

Titre: Décodage à seuil itératif des codes convolutionnels doublement orthogonaux perforés et application aux modulations multiniveaux
Title: Orthogonal Punctured Convolutional Codes Decoding by Iterative Thresholding and Application to Multilevel Modulations

Auteur: Frédéric Dru
Author:

Date: 2001

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Dru, F. (2001). Décodage à seuil itératif des codes convolutionnels doublement orthogonaux perforés et application aux modulations multiniveaux [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/6967/>

Document en libre accès dans PolyPublie Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/6967/>
PolyPublie URL:

Directeurs de recherche: David Haccoun
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

DÉCODAGE À SEUIL ITÉRATIF DES CODES CONVOLUTIONNELS
DOUBLEMENT ORTHOGONAUX PERFORÉS ET APPLICATION AUX
MODULATIONS MULTINIVEAUX

FRÉDÉRIC DRU

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES (M.Sc.A.)
(GÉNIE ÉLECTRIQUE)

JUIN 2001



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-65572-5

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

DÉCODAGE À SEUIL ITÉRATIF DES CODES CONVOLUTIONNELS
DOUBLEMENT ORTHOGONAUX PERFORÉS ET APPLICATION AUX
MODULATIONS MULTINIVEAUX

présenté par: DRU Frédéric

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de:

M. CONAN Jean, Ph.D., président

M. HACCOUN David, Ph.D., membre et directeur de recherche

M. GAGNON François, Ph.D., membre

À Christelle.

REMERCIEMENTS

Je tiens à remercier par ces quelques lignes les personnes sans qui ce travail de recherche n'aurait pas pu être mené à bien.

Mes premiers remerciements vont à mon directeur de recherches, le Dr. David Haccoun, qui a su me guider dans mon travail et poser un regard critique sur mes résultats. Je le remercie également de m'avoir accordé une aide financière pendant la majeure partie de mes recherches.

Je souhaite en outre exprimer ma reconnaissance envers l'École Supérieure d'Électricité pour m'avoir laissé entreprendre cette formation double-diplômante enrichissante à de nombreux points de vue.

Merci aussi à mes collègues de laboratoire, à la fois pour leurs conseils judicieux et pour leur bonne humeur quotidienne. Je pense particulièrement à Brice, Lionel, Christian, Khaled, Mahdi et Pierre.

Enfin, je tiens à remercier mes amis à Montréal et en France ainsi que ma famille pour leur soutien.

RÉSUMÉ

Le travail de recherche présenté dans ce mémoire concerne l'étude de l'évolution des performances de l'algorithme de décodage à seuil itératif lorsque l'on augmente l'efficacité spectrale du système de télécommunications dont il fait partie. Ce type de décodage itératif, permettant de protéger l'information contre d'éventuelles erreurs lors de la transmission, a été introduit il y a quelques années et représente une évolution importante du principe de décodage turbo. Au prix de performances certes inférieures à faible rapport signal sur bruit, ce type de décodage présente une complexité (théorique et matérielle) ainsi qu'un délai bien moindres. Cette amélioration est obtenue en reportant la propriété de décorrélation des observables, obtenue auparavant grâce aux entrelaceurs, sur la matrice génératrice du code convolutionnel utilisé qui doit alors vérifier certaines propriétés de double orthogonalité. Notre but a donc été d'étudier théoriquement et pratiquement le comportement de cet algorithme de décodage lorsqu'il est soumis à diverses contraintes occasionnées par l'augmentation de l'efficacité spectrale.

La première partie de notre étude a consisté à générer des codes de taux de codage élevés par perforation, c'est-à-dire en éliminant périodiquement certains symboles en sortie du codeur. L'influence de l'utilisation de ces codes perforés a été étudiée de manière théorique à l'aide d'une représentation spectrale et l'influence de nombreux paramètres a été quantifiée. On a pu par la suite dresser la liste de codes perforés optimaux du point de vue des performances d'erreur et certaines propriétés de ces codes ont été observées.

La deuxième partie de notre travail a quant à elle permis d'analyser les performances du décodage à seuil itératif des codes perforés générés dans la première partie lorsqu'une modulation QAM est utilisée. La nécessité de la mise en œuvre d'un calculateur de métrique est présentée ainsi que les performances de deux différents types de modulateurs numériques.

Au cours de ces deux étapes, des simulations ont permis de valider l'approche

théorique et de comparer les performances des différents codes perforés doublement orthogonaux utilisés. Notre travail nous a semblé fructueux du fait qu'il fournit des informations indispensables permettant de faire les compromis nécessaires entre la probabilité d'erreur souhaitée et l'efficacité spectrale que l'on désire atteindre.

Toutefois, il ne faut pas oublier que la grande majorité des codes présentant un comportement optimal peuvent avoir une longueur de contrainte élevée. C'est la raison pour laquelle un gros travail reste à accomplir en ce qui a trait à la fois à la recherche des codes optimaux de longueur de contrainte minimale et à l'amélioration des performances des codes non optimaux au sens de la probabilité d'erreur.

ABSTRACT

This study concerns the analysis of the behavior of the iterative threshold decoding algorithm when it is used in a communication system with high spectral efficiency. This type of iterative decoding was recently introduced as an improvement of the turbo codes decoding process.

The turbo decoding technique, which allows to achieve reliable communication at signal-to-noise ratios very close to that of the channel capacity, suffers from a substantial complexity and from an inherent latency which may prevent the technique from being used in some time-sensitive applications. The improvement which circumvents both the decoding complexity and interleaving requirements is based on new orthogonal threshold decodable convolutional codes and threshold decoding. The independence of the observables at each decoding iteration can be obtained thanks to codes which exhibit further orthogonal properties than the well-known CSOCs. Our goal was to study theoretically and practically the behavior of this algorithm when it is subjected to constraints due to the increasing of the spectral efficiency.

The first part of our study consisted in generating high-rated codes by a puncturing technique, i.e. by deleting periodically some symbols at the output of the encoder. Then, the influence of the puncturing technique was analyzed in a theoretical way using a spectral representation and the influence of several parameters was quantified. Some optimal codes (from the point of view of the probability of error) were listed and certain properties of these codes were presented.

The second part of our work was the analysis of the performance of the decoding of these punctured codes when QAM modulation is used. The need for an LLR computer is presented as well as the performance of two various types of mapping.

In both stages, simulations made it possible to validate the theoretical approach

and to compare the performance of the various punctured self-doubly orthogonal codes used. We think that this dissertation was profitable since it constitutes a tool making it possible to do the necessary compromises between the desired probability of error and the spectral efficiency that one wishes to reach.

However, it should not be forgotten that the large majority of the codes presenting an optimal behavior as far as the probability of error is concerned are not optimal with regard to the constraint length and that a large work remains to be achieved firstly with the search of codes whose constraint length is minimized and secondly with the improvement of the performance of non-optimal codes.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES TABLEAUX	xiv
LISTE DES FIGURES	xv
LISTE DES SIGLES ET DES SYMBOLES	xix
LISTE DES ANNEXES	xxi
CHAPITRE 1: INTRODUCTION	1
1.1 Motivations	1
1.2 Organisation du mémoire	3
1.3 Contributions	4
CHAPITRE 2: ÉLÉMENTS DE THÉORIE DU CODAGE	5
2.1 Généralités	5
2.1.1 Éléments d'un système de communications numériques	5
2.1.2 Historique	7
2.2 Codes en bloc	8
2.3 Codes convolutionnels	10
2.3.1 Définition	10
2.3.2 Types de représentation	14
2.3.3 Principe du décodage optimal de Viterbi	19
2.4 La technique de décodage turbo	20

2.4.1	Concaténation des codes convolutionnels	20
2.4.2	Le codage	21
2.4.3	L'entrelaceur	22
2.4.4	La structure du décodeur itératif	22
2.4.5	Les algorithmes de décodage	24
CHAPITRE 3:	DÉCODAGE À SEUIL ITÉRATIF ET CODAGE CONVOLUTIONNEL DOUBLEMENT ORTHOGONAL	25
3.1	Forces et faiblesses du décodage turbo	25
3.1.1	Avantages de la technique	25
3.1.2	Inconvénients	26
3.1.3	Évolutions apportées	27
3.1.4	Amélioration envisagée	27
3.2	Décodage à seuil et codes convolutionnels orthogonaux	28
3.2.1	Codes convolutionnels orthogonaux	28
3.2.2	Décodage à seuil	30
3.3	Décodage à seuil itératif et codes convolutionnels doublement orthogonaux	39
3.3.1	Décodage à seuil itératif	39
3.3.2	Codes doublement orthogonaux au sens large	41
3.3.3	Codes doublement orthogonaux au sens strict	43
3.3.4	Construction des codes doublement orthogonaux	45
CHAPITRE 4:	PERFORATION DES CODES DOUBLEMENT ORTHOGONAUX	47
4.1	Introduction	47
4.1.1	Motivations	47
4.1.2	Introduction à la perforation	48
4.2	Influence de la perforation des CSO2C-WS sur les performances du décodage à seuil itératif	49
4.2.1	Représentation spectrale des PCSO2C-WS de taux $R = \frac{k}{k+1}$	50
4.2.2	Étude des performances d'erreur	58

4.2.3	Cas des codes de taux $R = \frac{k}{n}$	69
4.3	Propriétés des PCSO2C-WS	74
4.3.1	Recherche des PCSO2C-WS optimaux	74
4.3.2	Codes imbriqués	77
4.3.3	Utilisation des coefficients de pondération pour le décodage des PCSO2C-WS	79
4.4	Cas des CSO2C-SS	81
4.4.1	Influence de la perforation: extension des PCSO2C-WS	81
4.4.2	Étude des performances d'erreur	84
4.5	Bilan	92
CHAPITRE 5: APPLICATION AUX MODULATIONS À EFFI- CACITÉS SPECTRALES ÉLEVÉES		94
5.1	Introduction aux modulations numériques	94
5.1.1	Modulation M -PAM	94
5.1.2	Modulation M -PSK	96
5.1.3	Modulation M -QAM	99
5.1.4	Autres types de modulations	101
5.2	Assignation des symboles de canal aux signaux de la constellation par codage de Gray	101
5.3	Performances du décodage à seuil itératif avec assignation par codage de Gray	104
5.3.1	Mise en œuvre d'un calculateur de métrique	104
5.3.2	Performances d'erreur et résultats	108
5.3.3	Influence de la sous-optimalité du calculateur de métrique	113
5.3.4	Entrelacement des bits au sein des trames	116
5.4	Assignation par partition d'ensemble	118
5.5	Utilisation de l'assignation par partition d'ensemble dans le cas de PCSO2C non optimaux	124
5.5.1	Protection inégale des symboles d'information	124
5.5.2	Étude des performances d'erreur	125
5.6	Bilan	127

CHAPITRE 6: CONCLUSION	128
6.1 Bilan de la recherche	128
6.2 Améliorations envisageables	129
6.3 Ouverture	130
RÉFÉRENCES	131

LISTE DES TABLEAUX

3.1	Meilleurs ensembles de positions obtenus	46
4.1	Meilleurs PCSO2C-WS de taux $R = \frac{2}{3}$	75
4.2	Meilleurs PCSO2C-WS de taux $R = \frac{3}{4}$	76
4.3	Meilleurs PCSO2C-WS de taux $R = \frac{4}{5}$	76
4.4	PCSO2C-WS imbriqués optimaux de taux $R = \frac{2}{3}$	78
4.5	PCSO2C-WS imbriqués optimaux de taux $R = \frac{3}{4}$	78
4.6	PCSO2C-WS imbriqués optimaux de taux $R = \frac{4}{5}$	78
5.1	Rapport signal sur bruit moyen $\frac{\bar{E}_b}{N_0}$ (en dB) pour diverses probabilités d'erreur dans le cas d'une information non codée transmise avec les modulations BPSK, 16-QAM et 64-QAM	112
5.2	Rapport signal sur bruit moyen $\frac{\bar{E}_b}{N_0}$ (en dB) pour diverses probabilités d'erreur dans le cas d'une information codée (avec PCSO2C-WS optimal de taux $R = \frac{2}{3}$, $J = 8$, quatrième itération) transmise avec les modulations BPSK, 16-QAM et 64-QAM	113
5.3	Temps de simulation en secondes pour les deux calculateurs de métriques dans le cas de la figure 5.10	115
5.4	Position des bits d'information utilisés à chaque itération au temps $i = 0$ par le décodeur à seuil correspondant au CSO2C-WS ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$.	118

LISTE DES FIGURES

2.1	Éléments de base d'un système de communications numériques	5
2.2	Codeur convolutionnel	11
2.3	Exemple de codeur convolutionnel ($R = \frac{1}{2}$, $K = 3$, $k = 1$)	13
2.4	Exemple de codeur convolutionnel ($R = \frac{1}{2}$, $K = 4$, $k = 1$, $\mathbf{g}_1 = [1000]$, $\mathbf{g}_2 = [1011]$)	14
2.5	Arbre correspondant à l'exemple de la figure 2.4	15
2.6	Treillis correspondant à l'exemple de la figure 2.4	17
2.7	Diagramme d'état correspondant à l'exemple de la figure 2.4	18
2.8	Concaténation parallèle de deux codeurs convolutionnels récursifs .	21
2.9	Décodeur itératif turbo	23
3.1	Exemple de codeur convolutionnel orthogonal	29
3.2	Notations relatives au canal de transmission	31
3.3	Schéma d'un décodeur à seuil de Type I	34
3.4	Schéma du décodeur à seuil de Type II à sortie pondérée correspondant au codeur de la figure 3.1	38
3.5	Schématisation du décodage à seuil itératif sans entrelacement . .	40
3.6	Exemple de codeur doublement orthogonal au sens strict de taux $\frac{2}{4}$	44
4.1	Illustration de la technique de perforation	49
4.2	Spectre de perforation du PCSO2C-WS de taux $R = \frac{3}{4}$ ayant $J = 10$ connexions aux positions $\{0, 29, 40, 43, 1020, 1328, 1495, 1606, 1696, 1698\}$	56
4.3	Spectre de perforation du PCSO2C-WS de taux $R = \frac{2}{3}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$	57
4.4	Performances à la première et à la troisième itération du décodage des PCSO2C-WS de taux $R = \frac{2}{3}$ et $R = \frac{3}{4}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$	59

4.5 Spectre de perforation des PCSO2C-WS de taux $R = \frac{2}{3}$ et $R = \frac{3}{4}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$	60
4.6 Performances à la première et à la troisième itération du décodage du PCSO2C-WS de taux $R = \frac{3}{4}$ ayant $J = 10$ connexions aux posi- tions $\{0, 29, 40, 43, 1020, 1328, 1495, 1606, 1696, 1698\}$ perforé suivant les trois matrices données en (4.32), (4.33) et (4.34)	62
4.7 Performances à la première et à la troisième itération du décodage de plusieurs PCSO2C-WS de taux $R = \frac{2}{3}$ dont le nombre de connexions varie entre $J = 8$ et $J = 11$	63
4.8 Spectres de perforation des PCSO2C-WS de taux $R = \frac{2}{3}$ utilisés à la figure 4.7	64
4.9 Performances à la première et à la troisième itération du décodage de plusieurs PCSO2C-WS ayant tous $J = 8$ connexions et dont les spectres sont représentés à la figure 4.10	66
4.10 Spectres de perforation de PCSO2C-WS de taux $R = \frac{2}{3}$ ayant tous $J = 8$ connexions mais dont les positions varient	67
4.11 Gain asymptotique de codage d'un décodeur à seuil classique de taux $R = \frac{1}{2}$	68
4.12 Performances à la première et à la troisième itération du décodage des PCSO2C-WS de taux $R = \frac{3}{5}$ et $R = \frac{7}{10}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$	72
4.13 Spectres de perforation des PCSO2C-WS de taux $R = \frac{3}{5}$ et $R = \frac{7}{10}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$	73
4.14 Performances à la quatrième itération des PCSO2C-SS de taux $R =$ $\frac{7}{14}$, $R = \frac{14}{21}$ et $R = \frac{21}{28}$ ayant $J = 7$ connexions optimaux du point de vue de la longueur de contrainte	85
4.15 Performances à la quatrième itération de deux PCSO2C-SS de taux $R = \frac{2J}{3J}$ ayant respectivement $J = 7$ et $J = 8$ registres	86
4.16 Spectres de perforation du PCSO2C-SS de taux $R = \frac{14}{21}$ ayant $J = 7$ registres optimal au sens de [1]	88

4.17 Spectres de perforation du PCSO2C-SS de taux $R = \frac{16}{24}$ ayant $J = 8$ registres optimal au sens de [1]	89
5.1 Constellation M -PAM pour $M = 4$	96
5.2 Constellations M -PSK pour $M = 2$, $M = 4$ et $M = 8$	98
5.3 Constellation M -QAM pour $M = 16$	100
5.4 Exemple d'assignation par codage de Gray pour la constellation 16-QAM	103
5.5 Schéma de principe d'un émetteur multiniveaux	107
5.6 Schéma de principe d'un récepteur multiniveaux	107
5.7 Performances à la quatrième itération du décodage des PCSO2C-WS de taux $R = \frac{2}{3}$ et $R = \frac{3}{4}$ ayant $J = 8$ connexions aux positions $\{0,$ $43, 139, 322, 422, 430, 441, 459\}$, modulation 16-QAM	108
5.8 Performances à la quatrième itération du décodage de plusieurs PCSO2C-WS de taux $R = \frac{2}{3}$ ayant $J = 8$ connexions, modulation 16-QAM	110
5.9 Performances à la quatrième itération du décodage du PCSO2C-WS de taux $R = \frac{2}{3}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139,$ $322, 422, 430, 441, 459\}$, modulation M -QAM	111
5.10 Performances à la quatrième itération du calculateur de métrique sous-optimal en utilisant le PCSO2C-WS de taux $R = \frac{2}{3}$ ayant $J =$ 8 connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$, modulation 64-QAM	115
5.11 Exemple d'assignation par partition d'ensemble pour la constellation 8-PSK	121
5.12 Exemple d'assignation par partition d'ensemble pour la constellation 16-QAM	123
5.13 Performances du décodage du PCSO2C-WS de taux $R = \frac{2}{3}$ ayant $J = 8$ connexions aux positions $\{0, 9, 21, 395, 584, 767, 871, 912\}$, modulation 16-QAM	126
I.1 Probabilité d'erreur par bit pour le PCSO2C-WS $J = 4$, $R = \frac{2}{3}$	135
I.2 Probabilité d'erreur par bit pour le PCSO2C-WS $J = 5$, $R = \frac{2}{3}$	136

I.3	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 6, R = \frac{2}{3}$	136
I.4	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 7, R = \frac{2}{3}$	137
I.5	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 8, R = \frac{2}{3}$	137
I.6	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 9, R = \frac{2}{3}$	138
I.7	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 10, R = \frac{2}{3}$	138
I.8	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 11, R = \frac{2}{3}$	139
I.9	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 12, R = \frac{2}{3}$	139
I.10	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 13, R = \frac{2}{3}$	140
I.11	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 14, R = \frac{2}{3}$	140
I.12	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 15, R = \frac{2}{3}$	141
I.13	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 6, R = \frac{3}{4}$	141
I.14	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 7, R = \frac{3}{4}$	142
I.15	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 8, R = \frac{3}{4}$	142
I.16	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 9, R = \frac{3}{4}$	143
I.17	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 10, R = \frac{3}{4}$	143
I.18	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 11, R = \frac{3}{4}$	144
I.19	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 12, R = \frac{3}{4}$	144
I.20	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 13, R = \frac{3}{4}$	145
I.21	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 14, R = \frac{3}{4}$	145
I.22	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 15, R = \frac{3}{4}$	146
I.23	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 8, R = \frac{4}{5}$	146
I.24	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 9, R = \frac{4}{5}$	147
I.25	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 10, R = \frac{4}{5}$	147
I.26	Probabilité d'erreur par bit pour le PCSO2C-WS $J = 11, R = \frac{4}{5}$	148
II.1	Probabilité d'erreur par bit pour le PCSO2C-SS $J = 4, R = \frac{8}{12}$	149
II.2	Probabilité d'erreur par bit pour le PCSO2C-SS $J = 7, R = \frac{14}{21}$	150
II.3	Probabilité d'erreur par bit pour le PCSO2C-SS $J = 9, R = \frac{18}{27}$	150
II.4	Probabilité d'erreur par bit pour le PCSO2C-SS $J = 6, R = \frac{18}{24}$	151

LISTE DES SIGLES ET DES SYMBOLES

Sigles

BPSK	Binary Phase Shift Keying
CSOC	Convolutional Self-Orthogonal Code
CSO2C	Convolutional Self-Doubly Orthogonal Code
CSO2C-SS	Convolutional Self-Doubly Orthogonal Code-Strict Sense
CSO2C-WS	Convolutional Self-Doubly Orthogonal Code-Wide Sense
MAP	Maximum <i>A posteriori</i> Probability
PAM	Pulse Amplitude Modulation
PCSO2C	Punctured Convolutional Self-Doubly Orthogonal Code
PCSO2C-SS	Punctured Convolutional Self-Doubly Orthogonal Code-Strict Sense
PCSO2C-WS	Punctured Convolutional Self-Doubly Orthogonal Code-Wide Sense
PSK	Phase Shift Keying
QPSK	Quadrature Phase Shift Keying
QAM	Quadrature Amplitude Modulation
SOVA	Soft Output Viterbi Algorithm
TCM	Trellis Coded Modulation

Symboles

\mathcal{E}_s	Énergie du signal $s(t)$ déterministe, réel et à énergie finie
f_c	Fréquence de porteuse
$\mathbf{v} = [v_1 \ v_2 \ \dots \ v_n]$	Vecteur de dimension n et ses composantes
$\ \mathbf{v}\ $	Norme euclidienne du vecteur \mathbf{v}
\mathbf{M}^T	Transposée de la matrice \mathbf{M}
M	Nombre de signaux dans une constellation
\mathbf{G} ou $\mathbf{G}(D)$	Matrice génératrice d'un code
\mathbf{H} ou $\mathbf{H}(D)$	Matrice de parité d'un code
K	Longueur de contrainte d'un code convolutionnel
R	Taux de codage
J	Nombre de connexions d'un code convolutionnel systématique de taux $R = \frac{1}{2}$
\mathbf{P}	Matrice de perforation
$L(\xi)$	Logarithme du rapport de vraisemblance de la variable aléatoire binaire ξ
$Q(x)$	$\frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-\frac{t^2}{2}) dt, \quad x \geq 0$
$P(A)$	Probabilité de l'événement A
$\delta_{i,j}$	1 si $i = j$, 0 sinon
\oplus	Somme modulo 2
\diamond	Opérateur add-min
$[x]$	Le plus grand entier inférieur à x

LISTE DES ANNEXES

Annexe I:	Performances de PCSO2C-WS optimaux	135
Annexe II:	Performances de certains PCSO2C-SS	149
Annexe III:	Représentation vectorielle des signaux	152

CHAPITRE 1

INTRODUCTION

1.1 Motivations

Depuis quelques années maintenant nous assistons à l'essor d'une théorie somme toute récente et des techniques qui lui sont associées. Les télécommunications sont entrées dans notre vie courante par la grande porte et aujourd'hui plus personne ne pourrait se passer qui du téléphone, qui du fax, qui de l'Internet. Les développements les plus spectaculaires sont à mettre d'une part sur le compte de la réseautique en général et plus particulièrement de l'Internet et d'autre part sur celui de la téléphonie cellulaire. On devrait à plus ou moins long terme assister à la convergence de ces deux technologies dans ce que l'on appelle déjà l'Internet mobile. Les télécommunications par satellite attendent quant à elles leur tour afin de jouer le rôle qu'elles méritent sur la grande scène mondiale des télécommunications.

Toutefois, un des facteurs majeurs limitant l'expansion de ces technologies est la largeur de bande disponible. Tant en ce qui concerne les réseaux fixes que les réseaux mobiles, elle constitue une ressource qui coûte très cher. Ainsi, l'un des défis majeurs des télécommunications à l'heure actuelle est de rentabiliser au maximum cette ressource dans le but de disposer du plus de capacité de transmission possible.

La technique du codage de canal est un élément indispensable à tout système de télécommunications numériques. Cette technique permet de lutter contre les erreurs qui apparaissent lors de la transmission et qui sont dues au bruit présent dans le canal de communication. Cette protection, qui permet une économie de puissance de transmission, est en contrepartie coûteuse en largeur de bande étant donné qu'elle nécessite également l'envoi de données supplémentaires que le décodeur devra exploiter de manière intelligente.

La technique de codage de canal la plus puissante à l'heure actuelle s'appelle

“codage turbo”. Elle fut découverte en 1993 par Messieurs Berrou et Glavieux, de l’École Nationale Supérieure des Télécommunications de Bretagne. Ses performances exceptionnelles en termes de probabilité d’erreur sont toutefois obtenues au prix d’une grande complexité matérielle et d’un délai conséquent, tant au niveau du codage que du décodage. À ces inconvénients s’ajoute une grande complexité théorique qui laisse à l’heure actuelle de nombreuses questions sans réponses.

Une évolution majeure a été apportée en 1997 par Messieurs Gagnon et Haccoun, respectivement de l’École de Technologie Supérieure et de l’École Polytechnique, qui proposèrent de supprimer les entrelaceurs tant au niveau du codage que du décodage et d’utiliser un algorithme de décodage plus simple que ceux mis en œuvre au sein des codes turbo. Une propriété de décorrélation intrinsèque aux nouveaux codes utilisés, appelés codes doublement orthogonaux, permettait d’obtenir d’excellentes performances en termes de probabilité d’erreur, certes moins bonnes que celles offertes par les codes turbo à faible rapport signal sur bruit, mais obtenues au prix d’une complexité matérielle bien moindre.

Toutefois, il subsiste encore quelques limitations à cette nouvelle technique qui n’est pas encore arrivée à maturité. En effet, même si la complexité a été réduite au niveau matériel, il reste que ce système de codage n’a été étudié que dans des schémas où la quantité de données transmise sur le lien de communications (et donc la largeur de bande) est doublée.

