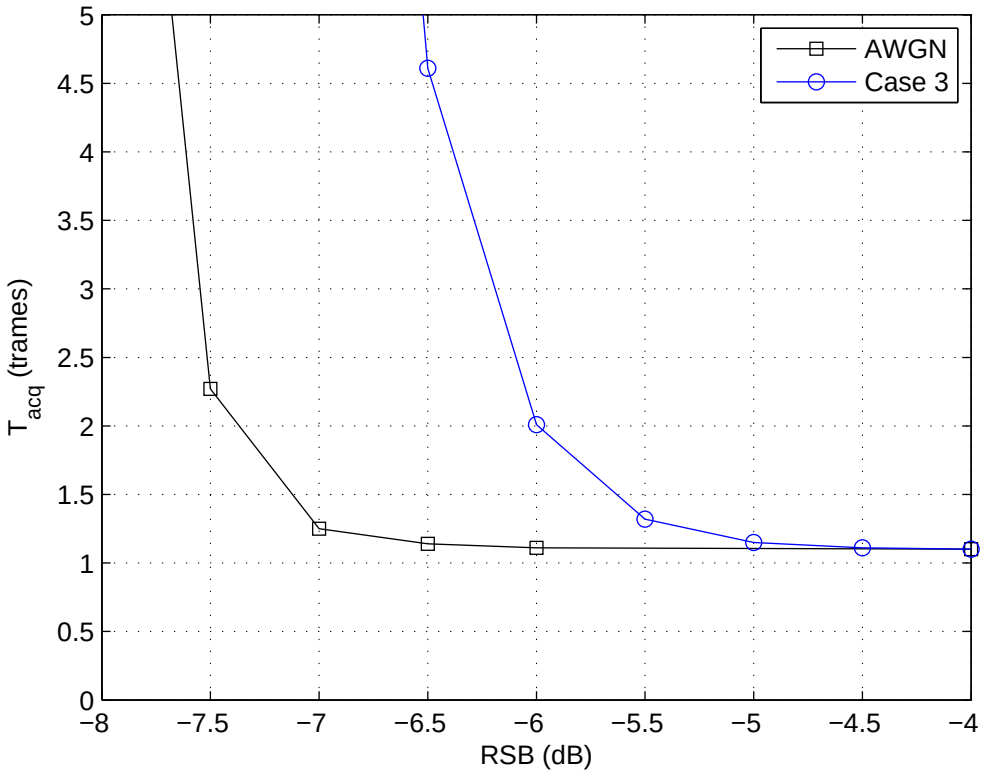

FIGURE 4.11 – Temps moyen d'acquisition. $\gamma_{norm} = 4.25$ et $L = 1024$

résultats de simulations montrent qu'il est possible d'obtenir une estimation fiable pour un RSB supérieur à -7 dB, dans un canal AWGN. Ainsi, il est possible de mettre en œuvre des algorithmes de suppression d'interférences au niveau d'une BS femto et d'éviter les trous de couverture qui apparaissent avec les interféreurs puissants. La robustesse de l'algorithme à un canal multi-trajets et un environnement multi-utilisateurs a aussi été établie. Nous avons enfin comparé les performances de notre algorithme avec celui de Kerr et Lodge [31]. Nous obtenons une amélioration de l'ordre de 5 dB, ce qui est très significatif.

4.3 Système CDMA2000

L'article de Kerr et Lodge s'intéresse uniquement à la détection du code d'embrouillage utilisé par les configuration radio (RC) 3 à 6. Ceci correspond aux services de données. Ils utilisent une modulation de type QPSK. Nous allons nous intéresser au cas de configurations RC 1 et 2 qui mettent en œuvre une modulation orthogonale à 64 états [4]. Ceci modifie les traitements à réaliser.

Kerr et Lodge ont mis en œuvre un algorithme de décodage qui partitionne les M bits reçus entre les r bits indépendants les plus fiables et les $M - r$ autres [83]. La matrice de parité

est alors modifiée pour être mise sous une forme systématique $[I_{M-r}C]$ où la matrice identité I_{M-r} correspond aux $M-r$ bits les moins fiables. L'algorithme de décodage suppose à chaque itération que les bits les plus fiables sont corrects, et sont utilisés pour corriger les bits les moins fiables. En effet, chaque bit 'moins fiable' est relié aux bits les plus fiables par une seule équation de parité. Si les bits les plus fiables sont corrects, le bit le moins fiable est le résultat de l'addition binaire des bits fiables. Cette technique est nommé 're-encodage'. L'algorithme essaye aussi de corriger des bits 'fiables' en détectant d'éventuelles erreurs. Dans ce cas, l'algorithme est itéré une nouvelle fois avec une nouvelle partition entre les bits les plus fiables et les moins fiables. Si aucune erreur n'est détectée, l'algorithme passe à une étape de validation. Le mot de code détecté est corrélé avec le signal reçu et comparé à un seuil de détection. Si le seuil est dépassé, le mot de code est accepté.

L'algorithme de décodage Kerr et Lodge nous est apparu moins performant et surtout ne reposant pas sur des bases théorique solide. Nous avons plutôt opté pour l'algorithme de décodage itératif par passage de messages, tel que décrit à la section 2.4.3.

4.3.1 Génération du signal

Dans cette section, la méthode de modulation et d'étalement employée pour générer les signaux de canaux RC 1 et 2 sera d'abord détaillée. La méthode de génération du code d'embrouillage sera ensuite expliquée. Ceci est tiré du standard définit par le 3GPP2 [4]. Ces informations seront mises à profit pour définir l'algorithme de détection du code d'embrouillage.

4.3.1.1 Modulation et étalement

La figure 4.12 détaille les opérations de modulation et d'étalement pour les canaux RC 1 et 2. Les données sont tout d'abord transposées sur une modulation orthogonale à 64 états. Les chips obtenus sont répétés 4 fois, puis le résultat est multiplié chip à chip par le code d'embrouillage long S_n. Le signal est enfin multiplié par une séquence complexe $C_I + jC_Q$, puis mis en forme avec le filtre $P(t)$.

La modulation orthogonale à 64 état consiste à associer 6 bits consécutifs $d_i, d_{i+1}, \cdots, d_{i+5}$ à l'une des séquences de Walsh-Hadamard de 64 chips (W_m). L'indice de la séquence est défini par la relation suivante : $m = d_i + 2d_{i+1} + 4d_{i+2} + 8d_{i+3} + 16d_{i+4} + 32d_{i+5}$. Ensuite, chaque chip de la séquence W_m est répété 4 fois, puis embrouillé par le code long S_n. L'indice n indique que le code est spécifique à chaque utilisateur. On obtient finalement le signal $V(k)$:

$$V(k) = S_n(k) W_m\left(\lfloor k/4 \rfloor\right) \tag{4.34}$$

$\lfloor x \rfloor$ est l'entier inférieur le plus proche de x.

Les variables S_n and W_m sont les modulations BPSK des séquence binaires s_n et w_m.

La séquence $V(k)$ est ensuite multipliée par la séquence complexe $C_I + jC_Q$. Les deux séquences c_I et c_Q sont des m-séquences de degré $r = 15$ et donc de période $2^{15} - 1 = 32767$ chips. Leur

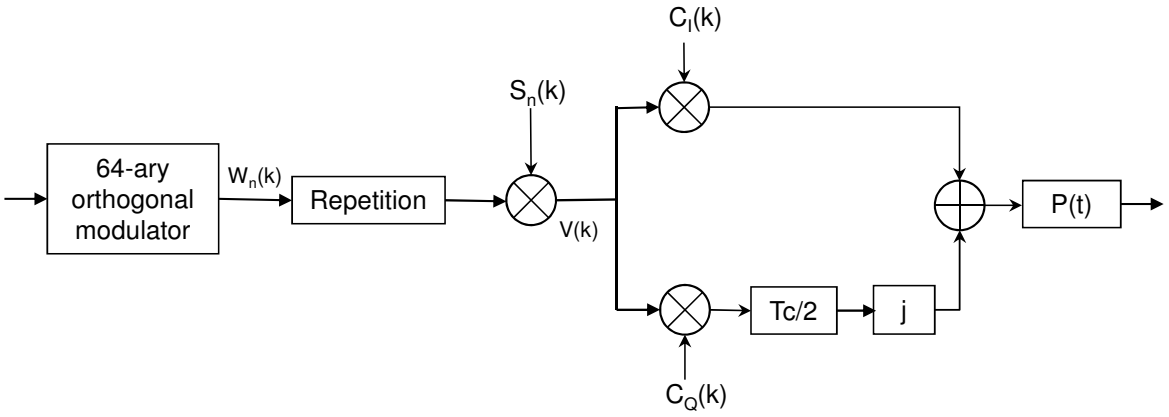

FIGURE 4.12 – Modulation et étalement

polynôme caractéristique est défini par :

$$\begin{aligned} g_I(x) &= x^{15} + x^{13} + x^9 + x^8 + x^7 + x^5 + 1 \\ g_Q(x) &= x^{15} + x^{12} + x^{11} + x^{10} + x^6 + x^5 + x^4 + x^3 + 1 \end{aligned}$$

Les séquence c_I et c_Q sont complétées par un '0' pour créer une séquence de $N = 32768$ chips. L'état initial de ces séquences au début de la trame est connu par le récepteur. En effet, le standard restreint l'utilisation de ces séquences à un sous-ensemble de 512 séquences $C_I + jC_Q$ utilisables. L'opérateur du réseau alloue à chaque cellule une séquence, qui est aussi utilisée par les mobiles pour détecter et se synchroniser sur le réseau de l'opérateur. On peut donc raisonnablement supposer qu'elle est connue du récepteur.

Dans cette étude, l'impact de la procédure de contrôle de puissance n'est pas pris en compte. Une trame est découpée en intervalles de temps (slot) dont la puissance d'émission peut être mise à zéro. Ainsi, le rapport entre le nombre de slots émis et le nombre de slots total (15) définit le ratio de fenêtrage. S'il vaut 1, tous les slots sont émis avec la puissance nécessaire. S'il vaut 1/2, un slot sur 2 est émis en moyenne. Si le motif de fenêtrage est aléatoire, cette procédure permet de diminuer les interférences entre utilisateurs au niveau de la station de base. Afin de simplifier la description du fonctionnement de l'algorithme de détection du code d'embrouillage, nous supposerons dans cette étude qu'il n'y a pas de fenêtrage (ratio de 1). S'il est plus petit, le principe de détection du code d'embrouillage sera le même, mais il faudrait paralléliser des instances de l'algorithme.

Après la multiplication par la séquence $C_I(k) + jC_Q(k)$, la partie réelle est filtrée par $P(t)$ alors que la partie imaginaire est retardée d'un demi-chip avant d'être filtrée. Cela implémente une modulation de type *offset QPSK*, qui diminue la rapport puissance crête sur puissance moyenne (PAPR) de l'émetteur. Ceci permet d'optimiser l'utilisation de la puissance au niveau du terminal et donc améliorer la durée de vie des batteries. Les spécifications du filtre $P(t)$ sont détaillées dans la norme [4].

Le signal transmis se modélise finalement de la manière suivante :

$$X(t) = \sum_k V(k) G_k(t - kT_c) \tag{4.35}$$

où $G_k(t) = C_I(k)P(t) + jC_Q(k)P(t - T_c/2)$.
T_c est la période chip et vaut 814 ns. $G_k(t)$ peut être considéré comme un filtre de mise en forme variable au cours du temps.

4.3.1.2 Génération du code d'embrouillage

Le code d'embrouillage est construit à partir d'une m-séquence notée a ayant une très longue période de répétition : 2^{42} chips. La Fig. 4.13 montre le générateur de la séquence, défini dans la représentation de Galois (cf. section 1.2). Son polynôme caractéristique est de degré $r = 42$. Il est défini par :

$$\begin{aligned} g_s(x) \quad = \quad & x^{42} + x^{35} + x^{33} + x^{31} + x^{27} + x^{26} + \\ & x^{25} + x^{22} + x^{21} + x^{19} + x^{17} + x^{16} + \\ & x^{10} + x^7 + x^6 + x^5 + x^3 + x^2 + x + 1 \end{aligned}$$

A l'instant k, le contenu du $i^{\text{ième}}$ registre est noté $u_i(k)$. Le code d'embrouillage est construit par le produit scalaire entre un masque de 42 bits et les 42 états des registres :

$$s_n(k) = \bigoplus_{i=0}^{r-1} u_i(k) m_n(i)$$

$\oplus$ représente l'addition modulo 2.