L’objectif de ce travail de recherche est donc d’étudier les performances de cet algorithme de décodage dans un environnement où l’on cherche à augmenter la largeur de bande “utile” du canal sur lequel on transmet l’information tout en conservant une latence au codage et au décodage faible. Cet objectif va être réalisé de deux manières: d’une part en augmentant le taux de codage des codes doublement orthogonaux utilisés et en déterminant quels sont ceux qui possèdent de bonnes propriétés intrinsèques pour être utilisés à un taux de codage donné; et d’autre part en insérant ces codes de taux élevés dans un schéma où des modulations multiniveaux sont employées.

1.2 Organisation du mémoire

Ce mémoire est structuré comme suit:

Le chapitre 2 résume les notions indispensables à la compréhension de ce mémoire en ce qui concerne des éléments de la théorie du codage de canal. Après une brève introduction et un rappel historique, on présente les deux grandes classes de codes que sont les codes en bloc et les codes convolutionnels. Ensuite, la technique de codage et de décodage turbo est brièvement exposée.

Le chapitre 3 présente la technique de décodage à seuil itératif des codes convolutionnels doublement orthogonaux comme une amélioration du décodage itératif turbo en termes de complexité. L'algorithme de décodage ainsi que les propriétés de double orthogonalité sont développés en détail.

Le chapitre 4 regroupe les différents travaux qui ont été entrepris en vue de la détermination de codes convolutionnels doublement orthogonaux de taux élevés. Une étude exhaustive de la performances de ces codes est présentée ainsi que leur sensibilité aux différents paramètres.

Le chapitre 5 concerne la mise en application des codes convolutionnels doublement orthogonaux de taux élevés dans un système où l'on utilise des modulations à efficacités spectrales élevées. Une modification du décodeur est présentée et l'influence des différents paramètres est étudiée.

Enfin, le chapitre 6 présente une conclusion sur l'ensemble des travaux qui ont été entrepris et propose quelques voies de recherche qui mériteraient d'être approfondies dans le cadre de travaux futurs.

1.3 Contributions

Les contributions apportées par ce travail de recherche sont les suivantes:

- Étude théorique du comportement des codes doublement orthogonaux perforés et détermination exhaustive de codes optimaux au sens de la probabilité d'erreur.
- Développement d'une nouvelle méthode de représentation spectrale des codes doublement orthogonaux perforés.
- Conception pratique d'un simulateur permettant l'étude des performances des codes de taux élevés et la validation des résultats théoriques.
- Mise en œuvre d'un calculateur de métrique pour le décodeur à seuil itératif afin de lui permettre de s'insérer dans un schéma où sont utilisées des modulations multiniveaux.
- Implémentation d'un simulateur permettant d'observer le comportement des codes doublement orthogonaux perforés utilisés conjointement avec des modulations M -QAM.

D'un point de vue pratique, toutes les simulations ont été effectuées sur un ordinateur personnel cadencé à 800 MHz et possédant 128 Mo de RAM. Les logiciels de programmation Visual[©] C++ ainsi que Matlab[©] ont servi à l'implémentation des différentes procédures sous environnement Windows[©].

CHAPITRE 2

ÉLÉMENTS DE THÉORIE DU CODAGE

2.1 Généralités

2.1.1 Éléments d'un système de communications numériques

La figure 2.1 présente, sous forme de diagramme fonctionnel, les éléments de base que l'on retrouve dans tout système de communications numériques.

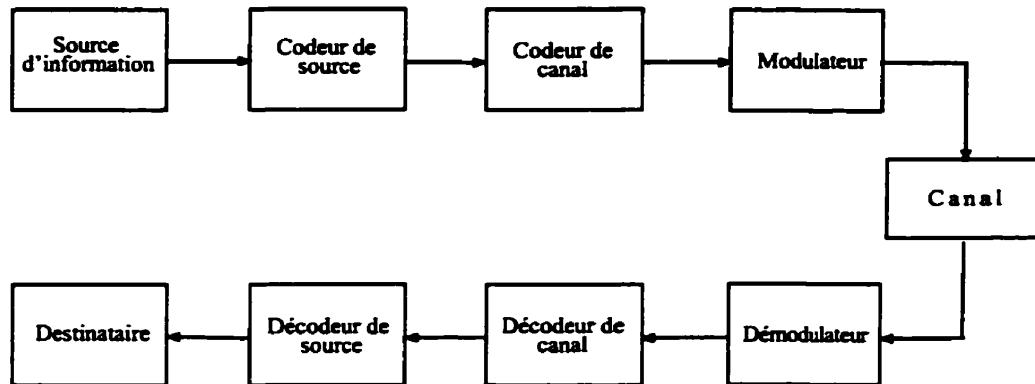


Figure 2.1: Éléments de base d'un système de communications numériques

La *source d'information* peut être soit analogique (par exemple un signal audio ou vidéo) soit discrète (par exemple les lettres de l'alphabet latin). En communications numériques, les messages produits par la source sont transformés en une suite de symboles binaires. Or, dans la pratique, on souhaite représenter ces messages avec le moins de symboles possibles en supprimant les redondances inhérentes à toute source. Cette tâche, appelée **compression**, incombe au *codeur de source*. Nous ne traiterons pas ce type de codage dans ce mémoire.

La séquence binaire d'information présente à la sortie du codeur de source doit alors être transmise dans le canal de transmission vers le destinataire. Le rôle du *codeur de canal* est d'ajouter de manière contrôlée de la redondance à cette séquence binaire transmise de telle manière que le décodeur puisse lutter contre les erreurs dues au bruit et aux interférences présentes dans le canal. Il existe deux grandes classes de codes: les codes en bloc et les codes convolutionnels présentés respectivement aux sections 2.2 et 2.3.

La séquence binaire ainsi codée est transmise au *modulateur numérique* qui sert d'interface entre le monde numérique (codeurs) et le monde analogique (canal). Son rôle principal est d'assigner à toute séquence binaire de longueur définie par le type de modulation un signal électrique qui sera transmis dans le canal.

Le *canal de transmission* est le support physique qui permet la transmission de l'information de l'émetteur au récepteur. Ce canal peut être l'atmosphère ambiante (dans le cas des communications sans fil), un fil de téléphone, un lien micro-ondes ou encore une fibre optique. Le signal véhiculé par ce canal est déformé du fait de nombreux phénomènes, tels le bruit thermique généré entre autres par les équipements électroniques ou les bruits atmosphériques (par exemple la foudre).

À la réception, les opérations inverses sont effectuées. Le *démodulateur* assigne à chaque signal reçu (déformé) une séquence de symboles binaires, le *décodeur de canal* tente de reconstruire la séquence binaire émise à l'origine par le codeur de source en utilisant la redondance initialement introduite par le codeur de canal. Une mesure de la qualité du système de codage/décodage de canal est la proportion d'erreur qui subsiste après l'opération de décodage, soit la probabilité d'erreur par bit. Enfin le *décodeur de source* reconstruit le message de source (discret ou analogique) correspondant à la séquence d'information fournie par le décodeur de canal et la transmet au destinataire.

Dans la suite de ce chapitre, après un bref rappel historique, nous allons introduire les fondements nécessaires à la compréhension de ce mémoire en ce qui

a trait aux techniques de codage de canal. Diverses techniques de modulations numériques seront abordées au chapitre 5. Quant aux autres éléments présentés dans cette section, on pourra se documenter à leur sujet en consultant divers ouvrages généraux de télécommunications numériques comme [27], [40].

2.1.2 Historique

Le premier système de télécommunications numériques à voir le jour fut le télégraphe. Il fut développé dès 1837 par Morse qui inventa un code binaire à longueur variable pour coder les lettres de l'alphabet anglais (codeur de source). Mais le véritable point de départ des télécommunications numériques modernes réside dans les travaux de Nyquist [25] qui chercha le débit maximum de signaux que l'on peut transmettre sur un câble de télégraphe de bande passante donnée W qui permettait d'éviter les interférences entre symboles. Ce taux, de valeur $2W$, est aujourd'hui appelé taux de Nyquist.

Ce résultat de Nyquist, parmi d'autres, permit à Shannon [33], [34] d'établir les fondements mathématiques de la transmission de l'information et de trouver les limites fondamentales de tout système de communications numériques. Dans ses travaux, Shannon posa de manière formelle le problème d'une transmission fiable de l'information en des termes probabilistes. Il démontra entre autres qu'un système de télécommunications ayant une puissance moyenne d'émission P et une bande passante W données et subissant l'effet d'un bruit additif peut être caractérisé par un paramètre appelé *capacité* s'appliquant au canal de transmission. Dans le cas d'un bruit blanc additif gaussien de densité spectrale de puissance N_0 , la capacité C est donnée en bits/s par:

$$C = W \log_2 \left(1 + \frac{P}{WN_0} \right) \quad (2.1)$$

La signification de la capacité est la suivante: si le débit d'information à la sortie de la source (ou du codeur de source), noté D , est inférieur à C ($D < C$), alors il est théoriquement possible d'obtenir une transmission parfaite (sans erreur)

à travers le canal et ceci par un codage de canal approprié. À l'inverse, si $D > C$, une transmission parfaite de l'information est impossible quelque soit le traitement effectué à la sortie du canal.

À la suite des publications de Shannon, beaucoup de travaux s'orientèrent vers la recherche de codes dits en bloc et sur les techniques de décodage algébrique associées dans le but de s'approcher le plus possible de la capacité. Nous allons présenter brièvement quelques notions relatives à cette technique de codage.

2.2 Codes en bloc

Les premiers travaux concernant les codes en bloc sont ceux de Hamming [18]. Ils furent les premiers d'une longue série parmi lesquels on peut citer ceux de Muller [24], Reed [28], Reed et Solomon [29], ou Bose et Ray-Chaudhuri [8], [9].

Un code en bloc est constitué d'un ensemble de vecteurs d'éléments binaires de longueur constante appelés *mots de code*. La longueur des mots de code est notée n . Il y a 2^n vecteurs binaires possibles pour un code en bloc de longueur n . Parmi ces 2^n vecteurs binaires, on peut en sélectionner 2^k , où $k \leq n$, pour former un code. Ainsi, à chaque bloc de k bits d'information on assigne un mot de code de longueur n choisi parmi les 2^k mots de code. On notera un tel code un code (n, k) et le rapport $R = \frac{k}{n}$ est appelé taux de codage et s'exprime en bits par symbole codé.

Soient $u_{m,1}, u_{m,2}, \dots, u_{m,k}$ les k bits d'information que l'on encode par le mot de code \mathbf{C}_m . On peut représenter ces vecteurs de la manière suivante:

$$\mathbf{U}_m = [u_{m,1} \ u_{m,2} \ \dots \ u_{m,k}] \quad (2.2)$$

et:

$$\mathbf{C}_m = [c_{m,1} \ c_{m,2} \ \dots \ c_{m,n}] \quad (2.3)$$

L'opération d'encodage peut être décrite par un ensemble de n équations de la

forme:

$$c_{m,j} = u_{m,1}g_{1,j} \oplus u_{m,2}g_{2,j} \oplus \dots \oplus u_{m,k}g_{k,j}, \quad j = 1, 2, \dots, n \quad (2.4)$$

où $g_{i,j} = 0$ ou 1 . Ce système d'équations linéaires peut être écrit sous forme matricielle de la manière suivante:

$$\mathbf{C}_m = \mathbf{U}_m \mathbf{G} \quad (2.5)$$

où \mathbf{G} est la *matrice génératrice* du code:

$$\mathbf{G} = \begin{pmatrix} g_{1,1} & g_{1,2} & \dots & g_{1,n} \\ g_{2,1} & g_{2,2} & \dots & g_{2,n} \\ \vdots & \vdots & \dots & \vdots \\ g_{k,1} & g_{k,2} & \dots & g_{k,n} \end{pmatrix} \quad (2.6)$$

Un code en bloc sera dit *systématique* si les mots de code se présentent sous la forme suivante:

$$[c_{m,1} \ c_{m,2} \ \dots \ c_{m,n}] = [u_{m,1} \ u_{m,2} \ \dots \ u_{m,k} \ p_{m,1} \ p_{m,2} \ \dots \ p_{m,n-k}] \quad (2.7)$$

La matrice génératrice d'un code en bloc systématique a la forme suivante:

$$\mathbf{G} = [\mathbf{I}_k \ \Gamma] \quad (2.8)$$

où \mathbf{I}_k est la matrice identité de taille $k * k$ et Γ est une matrice de taille $k * (n - k)$.

On peut associer à chaque code (n, k) son code *dual* qui possède 2^{n-k} mots de code (c'est un code $(n, n - k)$) orthogonaux aux mots de codes \mathbf{C}_m du code (n, k) . La matrice génératrice de ce code dual, appelée *matrice de parité* et notée \mathbf{H} , est donc telle que pour tout mot de code \mathbf{C}_m du code (n, k) :

$$\mathbf{C}_m \mathbf{H}^T = \mathbf{0} \quad (2.9)$$

Il s'ensuit que:

$$\mathbf{G}\mathbf{H}^T = \mathbf{0} \quad (2.10)$$

Dans le cas d'un code systématique dont la matrice génératrice est donnée par (2.8), la matrice de parité s'écrit:

$$\mathbf{H} = [\mathbf{\Gamma}^T \quad \mathbf{I}_{n-k}] \quad (2.11)$$

Nous ne présenterons pas ici plus de propriétés concernant les codes en bloc et nous renvoyons le lecteur à des ouvrages comme [22] pour plus de renseignements.

Une première évolution fondamentale de la technique de codage est apparue avec le développement des codes convolutionnels par Elias [13] et des algorithmes de décodage associés par Wozencraft et Rieffen [41], Fano [14], Jelinek [19] et Viterbi [37] [38]. Nous allons présenter dans la section suivante les définitions de base concernant ce type de codes.

2.3 Codes convolutionnels

2.3.1 Définition

Un code convolutionnel est généré par le passage de la séquence d'information (de manière continue et non plus en blocs) dans k registres à décalage de longueurs finies qui forment le codeur. Ceux-ci se composent de $K - 1$ délais et sont connectés à n additionneurs modulo 2, comme montré à la figure 2.2. K est appelé la *longueur de contrainte* du code.

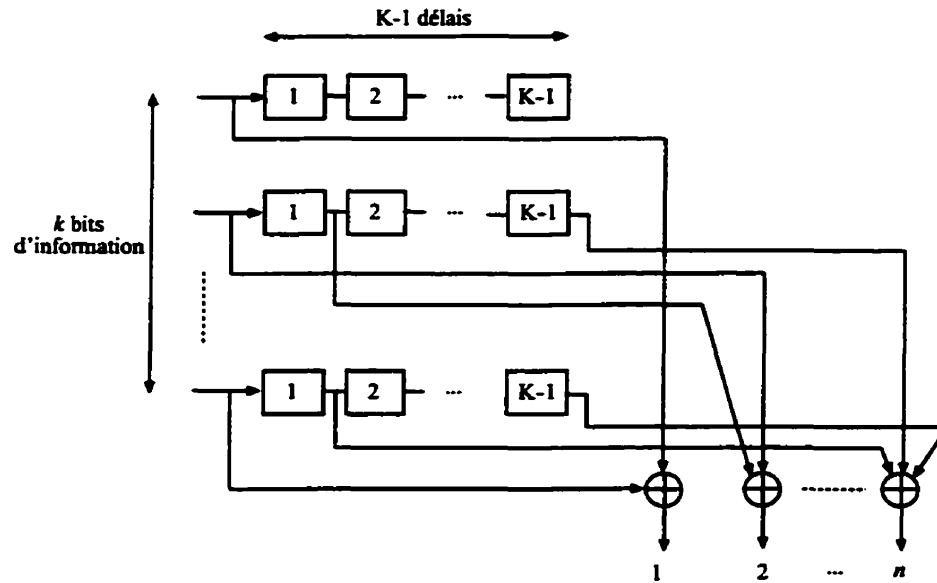


Figure 2.2: Codeur convolutionnel

Les éléments binaires de la séquence d'information entrent dans le codeur convolutionnel par séquences de k bits (k est en général faible) et le nombre de sorties pour chacune de ces séquences est de n . Ainsi, le taux de codage est défini comme étant $R = \frac{k}{n}$.

La matrice génératrice d'un code convolutionnel, si on la conçoit de la même manière que celle d'un code en bloc, est semi-infinie du fait que la séquence d'entrée est semi-infinie. On peut éviter ce problème en définissant différemment la matrice génératrice dans le cas d'un code convolutionnel. Elle sera entièrement spécifiée par k matrices de n vecteurs (un pour chaque additionneur modulo 2), chacun ayant K dimensions et contenant les connexions des additionneurs modulo 2 aux registres. Un vecteur a pour composantes 1 ou 0. Un 1 signifie que l'additionneur modulo 2 correspondant est connecté à l'étage du registre à décalage concerné. Au contraire, un 0 signifie l'absence de connexion entre ces deux entités. Ainsi:

$$\mathbf{G}^{(i)} = [\mathbf{g}_1^{(i)} \quad \mathbf{g}_2^{(i)} \quad \dots \quad \mathbf{g}_n^{(i)}], \quad i = 1, 2, \dots, k \quad (2.12)$$

et:

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}^{(1)} \\ \vdots \\ \mathbf{G}^{(k)} \end{pmatrix} \quad (2.13)$$

On peut représenter les générateurs sous forme polynomiale de la manière suivante:

$$\mathbf{g}_j^{(i)}(D) = \sum_{l=0}^{K-1} g_{j,l}^{(i)} D^l, \quad j = 1, 2, \dots, n, \quad i = 1, 2, \dots, k \quad (2.14)$$

où $g_{j,l}^{(i)} = 1$ si le j -ème additionneur modulo 2 est relié après le l -ème délai du i -ème registre à décalage et 0 sinon.

À ce moment-là, si l'on représente également les séquences d'information et les séquences codées sous forme polynomiale, c'est-à-dire:

$$\mathbf{u}(D) = [\mathbf{u}^{(1)}(D) \ \mathbf{u}^{(2)}(D) \ \dots \ \mathbf{u}^{(k)}(D)] \quad (2.15)$$

où $\mathbf{u}^{(i)}(D)$ est la partie de la séquence d'information codée par le i -ème registre et:

$$\mathbf{c}(D) = [\mathbf{c}_1(D) \ \mathbf{c}_2(D) \ \dots \ \mathbf{c}_n(D)] \quad (2.16)$$

où $\mathbf{c}_j(D)$ est la sortie codée du j -ème additionneur modulo 2, on a:

$$\mathbf{c}(D) = \mathbf{u}(D)\mathbf{G}(D) \quad (2.17)$$

Dans le cas d'un code convolutionnel systématique, c'est-à-dire un code pour lequel on retrouve la séquence d'information à la sortie du codeur accompagnée de symboles de parité, on a:

$$\mathbf{G}(D) = [\mathbf{I} \ \mathbf{T}(D)] \quad (2.18)$$

Illustrons ces notions à partir du code convolutionnel donné à la figure 2.3.

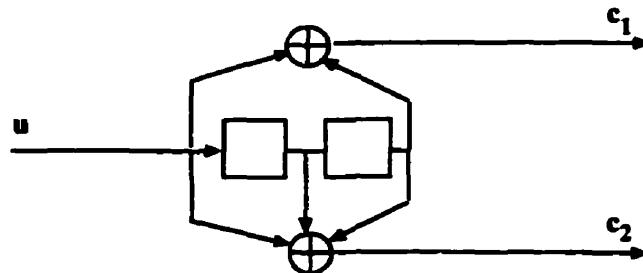


Figure 2.3: Exemple de codeur convolutionnel ($R = \frac{1}{2}$, $K = 3$, $k = 1$)

Pour ce code:

$$\mathbf{g}_1 = [101] \quad (2.19)$$

$$\mathbf{g}_2 = [111] \quad (2.20)$$

$$\mathbf{G} = [101 \ 111] \quad (2.21)$$

Sous forme polynomiale, on a:

$$\mathbf{g}_1(D) = 1 + D^2 \quad (2.22)$$

$$\mathbf{g}_2(D) = 1 + D + D^2 \quad (2.23)$$

et:

$$\mathbf{G}(D) = [1 + D^2 \ 1 + D + D^2] \quad (2.24)$$

De la même manière que l'on a défini la matrice de parité d'un code en bloc, on peut définir la matrice de parité d'un code convolutionnel par la relation suivante:

$$\mathbf{G}(D)\mathbf{H}^T(D) = \mathbf{0} \quad (2.25)$$

2.3.2 Types de représentation

Il existe trois manières de représenter le processus d'encodage d'un code convolutionnel: par son arbre, son treillis ou par son diagramme d'état. Nous allons développer ces trois types de représentation à travers l'exemple du codeur donné à la figure 2.4.

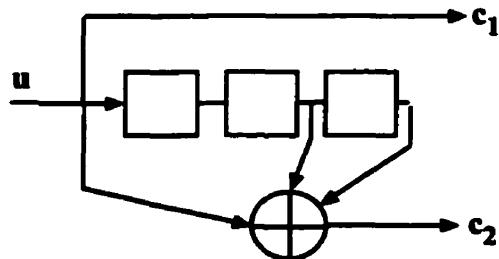
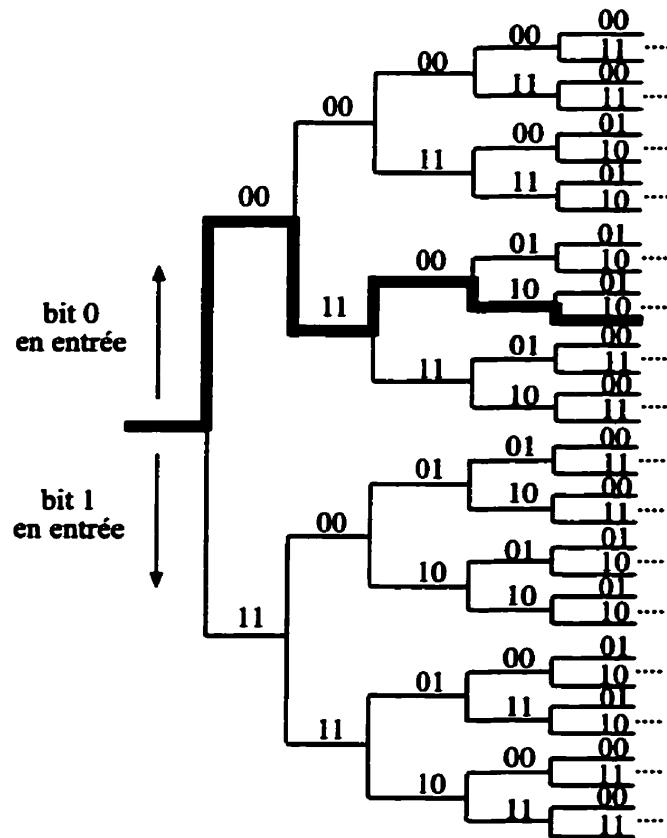


Figure 2.4: Exemple de codeur convolutionnel ($R = \frac{1}{2}$, $K = 4$, $k = 1$, $\mathbf{g}_1 = [1000]$, $\mathbf{g}_2 = [1011]$)

L'*arbre* correspondant à un code convolutionnel donné est un graphe pour lequel 2^k branches émanent de chaque noeud, chacune de ces branches correspondant à une séquence d'entrée de k bits possible. On écrit alors sur la branche les n symboles codés de sortie correspondant à cette séquence de k bits. Pour connaître la séquence codée correspondant à une séquence d'information donnée, la règle est de choisir la branche correspondant aux k bits présents en entrée et de répéter l'opération jusqu'à épuiser tous les bits appartenant à la séquence d'information. On parcourt ainsi un chemin spécifique à travers l'arbre pour obtenir la séquence de sortie. Par exemple, pour le codeur dessiné à la figure 2.4, l'arbre correspondant et un chemin spécifique sont donnés à la figure 2.5.



— Séquence d'entrée : 01011
Séquence de sortie : 0011001010

Figure 2.5: Arbre correspondant à l'exemple de la figure 2.4

Une observation attentive de l'arbre de la figure 2.5 révèle qu'il possède une certaine redondance à partir du quatrième étage (ce qui est cohérent avec le fait que $K = 4$). En fait, les n symboles codés en sortie sont déterminés par les k bits courants en entrée et par les $k(K - 1)$ bits précédents, c'est-à-dire les bits contenus dans les $k(K - 1)$ premières cases des registres. En effet, les derniers bits présents dans les registres à décalage sont éliminés lors de l'entrée des bits courants et donc n'interviennent pas dans le calcul des symboles codés de sortie. Ainsi, on peut dire que la séquence de sortie est déterminée par les k bits en entrée et l'un des $2^{k(K-1)}$ états du code. On peut donc obtenir une structure simplifiée pour la représentation du code en éliminant tous les noeuds de l'arbre étant dans le même état, car ils conduisent aux mêmes symboles en sortie lorsqu'ils sont soumis aux mêmes bits d'entrée. En faisant ceci, on obtient la structure dite en *treillis* du code, qui est plus compacte que la représentation en arbre. Le treillis correspondant au code de la figure 2.4 est représenté à la figure 2.6.

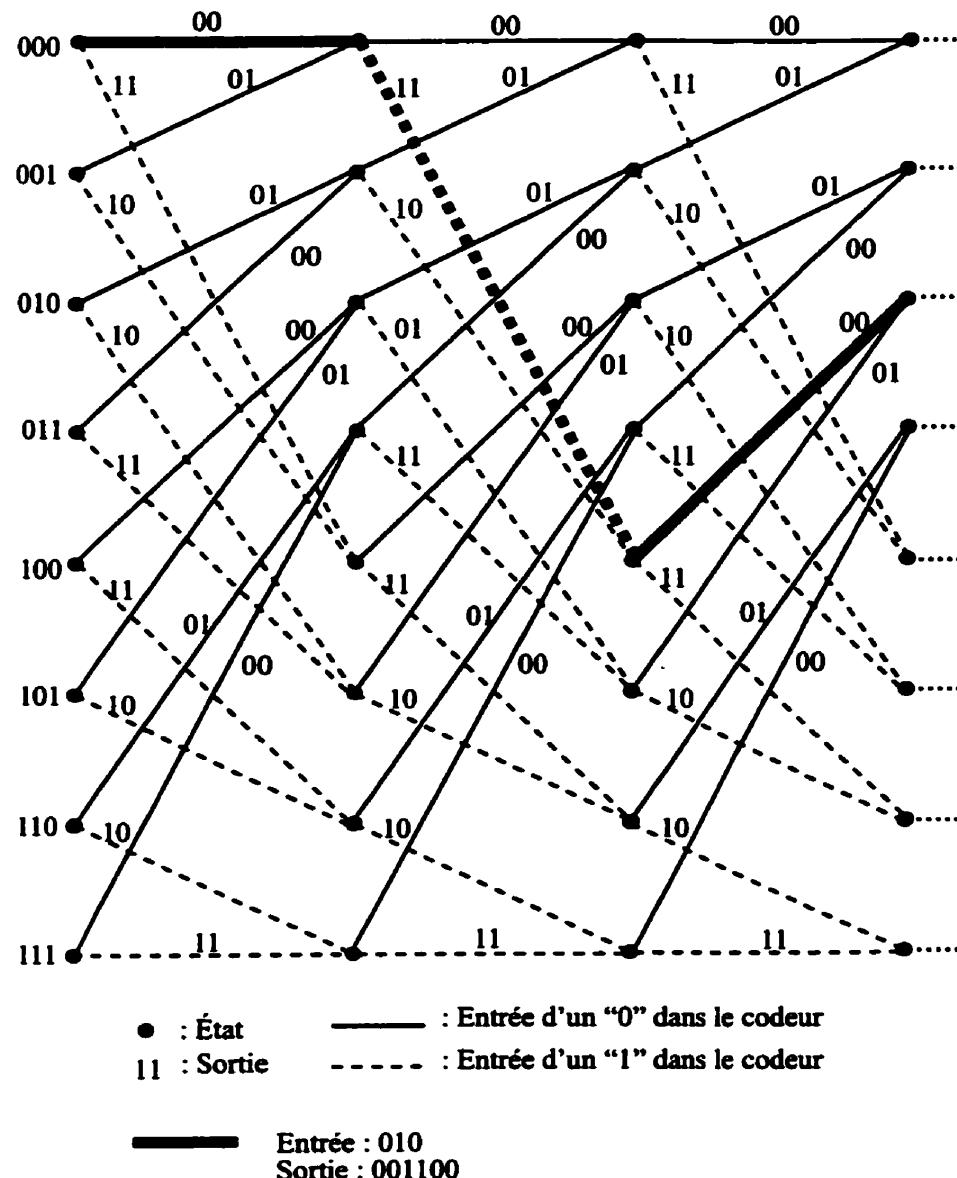


Figure 2.6: Treillis correspondant à l'exemple de la figure 2.4

Le treillis de la figure 2.6 comporte huit noeuds à chaque étage car le code représenté à la figure 2.4 possède $2^3 = 8$ états. Après le deuxième étage, chaque noeud possède quatre arcs incidents (deux qui mènent au noeud et deux qui en partent).

On a vu que la sortie du codeur ne dépend que de son état et de l'entrée. Une manière encore plus compacte de représenter le code est donc sous la forme d'un *diagramme d'état*, qui est simplement un graphe où les noeuds représentent tous les états possibles et les arcs représentent les transitions possibles entre les états. Toutefois, il faut noter que la représentation temporelle disparaît dans cette représentation. La représentation en diagramme d'état du code de la figure 2.4 est donnée à la figure 2.7.

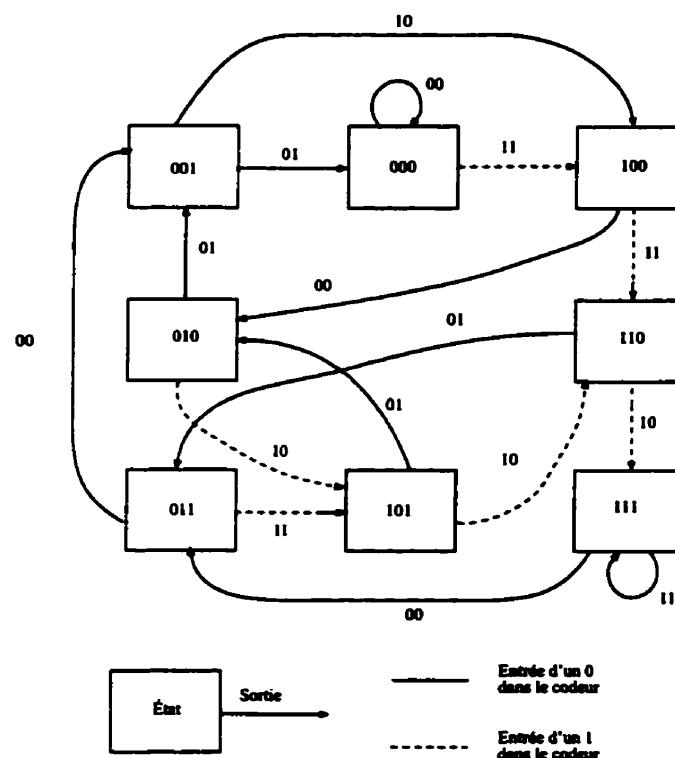


Figure 2.7: Diagramme d'état correspondant à l'exemple de la figure 2.4

Il existe d'autres manières de représenter les codes convolutionnels, comme par exemple par fonction de transfert, mais nous ne développerons pas ce genre de représentation ici.

Pour finir, nous allons présenter le principe de l'algorithme de décodage optimal des codes convolutionnels: l'algorithme de Viterbi [37].

2.3.3 Principe du décodage optimal de Viterbi

Étant donnée la séquence binaire reçue, cet algorithme consiste à rechercher le chemin le plus probable dans le treillis selon une métrique euclidienne (décision douce) ou de Hamming (décision dure). Plus précisément, la métrique de la j -ème branche du i -ème chemin au sein du treillis est définie par le logarithme de la probabilité de la séquence reçue $\mathbf{R}_j = \{r_{jm}\}$ (m est le rang du bit dans la séquence reçue) conditionnée sur la séquence transmise $\mathbf{T}_j^{(i)} = \{t_{jm}^{(i)}\}$ pour le i -ème chemin, soit:

$$\mu_j^{(i)} = \log P(\mathbf{R}_j | \mathbf{T}_j^{(i)}), \quad j = 1, 2, 3, \dots \quad (2.26)$$

Ainsi, une métrique pour le i -ème chemin composé de B branches à travers le treillis est définie par:

$$M^{(i)} = \sum_{j=1}^B \mu_j^{(i)} \quad (2.27)$$

Le chemin optimal est celui pour lequel la métrique est la plus grande. On minimise ainsi la probabilité d'erreur pour la séquence de bits initialement codée par le codeur [38].

De nombreux travaux portant sur les performances des codes convolutionnels et sur les algorithmes de décodage associés ont été entrepris. Ces années de recherches, et notamment la découverte des codes concaténés [15], ont permis l'invention en 1993 par Berrou *et al.* [7] de la technique de décodage turbo, qui

est aujourd’hui la plus performante en termes de probabilité d’erreur. Nous allons donc nous attarder un peu plus longuement sur cette technique dans la section 2.4.

2.4 La technique de décodage turbo

2.4.1 Concaténation des codes convolutionnels

Pour construire des codes convolutionnels performants, c'est-à-dire des codes pour lesquels la probabilité d'erreur après décodage optimal est faible, il est nécessaire d'avoir une longueur de contrainte K élevée. Toutefois, la complexité du décodage des codes convolutionnels croît exponentiellement avec K . Ainsi, dans le but d'obtenir des codes convolutionnels performants ne requérant pas une trop grande complexité au décodage, Forney [15] a introduit en 1966 le concept de codes concaténés.

Un code concaténé en série est tel que deux niveaux de codage sont appliqués. Le premier niveau de codage est effectué par un code appelé code extérieur tandis que le deuxième niveau de codage est effectué par un deuxième code appelé code intérieur dont le rôle est de coder les données en sortie du codeur extérieur. La faible complexité recherchée est atteinte grâce à un décodage séparé des deux codeurs. En pratique, un entrelaceur est introduit entre les codes intérieur et extérieur (et entre les décodeurs intérieur et extérieur) dans le but de mélanger les données fournies par le décodeur intérieur qui peuvent présenter des erreurs groupées. De plus, il possède d'autres propriétés qui sont détaillées à la section 2.4.3.

Dans le cas d'une concaténation parallèle, deux codeurs codent la même séquence d'information, mais celle présente à l'entrée du deuxième codeur a été préalablement entrelacée. C'est cette architecture qui est à la base de la technique du codage turbo que nous allons détailler à présent.

2.4.2 Le codage

Un codeur turbo est typiquement constitué de deux codeurs convolutionnels récursifs systématiques concaténés en parallèle séparés par un entrelaceur. La structure est représentée à la figure 2.8.

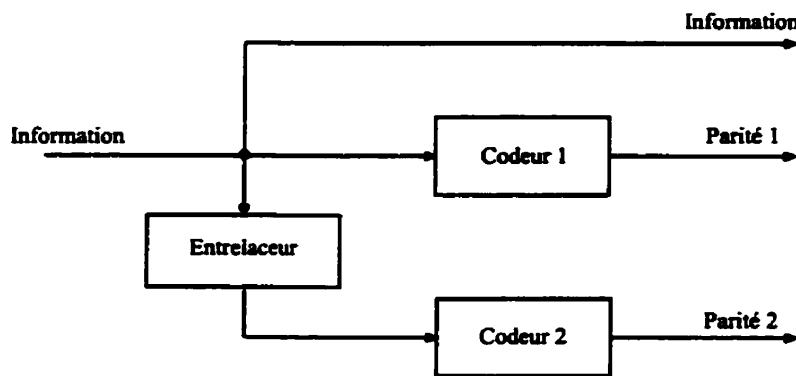


Figure 2.8: Concaténation parallèle de deux codeurs convolutionnels récursifs

La matrice génératrice d'un code convolutionnel récursif systématique de taux $R = \frac{1}{2}$ peut s'écrire sous la forme:

$$\mathbf{G}(D) = [1 \quad \frac{\mathbf{g}_1(D)}{\mathbf{g}_0(D)}] \quad (2.28)$$

où $\mathbf{g}_0(D)$ et $\mathbf{g}_1(D)$ sont les polynômes générateurs concernant respectivement la rétroaction et le calcul direct des symboles de parité.

Il est à noter que dans le cas du deuxième codeur, seuls les symboles de parité sont effectivement transmis étant donné que la séquence d'information est identique pour les deux codeurs (modulo l'entrelacement).

2.4.3 L'entrelaceur

Un entrelaceur de longueur N est défini comme étant une permutation de N éléments sans répétition. Il réalise donc de manière déterministe ou pseudo-aléatoire la modification du séquencement des symboles d'information.

Sa première fonction est de générer un code dont les performances sont équivalentes à celles d'un code de longueur de contrainte élevée tout en utilisant des codes constituants ayant une faible longueur de contrainte. Sa deuxième fonction est de décorrélérer les entrées des deux codeurs de telle manière qu'un algorithme de décodage sous-optimal basé sur l'échange d'information entre deux décodeurs puisse être mis en œuvre. En effet, si les séquences d'entrée des deux décodeurs sont décorrélées, il y a une forte probabilité pour qu'après correction de certaines erreurs par le premier décodeur, certaines erreurs résiduelles soient corrigibles par le deuxième décodeur (le deuxième décodeur bénéficie des corrections apportées par le premier décodeur).

2.4.4 La structure du décodeur itératif

Le décodeur itératif turbo est constitué de deux décodeurs concaténés en série séparés par un entrelaceur, identique à celui présent dans le codeur turbo. Il est représenté à la figure 2.9.

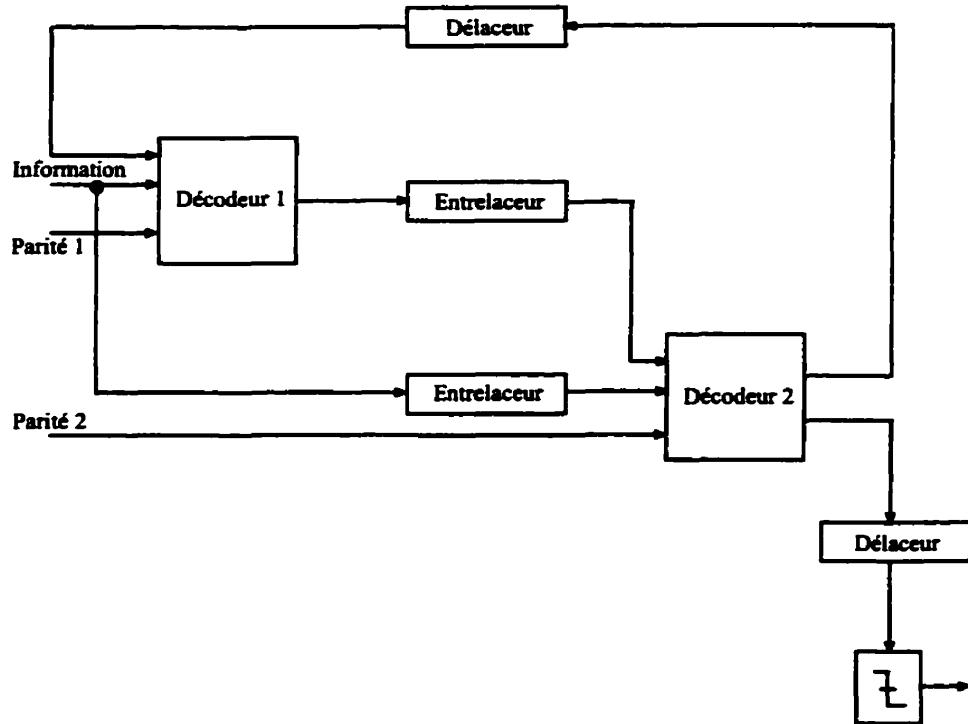


Figure 2.9: Décodeur itératif turbo

Le premier décodeur reçoit en entrée la séquence d'information reçue et la séquence de parité reçue qui avait été générée par le premier codeur. Ce décodeur produit alors une estimation de la séquence d'information émise, appelée *information extrinsèque*, qui est entrelacée et utilisée par le deuxième décodeur pour produire une estimation affinée des probabilités *a priori* de la séquence d'information émise.

Les deux autres entrées du second décodeur sont la séquence d'information reçue entrelacée et la séquence de parité reçue qui avait été générée par le deuxième codeur. Le second décodeur génère lui aussi une information extrinsèque qui est utilisée à l'itération suivante par le premier décodeur pour améliorer l'estimation de la probabilité *a priori* de la séquence d'information émise faite par le premier décodeur à l'itération précédente. Cette structure itérative permet d'améliorer grandement les performances par rapport à un décodeur qui n'utilisera qu'une

seule itération. De ce fait, la boucle de retour est un élément fondamental et caractéristique de la technique de décodage turbo d'où elle tire son nom.

Après un certain nombre d'itérations, les estimations des deux décodeurs ne s'améliorent presque plus d'une itération à l'autre. À ce moment-là, le décodeur effectue une décision dure sur la séquence décodée après l'avoir délacée.

2.4.5 Les algorithmes de décodage

À la vue de ce que l'on vient d'énoncer à la section 2.4.4, les algorithmes de décodage utilisés par les deux décodeurs doivent être des algorithmes à décision pondérée, c'est-à-dire fournissant une probabilité *a posteriori* pour chaque symbole décodé. L'algorithme de Viterbi [37] a été modifié [17], [21] pour produire des estimations pondérées sur les symboles reçus et peut donc être utilisé par le décodeur turbo. C'est ce que l'on nomme le SOVA (Soft Output Viterbi Algorithm). Comme l'algorithme de Viterbi, le SOVA minimise la probabilité d'erreur par séquence transmise. Un autre critère d'optimisation peut être la minimisation de la probabilité d'erreur par bit. L'algorithme réalisant ce critère d'optimisation est l'algorithme MAP. C'est l'adaptation que Bahl *et al.* [3] ont fait de cet algorithme qui est le plus couramment utilisé dans la technique de décodage turbo. Nous ne rentrerons pas plus en détail dans ces algorithmes développés dans des ouvrages plus spécialisés comme [39].

CHAPITRE 3

DÉCODAGE À SEUIL ITÉRATIF ET CODAGE CONVOLUTIONNEL DOUBLEMENT ORTHOGONAL

Nous avons présenté à la section 2.4 la technique de codage et de décodage turbo. Nous allons analyser dans ce chapitre quels sont ses avantages mais nous allons également mettre l'accent sur ses faiblesses et sur une technique qui a été proposée en 1997 [16] pour tenter d'y remédier.

3.1 Forces et faiblesses du décodage turbo

3.1.1 Avantages de la technique

Comme nous l'avons déjà mentionné dans la section 2.4, la technique du décodage turbo [7] est la plus performante à l'heure actuelle en termes de probabilité d'erreur. Originellement, on pouvait obtenir une probabilité d'erreur par bit de 10^{-5} pour un rapport signal à bruit de 0.7 dB (soit à 0.7 dB de la capacité du canal), en utilisant deux codeurs de longueur de contrainte $K = 5$, un entrelaceur de 65536 bits et un décodage par l'algorithme MAP utilisant 18 itérations. Ces performances, ainsi que sa robustesse (notamment dans des canaux très bruités), tiennent essentiellement à deux facteurs:

- un décodage itératif;
- une décorrélation des observables présentées aux deux codeurs obtenue grâce à l'entrelaceur.

Ces avantages (non exhaustifs) font que la technique de décodage turbo sera utilisée dans de nombreux systèmes de télécommunications, comme par exemple en téléphonie cellulaire dans la nouvelle norme CDMA de la troisième génération. Toutefois, cette technique présente deux inconvénients majeurs que nous allons

mettre en évidence dans la section suivante.

3.1.2 Inconvénients

On distingue principalement deux faiblesses du décodage turbo qui peuvent limiter son champ d'applications:

- **La complexité de l'algorithme de décodage**

On a exposé à la section 2.4.5 les deux algorithmes de décodage principalement utilisés par les deux décodeurs lors du décodage turbo: l'algorithme SOVA mais surtout l'algorithme MAP. Ce dernier souffre d'une grande complexité du point de vue du nombre d'opérations qu'il doit effectuer et donc du temps de calcul nécessaire à son exécution. On estime en effet que l'algorithme MAP est environ quatre fois plus complexe que l'algorithme de Viterbi.

- **Le délai induit par chaque itération**

D'après la figure 2.9, on peut observer qu'une itération du décodage turbo utilise deux décodeurs, un entrelaceur et un délaveur. Ainsi, si l'on note par $\delta_e(N)$ le délai imputable à un entrelaceur de N bits (égal à celui imputable à un délaveur de N bits) et par δ_d le délai induit par l'algorithme de décodage, on obtient une estimation de la durée du décodage δ en écrivant:

$$\delta = 2I(\delta_d + \delta_e(N)) \quad (3.1)$$

où I est le nombre d'itérations. Or, l'obtention de bonnes performances est conditionnée par une taille d'entrelaceur et par un nombre d'itérations élevés, ce qui par conséquent entraîne un temps de décodage lui aussi élevé.

3.1.3 Évolutions apportées

Pour tenter de remédier à ces inconvénients, diverses solutions portant sur des parties du décodeur ont été envisagées, parmi lesquelles on peut citer:

- **Modifications de l'algorithme de décodage**

De nombreux chercheurs se sont attaqués au problème de l'algorithme MAP dans le but de le simplifier. On peut citer entre autres le travail de Robertson [30] ou l'introduction de la procédure log-MAP [6], [26], [31].

- **Influence de la structure de l'entrelaceur**

Diverses structures d'entrelaceurs ont été étudiées (entrelaceurs déterministes ou pseudo-aléatoires) pour tenter d'améliorer la décorrélation des observables tout en maintenant une longueur réduite, car d'elle dépend en partie le délai au décodage. On pourra se reporter à [32] pour plus de détails.

3.1.4 Amélioration envisagée

Les évolutions mentionnées dans la section précédente n'ont pas remis fondamentalement en cause l'architecture même du décodage turbo mais ont plutôt cherché à améliorer la structure existante pour tenter de contrer les effets négatifs que sont la complexité et le délai.

Une amélioration a été apportée en 1997 [12], [16] dans laquelle non seulement l'algorithme de décodage mais aussi la structure même du décodage turbo ont été remis en cause. En effet, pour remédier aux deux inconvénients mis en évidence dans la section précédente, l'algorithme MAP a été remplacé par l'algorithme de décodage à seuil qui est un autre type de décodage par symbole de complexité bien plus faible. En outre, les entrelaceurs ont été éliminés de telle sorte que

cette source de délai au codage et au décodage est éliminée. Toutefois, pour maintenir des performances d'erreur proches de celles du décodage turbo, il a fallu d'une part conserver la structure itérative au décodage et d'autre part préserver l'indépendance entre les observables à chaque itération sans entrelacement. Nous allons revenir à la section 3.3 en détails sur cette technique qui sera développée tout au long de ce mémoire.

3.2 Décodage à seuil et codes convolutionnels orthogonaux

Avant de détailler l'amélioration apportée par [12], [16] mentionnée dans la section précédente, nous allons donner quelques rappels sur le décodage à seuil et sur la notion de codes orthogonaux.

Le concept de décodage à seuil pour les codes convolutionnels a été inventé en 1963 par J. L. Massey [23]. Son but était de concevoir un système correcteur d'erreurs simple et peu coûteux en matériel. Ce type de décodage est assez différent de l'algorithme de Viterbi présenté à la section 2.3 du fait que le décodage à seuil est un décodage fondé sur des considérations algébriques et non sur des considérations probabilistes. En outre c'est un décodage fonctionnant sur des symboles et non sur des séquences et qui peut donc remplacer l'algorithme MAP.

3.2.1 Codes convolutionnels orthogonaux

Nous allons considérer dans cette section des codes convolutionnels systématiques de taux $R = \frac{1}{2}$. La matrice génératrice de tels codes est de la forme:

$$\mathbf{G}(D) = [1 \ g_2(D)] \quad (3.2)$$

L'opération d'encodage consiste donc à générer un symbole de parité p_i pour chaque symbole d'information u_i que l'on souhaite transmettre à l'instant i , $i =$

$0, 1, 2, \dots$. Si l'on note $\{g_{2,\gamma_j}, j = 1, 2, \dots, J\}$ l'ensemble des coefficients de $\mathbf{g}_2(D)$ qui sont égaux à 1, on obtient:

$$p_i = \sum_{j=1}^J u_{i-\gamma_j}, \quad i = 0, 1, 2, \dots \quad (3.3)$$

Ainsi, les J positions $\gamma_j, j = 1, 2, \dots, J$ spécifient le code utilisé. Un code sera dit *orthogonal* [23] si toutes les différences entre deux des positions $\gamma_j, j = 1, 2, \dots, J$ sont différentes, soit:

$$\gamma_j - \gamma_k \neq \gamma_{j'} - \gamma_{k'}, \quad j > k, \quad j' > k', \quad (j, k) \neq (j', k') \quad (3.4)$$

La grandeur γ_J correspond à la mémoire du codeur. Prenons un exemple pour illustrer ces notions. Soit le codeur convolutionnel de la figure 3.1.

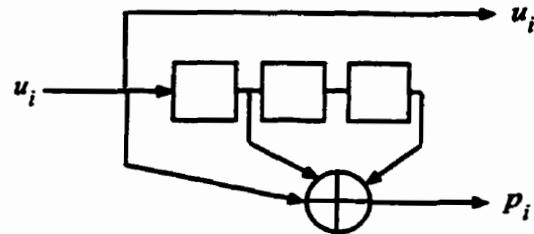


Figure 3.1: Exemple de codeur convolutionnel orthogonal

Pour ce code convolutionnel systématique de taux $R = \frac{1}{2}$, $J = 3$, $\mathbf{G}(D) = [1 \ 1 + D + D^3]$ et donc les positions des connexions sont $\{\gamma_1 = 0, \gamma_2 = 1, \gamma_3 = 3\}$. De plus, on a bien la loi:

$$p_i = u_i \oplus u_{i-1} \oplus u_{i-3} \quad (3.5)$$

Ce code est bien orthogonal car:

$$\begin{cases} \gamma_2 - \gamma_1 = 1 \\ \gamma_3 - \gamma_1 = 3 \\ \gamma_3 - \gamma_2 = 2 \end{cases}$$

3.2.2 Décodage à seuil

Les symboles d'information u_i et de parité p_i générés par un codeur orthogonal sont ensuite transformés respectivement en symboles de canal x_i^u et x_i^p d'énergie unitaire par un modulateur BPSK, ce qui peut s'écrire sous la forme:

$$x_i^u = 2u_i - 1 \quad (3.6)$$

$$x_i^p = 2p_i - 1 \quad (3.7)$$

Ces symboles sont transmis dans un canal à bruit blanc additif gaussien et on note par y_i^u et y_i^p les symboles reçus à l'instant i . Ceci revient à dire que:

$$y_i^u = x_i^u + n_i^u \quad (3.8)$$

$$y_i^p = x_i^p + n_i^p \quad (3.9)$$

où les n_i^u et n_i^p sont des variables aléatoires gaussiennes de moyenne nulle et de variance $\frac{N_0}{2}$ représentant les symboles de bruit.

Les décisions dures prises sur y_i^u et y_i^p et présentes en entrée du décodeur sont notées \tilde{u}_i et \tilde{p}_i de telle manière que:

$$\tilde{u}_i = u_i \oplus e_i^u \quad (3.10)$$

$$\tilde{p}_i = p_i \oplus e_i^p \quad (3.11)$$

où e_i^u et e_i^p sont les erreurs binaires qui affectent les symboles u_i et p_i respec-

tivement. Les décisions dures sur u_i et p_i après décodage seront notées \hat{u}_i et \hat{p}_i respectivement. La figure 3.2 permet d'illustrer ces notations.