$m_n(0), \cdots, m_n(r-1)$ est le masque du $n^{\text{ième}}$ utilisateur. Il dépend du numéro de série du terminal et est donc fixe. Le vecteur $u_0(k), u_1(k), \cdots, u_{r-1}(k)$ est connu à un instant donné par l'intermédiaire d'un message de signalisation qui est émis régulièrement (paramètre 'SYS TIME' indiqué dans le *synchronization channel message*). D'un autre côté, le masque est inconnu d'un utilisateur autre qu'une station de base. Notre récepteur ne le connaît donc pas. Il n'est par conséquent pas possible d'exploiter la connaissance de l'état des registres $u_0(k), u_1(k), \cdots, u_{r-1}(k)$ à un instant donné.

En invoquant la propriété d'addition des m-séquences, il apparaît que s_n est une m-séquence dont le polynôme caractéristique est $g_s(x)$. C'est une version décalée de la séquence a : $s_n(k) = a(k + \tau_n)$. τ_n est un retard spécifique au $n^{\text{ième}}$ utilisateur. Il dépend du masque $m_n(0), \cdots, m_n(r-1)$ et de $g_s(x)$ (cf.section A.3 de l'Annexe). En utilisant la représentation de Fibonacci, s_n peut être construite avec le générateur représenté à la Figure 1.1. Plutôt que d'estimer le masque de l'utilisateur, en supposant l'état initial de la séquence a connu, il est plus simple d'estimer directement l'état des registres de la séquence s_n dans sa représentation de Fibonacci. C'est l'objectif de notre algorithme d'estimation du code d'embrouillage.

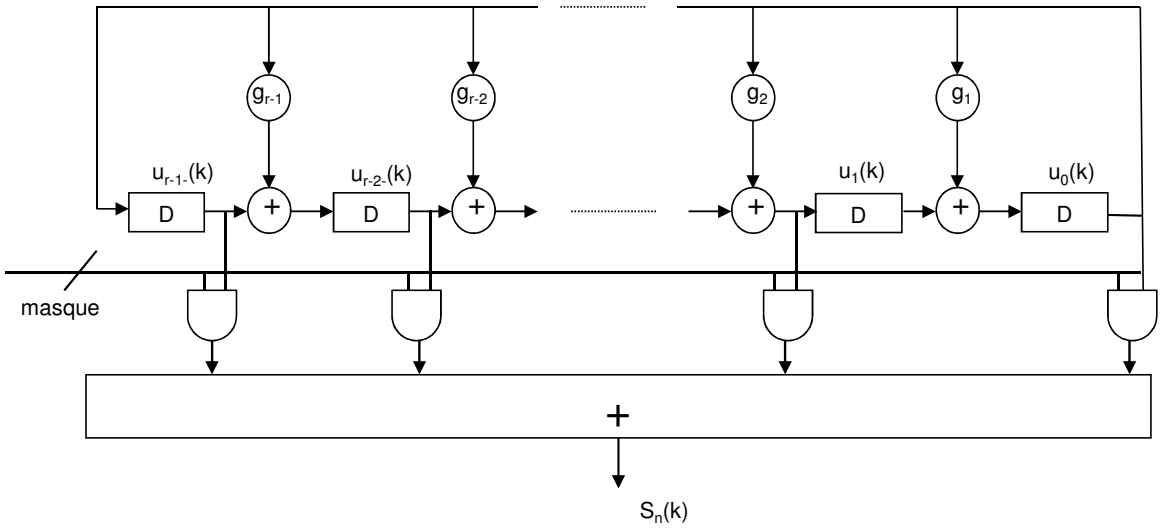

FIGURE 4.13 – Génération du code long

4.3.2 Détection du code d'embrouillage

Comme pour le système WCDMA décrit à la section 4.2, l'algorithme procède en 3 étapes :
- Étape 1 : Un pré-traitement qui permet d'observer une version décalée de la séquence s_n, notée $\tilde{s}$.
- Étape 2 : estimation de l'état initial de la séquence $\tilde{s}$ à l'aide d'un algorithme de décodage par passage de messages.
- Étape 3 : détermination de l'état initial de la séquence s_n à l'aide d'une matrice de transposition des états.

Afin de faciliter la compréhension de l'algorithme, la description sera restreinte à un canal AWGN.

4.3.2.1 Pré-traitement

Contrairement au cas du système WCDMA abordé dans la section 4.2, le signal reçu doit être surréchantillonné afin de permettre une égalisation du filtre $G_k(t)$. La période d'échantillonnage est donc $T = T_c/E$, où E est le facteur de surréchantillonnage. En appliquant cela au signal reçu, modélisé par l'Eq. 4.35, le signal échantillonné est :

$$R(q) = e^{j\theta} \sum_k V(k) G_k(qT - kT_c) + w(q) \tag{4.36}$$

où θ est une rotation de phase introduite par le canal et $w(q)$ le bruit additif modélisant le bruit thermique et les autres sources d'interférences (e.g ; autres utilisateurs).

Notre objectif est de pouvoir observer la séquence s_n. Il faut donc éliminer $G_k(t)$, θ et la séquence W_m. Le filtre de mise en forme $P(t)$ défini par le standard génère de l'interférence

entre chips, un simple filtre adapté n'est pas suffisant pour éliminer $G_k(t)$. Il est nécessaire d'implémenter un égaliseur du filtre variable $G_k(t)$. Ce dernier est modélisé par un filtre FIR (Finite Impulse Response) avec $2L+1$ coefficients : $F_k(-L), \cdots, F_k(0), \cdots, F_k(L)$. Il peut appliquer un critère de type ZF (Zero Forcing) ou MMSE. La méthode d'implémentation de l'égaliseur est détaillée dans la section B de l'Annexe.

Si le récepteur est synchronisé avec le signal émis, le signal après égalisation et sous-échantillonnage est correctement approximé par l'équation suivante :

$$J(k) = \sum_{l=-L}^{L} R(kE-l)F_k(l) \approx e^{j\theta}V(k) + w_{eq}(k) \tag{4.37}$$

où $w_{eq}(k)$ est le bruit après filtrage.

Si le récepteur n'est pas synchronisé avec le signal émis, la sortie de l'égaliseur peut être considéré comme une séquence de bruit aléatoire. En effet, l'égalisation va induire une multiplication du signal reçu avec les séquences c_I et c_Q, et si la synchronisation n'est pas correcte, cette opération va embrouiller le signal reçu. Ceci assure aussi une robustesse vis-à-vis des trajets multiples. Si un trajet est retardé de plus d'un chip, le signal qu'il transporte va être embrouillé par l'opération d'égalisation. Cela constituera une source additionnelle de bruit et va donc dégrader les performances, mais l'algorithme restera opérationnel.

La seconde étape consiste à supprimer l'impact de la séquence W_m et de la phase θ. Étant donné que les chips de la séquence W_m sont répétés 4 fois, une multiplication différentielle est réalisée parmi les échantillons contenus dans l'intervalle des 4 chips répétés. Par exemple, une multiplication différentielle entre les 2 premiers chips va donner le résultat suivant :

$$\begin{aligned} J(4k+1)J(4k)^* &= V(4k+1)V(4k) + w_{diff}(k) \\ &= S_n(4k+1)S_n(4k) + w_{diff}(k) \end{aligned}$$

où $w_{diff}(k)$ contient tous les termes de bruit.

Il y a en tout 7 combinaisons de produits possibles :

$$\begin{aligned} D_{i,j}(k) &= J(4k+i+j)J(4k+i)^* \\ &= S_n(4k+i+j)S_n(4k+i) + w_{i,j}(k) \end{aligned} \tag{4.38}$$

$i = 0, 1, 2$ and $j = i+1, \cdots, 3$.

Définissons la séquence binaire $\tilde{s}_{i,j}(k) = s_n(4k+i+j) \oplus s_n(4k+i)$. Il apparaît que $D_{i,j}$ est une observation bruitée de la séquence $\tilde{s}_{i,j}$ modulée en BPSK. De plus, si on applique 2 fois la propriété de décimation des m-séquences et leur propriété d'addition, on trouve que $\tilde{s}_{i,j}$ est une version décalée de la séquence s_n (cf. section 1.4). Par conséquent, $\tilde{s}_{i,j}$ possède le même polynôme caractéristique que s_n : $g_s(x)$. Il est ainsi possible d'estimer l'état initial de la séquence $\tilde{s}_{i,j}$ à l'aide d'un algorithme de décodage par passage de messages. Sa matrice de parité est construite, comme pour le système WCDMA, à partir de plusieurs équation de parité issues du polynôme caractéristique $g_s(x)$. Ce décodeur est modélisé par une fonction $dec(.)$ définie par les Eq. 4.22 et 2.20.

Le principe de l'algorithme est donc de décoder l'état initial de la séquence $\tilde{s}_{i,j}$, puis de retrouver celui de la séquence s_n grâce à une matrice de transposition des états. On retrouve les mêmes idées que celles qui ont été développées pour l'algorithme de détection du code d'embrouillage du système WCDMA. La différence réside dans le pré-traitement précédant le décodage.

4.3.2.2 Détermination de l'état initial de la séquence s_n

A partir du signal $D_{i,j}(k)$ obtenu après pré-traitements du signal reçu, le décodeur produit un vecteur de r bits noté $U_{\tilde{s}}$. Il représente l'état initial de la séquence $\tilde{s}_{i,j}$ au début de la trame. On souhaite trouver l'état initial de la séquence s_n : U_{s_n}. Cette tâche est réalisée en 2 étapes, similaires à celles qui ont été définies à la section 4.2.2.3 pour le système WCDMA.

La première étape consiste à trouver l'état initial de la séquence s_{decim}, noté $U_{s_{\text{decim}}}$, qui donne $\tilde{s}_{i,j}$ par décimation d'un facteur 4 :

$$\tilde{s}_{i,j}(k) = s_{\text{decim}}(4k)$$

Il existe une matrice B qui relie les états initiaux d'une m-séquence et de sa version décimée d'un facteur 2 [81]. Ceci est décrit dans la section A.4 de l'Annexe. En appliquant 2 fois cette opération, on obtient :

$$U_{s_{\text{decim}}} = B^2 U_{\tilde{s}}$$

La deuxième étape consiste à trouver U_{s_n} à partir de $U_{s_{\text{decim}}}$. On exploite la propriété d'addition entre les m-séquences s_{decim} et s_n :

$$s_{\text{decim}}(k) = s_n(k+i+j) \oplus s_n(k+i)$$

Soit G la matrice de transposition des états entre les instant k et $k+1$, dans la représentation de Fibonacci. Elle est définie par l'Eq. A.3 de la section A.2. On obtient :

$$U_{s_n} = (G^{i+j} + G^i)^{-1} U_{s_{\text{decim}}}$$

Les 2 étapes sont finalement regroupées dans une seule matrice de transposition T :

$$\begin{aligned} U_{s_n} &= T U_{\tilde{s}} \\ T &= (G^{i+j} + G^i)^{-1} B^2 \end{aligned} \tag{4.39}$$

Il est important de noter que la matrice T est calculée une seule fois, puis mémorisée.

4.3.3 Synthèse de l'algorithme

Le pseudo-code de l'algorithme est présenté ci-dessous. Contrairement au système WCDMA, il ne nécessite pas l'implémentation d'une étape de vérification. En effet, le poids des équations de parité utilisées par le décodeur est élevé, ce qui élimine les fausses alarmes.

Algorithm 5 Algorithme de détection du code d'embrouillage

$q = 0$
Décision = ÉCHEC
while Décision == ÉCHEC **do**
  Calcule $D_{i,j}(0), D_{i,j}(1), \cdots, D_{i,j}(M-1)$ d'après les Eq. 4.37 et 4.38
  $\{I_c, \hat{U}_{\tilde{s}}\} = dec(D_{i,j}(0), D_{i,j}(1), \cdots, D_{i,j}(M-1))$
  if $I_c = 1$ **then**
    Détermine l'état initial de la séquence s_n d'après l'Eq. 4.39
    Décision = SUCCÈS
  end if
  $q \leftarrow q + 1$
end while

4.3.4 Evaluation des performances

On utilise les 2 hypothèses de synchronisation H_0 et H_1, telles qu'elles sont définies à la section 4.2.2.2. Les performances de l'algorithme de détection sont mesurées par les probabilités de détection correcte P_{CD}, fausse alarme P_{FA} et d'erreur de détection P_e :

$$\begin{aligned} P_{CD} &= P(Ic = 1 \text{ and } \hat{U}_{s_n} = U_{s_n} | H_1) \\ P_{FA} &= P(Ic = 1 | H_0) \\ P_e &= 1 - P_{CD} \end{aligned} \tag{4.40}$$

$\hat{U}_{s_n}$ est l'estimation de l'état initial de la séquence s_n, donnée par l'algorithme de détection.

Les performances ont été mesurées d'après les configurations de simulation suivantes :

- Quand on mesure P_{CD}, le récepteur est synchronisé avec le début de la trame (i.e. l'hypothèse H_1 est satisfaite), alors que le récepteur n'est pas synchronisé lorsqu'on mesure P_{FA}.
- Le décodeur implémente un algorithme de type Min-Sum [54]. Il s'arrête si toutes les équations de parité sont satisfaites ou bien si le nombre maximal d'itérations N_{iter} est atteint.
- Le nombre de variables à l'entrée du décodeur est fixé à $M = 1500$ chips. Cette valeur a été sélectionnée afin que M soit inférieur à la taille d'un *Power Control Group* (PCG) qui contient 1536 chips [4]. Ceci assure que l'algorithme fonctionne indépendamment du motif de poinçonnage du mécanisme de contrôle de puissance.
- Le canal est supposé additif blanc et gaussien (AWGN).

Des simulations sur 10^7 tirages n'ont pas permis d'observer la moindre fausse alarme. Ceci est dû à la présence de très nombreux ensembles absorbants dans la matrice de parité (cf. section 3.4.1). En effet, le poids du polynôme $g_s(x)$ vaut 19, ce qui est extrêmement élevé. Les K équations de parité générées à partir des polynômes $g_l(x) = g_s(x^{2^l})$ ne sont pas suffisantes pour éliminer tous les ensembles absorbants. Les ensembles restants empêchent le décodeur de converger lorsqu'on l'alimente uniquement avec du bruit.

La Fig. 4.14 montre l'influence du choix du type d'égaliseur sur la probabilité d'erreur de détection P_e. Le nombre d'équations de parité est fixé à $n_{RGM} = 6$ et $N_{iter} = 100$. L'égaliseur MMSE offre un gain de l'ordre de 1 dB vis-à-vis du ZF et de 2 dB par rapport au filtre adapté (noté MRC, pour "Maximum Ratio Combining"). Ce dernier présente néanmoins un intérêt car il est beaucoup plus simple à mettre en œuvre.

La Fig. 4.15 montre l'influence du nombre d'équations de parité K sur la probabilité de détection ratée. Il apparaît qu'il n'est pas nécessaire d'utiliser plus de $K = 4$ équations.

On observe que l'algorithme est opérationnel pour un RSB assez élevé, de l'ordre de 2.5 dB. Ceci permet de détecter des interféreurs très puissants, mais certains ne pourront être détectés bien qu'ils vont brouiller la station de base. Ceci est dû au poids du polynôme caractéristique. Il faudrait employer des équations de parité ayant un poids plus faible. Si on applique le modèle de l'Eq. 3.15 pour les m-séquences, avec $N_3 \approx 2^{r-1}$ le nombre d'équations de parité de poids $t = 3$, on obtient : $m_0 \approx 2^{(r+1)/2}$. Pour $r = 42$, le degré minimal des équations de parité de poids 3 est donc très largement supérieur au nombre de chips alimentant le décodeur ($M = 1500$). Il est par conséquent peu probable de trouver des équations de parité de poids 3 qui puissent être utilisées par l'algorithme de détection. Il faudrait mener une étude supplémentaire afin de trouver les équations de poids 4 ou 5. On pourra s'inspirer des travaux menés dans le domaine de la cryptographie pour mettre au point des attaques sur les chiffreur par flot [23][26].

4.3.5 Conclusion

Nous avons proposé un algorithme de détection du code d'embrouillage de la liaison montante du système CDMA2000. Il est dédié au canaux de type RC 1 et 2 qui appliquent une modulation orthogonale à 64 états. L'algorithme obtenu présente une probabilité de fausse alarme au moins inférieure à 10^{-6}. Pour un canal AWGN, la probabilité de détection est bonne si le RSB est supérieur à 2.5 et que l'égaliseur applique un critère MMSE. Si on utilise un filtre adapté, on observe une dégradation des performances de l'ordre de 2 dB. Ceci est à mettre en balance avec une complexité d'implémentation qui est alors nettement réduite.

4.4 Conclusion

Nous avons proposé deux algorithmes de détection du code d'embrouillage pour les systèmes WCDMA et CDMA2000. Ils exploitent les propriétés des m-séquences et des séquences de Gold, ainsi que les mécanismes de décodage décrit au chapitre 2. Ces algorithmes montrent les vulnérabilités des transmissions par étalement de spectre. En particulier, les propriétés d'addition et de décimation des m-séquences sont exploitées pour trouver des pré-traitements qui permettent ensuite de décoder les séquence d'embrouillage. Ce type de traitements peut être envisagé pour de nombreux systèmes existants (e.g. GPS, Galileo, CCSDS). Ce travail a donné lieu à 2 publications et 2 brevets :

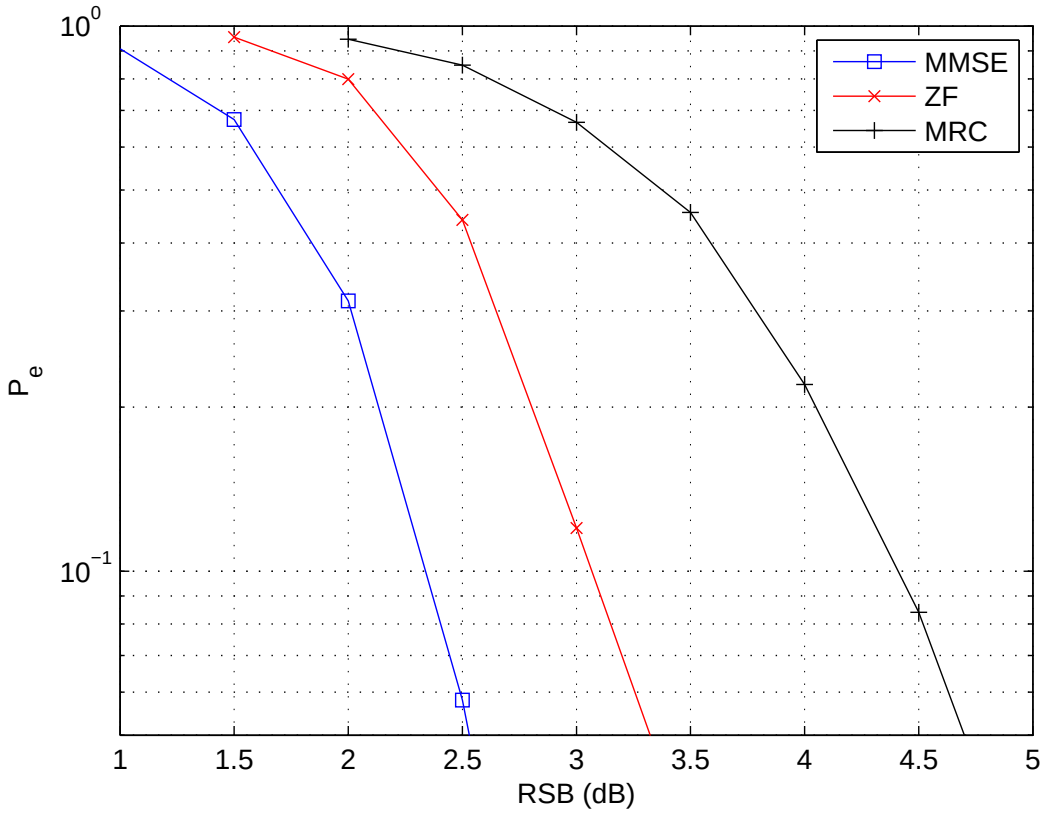

FIGURE 4.14 – Influence de l'égaliseur - canal gaussien - $N_{iter} = 100$, $M = 1500$ et $n_{\mathrm{RGM}} = 6$

- 'Blind Identification of the Uplink Scrambling Code Index of a WCDMA Transmission and Application to Femtocell Networks', ICC, Budapest, Juin 2013 [84]
- 'Blind Identification of the Scrambling Code of a Reverse Link CDMA 2000 Transmission', ICC, Budapest, Juin 2013 [85].
- 'Méthode d'estimation aveugle d'un code d'embrouillage d'une liaison montante WCDMA', E.N. 1252338.
- ' Méthode d'estimation aveugle d'un code d'embrouillage d'une liaison montante CDMA 2000', E.N. 1252340.

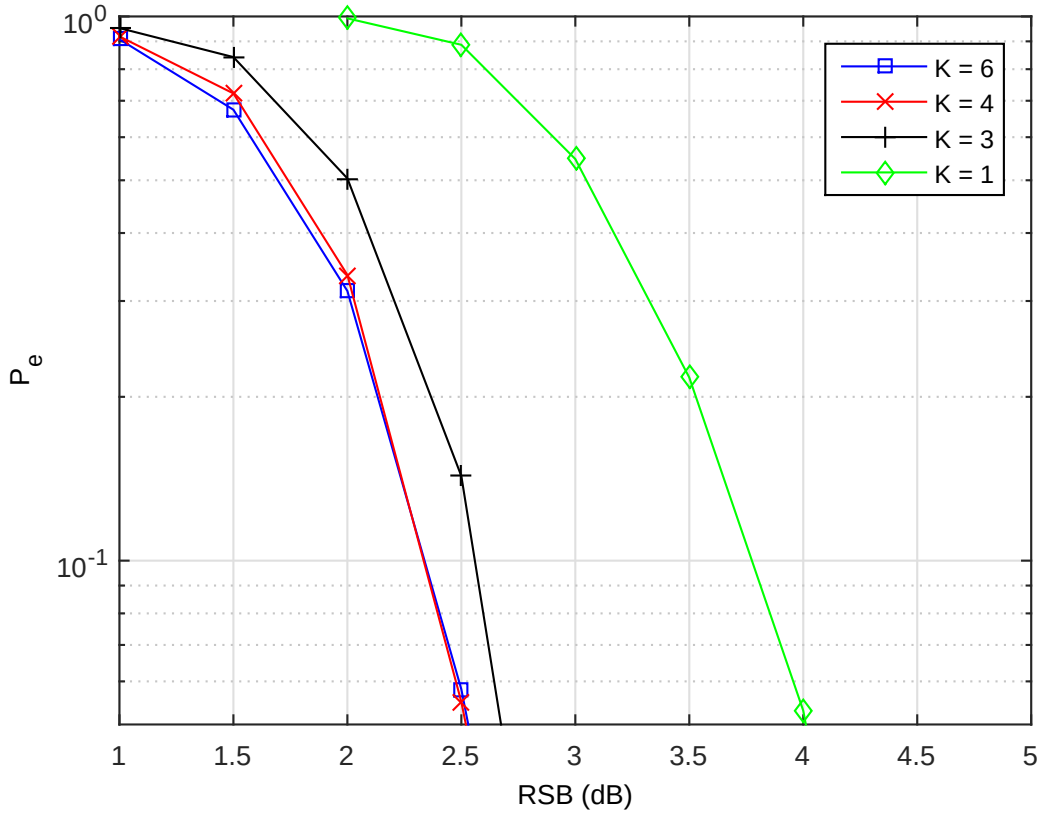

FIGURE 4.15 – Influence de K - canal gaussien - égaliseur MMSE, $N_{iter} = 100$ et $M = 1500$

Conclusion et perspectives

Cette thèse aborde le décodage des séquences pseudo-aléatoires. Cette technique permet de détecter des séquences longues (e.g. de période 2^{42}), contrairement aux méthodes par corrélation qui sont trop complexes à implémenter. Cela nécessite néanmoins que le récepteur connaisse au préalable le polynôme caractéristique de la séquence.