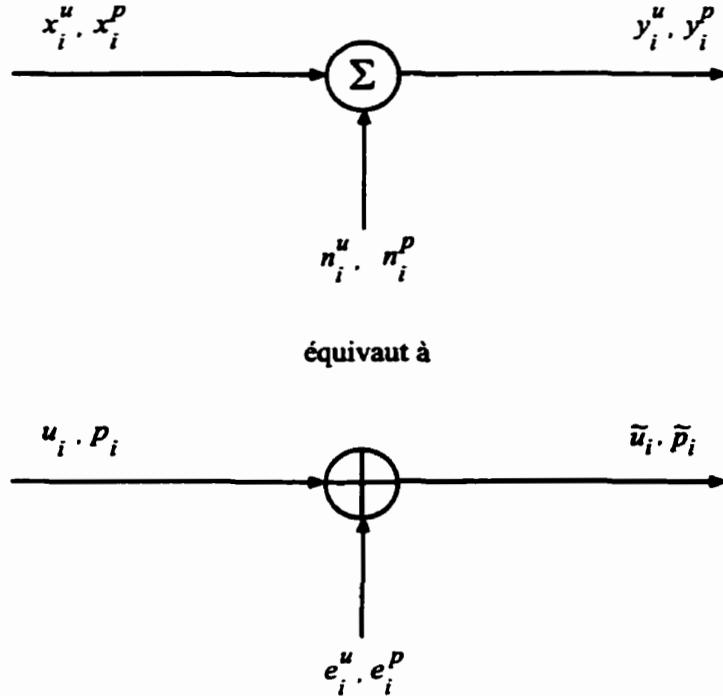


Figure 3.2: Notations relatives au canal de transmission

Le décodage à seuil est fondé sur le calcul de syndromes à partir des symboles de parité reçus, c'est-à-dire qu'il compare les symboles de parité reçus à ceux ré-encodés à partir des symboles d'information reçus (un décodeur à seuil contient donc le codeur orthogonal correspondant). Plus précisément, le syndrome relatif au bit de parité p_i vaut:

$$s_i = \tilde{p}_i \oplus \sum_{k=1}^J \tilde{u}_{i-\gamma_k} \quad (3.12)$$

$$= e_i^p \oplus \sum_{k=1}^J e_{i-\gamma_k}^u \quad (3.13)$$

Le décodeur à seuil calcule seulement les syndromes relatifs aux bits de parité $\{p_{i+\gamma_j}, j = 1, 2, \dots, J\}$, soit:

$$s_{i+\gamma_j} = \tilde{p}_{i+\gamma_j} \oplus \sum_{k=1}^J \tilde{u}_{i+\gamma_j - \gamma_k} = e_{i+\gamma_j}^p \oplus \sum_{k=1}^J e_{i+\gamma_j - \gamma_k}^u \quad (3.14)$$

Le traitement que le décodeur effectue à partir de ces syndromes dépend du type de sortie que l'on désire, à savoir à sortie pondérée ou dure. Deux types de décodeurs, baptisés respectivement Type I et Type II à sortie pondérée par Massey [23] vont être examinés successivement dans ce qui suit.

- **Décodeur à sortie dure de Type I**

Ce type de décodeur forme à partir des J syndromes donnés en (3.14) un ensemble de J équations de parité notées $\{A_{i,j}, j = 1, 2, \dots, J, i = 0, 1, 2, \dots\}$ orthogonales à e_i^u en ce sens que chaque équation de parité inclut le symbole e_i^u et que les autres symboles d'erreur n'apparaissent qu'une fois et une seule dans l'ensemble des équations. On a:

$$A_{i,j} = \tilde{p}_{i+\gamma_j} \oplus \tilde{u}_i \oplus \sum_{k=1}^{j-1} \tilde{u}_{i+\gamma_j - \gamma_k} \oplus \sum_{k=j+1}^J \tilde{u}_{i+\gamma_j - \gamma_k}, \quad j = 1, 2, \dots, J \quad (3.15)$$

ou:

$$A_{i,j} = e_{i+\gamma_j}^p \oplus e_i^u \oplus \sum_{k=1}^{j-1} e_{i+\gamma_j - \gamma_k}^u \oplus \sum_{k=j+1}^J (e_{i+\gamma_j - \gamma_k}^u \oplus \hat{e}_{i+\gamma_j - \gamma_k}^u), \quad j = 1, 2, \dots, J \quad (3.16)$$

La dernière partie constituant chaque équation $A_{i,j}$ correspond à une rétroaction de la décision et est nulle si celle-ci est idéale. On a alors:

$$A_{i,j} = e_{i+\gamma_j}^p \oplus e_i^u \oplus \sum_{k=1}^{j-1} e_{i+\gamma_j - \gamma_k}^u, \quad j = 1, 2, \dots, J \quad (3.17)$$

La règle de décision est alors basée sur une logique majoritaire de telle manière que l'on décide $\hat{u}_i = \bar{u}_i$ (on estime qu'il n'y a pas d'erreur) si et seulement si plus de $\lfloor \frac{J}{2} \rfloor$ équations de parité sont égales à 0. Ceci revient à dire que l'on choisit $\hat{u}_i = \bar{u}_i$ si et seulement si:

$$\sum_{j=1}^J A_{i,j} < \left\lfloor \frac{J}{2} \right\rfloor = T \quad (3.18)$$

où T est la valeur du seuil de décision dont la technique de décodage tire son nom.

Illustrons ces notions à l'aide du codeur de la figure 3.1 pour lequel $\gamma_1 = 0$, $\gamma_2 = 1$ et $\gamma_3 = 3$. Les équations formées par le décodeur à seuil de Type I pour cet exemple sont, dans le cas d'une rétroaction idéale:

$$\begin{cases} A_{i,1} = e_i^p \oplus e_i^u \\ A_{i,2} = e_{i+1}^p \oplus e_i^u \oplus e_{i+1}^u \\ A_{i,3} = e_{i+3}^p \oplus e_i^u \oplus e_{i+3}^u \oplus e_{i+2}^u \end{cases} \quad (3.19)$$

On obtient donc bien un ensemble d'équations orthogonales à e_i^u . Le schéma général d'un décodeur à seuil de Type I est donné à la figure 3.3.

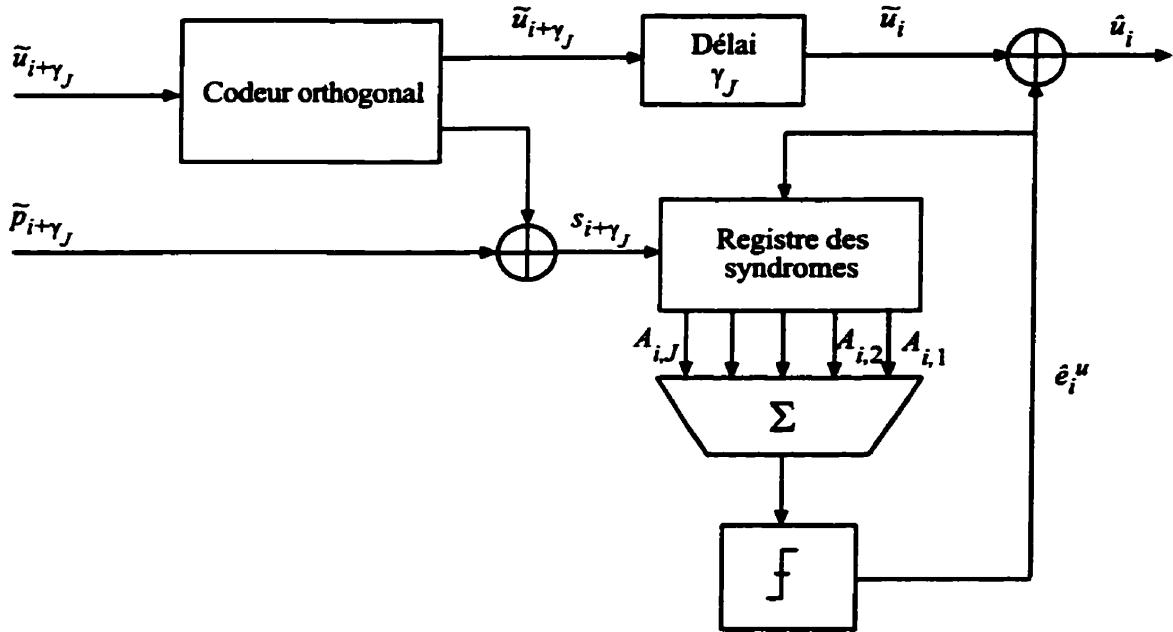


Figure 3.3: Schéma d'un décodeur à seuil de Type I

• Décodeur de Type II à sortie pondérée

Cette variante du décodeur de Type I permet d'obtenir directement une estimation du bit décodé, et c'est ce décodeur qui sera largement utilisé dans la suite de ce mémoire.

Dans ce cas, le décodeur forme $(J + 1)$ équations que l'on notera $\{B_{i,j}, j = 0, 1, \dots, J\}$ qui sont obtenues à partir des équations $\{A_{i,j}, j = 1, 2, \dots, J\}$ en éliminant le symbole \tilde{u}_i et en choisissant $B_{i,0} = \tilde{u}_i$. Ainsi, d'après (3.15), on a, pour $i = 0, 1, 2, \dots$:

$$B_{i,0} = \tilde{u}_i = u_i \oplus e_i^u \quad (3.20)$$

$$B_{i,j} = \tilde{p}_{i+\gamma_j} \oplus \sum_{k=1}^{j-1} \tilde{u}_{i+\gamma_j-\gamma_k} \oplus \sum_{k=j+1}^J \tilde{u}_{i+\gamma_j-\gamma_k}, \quad j = 1, 2, \dots, J \quad (3.21)$$

On peut réécrire les équations $\{B_{i,j}, j = 1, 2, \dots, J\}$ de la manière suivante:

$$B_{i,j} = u_i \oplus \sum_{k=1}^{j-1} e_{i+\gamma_j - \gamma_k}^u \oplus e_{i+\gamma_j}^p \oplus \sum_{k=j+1}^J (e_{i+\gamma_j - \gamma_k}^u \oplus \hat{e}_{i+\gamma_j - \gamma_k}^u), \quad j = 1, 2, \dots, J \quad (3.22)$$

On remarque grâce à cette dernière forme que toutes ces équations sont orthogonales non plus à l'erreur e_i^u mais au bit u_i .

L'algorithme de décodage à seuil à sortie pondérée va donc assigner au bit courant à décoder u_i la valeur qui maximise la probabilité conditionnelle $P(u_i | \{B_{i,j}\})$ pour $i = 0, 1, 2, \dots$. Ainsi, on décide $\hat{u}_i = 1$ si et seulement si:

$$P(u_i = 1 | \{B_{i,j}\}) \geq P(u_i = 0 | \{B_{i,j}\}) \quad (3.23)$$

ce qui revient à dire, en prenant le logarithme du rapport de vraisemblance (noté L) que l'on choisit $\hat{u}_i = 1$ si et seulement si:

$$L(u_i | \{B_{i,j}\}) = \ln \left(\frac{P(u_i = 1 | \{B_{i,j}\})}{P(u_i = 0 | \{B_{i,j}\})} \right) \geq 0 \quad (3.24)$$

En utilisant le théorème de Bayes et du fait que les équations $\{B_{i,j}, j = 1, 2, \dots, J\}$ sont indépendantes (cette indépendance étant obtenue grâce à la propriété d'orthogonalité du code), on obtient:

$$L(u_i | \{B_{i,j}\}) = \sum_{j=1}^J \ln \left(\frac{P(B_{i,j} | u_i = 1)}{P(B_{i,j} | u_i = 0)} \right) + \ln \left(\frac{P(B_{i,0} | u_i = 1)}{P(B_{i,0} | u_i = 0)} \right) + \ln \left(\frac{P(u_i = 1)}{P(u_i = 0)} \right) \quad (3.25)$$

Le dernier terme de cette équation représente l'information *a priori* que l'on supposera nulle du fait qu'en général les symboles binaires 0 et 1 émis par le codeur de source sont équiprobables. Nous pouvons simplifier l'équation ci-dessus de la manière suivante. D'après la structure des équations $\{B_{i,j}, j = 1, 2, \dots, J\}$ donnée par (3.22), on remarque que $B_{i,j} = 0$ si $u_i = 0$ et que $B_{i,j}$ contient un nombre pair de symboles d'erreur égaux à 1 ou si $u_i = 1$ et que $B_{i,j}$ contient un nombre impair de symboles d'erreur égaux à 1. Inversement, $B_{i,j} = 1$ si $u_i = 0$ et que $B_{i,j}$ contient un

nombre impair de symboles d'erreur à 1 ou si $u_i = 1$ et que $B_{i,j}$ contient un nombre pair de symboles d'erreur à 1. Soit $\sigma_{i,j} = 1 - \rho_{i,j}$, $j = 0, 1, \dots, J$, $i = 0, 1, 2, \dots$ la probabilité d'avoir un nombre impair de symboles d'erreur à 1 dans l'équation $B_{i,j}$. Ainsi:

$$\begin{aligned} P(B_{i,j} = 0|u_i = 0) &= P(B_{i,j} = 1|u_i = 1) = \rho_{i,j} \\ P(B_{i,j} = 1|u_i = 0) &= P(B_{i,j} = 0|u_i = 1) = \sigma_{i,j} \end{aligned} \quad (3.26)$$

d'où:

$$L(u_i|\{B_{i,j}\}) = \sum_{j=1}^J (2B_{i,j} - 1)w_{i,j} + (2B_{i,0} - 1)w_{i,0} \quad (3.27)$$

où les $w_{i,j} = \ln\left(\frac{\rho_{i,j}}{\sigma_{i,j}}\right)$ sont des facteurs de pondération de l'équation. On peut interpréter ces poids comme étant le logarithme du rapport de vraisemblance de l'équation $B_{i,j}$ en excluant le bit d'information u_i . On peut donc écrire:

$$\begin{aligned} w_{i,j} &= \ln\left(\frac{P(B_{i,j} \oplus u_i = 0)}{P(B_{i,j} \oplus u_i = 1)}\right) \\ &= -L(B_{i,j} \oplus u_i) \\ &= -L\left(e_{i+\gamma_j}^p \oplus \sum_{k=1}^{j-1} e_{i+\gamma_j-\gamma_k}^u \oplus \sum_{k=j+1}^J (e_{i+\gamma_j-\gamma_k}^u \oplus \hat{e}_{i+\gamma_j-\gamma_k}^u)\right) \end{aligned} \quad (3.28)$$

Nous allons à présent définir un opérateur fondamental pour la suite du mémoire: l'opérateur add-min [35]. Il sera noté par \diamond et est utilisé pour simplifier les calculs des logarithmes de rapports de vraisemblance de sommes de variables aléatoires binaires. Soient ξ_1 et ξ_2 deux variables aléatoires binaires, alors:

$$L(\xi_1 \oplus \xi_2) = \ln \left(\frac{P(\xi_1 \oplus \xi_2 = 1)}{P(\xi_1 \oplus \xi_2 = 0)} \right) \quad (3.29)$$

$$= -\ln \left(\frac{P(\xi_1 = 0)P(\xi_2 = 0) + P(\xi_1 = 1)P(\xi_2 = 1)}{P(\xi_1 = 1)P(\xi_2 = 0) + P(\xi_1 = 0)P(\xi_2 = 1)} \right) \quad (3.30)$$

$$= -\ln \left(\frac{1 + \frac{P(\xi_1=1)P(\xi_2=1)}{P(\xi_1=0)P(\xi_2=0)}}{\frac{P(\xi_1=1)}{P(\xi_1=0)} + \frac{P(\xi_2=1)}{P(\xi_2=0)}} \right) \quad (3.31)$$

soit:

$$L(\xi_1 \oplus \xi_2) = -\ln \left(\frac{1 + \exp(L(\xi_1)) \exp(L(\xi_2))}{\exp(L(\xi_1)) + \exp(L(\xi_2))} \right) \quad (3.32)$$

ce que l'on peut simplifier par:

$$L(\xi_1 \oplus \xi_2) \approx -\text{signe}(L(\xi_1))\text{signe}(L(\xi_2))\min(|L(\xi_1)|, |L(\xi_2)|) \triangleq L(\xi_1) \diamond L(\xi_2) \quad (3.33)$$

D'après [35], on peut estimer la perte induite par cette approximation à moins de 0,2 dB à faible rapport signal sur bruit. À fort signal sur bruit, la perte engendrée par cette approximation est négligeable. Le lecteur pourra vérifier dans [11] qu'en combinant (3.27), (3.28) et (3.33), on obtient une approximation du logarithme du rapport de vraisemblance que l'on notera λ_i et qui s'écrit:

$$\boxed{\lambda_i = y_i^u + \sum_{j=1}^J \left(y_{i+\gamma_j}^p \diamond \sum_{k=1}^{j-1} y_{i+\gamma_j-\gamma_k}^u \diamond \sum_{k=j+1}^J \lambda_{i+\gamma_j-\gamma_k} \right)} \quad (3.34)$$

ou encore:

$$\lambda_i = y_i^u + \sum_{j=1}^J \psi_{i,j} \quad (3.35)$$

$$= y_i^u + L_e(\hat{u}_i) \quad (3.36)$$

et le bit décodé $\hat{u}_i = 1$ si et seulement si $\lambda_i \geq 0$. La valeur de la somme $\sum_{j=1}^J \psi_{i,j}$ correspond à la valeur extrinsèque $L_e(\hat{u}_i)$.

Dans le cas du codeur représenté à la figure 3.1, les équations de parité $\psi_{i,j}$ s'écrivent:

$$\left\{ \begin{array}{l} \psi_{i,1} = y_i^p \diamond \lambda_{i-1} \diamond \lambda_{i-3} \\ \psi_{i,2} = y_{i+1}^p \diamond y_{i+1}^u \diamond \lambda_{i-2} \\ \psi_{i,3} = y_{i+3}^p \diamond y_{i+3}^u \diamond y_{i+2}^u \end{array} \right. \quad (3.37)$$

Le schéma du décodeur de Type II correspondant à cet exemple est présenté à la figure 3.4.

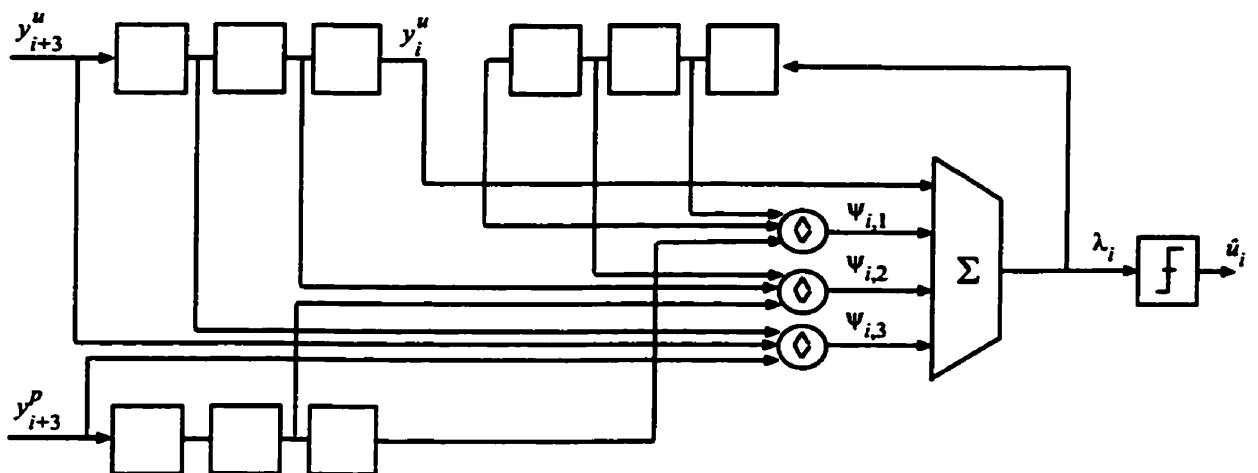


Figure 3.4: Schéma du décodeur à seuil de Type II à sortie pondérée correspondant au codeur de la figure 3.1

3.3 Décodage à seuil itératif et codes convolutionnels doublement orthogonaux

Nous pouvons maintenant présenter la technique de décodage itératif introduite à la section 3.1.4. L'algorithme de décodage des codes convolutionnels utilisé dans cette technique est le décodage à seuil présenté à la section précédente. Cette décision est motivée d'une part par le fait que c'est une technique de décodage symbole par symbole, d'autre part par la très faible complexité de ce type de décodage.

De plus, la procédure itérative est maintenue car elle permet un raffinement de l'estimation du symbole décodé courant du moment que les observables à l'entrée du décodeur à chaque itération sont indépendantes.

Or cette technique diffère fondamentalement de la technique de décodage turbo du fait qu'elle n'utilise aucun entrelaceur, ni au codage ni au décodage. Ainsi, on doit pouvoir obtenir l'indépendance des observables par un autre moyen. Nous allons voir que l'on peut y parvenir en imposant des conditions d'orthogonalité d'ordre supérieur à la simple orthogonalité introduite par Massey [23]. Ces conditions, que nous allons développer plus loin dans cette section, ont donné naissance à une nouvelle classe de codes: les codes doublement orthogonaux (en anglais CSO2C pour Convolutional Self Doubly Orthogonal Codes) [12], [16].

3.3.1 Décodage à seuil itératif

L'algorithme de décodage à seuil itératif sans entrelacement est obtenu par des applications successives de l'algorithme de décodage à seuil présenté à la section 3.2.2. La sortie pondérée du décodeur à seuil à l'itération $(\mu - 1)$, que l'on note $\lambda_i^{(\mu-1)}$, est utilisée avec les symboles de parité reçus pour produire l'information extrinsèque $L_e^{(\mu)}(\hat{u}_i)$. Le décodeur à l'itération μ utilise cette information extrinsèque avec le symbole d'information reçu y_i^u pour calculer la sortie pondérée $\lambda_i^{(\mu)}$.

D'après (3.34), la sortie pondérée $\lambda_i^{(\mu)}$ est donnée par:

$$\lambda_i^{(\mu)} = y_i^u + \sum_{j=1}^J \left(y_{i+\gamma_j}^p \diamond \sum_{k=1}^{j-1} \lambda_{i+\gamma_j-\gamma_k}^{(\mu-1)} \diamond \sum_{k=j+1}^J \lambda_{i+\gamma_j-\gamma_k}^{(\mu)} \right) \quad (3.38)$$

ou encore, si l'on note par:

$$\psi_{i,j}^{(\mu)} = y_{i+\gamma_j}^p \diamond \sum_{k=1}^{j-1} \lambda_{i+\gamma_j-\gamma_k}^{(\mu-1)} \diamond \sum_{k=j+1}^J \lambda_{i+\gamma_j-\gamma_k}^{(\mu)} \quad (3.39)$$

on a:

$$\lambda_i^{(\mu)} = y_i^u + \sum_{j=1}^J \psi_{i,j}^{(\mu)} = y_i^u + L_e^{(\mu)}(\hat{u}_i) \quad (3.40)$$

La figure 3.5 schématisé ce processus. Le délai γ_J qui y est représenté permet de compenser le délai présent au décodage de manière qu'à chaque itération, ce soient les mêmes symboles reçus qui seront présentés au décodeur à seuil.

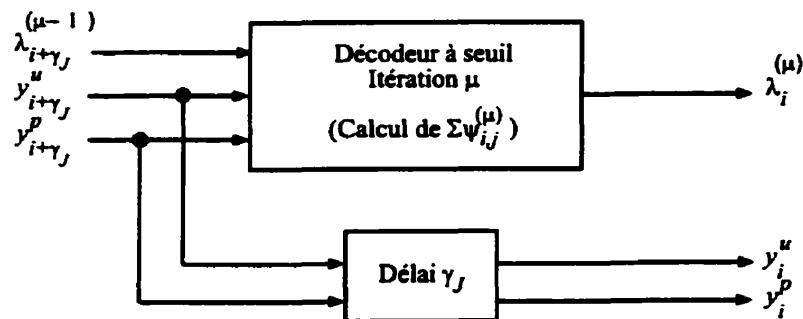


Figure 3.5: Schématisation du décodage à seuil itératif sans entrelacement

On remarque d'après (3.38) que $\lambda_i^{(\mu)}$ est une fonction d'observables corrélées du fait de l'absence d'entrelaceurs, ce qui ne permet pas au décodeur d'arriver à de bonnes performances en termes de probabilité d'erreur. Afin d'obtenir l'indépendance des observables requise, la propriété d'orthogonalité du code doit

être maintenue pour au moins deux itérations, ce qui amène la définition des codes doublement orthogonaux (CSO2C).

3.3.2 Codes doublement orthogonaux au sens large

Nous allons présenter dans cette section les propriétés mathématiques des codes qui doivent être satisfaites pour que les observables présentes à l'entrée du décodeur soient indépendantes d'une itération à la suivante. Nous allons nous intéresser ici à deux itérations seulement. L'écriture de (3.38) donne alors:

$$\begin{aligned} \lambda_i^{(2)} = & y_i^u + \sum_{j=1}^J \left(y_{i+\gamma_j}^p \diamond \sum_{k=j+1}^J \lambda_{i+\gamma_j-\gamma_k}^{(2)} \diamond \sum_{k=1}^{j-1} \left(y_{i+\gamma_j-\gamma_k}^u \right. \right. \\ & + \sum_{l=1, l \neq k}^J \left(y_{i+\gamma_j-\gamma_k+\gamma_l}^p \diamond \sum_{m=1, m \neq j}^{l-1} \diamond y_{i+(\gamma_j-\gamma_k)-(\gamma_m-\gamma_l)}^u \right. \\ & \left. \left. \diamond \sum_{m=l+1}^J \lambda_{i+(\gamma_j-\gamma_k)-(\gamma_m-\gamma_l)}^{(1)} \right) \right) \end{aligned} \quad (3.41)$$

Les deux conditions $l \neq k$ et $m \neq j$ sont ajoutées afin de supprimer certaines répétitions au niveau des observables. D'après (3.41), on peut remarquer que certaines conditions doivent être satisfaites pour obtenir l'indépendance des observables que nous recherchons. La condition principale est que les valeurs produites par les différences des différences des positions γ_j doivent être distinctes. Plus précisément, on peut formaliser la définition des codes doublement orthogonaux de la manière suivante:

Définition: Un code convolutionnel systématique de taux $R = \frac{1}{2}$ est doublement orthogonal si et seulement si les positions γ_j , $j = 1, 2, \dots, J$ satisfont les propriétés suivantes:

- a) les différences simples $\gamma_j - \gamma_k$ sont distinctes entre elles;
- b) les différences des différences $(\gamma_j - \gamma_k) - (\gamma_m - \gamma_l)$ sont distinctes des différences

simples;

c) les différences des différences $(\gamma_j - \gamma_k) - (\gamma_m - \gamma_l)$ sont distinctes entre elles; et ce pour tous (j, k, l, m) dans $\{1, 2, \dots, J\}$ tels que $j \neq k, l \neq m, l \neq k$ et $m \neq j$. La première condition correspond à la définition de simple orthogonalité donnée par (3.4). Les deux autres conditions correspondent à des conditions additionnelles qui caractérisent les codes doublement orthogonaux. Cette définition possède des redondances car on peut montrer que la troisième condition implique les deux premières [1].