Nous avons montré que le décodage d'une séquence pseudo-aléatoire est une problématique du type 'détecte et décode'. Le récepteur détecte la présence de la séquence et simultanément estime son état initial. Ceci correspond dans la théorie classique de la détection à un détecteur de type GLRT qui ne connaît pas la séquence émise, mais qui connaît sa méthode de construction (i.e. son polynôme caractéristique). L'algorithme implémente alors un GLRT qui utilise un décodeur pour estimer la séquence reçue. Le générateur de la séquence correspond à celui d'un code linéaire, les séquences sont donc des mots de code qui peuvent être estimés avec un algorithme approprié. Le décodeur peut appliquer un critère MLSE ou MAP. Le décodeur MLSE exact est implémenté par un test de corrélation lorsque la séquence est d'une taille 'raisonnable' ou un algorithme de Viterbi. Ce dernier s'avère impossible à réaliser pour les séquences employées habituellement, le nombre d'états du décodeur est très élevé et la complexité devient prohibitive. On applique plutôt un décodage par passage de message qui permet d'obtenir une bonne approximation du décodeur MLSE ou MAP. L'approximation du décodeur MLSE est obtenue avec l'algorithme Min-Sum, alors que l'algorithme Sum-Product fournit celle du décodeur MAP.

Les équations de parité sont un constituant indispensable du décodeur. Elles appartiennent au code dual de la séquence que l'on cherche à décoder. Le code dual d'un m-séquence est le code de Hamming construit avec le polynôme caractéristique de la séquence. Le nombre de mots de poids t du code de Hamming est connu, par conséquent le nombre d'équations de parité de poids t d'une m-séquence est lui aussi connu. Pour les séquences de Gold, Kasami a dénombré le nombre d'équations pour $t = 3$ et 4. Nous avons calculé le nombre d'équations de parité de poids $t = 5$ lorsque le degré du polynôme caractéristique r est impair. Ce calcul est important car il n'y a pas d'équations de parité de poids $t < 5$ lorsque r est impair. Le nombre d'équations de parité est aussi utilisé pour estimer le degré minimal des équations d'un poids t donné. Cette information peut être exploitée par l'algorithme de recherche des équations pour minimiser le nombre de calculs. Cela permet aussi de déterminer rapidement la taille du vecteur d'observation nécessaire pour le décodeur. Nous avons montré que le modèle de prédiction estime correctement la valeur moyenne du degré minimal de l'ensemble des séquences de Gold. Nous avons néanmoins mis en évidence une grande variabilité du degré minimal des séquences autour de cette valeur moyenne.

Nous avons ensuite identifié les ensembles absorbants de plus petite taille lorsque le décodeur emploie plusieurs polynômes de parité. Nous avons aussi montré que des cycles 'transverses' peuvent détruire ces ensembles absorbants, ce qui génère des fausses alarmes. Cette observation a servi d'élément de départ pour proposer un algorithme de sélection des polynômes de parité.

Ce dernier minimise le nombre de cycles transverses de longueur 6 et 8, ce qui minimise la probabilité de fausse alarme lorsque le poids des équations de parité vaut $t = 3$. Ce travail a nécessité de calculer le nombre de cycles de longueur 6 et 8 pour les matrices de parité concaténant plusieurs matrices de référence. Ce calcul a été mené en utilisant le modèle proposé par Chugg et Halford pour énumérer le nombre de cycles courts dans un graphe de décodage. Les résultats de simulation corroborent notre hypothèse sur l'impact des cycles transverses et l'algorithme permet de sélectionner les équations de parité qui minimisent la probabilité de fausse alarme. Le temps d'acquisition d'une séquence de Gold est ainsi très notablement réduit. Il reste néanmoins que notre hypothèse n'est pas démontrée formellement.

Nous avons enfin proposé deux algorithmes de détection du code d'embrouillage pour les systèmes WCDMA et CDMA2000. Ils exploitent les propriétés des m-séquences constituant les séquences de Gold, ainsi que les mécanismes de décodage par passage de messages. Ces algorithmes montrent les vulnérabilités des transmissions par étalement de spectre. En particulier, les propriétés d'addition et de décimation des m-séquences sont exploitées pour trouver des pré-traitements qui permettent ensuite de décoder les séquences d'embrouillage. Ce type de traitements peut être envisager pour de nombreux systèmes existants (e.g. GPS, Galileo, CCSDS). De nombreux systèmes à étalement de spectre, que l'on croyait difficilement détectables par des techniques de corrélation, deviennent vulnérables à une détection par décodage.

Ce travail de thèse ouvre de nouvelles pistes d'études pour le futur. Les m-séquences et les séquences de Gold sont des codes linéaires cycliques. Il est donc envisageable d'étendre le travail mené au chapitre 3 à d'autres codes cycliques. Si on suppose que le code est présent, les équations 3.30 et 3.31 peuvent être utilisées pour choisir les équations qui maximisent le nombre de cycles transverses et ainsi maximiser la probabilité de détection. Cela pourrait constituer une piste d'étude pour l'amélioration des performances des décodeurs itératifs pour les codes linéaires [86][87][59][46]. Ceci ouvrirait aussi l'opportunité d'estimer les séquences de saut de fréquence générées par des codes linéaires. Plus généralement, la détection et le décodage conjoint d'autres types de code est une autre piste à creuser. Les codes convolutif, les turbo-codes et les codes LDPC sont de bons candidats pour de futures études.

Le modèle d'estimation de la valeur moyenne du degré minimal des équations de parité de poids t ne rend pas compte correctement de la variabilité entre les séquences. Une modélisation de la variance est nécessaire. Ceci permettra par exemple d'affiner la recherche des équations de parité.

Lors de l'évaluation des performances de l'algorithme de détection du code d'embrouillage du système WCDMA, nous avons observé que l'implémentation du décodeur MS en dynamique finie diminuait significativement la probabilité de fausse alarme. Ceci s'explique par la distribution des messages émis par les variables en fonction de l'hypothèse H_0 ou $H1$. Lorsque le signal est absent (hypothèse H_0) et que le décodeur réalise une fausse alarme, les messages prennent des valeurs très élevées, typiquement supérieur à 10^{12}, alors que dans l'hypothèse H_1 ils restent à un niveau plus 'raisonnable', de l'ordre de 10^5. Cet écart entre les deux distributions peut être mis à profit pour limiter les fausses alarmes.

Dans l'exemple applicatif du chapitre 3, nous avons supposé que le récepteur observe direc-

tement la séquence de Gold que l'on souhaite détecter. En pratique, le signal sera altéré par l'écart de fréquence entre l'émetteur et le récepteur. Le signal sera donc une observation de la séquence ayant subi une rotation de phase s'incrémentant avec l'indice du chip de la séquence. Les algorithmes de décodage et synchronisation conjoints peuvent être implémentés pour décoder et estimer conjointement les paramètres de synchronisation [43]. Nous avons implémenté l'algorithme proposé par Herzet et al [45] pour décoder la séquence de Gold du système GPS en présence d'un offset de fréquence. La figure 4.16 montre qu'il est capable d'estimer cet offset et décoder convenablement jusqu'à un offset $\Delta f = 200$ Hz. D'autres algorithmes, [46] par exemple, pourraient être évalués.

Enfin, ces études pourraient être étendues à d'autres séquences LFSR. Les séquences de Kasami ressemblent aux séquences de Gold dans leur mode de construction. Le facteur de décimation entre les 2 m-séquences formant la paire préférentielle est différent. Le calcul du nombre d'équation de parité de poids t peut être facilement étendu aux séquences de Kasami. On pourrait aussi envisager le cas des m-séquences non-binaires qui pourraient être décodées avec des décodeurs non-binaires eux aussi.

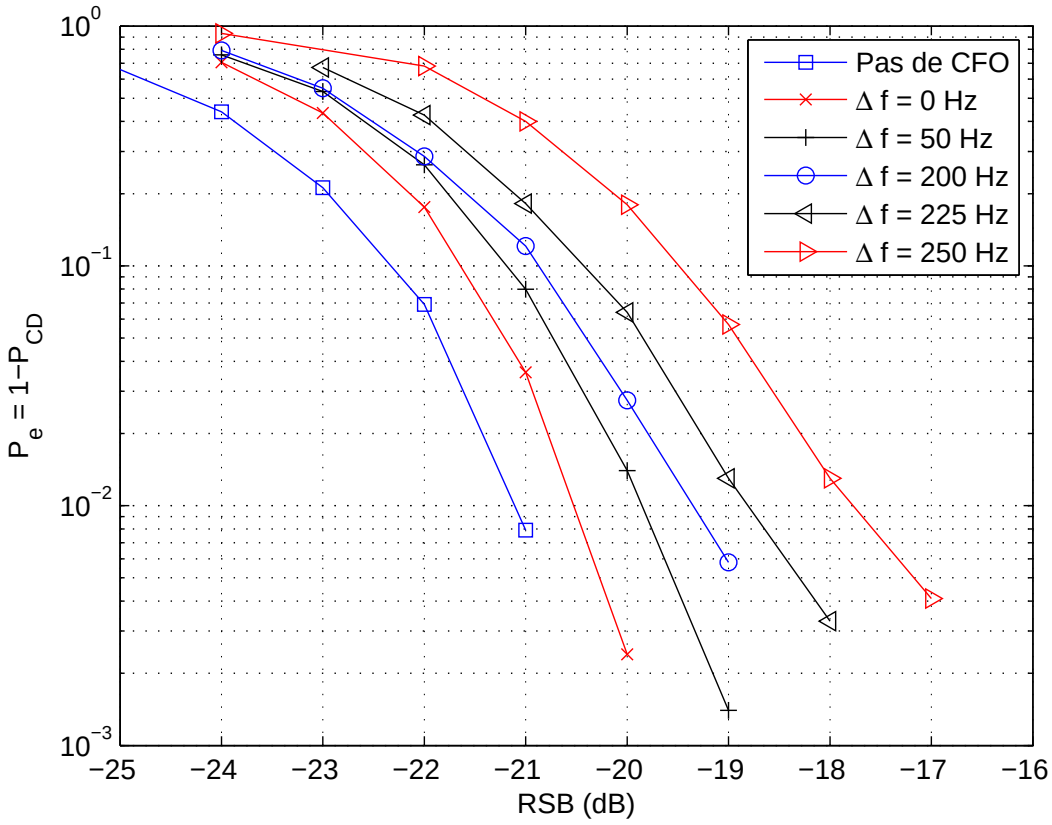

FIGURE 4.16 – Estimation de l'offset de fréquence et décodage

Annexe A

Générations des M-séquences décalées ou décimées

A.1 Relation entre les états initiaux de registres dans les représentation de Galois et Fibonacci

Dans la représentation de Galois, la séquence LFSR y est modélisée par la division polynomiale :

$$y(x) = \frac{u(x)}{g(x)} = y(0) + y(1)x + y(2)x^2 + \cdots + y(N-1)x^{N-1} \tag{A.1}$$

$u(x)$ est la représentation polynomiale de l'état initial des r registres :

$$u(x) = u(0) + u(1)x + \cdots + u(r-1)x^{r-1}$$

$g(x)$ est le polynôme caractéristique de la séquence :

$$g(x) = \sum_{k=0}^{r} g_k x^k$$

Il faut noter que la modélisation de l'Eq. (A.1) n'est pas valable pour la représentation de Fibonacci.

Le lien entre l'état des registres pour les 2 représentations est donné par la relation suivante [1] :

$$u_{\text{Galois}}(k) = \sum_{i=0}^{k} u_{\text{Fibonacci}}(i) g_{k-i} \quad k = 0, \ldots, r-1$$

Définissons les vecteurs $U_{\text{Fibonacci}}$ et U_{Galois} qui contiennent les coefficients des registres. Ils sont reliés par la relation matricielle $U_{\text{Galois}} = T_{\text{FibToGalois}} U_{\text{Fibonacci}}$, où la matrice de transposition des états $T_{\text{FibToGalois}}$ est construite de la manière suivante :

$$T_{\text{FibToGalois}} = \begin{cases} g_{i-j} & \text{si } j \leq i \\ 0 & \text{sinon} \end{cases}$$

La matrice de transposition qui réalise l'opération inverse

$$U_{\text{Fibonacci}} = T_{\text{GaloisToFib}} U_{\text{Galois}}$$

est obtenue par inversion de $T_{\text{FibToGalois}}$:

$$T_{\text{GaloisToFib}} = T_{\text{FibToGalois}}^{-1}$$

A.2 Relation entre les états d'une m-séquences retardée et sa version originale

L'objectif est de calculer l'état des registres de la séquence $y(k+i)$ sachant qu'on connaît l'état des registres de la séquence $y(k)$. Soit $U_y(k) = (u_y(0), \ldots, u_y(r))^T$ le vecteur contenant l'état des registres de la séquence y à l'instant k. Il existe une matrice de transition G entre les instant k et $k+1$ [1] :

$$U_y(k+1) = G U_y(k)$$

Dans la représentation de Galois ($g_0 = g_r = 1$) :

$$G = \begin{bmatrix} g_1 & 1 & 0 & 0 & \cdots & 0 \\ g_2 & 0 & 1 & 0 & \cdots & 0 \\ g_3 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & 0 & \cdots & \vdots \\ g_{r-1} & 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{A.2}$$

Dans la représentation de Fibonacci :

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & 0 & \cdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ 1 & g_{r-1} & g_{r-2} & g_{r-3} & \cdots & g_1 \end{bmatrix} \tag{A.3}$$

On a alors la relation :

$$U_y(k+i) = G^i U_y(k)$$

A.3 Génération d'une m-séquences retardée

Dans la section précédente, nous avons montré comment trouver l'état des registres d'une séquence décalée par rapport à la séquence d'origine. Dans cette section, nous allons présenter une méthode permettant de générer simultanément les séquences $y(k)$ et $y(k+i)$. Nous allons pour cela nous placer dans la représentation de Fibonacci 1.1. On note $u_i(k)$ l'état du $k^{\text{ième}}$ registre. On remarque tout d'abord que pour $i < r$: $y(k+i) = u_i(k)$. En utilisant la fonction trace (cf. section 1.3), la m-séquence $y(k)$ est générée de la manière suivante :

$$y(k) = \mathrm{Tr}(\alpha^k \theta)$$

où α est la racine primitive du polynôme caractéristique de la séquence, $g_y(x)$, dans $\mathrm{GF}(2^r)$ et θ est un élément non nul de $\mathrm{GF}(2^r)$ qui dépend de l'état initial des registres.

On a alors :

$$y(k+i) = \mathrm{Tr}(\alpha^{k+i}\theta) = \mathrm{Tr}(\alpha^k \alpha^i \theta)$$

Sachant que α^i se décompose sur la base $1, \alpha, \alpha^2, \ldots, \alpha^{r-1}$, il existe r coefficients binaires c_i tels que :

$$\alpha^i = \sum_{l=0}^{r-1} c_i(l)\alpha^l$$

Par conséquent :

$$y(k+i) = \sum_{l=0}^{r-1} c_i(l)\mathrm{Tr}(\alpha^k \alpha^l \theta) = \sum_{l=0}^{r-1} c_i(l) y(k+l) = \sum_{l=0}^{r-1} c_i(l) u_l(k)$$

Cette propriété est utilisée dans les systèmes WCDMA, GPS et Galileo pour générer des séquences de Gold.

A.4 Décimation d'une m-séquence

Soit une m-séquence $s(k)$ qui est décimée d'un facteur 2 : $y(k) = s(2k)$. D'après la propriété de décimation des m-séquences, y est une version décalée de la m-séquence s. Les deux séquences possèdent donc le même polynôme caractéristique $g(x)$, mais un état initial différent : $\exists\, \tau > 0$ tel que $y(k) = s(k+\tau)$. On cherche donc à trouver la relation entre les états initiaux des séquences s et y. On cherche une relation matricielle du type $U_y = BA_s$, où U_s est le vecteur contenant l'état des registres de la séquence d'origine et U_y le vecteur contenant l'état des registres de la séquence décimée. La méthode de construction de la matrice B est détaillée dans [81].

Soit α^{-1} la racine du polynôme réciproque du polynôme caractéristique de la séquence s ($g_{inv}(x^{-1}) = x^{-r}g(x)$). Soit B la matrice binaire de dimension $r \times r$, dont la $i^{\text{ième}}$ ligne est

donnée par les coefficients binaires $B_{i,k}$ de la décomposition de

$$\alpha^{-ip} = \sum_{k=0}^{r} B_{i,k}\alpha^{-k}$$

p est obtenu par la relation de congruence $2p = 1 \bmod (2^r - 1)$. Alors, $U_y = BU_s$.

On trouve tout d'abord facilement la valeur $p = 2^{r-1}$. On trouve ensuite les relations suivantes :

$$\begin{aligned} \alpha^{-2ip} &= \alpha^{-i} \\ \alpha^{-(2i+1)p} &= \alpha^{-p-i} \end{aligned}$$

Ce qui permet de calculer facilement les lignes paires de la matrice B :

$$B_{2i,k} = \begin{cases} 1 & k = i \\ 0 & k \neq i \end{cases}$$

Pour déterminer les lignes impaires, il faut calculer le reste de la division polynômiale de α^{-p-i} par le polynôme inverse $g_{inv}(x^{-1})$

Annexe B

Égaliseur du filtre $G_k(t)$

L'égaliseur décrit ici a pour but d'éliminer l'interférence entre chips causée par le filtre de mise en forme, mais n'agit pas sur l'interférence générée par un canal à trajets multiples. On se place donc dans un formalisme du canal AWGN. On suppose que l'interférence inter-chip se limite à un intervalle de 5 chips de part et d'autre du chip $V(k)$ qu'on souhaite égaliser. Le filtre de mise en forme contient 4 échantillons par chip et sa réponse impulsionnelle contient $W = 49$ coefficients[4].

Soit $R_k = [R(4k-20), \cdots, R(4k), \cdots, R(4k+20+W-1)]$ le vecteur contenant le signal reçu échantillonné à la fréquence $F_e = 4/T_c$. Il peut s'écrire de la manière suivante :

$$R_k = (A_1 I + jA_2 Q)V_k + n$$

où n est le vecteur contenant les échantillons de bruit, de variance σ^2. $V_k = [V(k-5), \cdots, V(k), \cdots, V(k+5)]$ est le vecteur des 11 chips centrés sur $V(k)$ (cf.Eq. 4.34). I et Q sont des matrices diagonales contenant les chips des séquences c_I et c_Q. A_1 et A_2 sont les matrices modélisant la mise en forme des voies en phase et en quadrature (élévation de cadence et filtrage). Elles contiennent 11 colonnes et $40 + W$ lignes. Elles sont modélisées de la manière suivante :

$$\begin{array}{ll} A_1(4j+i, j) = p(i) & i = 0, \cdots W-1; j = 0, \cdots, 10 \text{ et si } 4j+i < 40+W \\ A_2(4j+i+2, j) = p(i) & i = 0, \cdots W-1; j = 0, \cdots, 10 \text{ et si } 4j+i+2 < 40+W \end{array} \quad \text{(B.1)}$$

Chaque colonne est déduite de la précédente par un décalage de 2 éléments vers le bas. Les deux premières lignes de la matrice A_2 sont nulles afin de modéliser le décalage de $T_c/2$. On note $A = (A_1 I + jA_2 Q)$. L'égaliseur MMSE $F_k = [F_k(-5) \cdots F_k(0) \cdots F_k(5)]$ est obtenu par l'opération suivante : $F_k = e_k G$ où e_k est le vecteur ligne ayant un '1' en $k^{\text{ième}}$ position et des 0 ailleurs et :

$$G = (A^H A + \sigma^2 I)^{-1} A^H$$

L'égaliseur ZF est obtenu en appliquant $\sigma^2 = 0$. Le filtre adapté est obtenu en supprimant le terme $A^H A$ ($G = A$)

Annexe C

Calcul de I_6 et I_8

C.1 Calcul de $tr(L_{mod})$

En utilisant la propriété de commutativité de la trace ($tr(AB) = tr(BA)$) et en développant les termes en $Z(X)$, l'Eq. 3.29 peut être réarrangée de la manière suivante :

$$\begin{aligned} I_6 &= \frac{1}{6} tr(L_{mod}) \\ L_{mod} &= A^3 - \tilde{A}A^2 - 2\tilde{B}_m B^2 - \tilde{A}_m A^2 + (A\tilde{A}_m \circ I)A + (B\tilde{B}_m \circ I)B + \tilde{A}_m E \tilde{B}_m E^T \end{aligned} \tag{C.1}$$

$\tilde{A}$ est la matrice formée des éléments diagonaux de A. Par conséquent, on a $\tilde{A} = tI$ et $\tilde{A}_m = (t-1)I$, où I est la matrice identité de même taille que A. De la même manière, $\tilde{B} = KtI$ et $\tilde{B}_m = (Kt-1)I_M$. De plus, en appliquant la propriété de commutativité de la trace, on a $tr(A) = tr(EE^T) = tr(E^T E) = tr(B) = KNt$ et $(tr(B^2) = tr(A^2)$. En insérant tous ces résultats dans l'Eq. C.1, on obtient :

$$tr(L_{mod}) = tr\left(A^3\right) - (2(K+1)t-3)tr\left(A^2\right) + KNt\left((K^2+K+1)t^2 - 2(K+1)t + 1\right) \tag{C.2}$$

Le terme $tr\left(A^2\right)$ peut à son tour être calculé en exploitant la circularité des sous-matrices E_a. Soit la matrice $F_{ab} = E_a E_b^T$, $tr\left(A^2\right)$ peut s'écrire sous la forme suivante :

$$tr\left(A^2\right) = \sum_{i=0}^{K-1}\sum_{j=0}^{K-1}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F_{ij}(u,v)^2 \tag{C.3}$$

Puisque E_a et E_b sont circulante, c'est aussi le cas de F_{ab}. Par conséquent :

$$S_{i,j} = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F_{ij}(u,v)^2 = N\sum_{v=0}^{N-1} F_{ij}(0,v)^2 \tag{C.4}$$

De plus, on a $F_{ba} = F_{ab}^T$, et donc $S_{j,i} = S_{i,j}$. Cette propriété se répercute dans le calcul de $tr\left(A^2\right)$ de la manière suivante :

$$tr\left(A^2\right) = \sum_{i=0}^{K-1} S_{i,i} + \sum_{i=0}^{K-1}\left(\sum_{j=0}^{i-1} S_{i,j} + \sum_{j=i+1}^{K-1} S_{i,j}\right) \tag{C.5}$$

Si $i \neq j$, $F_{ij}(0,v)$ vaut soit 0 soit 1. Il y a au total t^2 éléments valant 1 lorsque $i \neq j$. Si $i = j$, $F_{ii}(0,0) = t$ et $F_{ii}(0,v)$ vaut soit 0 soit 1 pour $v > 0$. Il y a au total $t^2 - t$ éléments valant 1 et un éléments valant t^2. Ceci est expliqué dans la section C.3. On a donc :

$$\begin{array}{l} S_{i,j} = Nt^2 \ \text{ si } i \neq j \\ S_{i,i} = N(2t^2 - t) \ \text{ sinon} \end{array} \tag{C.6}$$

En insérant l'Eq.C.6 dans l'Eq. C.5, on obtient finalement :

$$tr\left(A^2\right) = NKt((K+1)t - 1) \tag{C.7}$$

Ce résultat est inséré dans l'Eq. C.2 pour donner l'Eq. 3.30.