Toutefois, pour un CSO2C de taux $R = \frac{1}{2}$, et malgré les conditions $l \neq k$ et $m \neq j$, on n'obtient pas un ensemble d'observables parfaitement décorrélées. En effet, la permutation des indices k et m produit des différences des différences $(\gamma_j - \gamma_k) - (\gamma_m - \gamma_l)$ identiques bien que la définition donnée ci-dessus soit respectée. Ces répétitions inévitables ne peuvent être éliminées du processus de décodage sans générer une propagation des erreurs d'une itération à la suivante. En présence de ces répétitions indésirables, la définition donnée ci-dessus se réfère aux codes convolutionnels *doublement orthogonaux au sens large* (en anglais CSO2C-WS).

Diverses tentatives pour améliorer la décorrélation des observables en présence de ces répétitions ont été entreprises, dans le but d'améliorer les performances du décodage, et notamment l'introduction de coefficients de pondération au sein des équations constitutives du décodage. Ces coefficients ont pour objectif d'accroître l'importance relative des équations $\psi_{i,j}^{(\mu)}$ qui contiennent peu de termes non orthogonaux et de diminuer l'importance de celles qui en contiennent beaucoup [11]. L'équation (3.40) s'écrit alors:

$$\lambda_i^{(\mu)} = a_0^{(\mu)} y_i^u + \sum_{j=1}^J a_j^{(\mu)} \psi_{i,j}^{(\mu)} \quad (3.42)$$

Diverses méthodes (empiriques ou théoriques) ayant pour objectif de trouver les coefficients $a_j^{(\mu)}$ optimaux ont été entreprises et on se reportera à [11] pour plus de détails sur ces méthodes ainsi que sur les résultats obtenus.

Ces tentatives n'étant pas totalement satisfaisantes, il a fallu recourir à une modification du codeur afin d'obtenir une double orthogonalité au sens strict qui permet d'éviter en totalité les répétitions indésirables d'observables [16]. Ce type de codeur est présenté à la section suivante.

3.3.3 Codes doublement orthogonaux au sens strict

L'obtention de codes doublement orthogonaux au sens strict peut être intéressante du point de vue des performances d'erreur et du nombre d'itérations nécessaires. Ce type de code est obtenu en utilisant un taux de codage de $R = \frac{J}{2J}$.

Dans cette architecture parallèle du codeur, chaque registre à décalage est connecté à un et un seul additionneur modulo 2 donné de telle manière que les équations de syndromes n'utilisent pas les mêmes observables. Le code est alors spécifié par une matrice $(\gamma_{v,w})$ de taille $J * J$. Chaque $\gamma_{v,w}$ correspond à une case du registre à décalage numéro v qui est connectée au symbole de parité numéro w . De la même manière que dans (3.3), on a:

$$p_{w,i} = \sum_{k=1}^J u_{k,i-\gamma_{k,w}}, \quad w = 1, 2, \dots, J, \quad i = 0, 1, 2, \dots \quad (3.43)$$

Par un processus similaire à celui auquel nous avons abouti à (3.41), on obtient ici les J valeurs MAP approximatives associées aux J entrées du codeur [11]:

$$\begin{aligned} \lambda_{v,i}^{(2)} &= y_{v,i}^u + \sum_{n=1}^J \left(y_{n,i+\gamma_{v,n}}^p \diamond \sum_{k=j+1}^J \lambda_{k,i+\gamma_{v,n}-\gamma_{k,n}}^{(2)} \diamond \sum_{l=1}^{j-1} \left(y_{k,i+\gamma_{v,n}-\gamma_{k,n}}^u \right. \right. \\ &\quad + \sum_{s=1, s \neq n}^J \left(y_{s,i+\gamma_{v,n}-\gamma_{k,n}+\gamma_{k,s}}^p \diamond \sum_{l=1}^{k-1} y_{l,i+(\gamma_{v,n}-\gamma_{k,n})-(\gamma_{l,s}-\gamma_{k,s})}^u \right. \\ &\quad \left. \left. \diamond \sum_{l=k+1}^J \lambda_{l,i+(\gamma_{v,n}-\gamma_{k,n})-(\gamma_{l,s}-\gamma_{k,s})}^{(1)} \right) \right) \quad v = 1, 2, \dots, J \end{aligned} \quad (3.44)$$

Ainsi, un code doublement orthogonal au sens strict doit satisfaire la définition suivante:

Définition: Un code convolutionnel systématique de taux $R = \frac{J}{2J}$ est dit doublement orthogonal au sens strict (CSO2C-SS) si et seulement si:

- a) les différences $\gamma_{v,n} - \gamma_{k,n}$ sont distinctes;
- b) les différences doubles $(\gamma_{v,n} - \gamma_{k,n}) - (\gamma_{l,s} - \gamma_{k,s})$ sont distinctes des différences simples;
- c) les différences doubles $(\gamma_{v,n} - \gamma_{k,n}) - (\gamma_{l,s} - \gamma_{k,s})$ sont distinctes entre elles.

et ce pour tout (v, n, s, k, l) dans $\{1, 2, \dots, J\}$ tels que $s \neq n$, $k \neq v$ et $k \neq l$. D'après cette définition, on se rend compte que toutes les répétitions inévitables et indésirables sont exclues et ainsi la deuxième itération du décodage s'effectue à partir d'un ensemble d'observables totalement indépendant de celui de la première itération.

Un exemple de codeur doublement orthogonal au sens strict est donné à la figure 3.6.

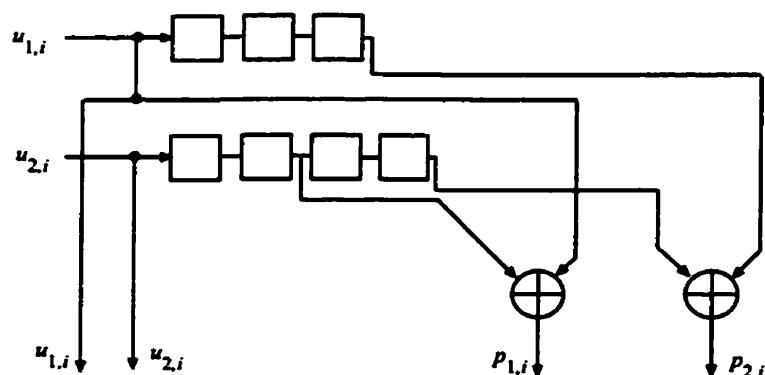


Figure 3.6: Exemple de codeur doublement orthogonal au sens strict de taux $\frac{2}{4}$

La matrice des positions correspondant à ce code est donnée par:

$$(\gamma_{v,w}) = \begin{pmatrix} 0 & 3 \\ 2 & 4 \end{pmatrix}$$

Ce code est bien doublement orthogonal du fait que chaque additionneur modulo 2 est relié à exactement un seul registre à décalage.

3.3.4 Construction des codes doublement orthogonaux

En utilisant les deux définitions données dans les sections précédentes, les auteurs de [2] ont recherché des ensembles de nombres vérifiant les conditions propres à chaque type de code. Le critère d'optimisation était la réduction de γ_J pour les codes au sens large et de $\max_{v,w}(\gamma_{v,w})$ pour les codes au sens strict, ces deux grandeurs caractérisant la latence du codeur. Deux techniques ont été employées, passant toutes deux par deux étapes: la première consiste à générer un ensemble de positions valides et la deuxième étant la réduction de cet ensemble.

Cette recherche des codes peut être effectuée en utilisant la géométrie projective. En effet, dans le cas d'un CSO2C-WS, les J positions γ_j peuvent être obtenues en déterminant les $J = p^s + 1$ exposants possibles des points d'une ligne $\eta_1 + \eta_2\gamma$ définie dans une géométrie projective $PG(4, p^s)$. Cette méthode n'étant pas satisfaisante car les longueurs de contrainte γ_J demeuraient trop élevées, une méthode pseudo-aléatoire fondée sur l'utilisation d'un seuil de rejet [2] a permis d'améliorer ces résultats. Le tableau 3.1 présente les meilleurs résultats obtenus pour les CSO2C-WS. Nous renvoyons le lecteur à [1] pour les méthodes et les résultats obtenus pour les CSO2C-SS.

Tableau 3.1: Meilleurs ensembles de positions obtenus

J	$\{\gamma_j, j = 1, 2, \dots, J\}$
2	$\{0, 1\}$
3	$\{0, 2, 5\}$
4	$\{0, 4, 14, 15\}$
5	$\{0, 1, 24, 37, 41\}$
6	$\{0, 1, 17, 70, 95, 100\}$
7	$\{0, 1, 53, 128, 207, 216, 222\}$
8	$\{0, 43, 139, 322, 422, 430, 441, 459\}$
9	$\{0, 9, 21, 395, 584, 767, 871, 899, 912\}$
10	$\{0, 29, 40, 43, 1020, 1328, 1495, 1606, 1696, 1698\}$
11	$\{0, 220, 521, 695, 908, 926, 1059, 2457, 3367, 3458, 3490\}$
12	$\{0, 48, 212, 1014, 1381, 2217, 4198, 4373, 4766, 4885, 4914, 5173\}$
13	$\{0, 335, 594, 639, 1476, 1778, 3034, 5637, 6584, 9682, 9934, 10138, 10201\}$
14	$\{0, 575, 1253, 4863, 6407, 6660, 7285, 8443, 11884, 13891, 15065, 15484, 15550, 16285\}$
15	$\{0, 576, 1331, 1897, 2222, 4684, 6502, 7293, 10785, 18999, 20901, 22157, 24372, 27107, 29532\}$

CHAPITRE 4

PERFORATION DES CODES DOUBLEMENT ORTHOGONAUX

4.1 Introduction

4.1.1 Motivations

Nous avons présenté dans le chapitre précédent une technique de décodage itératif sans entrelacement fondée sur le décodage à seuil. Cette technique semble prometteuse du fait qu'elle permet de contourner les deux défauts inhérents à la technique du décodage itératif turbo que sont la complexité de l'algorithme de décodage et sa latence. De plus elle permet d'atteindre de bonnes performances d'erreur comparée aux autres techniques de décodage existantes.

Toutefois, cette technique n'a été conçue que pour des codes dont le taux de codage est égal à $R = \frac{1}{2}$ (CSO2C-WS) ou à $R = \frac{J}{2J}$ (CSO2C-SS). Rappelons que pour de tels taux de codage, le nombre de symboles binaires effectivement transmis sur le canal est le double du nombre de symboles binaires d'information émis par le codeur de source. Il s'ensuit que la largeur de bande requise pour transmettre ces symboles binaires d'information est deux fois plus importante que dans le cas d'une information non codée (on se place ici et dans tout ce chapitre dans le cas d'une modulation antipodale BPSK).

Or, comme nous l'avons mentionné lors de l'introduction, la bande passante est une ressource précieuse que l'on essaie de rentabiliser au maximum lorsque l'on effectue le design des systèmes de télécommunications modernes. Le recours à des codes de taux élevés $R = \frac{k}{n}$, $R > \frac{1}{2}$, permet de diminuer l'expansion de largeur de bande liée au codage de canal et donc d'augmenter l'efficacité spectrale de la communication.

Dans ce but, nous allons présenter dans ce chapitre une technique basée sur

la perforation qui va nous permettre d'augmenter le taux de codage des codes doublement orthogonaux au sens large et au sens strict. Les codes obtenus seront notés PCSO2C pour Punctured Convolutional Self-Doubly Orthogonal Codes. Nous étudierons par la suite les conséquences sur les performances du décodeur de l'emploi de ces codes perforés.

4.1.2 Introduction à la perforation

La perforation d'un code convolutionnel [4], [5], [10], [42] est une technique qui permet d'obtenir de manière très simple des codes de taux élevés $R = \frac{k}{n}$ à partir d'un code de taux $R = \frac{1}{2}$ (appelé code origine) et qui en outre a le mérite d'être souple car des codes de différents taux de codage peuvent être obtenus à partir du même code origine. La technique consiste à éliminer périodiquement des symboles émis par le code origine de taux $R = \frac{1}{2}$ de telle manière qu'en moyenne la proportion de symboles de parité est égale à $(1 - \frac{k}{n}) < \frac{1}{2}$.

La perforation d'un code origine de taux $R = \frac{1}{2}$ est uniquement caractérisée par une matrice notée \mathbf{P} , possédant deux lignes et k colonnes et constituée d'éléments binaires (0 ou 1). Un taux de codage égal à $R = \frac{k}{n}$ sera atteint si \mathbf{P} contient exactement n uns et $(2k - n)$ zéros. Un exemple de matrice de perforation est donné par (4.1), où $k = 3$ et $n = 4$.

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad (4.1)$$

La signification de cette matrice est la suivante: à chaque sortie du codeur correspond une ligne de la matrice. Les 1 présents sur chaque ligne de la matrice signifient que les symboles correspondants seront conservés après perforation alors que les 0 correspondent aux symboles éliminés. La correspondance entre les symboles et les colonnes de \mathbf{P} s'effectue suivant une opération de modulo sur le nombre de colonnes. Ainsi, si l'on note par $\{c_{1,i}, i = 0, 1, 2, \dots\}$ les symboles présents à la première sortie du codeur et par $\{c_{2,i}, i = 0, 1, 2, \dots\}$ ceux présents à la deuxième

sortie de ce même codeur, les symboles conservés après perforation suivant la matrice donnée en (4.1) seront ceux qui appartiennent à:

$$\{c_{1,qk}\} \cup \{c_{1,2+qk}\}, \quad q = 0, 1, 2, \dots \quad (4.2)$$

et à:

$$\{c_{2,qk}\} \cup \{c_{2,1+qk}\}, \quad q = 0, 1, 2, \dots \quad (4.3)$$

car ce sont les symboles pour lesquels l'élément correspondant dans la matrice de perforation est égal à 1. Un schéma illustrant ces notions est donné à la figure 4.1.

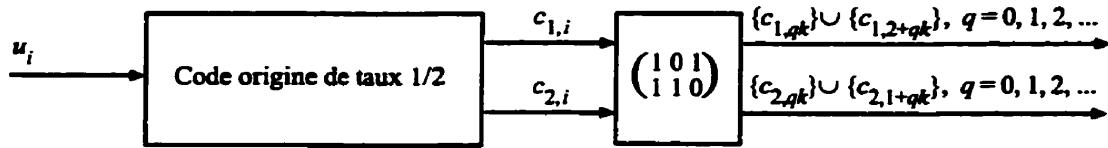


Figure 4.1: Illustration de la technique de perforation

On peut alors vérifier que le taux de codage obtenu par application de la matrice de perforation donnée en (4.1) est bien $R = \frac{k}{n}$. Pour $k = 3$ bits en entrée du codeur, il y aura $2k = 6$ symboles codés délivrés par le code origine et après perforation, il en restera $n = 4$ qui seront effectivement transmis. Le taux de codage après perforation est donc bien $R = \frac{k}{n} = \frac{3}{4}$.

4.2 Influence de la perforation des CSO2C-WS sur les performances du décodage à seuil itératif

Dans cette section, nous allons étudier l'effet de la perforation des CSO2C-WS sur les performances du décodage à seuil itératif. Considérons comme code origine un CSO2C-WS de taux $R = \frac{1}{2}$ spécifié par l'ensemble de positions $\{\gamma_j, j = 1, 2, \dots, J\}$. Ce code étant systématique, les symboles en sortie du

codeur sont d'une part les symboles d'information générés par le codeur de source ($c_{1,i} = u_i, i = 0, 1, \dots$) et d'autre part les symboles de parité qui ont été calculés selon (3.3) ($c_{2,i} = p_i, i = 0, 1, \dots$).

Du fait que les symboles d'information ne peuvent être perforés, la première ligne de la matrice de perforation ne doit comporter que des 1. Ainsi, la matrice de perforation aura l'allure suivante:

$$\mathbf{P} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 1 & 0 & 1 & \dots & 0 & 1 & 0 \end{pmatrix} \quad (4.4)$$

Les positions des uns sur la seconde ligne seront dénotées $\pi_1, \pi_2, \dots, \pi_{n-k}$ (il doit y avoir $(n - k)$ uns sur la deuxième ligne pour obtenir un taux de codage égal à $R = \frac{k}{n}$).

4.2.1 Représentation spectrale des PCSO2C-WS de taux $R = \frac{k}{k+1}$

Nous allons nous intéresser en premier lieu à la génération de PCSO2C-WS de taux $R = \frac{k}{k+1}$. En conséquence, la matrice de perforation aura l'allure suivante:

$$\mathbf{P} = \begin{pmatrix} 1 & 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{pmatrix} \quad (4.5)$$

La longueur de cette matrice est égale à k et donc on génère un bit de parité à chaque k bits d'information. Le taux de codage qui en résulte est donc bien $R = \frac{k}{k+1}$. La position du 1 sur la deuxième ligne sera noté π comme spécifié ci-dessus.

En premier lieu, rappelons l'expression de la valeur MAP approximative qui est calculée par le décodeur à seuil itératif.

- À la première itération:

$$\lambda_i = y_i^u + \sum_{j=1}^J \left(y_{i+\gamma_j}^p \diamond \sum_{k=1}^{j-1} y_{i+\gamma_j-\gamma_k}^u \diamond \sum_{k=j+1}^J \lambda_{i+\gamma_j-\gamma_k} \right) \quad (4.6)$$

$$= y_i^u + \sum_{j=1}^J \psi_{i,j} \quad (4.7)$$

- À l'itération $\mu \geq 2$:

$$\lambda_i^{(\mu)} = y_i^u + \sum_{j=1}^J \left(y_{i+\gamma_j}^p \diamond \sum_{k=1}^{j-1} \lambda_{i+\gamma_j-\gamma_k}^{(\mu-1)} \diamond \sum_{k=j+1}^J \lambda_{i+\gamma_j-\gamma_k}^{(\mu)} \right) \quad (4.8)$$

$$= y_i^u + \sum_{j=1}^J \psi_{i,j}^{(\mu)} \quad (4.9)$$

On rappelle d'après (4.6) et (4.8) que pour décoder le bit courant u_i à l'itération μ , le décodeur utilise J équations $\psi_{i,j}^{(\mu)}$ dont la somme forme l'information extrinsèque. Or, quelle que soit l'itération μ , chaque équation $\psi_{i,j}^{(\mu)}$ utilise un seul symbole de parité $y_{i+\gamma_j}^p$ propre à chaque équation mais identique pour toutes les itérations, et donc l'ensemble des symboles de parité nécessaires pour décoder le bit u_i est:

$$\{y_{i+\gamma_j}^p, \quad j = 1, \dots, J\} \quad (4.10)$$

Or si l'on considère la matrice de perforation donnée par (4.5), les seuls symboles qui ne seront pas éliminés par le processus de perforation sont ceux dont la position en sortie du codeur est égale à π , et ce, modulo k . Ce sont donc ceux qui appartiennent à l'ensemble donné par (4.11):

$$\{y_{\pi+qk}^p, \quad q = 0, 1, 2, \dots\} \quad (4.11)$$

Ainsi, les seuls symboles de parité que le décodeur a à sa disposition pour effectuer le décodage de u_i sont ceux pour lesquels l'égalité suivante est vérifiée:

$$i + \gamma_j = \pi + qk, \quad q = 0, 1, 2, \dots, \quad j = 1, \dots, J \quad (4.12)$$

soit:

$$\boxed{\gamma_j = (\pi - i) \bmod(k), \quad j = 1, \dots, J} \quad (4.13)$$

Ainsi, l'équation $\psi_{i,j}^{(\mu)}$ verra son symbole de parité conservé par le processus de perforation si la position correspondante γ_j vérifie (4.13), et ce quel que soit μ . Pour les équations dont la position correspondante ne vérifie pas (4.13), elles ne pourront pas délivrer un résultat correct et donc on doit les forcer à la valeur 0 pour qu'elles n'influent pas la valeur de l'information extrinsèque $L_e^{(\mu)}(\hat{u}_i)$. Ainsi, le décodeur à seuil itératif opérera de la manière suivante:

$$\lambda_i^{(\mu)} = y_i^u + \sum_{j=1}^J \psi_{i,j}^{(\mu)}, \quad \mu = 1, 2, \dots, \quad i = 0, 1, 2, \dots \quad (4.14)$$

où:

- si $\gamma_j = (\pi - i) \bmod(k)$,

- À la première itération:

$$\psi_{i,j} = y_{i+\gamma_j}^p \diamond \sum_{k=1}^{j-1} y_{i+\gamma_j-\gamma_k}^u \diamond \sum_{k=j+1}^J \lambda_{i+\gamma_j-\gamma_k} \quad (4.15)$$

- À l'itération $\mu \geq 2$:

$$\psi_{i,j}^{(\mu)} = y_{i+\gamma_j}^p \diamond \sum_{k=1}^{j-1} \lambda_{i+\gamma_j-\gamma_k}^{(\mu-1)} \diamond \sum_{k=j+1}^J \lambda_{i+\gamma_j-\gamma_k}^{(\mu)} \quad (4.16)$$

- sinon, quelle que soit l'itération:

$$\psi_{i,j}^{(\mu)} = 0 \quad (4.17)$$

Définissons l'ensemble de positions \mathcal{J}_i de la manière suivante:

$$\mathcal{J}_i = \{\gamma_j \text{ vérifiant } \gamma_j = (\pi - i) \bmod(k)\} \quad (4.18)$$

$$= \{\gamma_j \text{ tel que } \psi_{i,j}^{(\mu)} \neq 0, \mu = 1, 2, \dots\} \quad (4.19)$$

et:

$$J_i = \text{card}(\mathcal{J}_i) \leq J \quad (4.20)$$

J_i correspond au nombre de positions qui vérifient (4.13) et donc au nombre d'équations qui seront effectivement utilisées par le décodeur à seuil itératif pour décoder le bit courant u_i . Ainsi, cette quantité détermine la performance effective du décodeur à seuil itératif en ce qui concerne le bit courant à l'instant i .

Supposons qu'à un instant i fixé, on ait $J_i = J$. Cela signifie que le décodeur va disposer de J équations non nulles pour décoder le bit u_i , et donc la valeur MAP approximative $\lambda_i^{(\mu)}$ correspondant à ce bit va être la même que celle calculée dans le cas où le code n'est pas perforé. Ceci est également vrai pour les bits u_{i+k}, u_{i+2k}, \dots . Toutefois, si $J_i = J$, cela signifie que:

$$\gamma_j = (\pi - i) \bmod(k), \quad \forall j \in \{1, \dots, J\} \quad (4.21)$$

Or, si une telle égalité est vérifiée, à l'instant $(i+1)$ on aura de manière évidente:

$$\gamma_j \neq (\pi - i - 1) \bmod(k), \quad \forall j \in \{1, \dots, J\} \quad (4.22)$$

et donc $J_{i+1} = 0$. On voit donc apparaître un phénomène caractéristique de l'effet de la perforation sur ce type de décodage, soit le fait qu'elle engendre des capacités de correction au décodage inégales d'un bit à l'autre. Il semble possible *a priori* d'obtenir des capacités de correction égales pour tous les bits (i.e. quel que soit i). Il est important de mentionner ici que les performances du décodeur à seuil, soit $\{J_i, i = 0, 1, 2, \dots\}$, sont entièrement déterminées par l'ensemble des positions $\{\gamma_j, j = 1, 2, \dots, J\}$. Il est donc possible de prévoir les performances

d'un PCSO2C-WS donné uniquement à partir de son ensemble de connexions.

Ces performances peuvent alors être représentées très simplement de manière graphique en utilisant une représentation spectrale. Tout d'abord, nous pouvons remarquer que les valeurs de J_i sont périodiques de période k car (4.13) l'est également. En effet, pour tout $i = 0, 1, 2, \dots$ et tout $q = 0, 1, 2, \dots$ on a:

$$\begin{aligned} \mathcal{J}_{i+qk} &= \{\gamma_j \text{ vérifiant } \gamma_j = (\pi - i - qk) \bmod(k)\} \\ &= \{\gamma_j \text{ vérifiant } \gamma_j = (\pi - i) \bmod(k)\} \\ &= \mathcal{J}_i \end{aligned} \quad (4.23)$$

d'où:

$$J_{i+qk} = J_i \quad (4.24)$$

Ainsi, seules les k premières valeurs J_0, J_1, \dots, J_{k-1} sont nécessaires pour déterminer les performances du décodeur. Parmi ces k valeurs, il y en a un certain nombre qui sont distinctes. La représentation spectrale consiste donc à placer aux abscisses correspondant à ces valeurs qui sont distinctes des raies d'amplitude la proportion des J_i qui sont égales à ces valeurs, soit la proportion des bits d'information qui seront décodés avec J_i équations.