C.2 Calcul de I_8

On applique ici aussi la méthode proposée par Halford et Chugg [66] pour l'évaluation des cycles de longueur 8 lorsque la circonférence du graphe (*girth*) vaut 6. On obtient la formule suivante :

$$\begin{array}{l} I_8 = \dfrac{1}{8} tr(L_8) \\ L_8 = P_5^U E^T A - L_{(0,6)}^U A - L_{(4,2)}^U A - L_{(1,6)}^U E^T - L_{(5,2)}^U E^T \end{array} \tag{C.8}$$

où $L_{(0,6)}^U$, $L_{(4,2)}^U$, $L_{(1,6)}^U$ et $L_{(5,2)}^U$ sont des matrices définies dans [66].

La trace de ces matrices peut être calculée d'une manière analogue à ce qui a été fait dans la section C.1 pour l'évaluation de I_6 :

$$\begin{array}{l} I_8 = \dfrac{1}{8}\left(tr(A^4) + \alpha_3 tr(A^3) + \alpha_2 tr(A^2) + \alpha_1 tr(A)\right) \\ \alpha_3 = 4(2 - Kt - t) \\ \alpha_2 = (5K^2 + 8K + 5)t^2 - 17(K+1)t + 15 \\ \alpha_1 = -(2K^3 + 3K^2 + 3K + 2)t^3 + 2(4K^2 + 5K + 4)t^2 - 10(K+1)t + 4 \end{array} \tag{C.9}$$

Le résultat final est donné par l'Eq. 3.31.

C.3 Calcul de F_{ab}

$F_{ab} = E_a E_b^T$ est une matrice circulante définie par les 2 polynômes $c_a(x) = \sum_{k=0}^{r_a} c_{a,k} x^k$ and $c_b(x) = \sum_{k=0}^{r_b} c_{b,k} x^k$. On suppose que $r_b > r_a$. Chaque élément de F_{ab} est défini par la première ligne de la matrice : $F_{ab}(i,j) = F_{ab}(0, \phi(j-i))$, où $\phi(u)$ est défini par l'Eq. 3.36.

Notre objectif est d'évaluer $F_{ab}(0,j)$ pour $0 \leq j \leq N-1$. Il y a deux cas à distinguer, en fonction de la somme $r_a + r_b$.

Si $r_a + r_b < N$:

$$F_{ab}(0,j) = \begin{cases} \sum_{k=j}^{r_a} c_{a,k} c_{b,k-j} & \text{si } 0 \geq j \geq r_a \\ \sum_{k=0}^{j-N+r_b} c_{a,k} c_{b,N+k-j} & \text{si } N - r_b \geq j \geq N + r_a - r_b \\ \sum_{k=0}^{r_a} c_{a,k} c_{b,N+k-j} & \text{si } N - r_b + r_a \geq j \geq N - 1 \end{cases} \tag{C.10}$$

Si $r_a + r_b \geq N$:

$$F_{ab}(0,j) = \begin{cases} \sum_{k=j}^{r_a} c_{a,k} c_{b,k-j} & \text{si } 0 \geq j \geq N - r_b - 1 \\ \sum_{k=0}^{j-N+r_b} c_{a,k} c_{b,N+k-j} + \sum_{k=0}^{r_a-j} c_{a,k+j} c_{b,k} & \text{si } N - r_b \geq j \geq r_a \\ \sum_{k=0}^{j-N+r_b} c_{a,k} c_{b,N+k-j} & \text{si } r_a + 1 \geq j \geq N + r_a - r_b - 1 \\ \sum_{k=0}^{r_a} c_{a,k} c_{b,N+k-j} & \text{si } N - r_b + r_a \geq j \geq N - 1 \end{cases} \tag{C.11}$$

Si $c_a(x)$ et $c_b(x)$ sont trinomiaux : $c_{a,0} = c_{a,r_a} = c_{a,i_a} = c_{b,0} = c_{b,r_b} = c_{b,i_b} = 1$ pour des coefficients i_a et i_b vérifiant $0 < i_a < r_a$ et $0 < i_b < r_b$. On suppose que la matrice de parité E ne contient pas de cycles de longueur 4 (cf. section 3.4.2). Par conséquent, si $a \neq b$, chaque somme dans les Eq. C.10 ou C.11 vaut 0 ou 1. Si $a = b$, on a la même propriété, excepté lorsque $j = 0$: $F_{aa}(0,0) = t$. Il est donc possible de déterminer l'ensemble des éléments non-nuls de $F_{ab}(0,j)$: $\Omega_{ab} = \{j \text{ tel que } F_{ab}(0,j) > 0 \text{ et } 0 \leq j < N\}$. En détaillant les cas possibles, on trouve :

$$\Omega_{ab} = \left\{ \begin{array}{l} j = 0, i_a, r_a, N - i_b, N - r_b, N - r_b + i_a, N - r_b + r_a \\ \text{si } i_a \geq i_b : j = i_a - i_b \text{ sinon } j = N + i_a - i_b \\ \text{si } r_a \geq i_b : j = r_a - i_b \text{ sinon } j = N + r_a - i_b \end{array} \right\} \tag{C.12}$$

Ω_{ab} contient $t^2 = 9$ indices si $a \neq b$, et $t^2 - t + 1 = 7$ si $a = b$.

Annexe D

Calcul du temps moyen d'acquisition

La Figure D.1 montre le diagramme d'état du mécanisme d'acquisition en série. Le récepteur possède $N+2$ états possible :

- ACQ : l'acquisition est correcte.
- FA : le récepteur est victime d'une fausse alarme.
- S_0 : le récepteur est synchronisé avec la séquence.
- S_j $(j = 1, \ldots, N-1)$: le récepteur n'est pas synchronisé avec la séquence.

On suppose que le récepteur n'a pas d'information a priori sur le début de la séquence. Par conséquent, les états S_j $(j = 1, \ldots, N-1)$ sont équiprobables, avec une probabilité $\pi_j = 1/N$.

Si le récepteur est dans l'état S_0 et que la décision est correcte, le récepteur passe dans l'état ACQ. Si la détection est erronée, il passe dans l'état S_1 après un temps de pénalité valant $\kappa_p T_c$. S'il ne détecte rien, il passe dans l'état S_1. Si le récepteur est dans l'état S_j et que la synchronisation est décidée, le récepteur passe dans l'état FA ('Fausse Alarme'). Après un temps de pénalité $\kappa_p T_c$ pour détecter l'absence de synchronisation, le récepteur passe dans l'état S_{j+1}.

Les fonctions de transfert $H_{NFA}(z)$, $H_{FA}(z)$, $H_P(z)$, $H_M(z)$ et $H_D(z)$ modélisent la transition entre les états, en supposant qu'on a un processus Markovien. La fonction de transfert entre les états i et j d'un processus Markovien est définie par :

$$p_{ij}(z) = \sum_{n=0}^{\infty} p_{ij}(n) z^n$$

où $p_{ij}(n)$ est la probabilité de passer de l'état i à l'état j en n coups d'horloge. Avec cette définition, on a :

$$\begin{aligned} H_{NFA}(z) &= (1 - P_{FA}) z^{T_c} \\ H_{FA}(z) &= P_{FA} z^{T_c} \\ H_D(z) &= P_{CD} z^{T_c} \\ H_M(z) &= P_{MD} z^{T_c} + P_{WD} z^{(\kappa_p+1)T_c} \\ H_P(z) &= z^{\kappa_p T_c} \end{aligned} \tag{D.1}$$

On définit $H_0(z)$ la fonction de transfert équivalente pour passer de l'état S_j à S_{j+1} :

$$\begin{aligned} H_0(z) &= H_{NFA}(z) + H_{FA}(z) H_P(z) \\ H_0(z) &= (1 - P_{FA}) z^{T_c} + P_{FA} z^{(\kappa_p+1)T_c} \end{aligned} \tag{D.2}$$

Le temps moyen d'acquisition et sa variance sont définis par :

$$\begin{aligned} E[T_{acq}] &= \frac{dP_{ACQ}(z)}{dz}\Big|_{z=1} \\ Var[T_{acq}] &= \left[\frac{d^2P_{ACQ}(z)}{dz^2} + \frac{dP_{ACQ}(z)}{dz} - \left(\frac{dP_{ACQ}(z)}{dz}\right)^2\right]_{z=1} \end{aligned} \tag{D.3}$$

$P_{ACQ}(z)$ est la fonction génératrice :

$$P_{ACQ}(z) = \sum_{n=0}^{\infty} p_{ACQ}(n) z^n$$

où $p_{ACQ}(n)$ est la probabilité de passer de l'un des état S_i à l'état ACQ en n coups d'horloge.

$P_{ACQ}(z)$ peut se réécrire de la manière suivante :

$$P_{ACQ}(z) = \sum_{j=1}^{N-1} \pi_j P_{j,ACQ}(z) \tag{D.4}$$

où $P_{j,ACQ}$ est la fonction de transfert entre l'état S_j et l'état ACQ, et π_j la probabilité d'être dans l'état S_j. Étant donné que $\pi_j = 1/N$, en s'aidant de la Figure (D.1), on obtient [70] :

$$P_{ACQ}(z) = \frac{1}{N} \frac{H_D(z)(1 - H_0(z)^N)}{(1 - H_M(z)H_0(z)^{N-1})(1 - H_0(z))} \tag{D.5}$$

Par conséquent, le temps moyen d'acquisition vaut :

$$E[T_{acq}] = \left(\frac{1 + \kappa_p P_{WD} + (1 + \kappa_p P_{FA})\left(\frac{N-1}{2}\right)(2 - P_{CD})}{P_{CD}}\right) T_c \tag{D.6}$$

où nous avons employé la relation $P_{CD} + P_{MD} + P_{WD} = 1$.

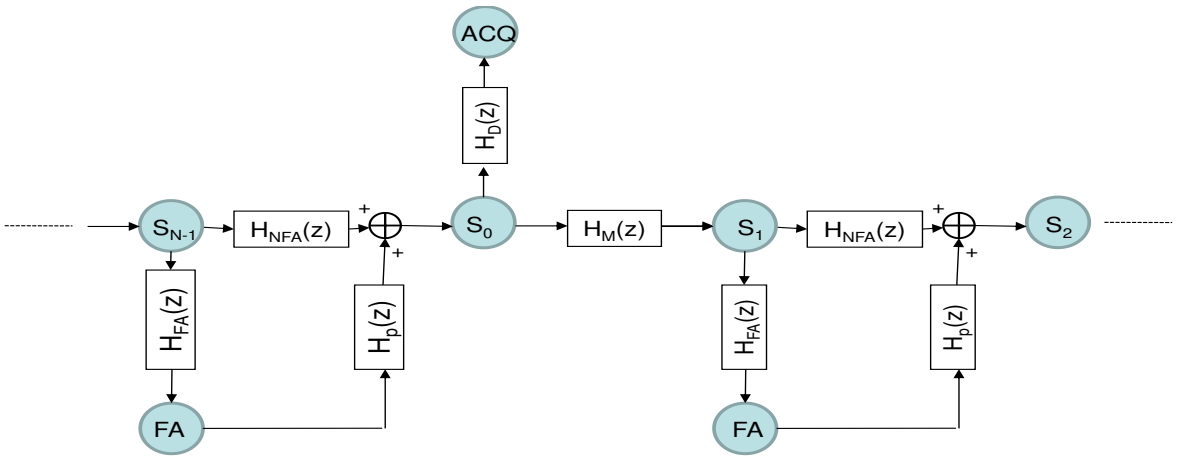

FIGURE D.1 – Diagramme d'état du mécanisme d'acquisition en série

Annexe E

Liste des équations de parité des séquences 4445 et $4005 - 4445$

Les équations de parité sont notées $c_l(x)$ et sont identifiées de la manière suivantes : $c_l(x) = 1 + x^k + x^j + x^i + x^m$ pour $t = 5$ $c_l(x) = 1 + x^i + x^m$ pour $t = 3$. La colonne 'Rang' indique le rang de la matrice $\mathbf{E}_a$ construite avec l'équation de parité considérée (cf. Eq. (3.18). Celle-ci est de rang plein si elle vaut $N - r$ pour une m-séquence et $N - 2r$ pour une séquence de Gold. Les matrices de rang plein sont identifiées par le symbole $^{(*)}$.