Prenons deux exemples pour illustrer ces notions. Premièrement, considérons le CSO2C-WS ($J = 10$) dont les connexions sont aux positions $\{0, 29, 40, 43, 1020, 1328, 1495, 1606, 1696, 1698\}$ que nous perforons à un taux $R = \frac{3}{4}$ suivant la matrice ($\pi = 0$):

$$\mathbf{P} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad (4.25)$$

Nous allons déterminer les capacités de correction pour chaque bit. Commençons par $i = 0$. On doit alors calculer les restes des positions γ_j par k et les comparer à $(\pi - i) = 0$, ce qui donne:

$$\left\{ \begin{array}{lll} \gamma_1 & = 0 & = 0 \pmod{3} \\ \gamma_2 & = 29 & \neq 0 \pmod{3} \\ \gamma_3 & = 40 & \neq 0 \pmod{3} \\ \gamma_4 & = 43 & \neq 0 \pmod{3} \\ \gamma_5 & = 1020 & = 0 \pmod{3} \\ \gamma_6 & = 1328 & \neq 0 \pmod{3} \\ \gamma_7 & = 1495 & \neq 0 \pmod{3} \\ \gamma_8 & = 1606 & \neq 0 \pmod{3} \\ \gamma_9 & = 1696 & \neq 0 \pmod{3} \\ \gamma_{10} & = 1698 & = 0 \pmod{3} \end{array} \right. \quad (4.26)$$

Donc $\mathcal{J}_0 = \{\gamma_1, \gamma_5, \gamma_{10}\}$. Ainsi, $J_0 = 3$ et le décodeur dispose à chaque itération de trois équations de parité non nulles pour décoder u_0 . Si l'on répète ce processus pour J_1 , on doit calculer les restes des γ_j par k et les comparer à $(\pi - i) = -1 \pmod{3} = 2 \pmod{3}$. On obtient alors $J_1 = 2$. De la même manière, on obtient $J_2 = 5$. Si l'on poursuivait plus loin le processus, on obtiendrait $J_3 = 3$, $J_4 = 2$, etc. Ainsi, les valeurs trouvées pour J_i prennent trois valeurs différentes lorsque i varie, $i = 0, 1, 2$. La représentation spectrale de ce code à un taux $R = \frac{3}{4}$ comporte donc trois raies (une à $J_0 = 3$, une deuxième à $J_1 = 2$ et une troisième à $J_2 = 5$) et un tiers des bits d'information seront décodés avec trois équations de parité, un autre tiers avec deux équations de parité et enfin un dernier tiers avec cinq équations de parité. Ainsi, toutes les raies sont d'amplitude $\frac{1}{3}$ et la représentation spectrale du code est donnée à la figure 4.2.

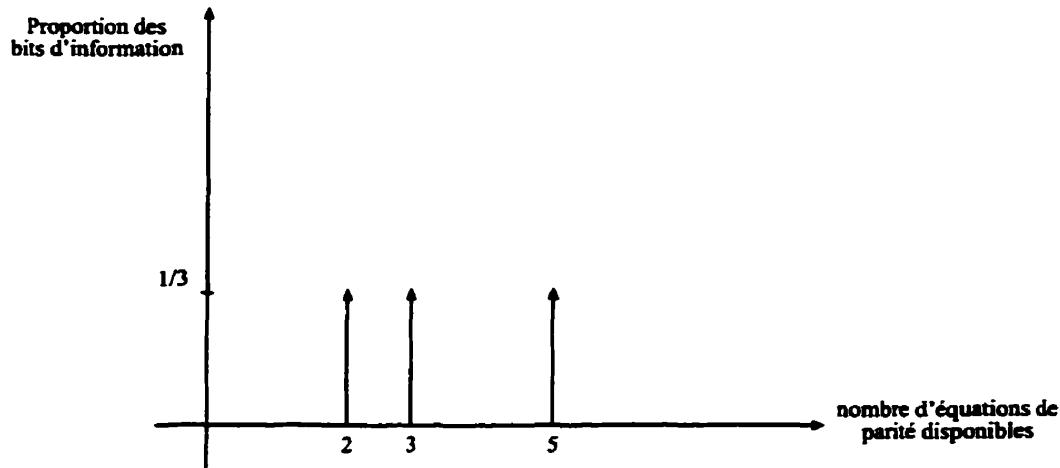


Figure 4.2: Spectre de perforation du PCSO2C-WS de taux $R = \frac{3}{4}$ ayant $J = 10$ connexions aux positions $\{0, 29, 40, 43, 1020, 1328, 1495, 1606, 1696, 1698\}$

Prenons à présent un deuxième exemple. Soit le CSO2C-WS ayant $J = 8$ connexions dont les positions sont $\{0, 43, 139, 322, 422, 430, 441, 459\}$ que nous perforons à un taux $R = \frac{2}{3}$ suivant la matrice suivante ($\pi = 1$):

$$\mathbf{P} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad (4.27)$$

De la même manière que dans l'exemple précédent, pour $i = 0$, on a $(\pi - i) = 1$ et:

$$\left\{ \begin{array}{ll} \gamma_1 &= 0 \neq 1 \pmod{2} \\ \gamma_2 &= 43 = 1 \pmod{2} \\ \gamma_3 &= 139 = 1 \pmod{2} \\ \gamma_4 &= 322 \neq 1 \pmod{2} \\ \gamma_5 &= 422 \neq 1 \pmod{2} \\ \gamma_6 &= 430 \neq 1 \pmod{2} \\ \gamma_7 &= 441 = 1 \pmod{2} \\ \gamma_8 &= 459 = 1 \pmod{2} \end{array} \right. \quad (4.28)$$

Ainsi, $J_0 = 4$ et de la même manière $J_1 = 4$. Comme $k = 2$, les J_i sont

périodiques de période 2 et seuls J_0 et J_1 sont nécessaires à la description du comportement du code. Ce code est donc tel que tous les bits d'information seront décodés avec le même nombre d'équations. Son spectre est représenté à la figure 4.3.

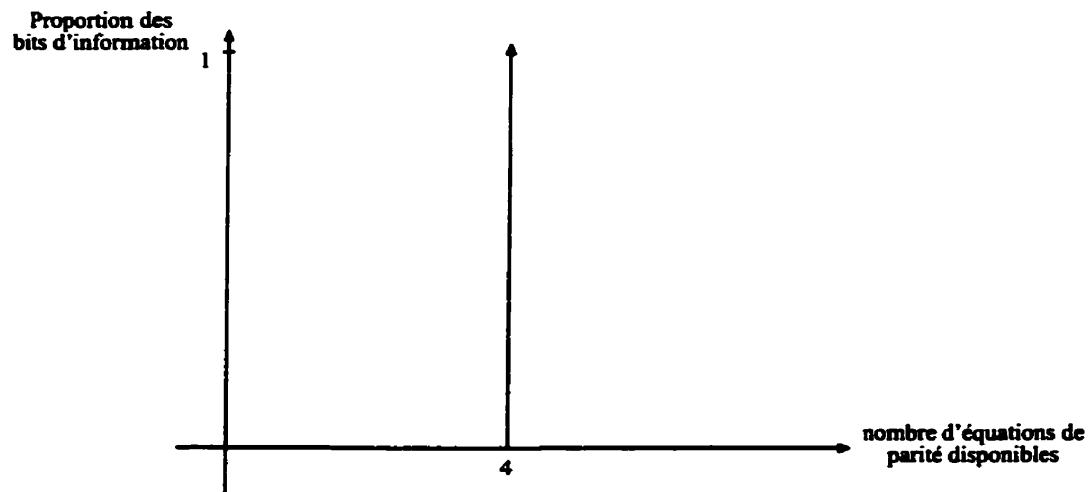


Figure 4.3: Spectre de perforation du PCSO2C-WS de taux $R = \frac{2}{3}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$

On peut remarquer que l'on peut calculer le spectre d'un code à un taux de perforation donné directement en calculant les restes des différentes positions par k du fait que l'opération de division des γ_j par k est commune à tous les instants i . En identifiant les restes obtenus avec $(\pi - i)$, on peut calculer en une seule fois les J_i qui nous intéressent. Par exemple, si l'on calcule les restes de manière explicite dans le premier exemple que nous avons traité, on a:

$$\left\{ \begin{array}{lll} \gamma_1 & = 0 & = 0 \pmod{3} \\ \gamma_2 & = 29 & = 2 \pmod{3} \\ \gamma_3 & = 40 & = 1 \pmod{3} \\ \gamma_4 & = 43 & = 1 \pmod{3} \\ \gamma_5 & = 1020 & = 0 \pmod{3} \\ \gamma_6 & = 1328 & = 2 \pmod{3} \\ \gamma_7 & = 1495 & = 1 \pmod{3} \\ \gamma_8 & = 1606 & = 1 \pmod{3} \\ \gamma_9 & = 1696 & = 1 \pmod{3} \\ \gamma_{10} & = 1698 & = 0 \pmod{3} \end{array} \right. \quad (4.29)$$

donc, comme $\pi = 0$ on a bien $J_0 = 3$, $J_1 = 2$ et $J_2 = 5$. On peut remarquer grâce à cette manière de calculer le spectre que l'ensemble des positions doit se "répartir" entre les k restes possibles et donc on peut avoir une répartition plus ou moins inégales des positions. De cette remarque, on peut également écrire:

$$J = \sum_{i=0}^{k-1} J_i \quad (4.30)$$

De plus, dans le cas où J_i est constant en fonction de i , c'est-à-dire dans le cas où tous les bits d'information sont décodés avec le même nombre d'équations de parité, on a:

$$J_i = \frac{J}{k}, \quad i = 0, 1, \dots, k-1 \quad (4.31)$$

On peut remarquer que pour qu'une telle condition soit remplie, il faut que J soit un multiple de k .

4.2.2 Étude des performances d'erreur

Nous allons présenter dans cette section les performances obtenues par simulation à l'ordinateur lors du décodage à seuil itératif de codes doublement orthogonaux perforés. Pour ce faire, nous allons étudier l'influence de divers

paramètres que sont le taux de codage R , la matrice de perforation \mathbf{P} , le nombre de connexions J et pour finir la position des connexions $\gamma_j, j = 1, 2, \dots, J$.

- **Influence du taux de codage R**

La figure 4.4 présente les performances en termes de probabilité d'erreur de PCSO2C-WS de taux de codage $R = \frac{2}{3}$ et $R = \frac{3}{4}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$. Comme prévu, plus le taux de codage augmente, plus les performances se dégradent.

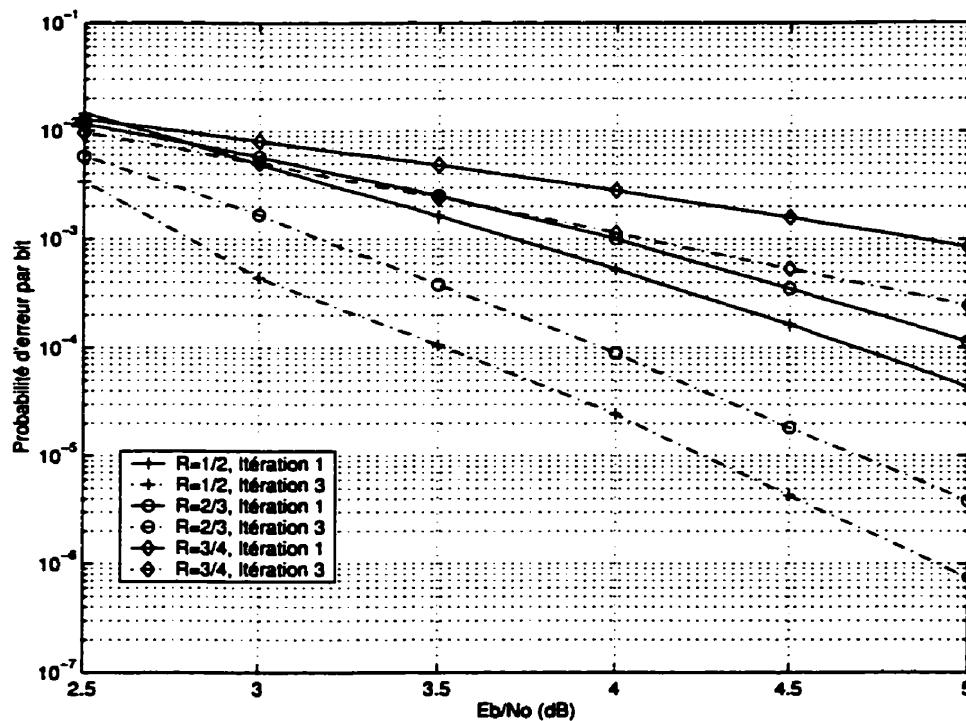


Figure 4.4: Performances à la première et à la troisième itération du décodage des PCSO2C-WS de taux $R = \frac{2}{3}$ et $R = \frac{3}{4}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$

Si l'on se réfère à la section précédente où a été analysée théoriquement l'influence de la perforation, on peut remarquer que plus le taux de codage

augmente et plus le nombre d'équations que va calculer le décodeur à seuil à chaque itération est faible, étant donné que J reste constant et que k augmente. En effet, les J positions doivent se "répartir" sur un ensemble de cardinalité plus grande, ce qui fait chuter les performances en terme de probabilité d'erreur. De manière plus quantitative, la figure 4.5 représente les spectres de perforation des codes utilisés. On voit donc bien à travers cette représentation l'effet du taux de codage sur les performances.

De plus, plus le nombre d'itérations augmente, plus l'écart de performances est flagrant. En effet, à la première itération, l'écart de performances entre les codes de taux $R = \frac{1}{2}$ et $R = \frac{3}{4}$ à une probabilité d'erreur de 10^{-3} est à peine supérieur à 1 dB alors qu'à la troisième itération, il est de l'ordre de 1,5 dB. Ceci vient du fait qu'à chaque itération, le décodeur à seuil se comporte de la même manière et donc ses effets sont amplifiés.

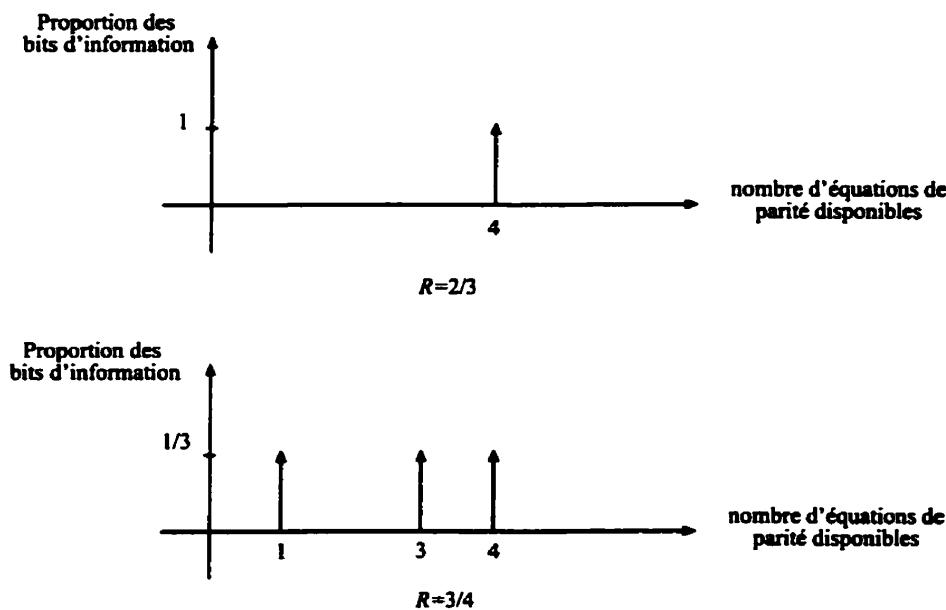


Figure 4.5: Spectre de perforation des PCSO2C-WS de taux $R = \frac{2}{3}$ et $R = \frac{3}{4}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$

● Influence de la matrice de perforation \mathbf{P}

La figure 4.6 permet de visualiser les performances du décodage à seuil itératif du PCSO2C-WS de taux $R = \frac{3}{4}$ ayant $J = 10$ connexions aux positions $\{0, 29, 40, 43, 1020, 1328, 1495, 1606, 1696, 1698\}$ perforé suivant les trois matrices de perforations possibles données par (4.32), (4.33) et (4.34).

$$\mathbf{P} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad (4.32)$$

$$\mathbf{P} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (4.33)$$

$$\mathbf{P} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.34)$$

On remarque que les performances au décodage sont parfaitement identiques quelle que soit la matrice de perforation utilisée et quelle que soit l'itération que l'on considère. Ceci s'explique très bien si l'on se réfère encore une fois à la représentation spectrale du code donné à la figure 4.2. En effet, la seule différence engendrée par le choix d'une matrice de perforation particulière porte uniquement sur la valeur de π . Or cela ne modifie en aucune manière le spectre de perforation du PCSO2C-WS utilisé et donc ne change aucunement ses performances. La seule différence que l'on peut observer est un décalage dans le temps du nombre d'équations utilisées pour décoder les différents bits d'information. Par exemple, pour la matrice de perforation donnée en (4.32), on avait obtenu $J_0 = 3$, $J_1 = 2$ et $J_2 = 5$ (le spectre de ce code est représenté à la figure 4.2). Dans le cas de la matrice donnée en (4.33), on obtiendrait $J_0 = 5$, $J_1 = 3$ et $J_2 = 2$. Ainsi, ce ne sont pas les mêmes bits qui sont décodés avec un nombre d'équations fixé. Ceci peut s'avérer utile dans le cas où l'on chercherait à protéger certains bits plus que d'autres.

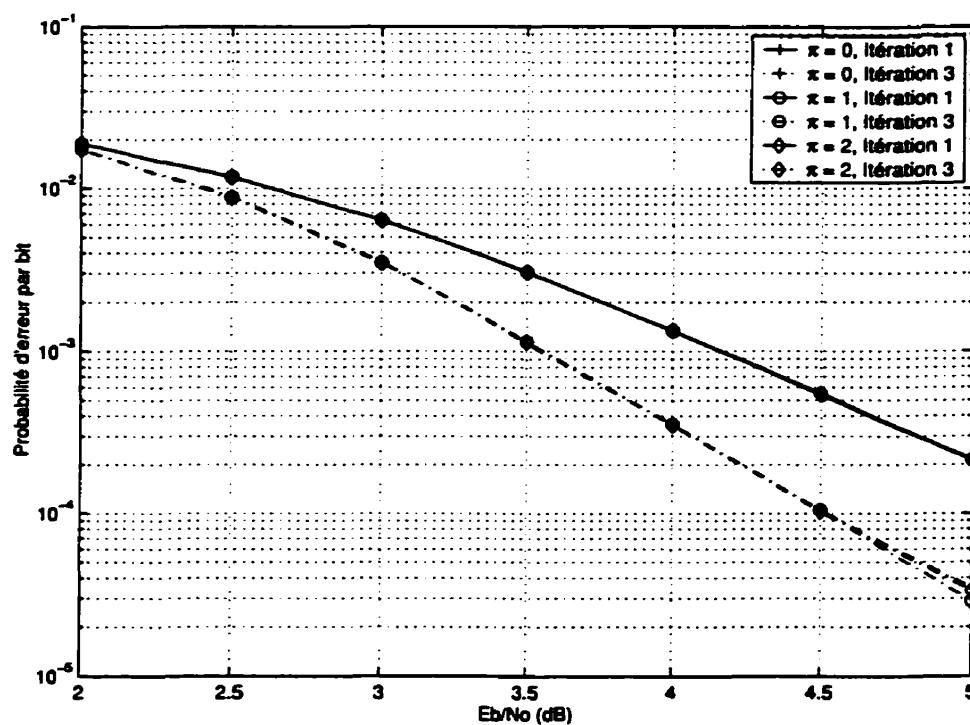


Figure 4.6: Performances à la première et à la troisième itération du décodage du PCSO2C-WS de taux $R = \frac{3}{4}$ ayant $J = 10$ connexions aux positions $\{0, 29, 40, 43, 1020, 1328, 1495, 1606, 1696, 1698\}$ perforé suivant les trois matrices données en (4.32), (4.33) et (4.34)

• Influence du nombre de connexions J

La figure 4.7 présente l'influence du nombre de connexions J sur les performances du décodeur à seuil itératif à un taux de codage fixe égal à $R = \frac{2}{3}$.

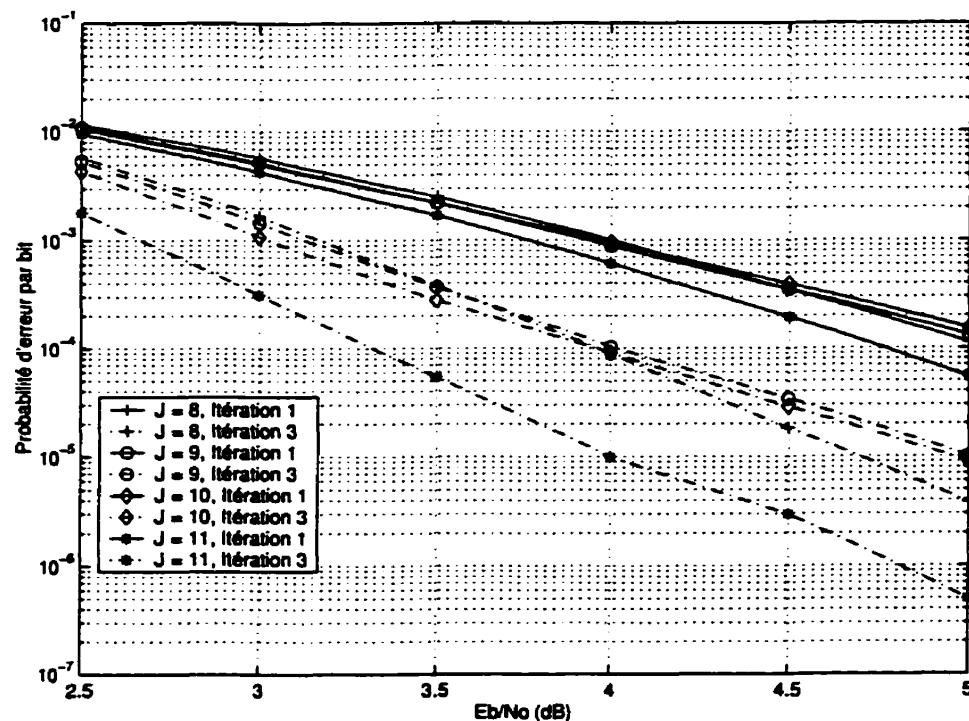


Figure 4.7: Performances à la première et à la troisième itération du décodage de plusieurs PCSO2C-WS de taux $R = \frac{2}{3}$ dont le nombre de connexions varie entre $J = 8$ et $J = 11$

Quatre codes de taux $R = \frac{2}{3}$ dont le nombre de connexions varie de $J = 8$ à $J = 11$ ont été testés et leurs spectres de perforation sont représentés à la figure 4.8.

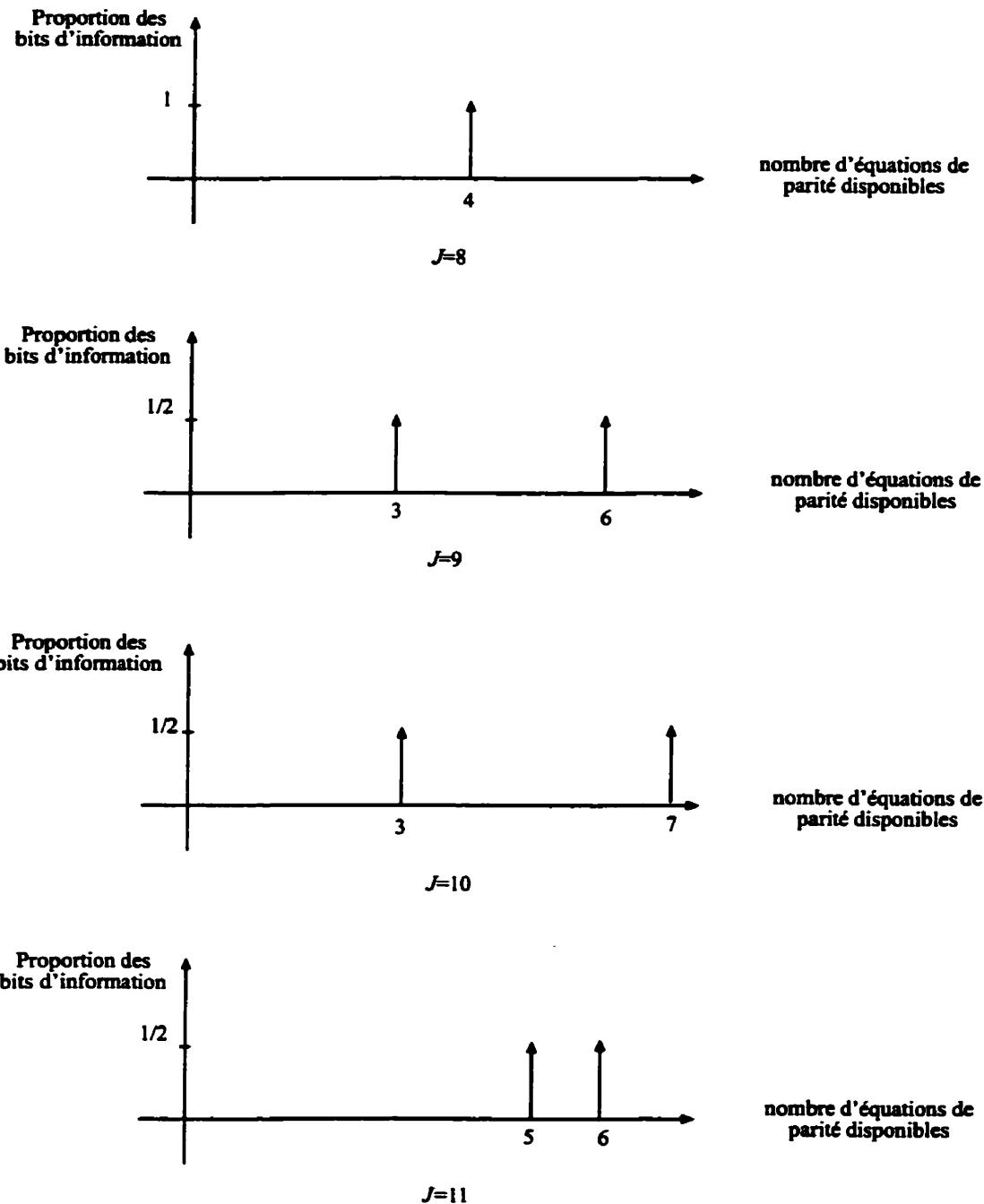


Figure 4.8: Spectres de perforation des PCSO2C-WS de taux $R = \frac{2}{3}$ utilisés à la figure 4.7

Étonnamment, d'après la figure 4.7, on remarque qu'à fort rapport signal sur bruit on peut dégrader les performances en utilisant des codes possédant un plus grand nombre de connexions, et ce quel que soit l'itération considérée. En effet, on remarque que le code possédant $J = 8$ connexions est meilleur que les codes possédant $J = 9$ ou $J = 10$ connexions au delà de 4 dB. Toutefois, c'est le code possédant $J = 11$ connexions qui reste le meilleur.