Table E.1 – Équations de parité pour la séquence $(4005, 4445)$.

	m	i	j	k	Rang
c_1	114	78	49	37	2025 $^{(*)}$
c_2	359	344	148	69	2025 $^{(*)}$
c_3	361	174	147	116	2025 $^{(*)}$
c_4	362	311	109	56	2025 $^{(*)}$
c_5	366	322	298	4	2025 $^{(*)}$
c_6	367	257	133	81	2025 $^{(*)}$
c_7	946	833	463	121	2025 $^{(*)}$
c_8	981	935	774	164	2025 $^{(*)}$

TABLE E.2 – Équations de parité pour la séquence (4445).

	m	i	
c_1	49	4	2036 (*)
c_2	73	22	2036 (*)
c_3	93	56	2036 (*)
c_4	98	8	2036 (*)
c_5	114	83	2036 (*)
c_6	146	44	2036 (*)
c_7	186	112	2036 (*)
c_8	196	16	2036 (*)
c_9	228	166	2036 (*)
c_{10}	261	80	2036 (*)
c_{11}	372	224	2036 (*)
c_{12}	465	136	2036 (*)
c_{13}	866	339	2036 (*)

Bibliographie

[1] R.L. Peterson, R.E. Ziemer, and D.E. Borth, *Introduction to spread-spectrum communications*, Prentice Hall, 1995. (Cité en pages 1, 7, 10, 103 et 104.)

[2] S. Vaudenay, *A classical introduction to cryptography : Applications for communications security*, Springer-Verlag New York Inc, 2006. (Cité en pages 1, 7 et 14.)

[3] 3GPP TS25.213 v.4.4.0, "3rd Generation Partnership Project ; Technical Specification Group Radio Access Network ; Spreading and modulation (FDD)," 2004. (Cité en pages 1, 7, 12, 39, 70, 71 et 72.)

[4] 3GPP2 C.S0002-A, "Physical Layer Standard for CDMA2000 Spread Spectrum Systems - Release A," 2000. (Cité en pages 1, 7, 69, 70, 88, 89, 90, 95 et 107.)

[5] H. Holma and A. Toskala, *LTE for UMTS : OFDMA and SC-FDMA based radio access*, Wiley, 2009. (Cité en page 1.)

[6] E.D. Kaplan and C.J. Hegarty, *Understanding GPS : principles and applications*, Artech House Publishers, 2006. (Cité en pages 1 et 21.)

[7] J. Golic, V. Bagini, and G. Morgari, "Linear cryptanalysis of bluetooth stream cipher," in *Advances in Cryptology - EUROCRYPT 2002*. Springer, 2002, pp. 238–255. (Cité en pages 1 et 14.)

[8] R.J. McEliece, *Finite fields for computer scientists and engineers*, Springer, 1987. (Cité en pages 1, 9, 10, 11 et 34.)

[9] T. Kasami, S. Lin, and W. Peterson, "Some results on cyclic codes which are invariant under the affine group and their applications," *Information and Control*, vol. 11, no. 5, pp. 475–496, 1967. (Cité en pages 1, 4, 32, 34 et 35.)

[10] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding : Turbo-codes," in *Proceedings of the IEEE International Conference on Communications (ICC'93)*, 1993, vol. 2, pp. 1064–1070. (Cité en page 1.)

[11] R. Gallager, "Low-density parity-check codes," *IRE Trans. on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962. (Cité en pages 1, 2, 44 et 49.)

[12] D. MacKay and R. Neal, "Good codes based on very sparse matrices," *Cryptography and Coding*, pp. 100–111, 1995. (Cité en pages 1 et 2.)

[13] F.J. MacWilliams and N.J.A. Sloane, *The theory of error-correcting codes*, vol. 16, Elsevier, 1981. (Cité en pages 1, 23, 24, 25, 32, 33 et 34.)

[14] J. Pearl, *Probabilistic reasoning in intelligent systems : networks of plausible inference*, Morgan Kaufmann, 1988. (Cité en page 2.)

[15] F.R. Kschischang, B.J. Frey, and H.A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001. (Cité en pages 2, 25, 26, 27, 42 et 80.)

[16] M. Mezard and A. Montanari, *Information, physics, and computation*, Oxford University Press, USA, 2009. (Cité en page 2.)

[17] A.T. Ihler, J.W. Fisher III, R.L. Moses, and A.S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 809–819, 2005. (Cité en page 2.)

[18] R.J. McEliece, D.J.C. MacKay, and J.F. Cheng, "Turbo decoding as an instance of pearl's belief propagation algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 140–152, 1998. (Cité en page 2.)

[19] Robert B Ward, "Acquisition of pseudonoise signals by sequential estimation," *IEEE Trans. on Communication Technology*, vol. 13, no. 4, pp. 475–483, 1965. (Cité en page 2.)

[20] C.C. Kilgus, "Pseudonoise code acquisition using majority logic decoding," *IEEE Trans. on Communications*, vol. 21, no. 6, pp. 772–774, 1973. (Cité en page 2.)

[21] James L Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969. (Cité en page 2.)

[22] W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1, no. 3, pp. 159–176, 1989. (Cité en pages 2, 14 et 31.)

[23] A. Canteaut and M. Trabbia, "Improved fast correlation attacks using parity-check equations of weight 4 and 5," in *Advances in Cryptology - EUROCRYPT 2000*. Springer, 2000, pp. 573–588. (Cité en pages 2, 25, 28, 31, 39, 41 et 96.)

[24] N. Santhi and A. Vardy, "On the effect of parity-check weights in iterative decoding," in *Proceedings of the International Symposium on Information Theory (ISIT)*. IEEE, 2004. (Cité en pages 2, 4, 25, 28, 31 et 41.)

[25] W. T. Penzhorn and G.J. Kühn, "Computation of low-weight parity checks for correlation attacks on stream ciphers," in *Cryptography and Coding*, pp. 74–83. Springer, 1995. (Cité en pages 2, 4, 32, 37 et 41.)

[26] P. Chose, A. Joux, and M. Mitton, "Fast correlation attacks : An algorithmic point of view," in *Advances in Cryptology - EUROCRYPT 2002*. Springer, 2002, pp. 209–221. (Cité en pages 2, 4, 25, 28, 31, 39, 41 et 96.)

[27] L.L. Yang and L. Hanzo, "Differential acquisition of m-sequences using recursive soft sequential estimation," *IEEE Trans. on Wireless Communications*, vol. 4, no. 1, pp. 128–136, 2005. (Cité en page 2.)

[28] K.M. Chugg and M. Zhu, "A new approach to rapid PN code acquisition using iterative message passing techniques," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 5, pp. 884–897, 2005. (Cité en pages 2, 15 et 25.)

[29] F. Principe, K.M. Chugg, and M. Luise, "Rapid acquisition of Gold codes and related sequences using iterative message passing on redundant graphical models," in *Proceedings of the IEEE Military Communications Conference (MILCOM'06)*, 2006, pp. 1–7. (Cité en pages 2, 3, 15, 25, 27 et 32.)

[30] O.W. Yeung and K.M. Chugg, "An iterative algorithm and low complexity hardware architecture for fast acquisition of long PN codes in UWB systems," *The Journal of VLSI Signal Processing*, vol. 43, no. 1, pp. 25–42, 2006. (Cité en pages 2, 3, 15, 27 et 28.)

[31] R. Kerr and J. Lodge, "Iterative signal processing for blind code phase acquisition of CDMA 1x signals for radio spectrum monitoring," *Journal of Electrical and Computer Engineering*, vol. 2010, pp. 3, 2010. (Cité en pages 2, 70 et 88.)

[32] N. Weinberger and N. Merhav, "Codeword or noise ? exact random coding exponents for joint detection and decoding," *IEEE Trans. on Information Theory*, vol. 60, no. 9, pp. 5077–5094, 2014. (Cité en pages 3, 15 et 29.)

[33] T. Richardson, "Error floors of LDPC codes," in *Proceedings of the annual Allerton conference on communication control and computing*. The University ; 1998, 2003, vol. 41, pp. 1426–1435. (Cité en pages 4, 28 et 49.)

[34] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. Wainwright, "Investigation of error floors of structured low-density parity-check codes by hardware emulation," in *Global Telecommunications Conference (GLOBECOM'06)*. IEEE, 2006, pp. 1–6. (Cité en pages 4, 28 et 43.)

[35] K. Borre, D.M. Akos, N. Bertelsen, P. Rinder, and S.H. Jensen, *A software-defined GPS and Galileo receiver : a single-frequency approach*, Birkhauser, 2007. (Cité en pages 7, 13 et 14.)

[36] R. Gold, "Optimal binary sequences for spread spectrum multiplexing," *IEEE Trans. on Information Theory*, vol. 13, pp. 619–621, 1967. (Cité en pages 11 et 75.)

[37] T. Johansson and F. Jonsson, "Improved fast correlation attacks on stream ciphers via convolutional codes," in *Advances in Cryptology - EUROCRYPT 1999*. Springer, 1999, pp. 347–362. (Cité en page 14.)

[38] M. Mihaljevic, M. Fossorier, and H. Imai, "A low-complexity and high-performance algorithm for the fast correlation attack," in *Fast Software Encryption*. Springer, 2001, pp. 45–60. (Cité en page 14.)

[39] M.P.C. Fossorier, M.J. Mihaljevic, and H. Imai, "Modeling block decoding approaches for the fast correlation attack," *IEEE Trans. on Information Theory*, vol. 53, no. 12, pp. 4728–4737, 2007. (Cité en page 14.)

[40] SM Kay, *Fundamentals of Statistical Signal Processing : Detection Theory*, vol. 2, Prentice Hall, 1998. (Cité en pages 16, 17 et 20.)

[41] M.A. Richards, *Fundamentals of radar signal processing*, McGraw-Hill, 2005. (Cité en pages 18, 19, 44 et 78.)

[42] J.G. Proakis, *Digital communications*, vol. 1221, McGraw-hill, 1987. (Cité en page 21.)

[43] C. Herzet, N. Noels, V. Lottici, H. Wymeersch, M. Luise, M. Moeneclaey, and L. Vandendorpe, "Code-aided turbo synchronization," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1255–1271, 2007. (Cité en pages 22 et 101.)

[44] N. Nele, S. Heidi, and M. Moeneclaey, "Carrier phase tracking from turbo and LDPC coded signals affected by a frequency offset," *IEEE Communications Letters*, vol. 9, no. 10, pp. 915–917, 2005. (Cité en page 22.)

[45] C. Herzet, V. Ramon, and L. Vandendorpe, "A theoretical framework for iterative synchronization based on the sum–product and the expectation-maximization algorithms," *IEEE Trans. on Signal Processing*, vol. 55, no. 5, pp. 1644–1658, 2007. (Cité en pages 22 et 101.)

[46] R. Imad, S. Houcke, and M. Ghogho, "Blind estimation of the phase and carrier frequency offsets for LDPC-coded systems," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 83, 2010. (Cité en pages 22, 100 et 101.)

[47] R. Raheli, A. Polydoros, and C.K. Tzou, "Per-survivor processing : A general approach to MLSE in uncertain environments," *IEEE Trans. on Communications*, vol. 43, no. 234, pp. 354–364, 1995. (Cité en page 22.)

[48] J. Dauwels and H.A. Loeliger, "Phase estimation by message passing," in *IEEE International Conference on Communications*. IEEE, 2004, vol. 1, pp. 523–527. (Cité en page 22.)

[49] A. Barbieri, G. Colavolpe, and G. Caire, "Joint iterative detection and decoding in the presence of phase noise and frequency offset," *IEEE Trans. on Communications*, vol. 55, no. 1, pp. 171–179, 2007. (Cité en page 22.)