Ces résultats peuvent s'expliquer de part les spectres représentés à la figure 4.8. En effet, les deux codes possédant $J = 9$ ou $J = 10$ connexions ont tous deux la moitié des bits qui est décodée avec seulement trois équations de parité alors que tous les bits du code possédant $J = 8$ connexions sont décodés avec quatre équations de parité. Cela signifie que c'est le nombre d'équations minimal avec lequel certains bits sont décodés qui dicte majoritairement les performances asymptotiques du décodage, alors que les autres valeurs de J_i engendre des modifications du second ordre sur les performances. Nous allons revenir sur ce point dans le paragraphe suivant où ce phénomène est bien plus flagrant.

- **Influence de la position des connexions $\gamma_j, j = 1, 2, \dots, J$**

Afin de vérifier l'intuition décrite au paragraphe précédent, on a choisi de comparer les performances de PCSO2C-WS ayant tous $J = 8$ connexions mais dont les matrices génératrices (et donc les spectres) varient. La figure 4.9 synthétise ces résultats pour les spectres représentés à la figure 4.10. La figure 4.9 permet de confirmer que le fait d'avoir une fois sur deux un très faible nombre d'équations de parité détruit bien plus les performances du décodeur que le fait d'en avoir une fois sur deux un très grand nombre ne les améliore.

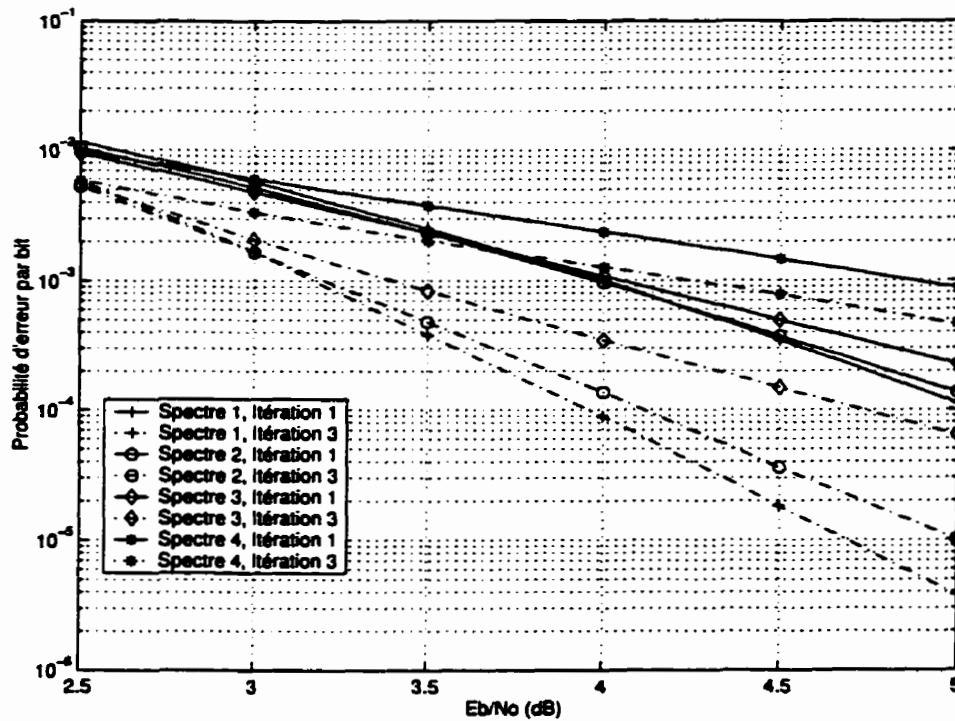


Figure 4.9: Performances à la première et à la troisième itération du décodage de plusieurs PCSO2C-WS ayant tous $J = 8$ connexions et dont les spectres sont représentés à la figure 4.10

Ceci peut s'expliquer si l'on analyse le gain de codage asymptotique de codage du décodeur à seuil. On a [11]:

$$G_c = 10 \log(R(J + 1)) \quad (4.35)$$

La courbe donnant le gain asymptotique de codage en fonction de J est donnée à la figure 4.11.

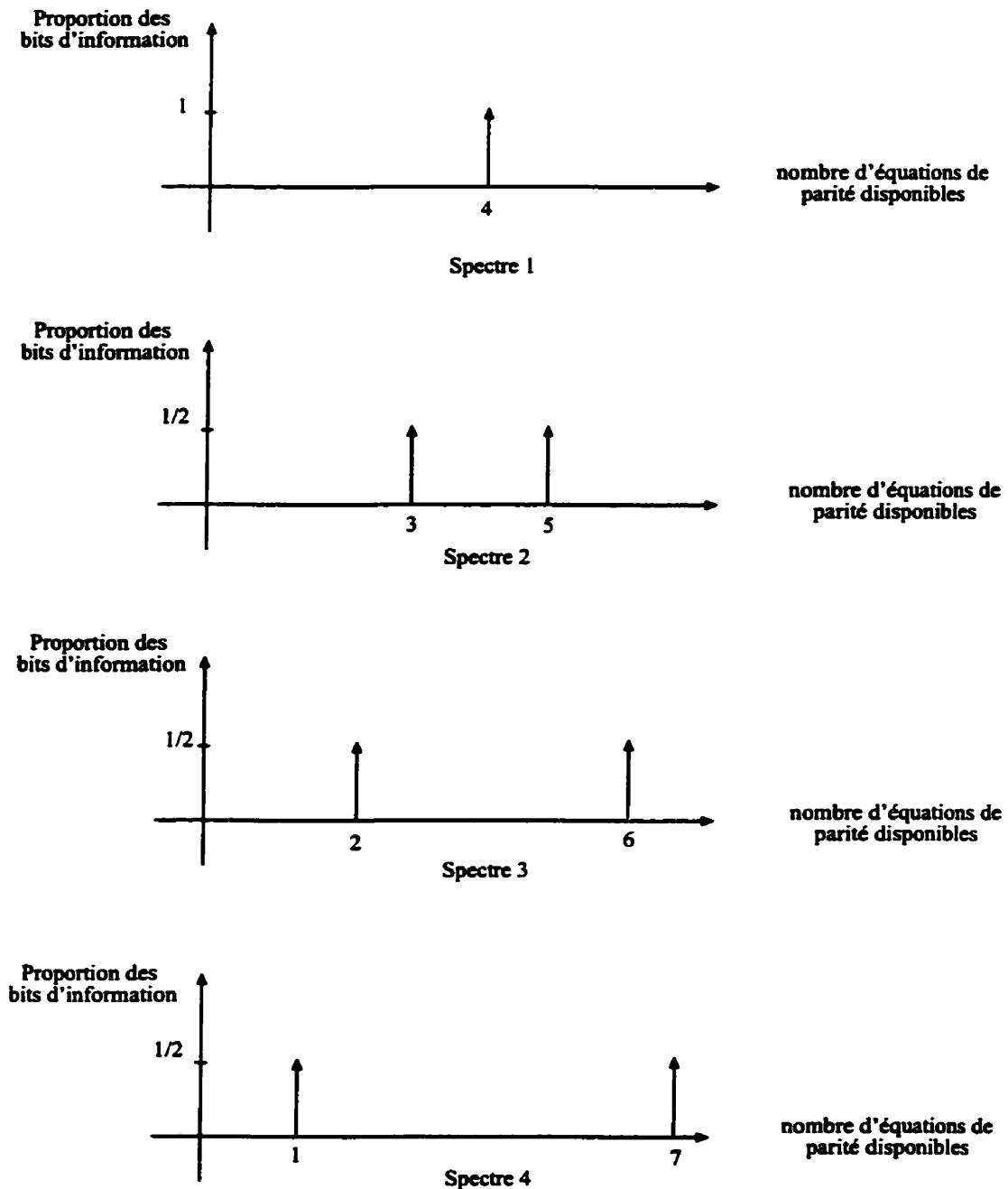


Figure 4.10: Spectres de perforation de PCSO2C-WS de taux $R = \frac{2}{3}$ ayant tous $J = 8$ connexions mais dont les positions varient

On peut se rendre compte à la lecture de cette courbe que les variations de gain asymptotique de codage sont beaucoup plus fortes pour de faibles valeurs de J que pour de fortes valeurs de J . Ceci vient du fait que:

$$\frac{d}{dJ}(10 \log(R(J + 1))) \propto \frac{1}{J + 1} \quad (4.36)$$

et donc la dérivée du gain asymptotique de codage prend des valeurs bien plus élevées pour les faibles J . Par conséquent, le fait de perdre des équations de décodage sur certains bits est bien plus dommageable du point de vue des performances que l'apport engendré par le gain d'équations sur d'autres bits.

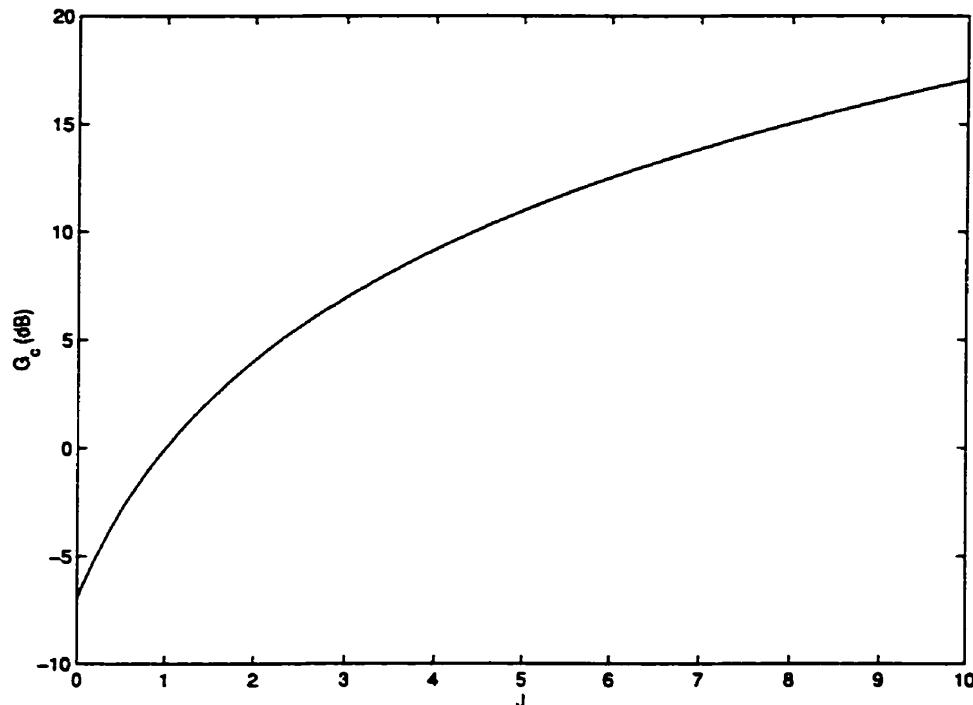


Figure 4.11: Gain asymptotique de codage d'un décodeur à seuil classique de taux $R = \frac{1}{2}$

Donc, pour décrire les performances du décodage à seuil itératif d'un PCSO2C-WS plus succinctement que par son spectre de perforation (mais néanmoins plus

incomplètement), on définit J_{\min}^{WS} comme étant le nombre minimal d'équations de parité dont dispose le décodeur:

$$J_{\min}^{WS} = \min_{i=0,1,\dots,k-1} \{J_i\} \quad (4.37)$$

4.2.3 Cas des codes de taux $R = \frac{k}{n}$

Durant toute la section précédente, nous n'avons parlé que de l'obtention de codes de taux $R = \frac{k}{k+1}$. Nous allons aborder à présent le cas des codes de taux quelconque $R = \frac{k}{n}$ où $(k+1) \leq n \leq 2k$ et analyser leur performance lors du décodage à seuil itératif.

Dans le but d'obtenir un code de taux $R = \frac{k}{n}$, on doit utiliser une matrice de perforation de longueur k ayant l'allure suivante:

$$\mathbf{P} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 1 & 0 & 1 & \dots & 0 & 1 & 0 \end{pmatrix} \quad (4.38)$$

On rappelle que les positions des 1 au sein de cette matrice sont notées $\pi_1, \pi_2, \dots, \pi_{n-k}$. De la même manière que celle qui nous a permis d'aboutir à (4.13), on a cette fois-ci l'ensemble des symboles conservés par l'opération de perforation qui s'écrit:

$$\begin{aligned} & \{y_{\pi_1+q_1k}^p, q_1 = 0, 1, \dots\} \cup \{y_{\pi_2+q_2k}^p, q_2 = 0, 1, \dots\} \cup \dots \\ & \cup \{y_{\pi_{n-k}+q_{n-k}k}^p, q_{n-k} = 0, 1, \dots\} \end{aligned} \quad (4.39)$$

Ainsi, les équations de parité qui seront conservées sont celles pour lesquelles leur indice j vérifie la relation suivante:

$$\left\{ \begin{array}{l} \gamma_j = (\pi_1 - i) \bmod(k), \text{ ou} \\ \gamma_j = (\pi_2 - i) \bmod(k), \text{ ou} \\ \vdots \\ \gamma_j = (\pi_{n-k} - i) \bmod(k) \end{array} \right. \quad (4.40)$$

Ainsi, et comme on devait s'y attendre *a priori*, le fait de considérer un taux de codage plus faible que $R = \frac{k}{k+1}$ améliore les performances du fait qu'une équation de parité a $(n - k)$ fois plus de chance d'être conservée par le processus de perforation.

Nous allons donc étendre l'étude théorique faite précédemment au cas des PCSO2C-WS de taux $R = \frac{k}{n}$. Considérons par exemple le code ayant $J = 10$ connexions dont les positions étaient données par $\{0, 29, 40, 43, 1020, 1328, 1495, 1606, 1696, 1698\}$. Nous avons déjà déterminé son spectre de perforation à un taux $R = \frac{3}{4}$ qui est représenté à la figure 4.2. Nous avons déterminé les restes de ces positions par la division modulo 3 et les résultats sont rappelés ci-dessous:

$$\left\{ \begin{array}{ll} \gamma_1 & = 0 \quad = 0 \bmod(3) \\ \gamma_2 & = 29 \quad = 2 \bmod(3) \\ \gamma_3 & = 40 \quad = 1 \bmod(3) \\ \gamma_4 & = 43 \quad = 1 \bmod(3) \\ \gamma_5 & = 1020 \quad = 0 \bmod(3) \\ \gamma_6 & = 1328 \quad = 2 \bmod(3) \\ \gamma_7 & = 1495 \quad = 1 \bmod(3) \\ \gamma_8 & = 1606 \quad = 1 \bmod(3) \\ \gamma_9 & = 1696 \quad = 1 \bmod(3) \\ \gamma_{10} & = 1698 \quad = 0 \bmod(3) \end{array} \right. \quad (4.41)$$

Considérons à présent la perforation de ce même code à un taux $R = \frac{3}{5}$. La matrice de perforation est choisie égale à:

$$\mathbf{P} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad (4.42)$$

Les équations qui seront conservées lors du décodage du bit u_0 sont celles pour lesquelles le reste de la division des positions correspondantes par $k = 3$ vaut $\pi_1 = 0$ ou $\pi_2 = 1$. Ainsi, $J_0 = 8$ et de la même manière $J_1 = 5$ et $J_2 = 7$. Donc pour obtenir le spectre de perforation d'un code à un taux $R = \frac{k}{n}$, il suffit de définir son

spectre à un taux $R = \frac{k}{k+1}$ et de prendre toutes les combinaisons de $(n - k)$ raies spectrales parmi les k raies présentes pour obtenir le spectre du code de taux $R = \frac{k}{n}$.

Un exemple de performances à deux taux de codage différents $R = \frac{3}{5}$ et $R = \frac{7}{10}$ est présenté à la figure 4.12. Comme prévu, on retrouve une dégradation progressive des performances au fur et à mesure que le taux de codage R augmente pour les mêmes raisons que dans le cas des codes de taux $R = \frac{k}{k+1}$. La grande différence de performance que l'on peut observer entre deux taux de codage somme toute assez proches peut s'expliquer si l'on observe les spectres des codes utilisés représentés à la figure 4.13. La mauvaise performance du code de taux $R = \frac{7}{10}$ vient du fait que certains bits seront décodés avec deux équations au décodage contre quatre au pire dans le cas du code de taux $\frac{3}{5}$.

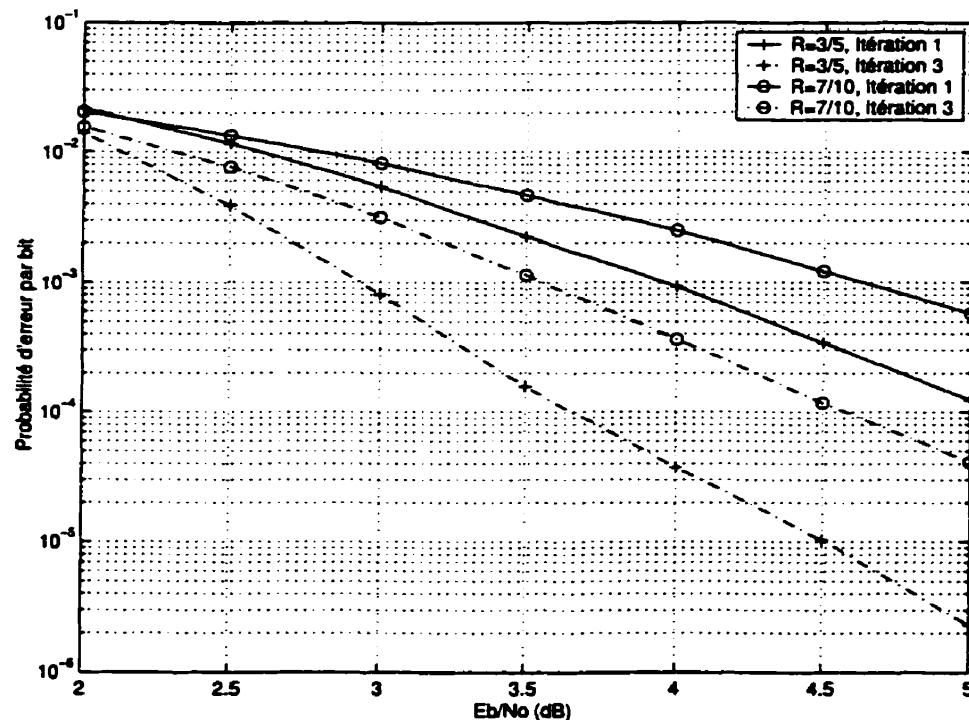


Figure 4.12: Performances à la première et à la troisième itération du décodage des PCSO2C-WS de taux $R = \frac{3}{5}$ et $R = \frac{7}{10}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$

De la même manière que dans le cas de la perforation à un taux $R = \frac{k}{k+1}$, la position des 1 au sein de la matrice de perforation n'a aucune influence sur les performances du décodage. De plus, on peut généraliser l'équation (4.30) en écrivant que:

$$\sum_{i=0}^{k-1} J_i = (n - k)J \quad (4.43)$$

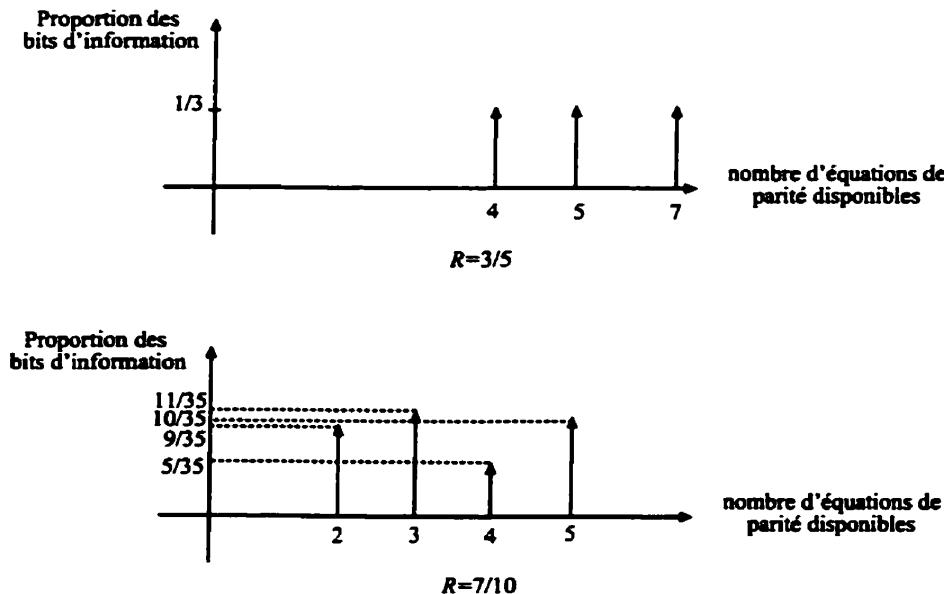


Figure 4.13: Spectres de perforation des PCSO2C-WS de taux $R = \frac{3}{5}$ et $R = \frac{7}{10}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$

En outre, le rôle de J_{\min}^{WS} est encore une fois valide et de manière plus générale, les codes de taux $R = \frac{k}{n}$ se comportent de la même manière que les codes de taux $R = \frac{k}{k+1}$. On avait établi que dans le cas de ce dernier, pour des codes présentant un nombre d'équations au décodage constant, on avait $J_{\min}^{WS} = \frac{J}{k}$. On peut généraliser cette formule en disant que dans le cas de codes ayant un spectre de perforation ne comportant qu'une seule raie:

$$J_{\min}^{WS} = J \left(\frac{1}{R} - 1 \right) \quad (4.44)$$

4.3 Propriétés des PCSO2C-WS

4.3.1 Recherche des PCSO2C-WS optimaux

Nous allons présenter dans cette section les meilleures performances d'erreur que l'on peut obtenir lors du décodage à seuil itératif des PCSO2C-WS ainsi que les codes qui permettent d'obtenir ces performances.

On a déjà mentionné précédemment que les meilleures performances sont obtenues pour des PCSO2C-WS dont le spectre de perforation ne comporte qu'une seule raie à $J_{\min}^{WS} = \frac{J}{k}$, car J_{\min}^{WS} est ainsi maximisé. Le critère principal d'optimisation pour la recherche de bons codes sera donc de rechercher parmi l'ensemble des PCSO2C-WS connus [1] ceux pour lesquels J_i est constant en fonction de i à un taux de codage donné. Pour ce faire, il suffit d'effectuer la division des positions γ_j par k et d'en regarder les restes comme on l'a expliqué dans les sections précédentes.

Parmi l'ensemble des PCSO2C-WS connus dont le spectre ne comporte qu'une seule raie à $J_{\min}^{WS} = \frac{J}{k}$, on choisira celui dont la longueur de contrainte est la plus faible, de telle sorte que le délai généré au codage et au décodage soit minimal. Les tableaux 4.1, 4.2 et 4.3 présentent les meilleurs PCSO2C-WS pour les trois taux de codage $R = \frac{2}{3}$, $R = \frac{3}{4}$ et $R = \frac{4}{5}$ respectivement.

Tableau 4.1: Meilleurs PCSO2C-WS de taux $R = \frac{2}{3}$

J	J_{\min}^{WS}	$\{\gamma_j, j = 1, 2, \dots, J\}$
4	2	$\{0, 1, 24, 37\}$
5	2	$\{0, 1, 24, 37, 41\}$
6	3	$\{0, 1, 17, 70, 95, 100\}$
7	3	$\{0, 1, 53, 128, 207, 216, 222\}$
8	4	$\{0, 43, 139, 322, 422, 430, 441, 459\}$
9	4	$\{0, 29, 75, 404, 864, 983, 1047, 1048, 1051\}$
10	5	$\{0, 220, 521, 695, 908, 926, 1059, 2457, 3367, 3458\}$
11	5	$\{0, 220, 521, 695, 908, 926, 1059, 2457, 3367, 3458, 3490\}$
12	6	$\{0, 576, 1331, 1897, 2222, 4684, 6502, 7293, 10785, 18999, 20901, 24372\}$
13	6	$\{0, 576, 1331, 1897, 2222, 4684, 6502, 7293, 10785, 18999, 20901, 22157, 24372\}$
14	7	$\{0, 576, 1331, 1897, 2222, 4684, 6502, 7293, 10785, 18999, 20901, 22157, 24372, 29532\}$
15	7	$\{0, 576, 1331, 1897, 2222, 4684, 6502, 7293, 10785, 18999, 20901, 22157, 24372, 27107, 29532\}$

En ce qui concerne le taux de codage $R = \frac{4}{5}$, on n'avait pas à notre disposition de codes doublement orthogonaux de longueur de contrainte $K < 10^5$ qui satisfaisait $J = 12$ et $J_{\min}^{WS} = 3$. C'est la raison pour laquelle le tableau 4.3 ne présente pas de résultat pour $J > 11$. De plus, on rappelle qu'aucune garantie n'est donnée ici quant à l'optimalité de la longueur de contrainte des codes présentés étant donné que certains des codes présentés n'appartiennent pas à la liste des codes optimaux dressée dans [1]. On rappelle que la prospection de ces codes ne s'est faite que parmi ceux qui sont connus et aucun nouveau CSO2C-WS n'a été découvert ici.

Les performances exhaustives de ces codes sont données en annexe I.

Tableau 4.2: Meilleurs PCSO2C-WS de taux $R = \frac{3}{4}$

J	J_{\min}^{WS}	$\{\gamma_j, j = 1, 2, \dots, J\}$
6	2	{0, 1, 5, 23, 93, 340}
7	2	{0, 1, 5, 23, 93, 197, 340}
8	2	{0, 49, 67, 173, 420, 606, 617, 620}
9	3	{0, 220, 521, 695, 908, 1059, 2457, 3367, 3490}
10	3	{0, 220, 521, 695, 908, 926, 1059, 2457, 3367, 3490}
11	3	{0, 1, 5, 7982, 15760, 16210, 22066, 24392, 38186, 50004, 66165}
12	4	{0, 1, 5, 7982, 15760, 16210, 22066, 24392, 38186, 50004, 66165, 81999}
13	4	{0, 1, 5, 7982, 15760, 16210, 22066, 23146, 24392, 38186, 50004, 66165, 81999}
14	4	{0, 1, 5, 7982, 15760, 16210, 22066, 23146, 24392, 38186, 50004, 66165, 76709, 81999}
15	5	{0, 1, 5, 7982, 15760, 16210, 22066, 23146, 24392, 38186, 50004, 66165, 76709, 81999, 82344}

Tableau 4.3: Meilleurs PCSO2C-WS de taux $R = \frac{4}{5}$

J	J_{\min}^{WS}	$\{\gamma_j, j = 1, 2, \dots, J\}$
8	2	{0, 220, 521, 695, 926, 1059, 2457, 3458}
9	2	{0, 220, 521, 695, 908, 926, 1059, 2457, 3458}
10	2	{0, 220, 521, 695, 908, 926, 1059, 2457, 3367, 3458}
11	2	{0, 220, 521, 695, 908, 926, 1059, 2457, 3367, 3458, 3490}

4.3.2 Codes imbriqués

À la section précédente, on a déterminé les meilleurs codes que l'on avait à notre disposition pour la perforation à un taux de codage donné. Le choix des connexions était guidé en premier lieu par l'obtention des meilleures performances possibles en termes de probabilité d'erreur et en second lieu par une longueur de contrainte minimale.