[50] A. Canteaut and .F Chabaud, "A new algorithm for finding minimum-weight words in a linear code : Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511," *IEEE Trans. on Information Theory*, vol. 44, no. 1, pp. 367–378, 1998. (Cité en page 25.)

[51] T. M. Cassaro and C. N. Georghiades, "Frame synchronization for coded systems over awgn channels," *IEEE Trans. on Communications*, vol. 52, no. 3, pp. 484–489, 2004. (Cité en page 25.)

[52] H. Wymeersch, H. Steendam, H. Bruneel, and M. Moeneclaey, "Code-aided frame synchronization and phase ambiguity resolution," *IEEE Trans. on Signal Processing*, vol. 54, no. 7, pp. 2747–2757, 2006. (Cité en page 25.)

[53] R. Imad and S. Houcke, "Theoretical analysis of a MAP based blind frame synchronizer," *IEEE Trans. on Wireless Communications*, vol. 8, no. 11, pp. 5472–5476, 2009. (Cité en page 25.)

[54] N. Wiberg, *Codes and decoding on general graphs*, Ph.D. dissertation, Linköping University, Sweden, 1996. (Cité en pages 26, 27, 42, 76, 79, 82 et 95.)

[55] V. Savin, "Self-corrected min-sum decoding of LDPC codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, 2008, pp. 146–150. (Cité en page 27.)

[56] Marc PC Fossorier, Miodrag Mihaljevic, and Hideki Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. on Communications*, vol. 47, no. 5, pp. 673–680, 1999. (Cité en page 27.)

[57] Y. Han and W. E. Ryan, "Low-floor decoders for LDPC codes," *IEEE Trans. on Communications*, vol. 57, no. 6, pp. 1663–1673, 2009. (Cité en pages 28 et 44.)

[58] L. Dolecek, P. Lee, Z. Zhang, V. Anantharam, B. Nikolic, and M. Wainwright, "Predicting error floors of structured LDPC codes : Deterministic bounds and estimates," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 6, pp. 908–917, 2009. (Cité en pages 28 et 43.)

[59] T. Hehn, J. B Huber, O. Milenkovic, and S. Laendner, "Multiple-bases belief-propagation decoding of high-density cyclic codes," *IEEE Trans. on Communications*, vol. 58, no. 1, pp. 1–8, 2010. (Cité en pages 28, 42 et 100.)

[60] S. Maitra, K. C. Gupta, and A. Venkateswarlu, "Results on multiples of primitive polynomials and their products over GF(2)," *Theoretical Computer Science*, vol. 341, no. 1, pp. 311–343, 2005. (Cité en pages 32 et 37.)

[61] K. Huber, "Some comments on zech's logarithms," *IEEE Trans. on Information Theory*, vol. 36, no. 4, pp. 946–950, 1990. (Cité en page 33.)

[62] V. Pless, "Power moment identities on weight distributions in error correcting codes," *Information and Control*, vol. 6, no. 2, pp. 147–152, 1963. (Cité en page 34.)

[63] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions : with formulas, graphs, and mathematical tables*, Courier Dover Publications, 2012. (Cité en page 35.)

[64] T. Tian, C.R. Jones, J.D. Villasenor, and R.D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. on Communications*, vol. 52, no. 8, pp. 1242–1247, 2004. (Cité en page 42.)

[65] AW Ingleton, "The rank of circulant matrices," *Journal of the London Mathematical Society*, vol. 1, no. 4, pp. 445–460, 1956. (Cité en page 42.)

[66] T. R. Halford and K. M. Chugg, "An algorithm for counting short cycles in bipartite graphs," *IEEE Trans. on Information Theory*, vol. 52, no. 1, pp. 287–292, 2006. (Cité en pages 43, 52 et 110.)

[67] L. Dolecek, Z. Zhang, V. Anantharam, M. J Wainwright, and B. Nikolic, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. on Information Theory*, vol. 56, no. 1, pp. 181–201, 2010. (Cité en pages 43 et 44.)

[68] M. Karimi and A. H. Banihashemi, "On characterization of elementary trapping sets of variable-regular LDPC codes," *arXiv preprint arXiv :1308.1259*, 2013. (Cité en page 48.)

[69] RH Zarrabizadeh and Elvino S Sousa, "A differentially coherent PN code acquisition receiver for CDMA systems," *IEEE Trans. on Communications*, vol. 45, no. 11, pp. 1456–1465, 1997. (Cité en page 56.)

[70] A. Polydoros and C. Weber, "A unified approach to serial search spread-spectrum code acquisition–part I : general theory," *IEEE Trans. on Communications*, vol. 32, no. 5, pp. 542–549, 1984. (Cité en pages 56, 59, 86 et 114.)

[71] L. Milstein, J. Gevargiz, and P. Das, "Rapid acquisition for direct sequence spread-spectrum communications using parallel SAW convolvers," *IEEE Trans. on Communications*, vol. 33, no. 7, pp. 593–600, 1985. (Cité en page 56.)

[72] F. Principe, K.M. Chugg, and M. Luise, "Performance evaluation of message-passing-based algorithms for fast acquisition of spreading codes with application to satellite positioning," in *NAVITEC, Noordwijk, The Netherlands*, December 2006. (Cité en page 57.)

[73] H. Holma and A. Toskala, *WCDMA for UMTS*, vol. 4, Wiley, 2000. (Cité en page 69.)

[74] V. Chandrasekhar, J. Andrews, and A. Gatherer, "Femtocell networks : a survey," *IEEE Communications Magazine*, vol. 46, no. 9, pp. 59–67, 2008. (Cité en page 69.)

[75] P. Xia, V. Chandrasekhar, and J.G. Andrews, "Open vs. closed access femtocells in the uplink," *IEEE Trans. on Wireless Communications*, vol. 9, no. 12, pp. 3798–3809, 2010. (Cité en page 70.)

[76] S. Verdu, *Multiuser detection*, Cambridge University Press, 1998. (Cité en page 70.)

[77] Z. Shi, H. Wang, M. Zhao, and M. C. Reed, "An uplink analytical model for two-tiered 3G femtocell networks," in *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*. IEEE, 2010. (Cité en page 70.)

[78] J.G. Andrews, "Interference cancellation for cellular systems : a contemporary overview," *IEEE Wireless Communications*, vol. 12, no. 2, pp. 19–29, 2005. (Cité en page 70.)

[79] 3GPP TS 25.101 V5.2.0, "Radio Transmission and Reception (FDD)," 2002. (Cité en page 74.)

[80] 3GPP TS 25.141 V5.15.1, "Base Station (BS) Conformance Testing (FDD)," 2007. (Cité en pages 74 et 84.)

[81] B. Arazi, "Decimation of m-sequences leading to any desired phase shift," *Electronics Letters*, vol. 13, no. 7, pp. 213–215, 1977. (Cité en pages 77, 94 et 105.)

[82] 3GPP TS25.104 v.5.8.0, "3rd Generation Partnership Project ; Technical Specification Group Radio Access Network ; BS Radio Transmission and Reception (FDD)," 2007. (Cité en pages 79, 86 et 87.)

[83] R. Kerr and J. Lodge, "Near ML performance for linear block codes using an iterative vector SISO decoder," in *4th International Symposium on Turbo Codes&Related Topics, Munich, Germany*. VDE, 2006. (Cité en pages 83 et 88.)

[84] M. des Noes, V. Savin, L. Ros, and J.M. Brossier, "Blind identification of the uplink scrambling code index of a WCDMA transmission and application to femtocell networks," in *IEEE International Conference on Communications, Budapest, Hungary*, 2013. (Cité en page 97.)

[85] M. des Noes, V. Savin, L. Ros, and J.M. Brossier, "Blind identification of the scrambling code of a reverse link CDMA 2000 transmission," in *IEEE International Conference on Communications Budapest, Hungary*, 2013. (Cité en page 97.)

[86] J. Jiang and K. R. Narayanan, "Iterative soft decoding of reed-solomon codes," *IEEE Communications Letters*, vol. 8, no. 4, pp. 244–246, 2004. (Cité en page 100.)

[87] T. R. Halford and K. M. Chugg, "Random redundant soft-in soft-out decoding of linear block codes," in *IEEE International Symposium on Information Theory*. IEEE, 2006, pp. 2230–2234. (Cité en page 100.)

Résumé — Le décodage des séquences pseudo-aléatoire est un sujet relativement peu traité dans la littérature. Les études menées dans le domaine de la cryptographie se sont focalisées principalement sur la recherche d'équations de parité de poids faible, qui est le point le plus problématique dans ce contexte. Dans le domaine des communications numériques, ces séquences ne sont pas employées pour leurs propriétés de correction d'erreurs. Par conséquent, il y a peu d'études sur les décodeurs adaptés, ainsi que l'analyse de leurs performances. Une des contributions de cette thèse est d'améliorer la compréhension des mécanismes de décodage, en se focalisant sur les performances des décodeurs itératifs par passage de messages. L'objectif est de comprendre comment le choix des équations de parité employées par le décodeur influence les performances (e.g. probabilité de détection correcte ou de fausse alarme). Nous allons tout d'abord établir un lien entre la théorie de la détection conventionnelle et le décodage des m-séquences. Nous allons ensuite détailler les propriétés des codes duaux des m-séquences et des séquences de Gold afin de déterminer le nombre d'équations de parité disponibles pour le décodeur. Nous nous intéressons aussi au choix de ces équations et à leur impact sur le graphe de décodage. L'identification de certaines structures topologiques (i.e. *absorbing set*) permet de comprendre l'origine des fausses alarmes et d'en tirer un algorithme de sélection des équations de parité qui minimise le taux de fausses alarmes.

Nous proposons enfin deux algorithmes de détection du code d'embrouillage pour les systèmes WCDMA et CDMA2000. Ils exploitent la structure des trames, ainsi que les propriétés des m-séquences constituant les séquences de Gold. Ces algorithmes montrent les nouvelles vulnérabilités des transmissions par étalement de spectre.

Mots clés : séquences binaires, m-séquence, Gold, détection, décodage, ensembles absorbants, cycles, code d'embrouillage, WCDMA, CDMA2000.

Abstract — The decoding of pseudo-random binary sequences is not well covered in the literature. In cryptography, researches mainly focused on the search of small weight parity check equations. In digital communications, these sequences are not used for their error correction capabilities. As a result, there are few studies about well suited decoder and the analysis of their performances. One contribution of this thesis is to improve the understanding of decoding mecanism for such coding schemes. More specifically, we focus on iterative message-passing algorithm. One objective is to understand how the selection of parity check equations affects decoding performance (probability of correct detection, wrong detection and false alarm).

First of all, a link is established between conventional detection theory and decoding of pseudo-random sequences. Then, the properties of dual codes of m-sequence and Gold codes are detailed. This allows to evaluate the number of parity check equation of weight t. More specifically, the number of parity check equations of weight $t = 5$ is evaluated for Gold sequence having

an odd degree. This is important because there is no parity check equations of weight $t = 2, 3$ or 4 in this case. Subsequently, the impact of the selected equations on the decoding graph. It is showed that absorbing sets prevent from the apparition of false alarms, and must not be eliminated as it is done in conventional LDPC code design. A method for identifying these absorbing sets for a redundant parity check matrix is proposed. It is found that 'transverse' cycles may destroy these absorbing sets and hence modify the decoding performances. An algorithm for selecting parity check equations minimizing the probability of false alarm is thus derived. It is based on the minimization of cycles of length 6 and 8, which also minimizes the number of transverse cycles. This algorithm is proved to be very effective for detecting m-sequences.

Then, the usefulness of detecting m-sequence or Gold sequence with a decoding technique is emphasized. Exploiting the properties of m-sequence, it is detailed how the scrambling code of WCDMA and CDMA2000 systems may be estimated blindly. This could be used for instance for detecting a strong interferer in a femtocell system.

Keywords : binary sequences , m-sequence, Gold, detection, iterative decoding, absorbing set, cycles, WCDMA, CDMA2000, scrambling code.

Commissariat à l'Energie Atomique (CEA), 17 rue des Martyrs
38054 Grenoble cedex 9