Une observation attentive des tableaux 4.1, 4.2 et 4.3 montre que le code ayant $J = 11$ connexions dont les positions sont $\{0, 220, 521, 695, 908, 926, 1059, 2457, 3367, 3458, 3490\}$ se révèle excellent pour les trois taux de codage considérés. On peut donc penser créer plusieurs codes imbriqués à partir de ce code et obtenir une structure souple permettant d'utiliser un code optimal en termes de probabilité d'erreur pour un taux de codage R donné à partir de ce seul code. Les tableaux 4.4, 4.5 et 4.6 présentent les positions des connexions que l'on peut utiliser pour ces codes imbriqués.

Ce genre de propriété peut s'avérer extrêmement avantageuse au niveau de la souplesse d'une implémentation éventuelle en VLSI, du fait qu'un seul codeur ayant $J = 10$ connexions est nécessaire. Les connexions sont alors activées ou non selon le compromis que l'on désire faire entre le niveau de performances et le délai du codage et du décodage et le taux de codage choisi.

Tableau 4.4: PCSO2C-WS imbriqués optimaux de taux $R = \frac{2}{3}$

J	$\{\gamma_j, j = 1, 2, \dots, J\}$
4	{0, 220, 521, 695}
5	{0, 220, 521, 695, 908}
6	{0, 220, 521, 695, 908, 1059}
7	{0, 220, 521, 695, 908, 926, 1059}
8	{0, 220, 521, 695, 908, 926, 1059, 2457}
9	{0, 220, 521, 695, 908, 926, 1059, 2457, 3367}
10	{0, 220, 521, 695, 908, 926, 1059, 2457, 3367, 3458}

Tableau 4.5: PCSO2C-WS imbriqués optimaux de taux $R = \frac{3}{4}$

J	$\{\gamma_j, j = 1, 2, \dots, J\}$
6	{0, 220, 521, 695, 1059, 3367}
7	{0, 220, 521, 695, 908, 1059, 3367}
8	{0, 220, 521, 695, 908, 1059, 2457, 3367}
9	{0, 220, 521, 695, 908, 1059, 2457, 3367, 3490}
10	{0, 220, 521, 695, 908, 926, 1059, 2457, 3367, 3490}

Tableau 4.6: PCSO2C-WS imbriqués optimaux de taux $R = \frac{4}{5}$

J	$\{\gamma_j, j = 1, 2, \dots, J\}$
8	{0, 220, 521, 695, 926, 1059, 2457, 3458}
9	{0, 220, 521, 695, 908, 926, 1059, 2457, 3458}
10	{0, 220, 521, 695, 908, 926, 1059, 2457, 3367, 3458}

4.3.3 Utilisation des coefficients de pondération pour le décodage des PCSO2C-WS

Jusqu'à présent, on a analysé les performances du décodeur à seuil itératif lors du décodage des PCSO2C-WS de divers taux de codage et on a présenté des codes qui permettaient d'obtenir les meilleures performances possibles au niveau du décodage. Toutefois, la recherche des codes optimaux pour la perforation (ceux dont le spectre de perforation ne comporte qu'une seule raie) s'est souvent faite au détriment de la longueur de contrainte du fait que les meilleurs PCSO2C-WS du point de vue du spectre de perforation trouvés ont rarement une longueur de contrainte minimale. Il peut donc être avantageux de pondérer les équations constitutives du décodeur afin d'améliorer les performances des codes qui ne sont pas optimaux mais dont la longueur de contrainte K est réduite. On pourrait y gagner en latence au niveau du codage et du décodage.

En effet, en analysant la manière dont la valeur MAP approximative $\lambda_i^{(\mu)}$ est passée d'une itération à la suivante, il existe un moyen d'améliorer les performances du décodeur en introduisant des coefficients de pondération au sein des équations $\psi_{i,j}^{(\mu)}$ de la manière suivante:

$$\lambda_i^{(\mu)} = a_0^{(\mu)} y_i^u + \sum_{j=1}^J a_j^{(\mu)} \psi_{i,j}^{(\mu)} \quad (4.45)$$

On rappelle que:

$$\psi_{i,j}^{(\mu)} = y_{i+\gamma_j}^p \diamond \sum_{k=1}^{j-1} \lambda_{i+\gamma_j-\gamma_k}^{(\mu-1)} \diamond \sum_{k=j+1}^J \lambda_{i+\gamma_j-\gamma_k}^{(\mu)} \quad (4.46)$$

si (4.13) est vérifiée et $\psi_{i,j}^{(\mu)} = 0$ sinon. Ainsi, pour décoder le bit u_i à l'itération μ , le décodeur à seuil va calculer J_i équations de parité à partir d'un certain nombre de valeurs MAP de l'itération précédente. Certaines de ces valeurs MAP $\lambda_{i+\gamma_j-\gamma_k}^{(\mu-1)}$ ont été calculées avec beaucoup d'équations de parité et d'autres non. L'idée est de favoriser par un coefficient de pondération plus élevé les équations qui utilisent davantage de ces valeurs MAP qui ont été calculées à l'itération précédente avec

beaucoup d'équations de parité afin de "rééquilibrer" les fiabilités des valeurs MAP. Regardons un exemple pour illustrer ces propos.

Soit le PCSO2C-WS de taux $R = \frac{2}{3}$ ayant $J = 9$ connexions aux positions $\{0, 9, 21, 395, 584, 767, 871, 899, 912\}$. Ce code est le meilleur obtenu jusqu'à présent en ce qui concerne la longueur de contrainte [1] mais il n'est pas optimal en ce qui concerne la probabilité d'erreur. En effet, si l'on considère la matrice de perforation suivante ($\pi = 0$):

$$\mathbf{P} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \quad (4.47)$$

et sachant que:

$$\left\{ \begin{array}{lll} \gamma_1 & = 0 & = 0 \bmod(2) \\ \gamma_2 & = 9 & = 1 \bmod(2) \\ \gamma_3 & = 21 & = 1 \bmod(2) \\ \gamma_4 & = 395 & = 1 \bmod(2) \\ \gamma_5 & = 584 & = 0 \bmod(2) \\ \gamma_6 & = 767 & = 1 \bmod(2) \\ \gamma_7 & = 871 & = 1 \bmod(2) \\ \gamma_8 & = 899 & = 1 \bmod(2) \\ \gamma_9 & = 912 & = 0 \bmod(2) \end{array} \right. \quad (4.48)$$

on a $\mathcal{J}_0 = \{\gamma_1, \gamma_5, \gamma_9\}$ et $\mathcal{J}_1 = \{\gamma_2, \gamma_3, \gamma_4, \gamma_6, \gamma_7, \gamma_8\}$ soit $J_0 = 3$ et $J_1 = 6$.

Ainsi, les valeurs MAP λ_i pour lesquelles i est pair auront été calculées avec peu d'équations de parité alors que les valeurs MAP pour lesquelles i est impair auront été calculées avec beaucoup d'équations de parité. L'idée est donc de favoriser à la deuxième itération les équations qui sont calculées avec des λ_i où i est impair.

Ainsi, lors de la deuxième itération, les coefficients des équations comportant plus de valeurs MAP pour lesquelles i est impair seront choisis plus élevés que ceux

des équations comportant plus de valeurs MAP pour lesquelles i est pair. Nous ne rentrerons pas plus dans les détails de ce concept étant donné que la détermination des coefficients $a_j^{(\mu)}$ utilisés au sein des équations dépasse le cadre de ce mémoire [11].

4.4 Cas des CSO2C-SS

4.4.1 Influence de la perforation: extension des PCSO2C-WS

Jusqu'à présent, nous n'avons considéré que le cas des CSO2C-WS, c'est-à-dire le cas de codes de taux $R = \frac{1}{2}$ ne permettant pas de satisfaire les exigences d'indépendance stricte des observables entre les itérations. Nous allons aborder dans cette section le cas de la perforation des CSO2C-SS introduits à la section 3.3.3. On rappelle que ces codes diffèrent majoritairement de leurs homologues au sens large du fait qu'ils sont constitués de J registres à décalage en parallèle. Par conséquent, le taux de codage est égal à $R = \frac{J}{2J} = \frac{1}{2}$.

Comme mentionné dans la section 3.3.3, un CSO2C-SS est spécifié non plus par un vecteur de position de connexions au sein d'un registre à décalage mais par une matrice de positions notée $(\gamma_{v,w})$ avec $v = 1, 2, \dots, J$ et $w = 1, 2, \dots, J$. Rappelons d'abord l'expression de la valeur MAP approximative obtenue par le décodeur à seuil [12], [16]:

- À la première itération:

$$\begin{aligned}\lambda_{v,i} &= y_{v,i}^u + \sum_{l=1}^J \left(y_{l,i+\gamma_{v,l}}^p \diamond \sum_{k=1}^{j-1} y_{k,i+\gamma_{v,l}-\gamma_{k,l}}^u \diamond \sum_{k=j+1}^J \lambda_{k,i+\gamma_{v,l}-\gamma_{k,l}} \right) \\ &= y_{v,i}^u + \sum_{l=1}^J \psi_{v,i,l}, \quad v = 1, 2, \dots, J\end{aligned}\tag{4.49}$$

- À l'itération $\mu \geq 2$:

$$\begin{aligned}\lambda_{v,i}^{(\mu)} &= y_{v,i}^u + \sum_{l=1}^J \left(y_{l,i+\gamma_{v,l}}^p \diamond \sum_{k=1}^{j-1} \lambda_{k,i+\gamma_{v,l}-\gamma_{k,l}}^{(\mu-1)} \diamond \sum_{k=j+1}^J \lambda_{k,i+\gamma_{v,l}-\gamma_{k,l}}^{(\mu)} \right) \\ &= y_{v,i}^u + \sum_{l=1}^J \psi_{v,i,l}^{(\mu)}, \quad v = 1, 2, \dots, J\end{aligned}\tag{4.50}$$

Nous allons suivre ici exactement le même raisonnement qui nous a permis d'aboutir à (4.13). D'après (4.49) et (4.50), le décodeur a besoin de l'ensemble des symboles de parité suivants pour décoder le bit $u_{v,i}$:

$$\{y_{l,i+\gamma_{v,l}}^p, \quad l = 1, 2, \dots, J\}\tag{4.51}$$

Intéressons-nous à présent à la matrice de perforation que nous allons devoir utiliser pour générer un code dont le taux de codage est équivalent à $R = \frac{k}{k+1}$. Celle-ci doit posséder k colonnes et $2J$ lignes (une pour chaque sortie du codeur). Les J premières lignes correspondent au bits d'information tandis que les J dernières correspondent aux symboles de parité. Comme dans le cas des PCSO2C-WS, on ne peut perforer les bits d'information et donc les J premières lignes de la matrice de perforation ne comportent que des uns. Dans le but d'atteindre un taux de codage équivalent à $R = \frac{k}{k+1}$, chacune des J dernières lignes ne doit comporter qu'un seul un. Par exemple, considérons la matrice de perforation suivante:

$$\mathbf{P} = \left(\begin{array}{cccc} 1 & 1 & \dots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \dots & 1 \\ \hline 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 1 \\ \vdots & \vdots & & \vdots \\ 0 & 1 & \dots & 0 \end{array} \right)\tag{4.52}$$

Vérifions que le taux de codage que permet d'atteindre cette matrice est conforme à nos attentes. Sur une période égale à la longueur de la matrice, c'est-à-dire

sur une période égale à k , le nombre de bits en entrée du codeur est égal à kJ et en sortie, on retrouve les kJ bits d'information (le codeur est systématique) ainsi que J symboles de parité (un par ligne de la matrice de perforation). Ainsi le taux de codage vaut:

$$R = \frac{kJ}{kJ + J} = \frac{kJ}{(k+1)J} \quad (4.53)$$

et est bien équivalent à $R = \frac{k}{k+1}$.

En considérant une telle matrice de perforation, les symboles de parité qui seront conservés seront ceux de la forme $\{y_{v,\pi_v+qk}^p, l = 1, 2, \dots, J, q = 0, 1, 2, \dots\}$ où π_v est la position du 1 sur la ligne $J + v$ de la matrice de perforation. Donc parmi ceux qui sont nécessaires au décodage de $u_{v,i}$, les seuls symboles de parité disponibles sont ceux qui vérifient:

$$i + \gamma_{v,l} = \pi_v + qk, \quad v = 1, 2, \dots, J, \quad q = 0, 1, 2, \dots \quad (4.54)$$

soit:

$$\boxed{\gamma_{v,l} = (\pi_v - i) \bmod(k), \quad l = 1, 2, \dots, J} \quad (4.55)$$

D'après (4.55), on peut remarquer une propriété fondamentale régissant le comportement des PCSO2C-SS. En effet, pour v fixé $v = 1, 2, \dots, J$, l'équation (4.55) est parfaitement équivalente à (4.13). Cela signifie que du point de vue des performances, les PCSO2C-SS sont complètement caractérisés par J spectres de perforation correspondant aux J codeurs constitutifs du CSO2C-SS.

Par conséquent, pour analyser les performances d'un PCSO2C-SS, il suffit d'analyser les spectres des J PCSO2C-WS spécifiés par $\{\gamma_{v,w}, w = 1, 2, \dots, J\}$ pour $v = 1, 2, \dots, J$ et les performances du PCSO2C-SS s'en déduisent immédiatement. Nous allons valider cette approche dans la section suivante.

4.4.2 Étude des performances d'erreur

Nous allons dans cette section étudier les performances d'erreur de divers PCSO2C-SS que nous analyserons à la lumière des résultats trouvés à la section précédente, soit que le comportement d'un PCSO2C-SS est équivalent à celui de J PCSO2C-WS.

- **Influence du taux de codage R**

Comme dans le cas des PCSO2C-WS, on remarque que les performances d'erreurs se dégradent au fur et à mesure que l'on augmente le taux de codage. La figure 4.14 présente les résultats pour un code origine de taux $R = \frac{7}{14}$ que l'on a perforé au taux $R = \frac{14}{21} = \frac{2*7}{3*7}$ et $R = \frac{21}{28} = \frac{3*7}{4*7}$ à la quatrième itération du décodage.

Ces résultats peuvent s'analyser grâce aux spectres de perforation de ce code aux différents taux de codage envisagés, c'est-à-dire que pour chaque codeur constitutif du PCSO2C-SS, la hausse du taux de codage entraîne une diminution du nombre d'équations disponibles au décodage. Ainsi, le phénomène est visible pour le PCSO2C-SS. Il semble toutefois d'après la figure 4.14 que les écarts de performances entre différents taux de codage soient plus importants que dans le cas des PCSO2C-WS. Nous allons analyser ce phénomène dans la section suivante où il apparaîtra comme une conséquence de l'influence des spectres de perforation des J codeurs constitutifs du CSO2C-SS.

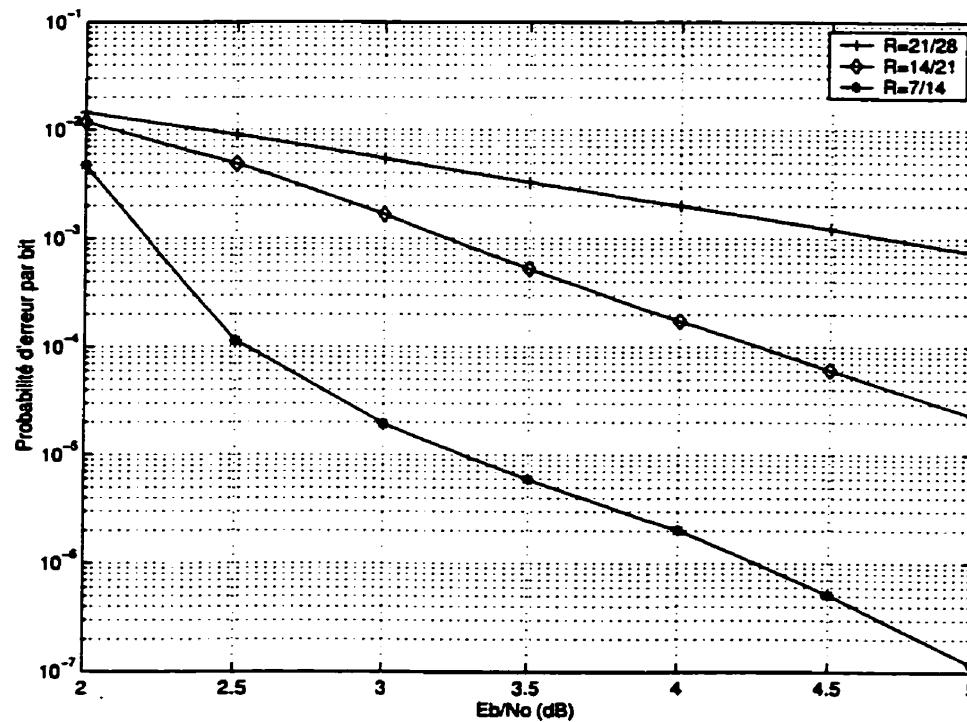


Figure 4.14: Performances à la quatrième itération des PCSO2C-SS de taux $R = \frac{7}{14}$, $R = \frac{14}{21}$ et $R = \frac{21}{28}$ ayant $J = 7$ connexions optimales du point de vue de la longueur de contrainte

- **Influence de la matrice de perforation P**

Comme dans le cas des PCSO2C-WS, on peut également se rendre compte que la matrice de perforation n'influe en aucun cas sur les performances du code étant donné que l'analyse par spectre est toujours valide. En effet, le changement de matrice de perforation n'agit que comme un décalage temporel sur les différentes raies des J spectres de perforation du PCSO2C-SS et donc les performances globales restent inchangées.

- **Influence de la position des connexions** $\gamma_j, j = 1, 2, \dots, J$

Nous allons ici étudier l'influence des J spectres du PCSO2C-SS sur ses performances. On peut s'attendre, comme dans le cas des PCSO2C-WS, à ce que plus les raies spectrales sont proches, meilleures sont les performances.

Ceci est en effet vérifié et est illustré par la figure 4.15. Les spectres de perforations des deux codes considérés sont représentés aux figures 4.16 et 4.17.

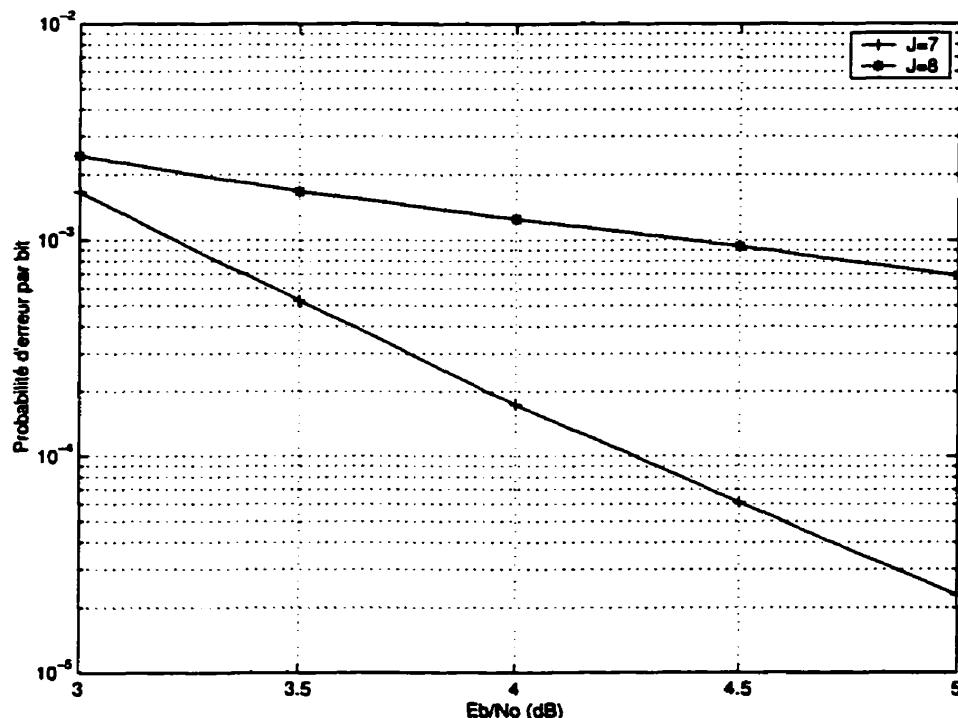


Figure 4.15: Performances à la quatrième itération de deux PCSO2C-SS de taux $R = \frac{2J}{3J}$ ayant respectivement $J = 7$ et $J = 8$ registres

On remarque que le code de taux $R = \frac{7}{14}$ est incomparablement meilleur que le code de taux $R = \frac{8}{16}$. Ceci peut s'expliquer si l'on regarde les spectres des deux PCSO2C-SS. La faible performance du PCSO2C-SS à $J = 8$ connexions est due au fait que l'un des codeurs constitutifs du CSO2C-SS présente un spectre pour lequel la moitié des bits d'informations est décodée avec aucune équation. Donc, pour qu'un PCSO2C-SS soit performant au niveau de la perforation à un taux donné, il faut que l'ensemble des spectres qui le caractérise soit bon. Ces conditions sont bien entendu bien plus difficiles à remplir que pour un PCSO2C-WS du fait qu'un seul spectre irrégulier détruit complètement les performances du code. Ceci explique les écarts importants que l'on a pu observer dans le paragraphe précédent à la figure 4.14.

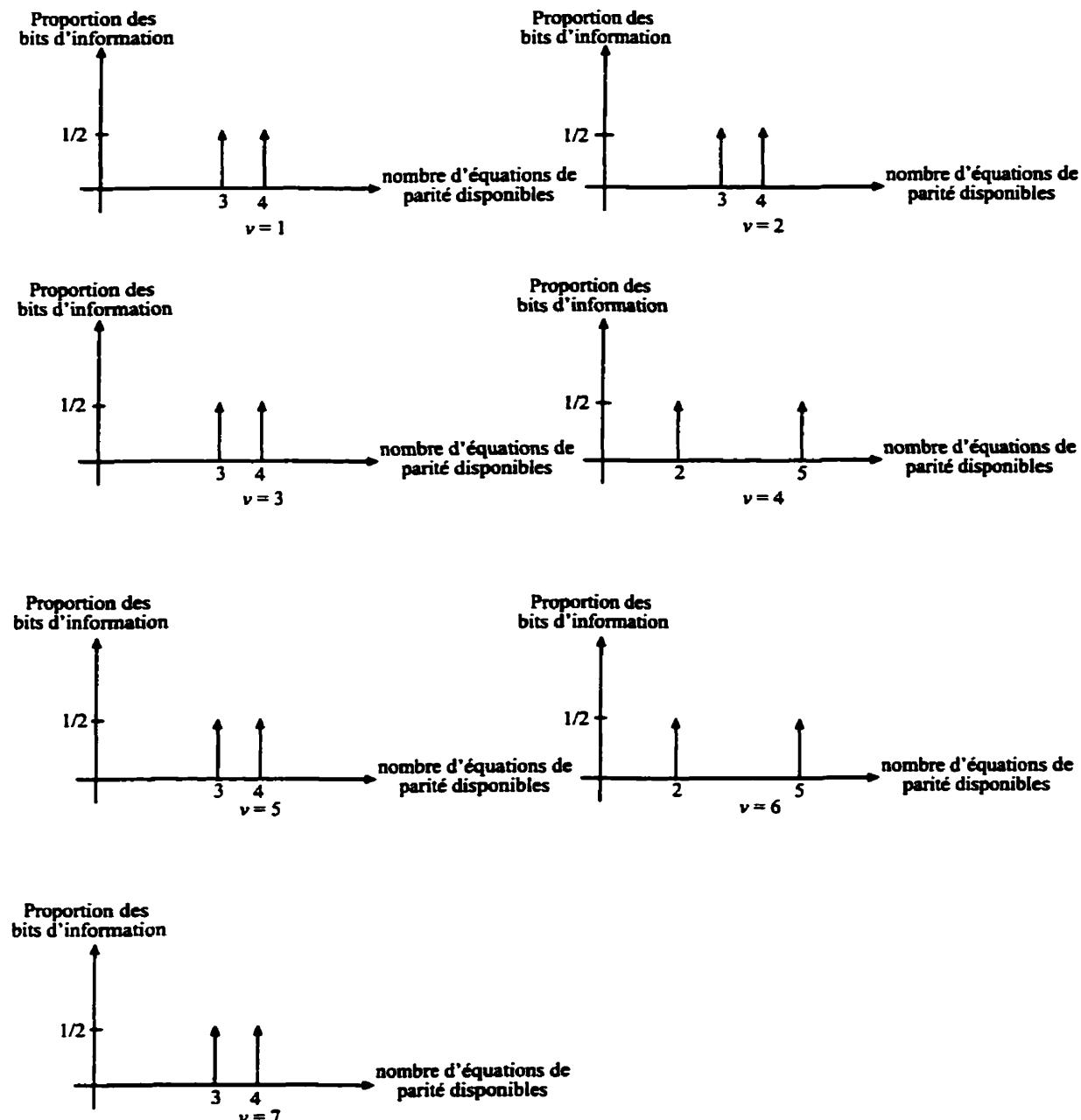


Figure 4.16: Spectres de perforation du PCSO2C-SS de taux $R = \frac{14}{21}$ ayant $J = 7$ registres optimal au sens de [1]

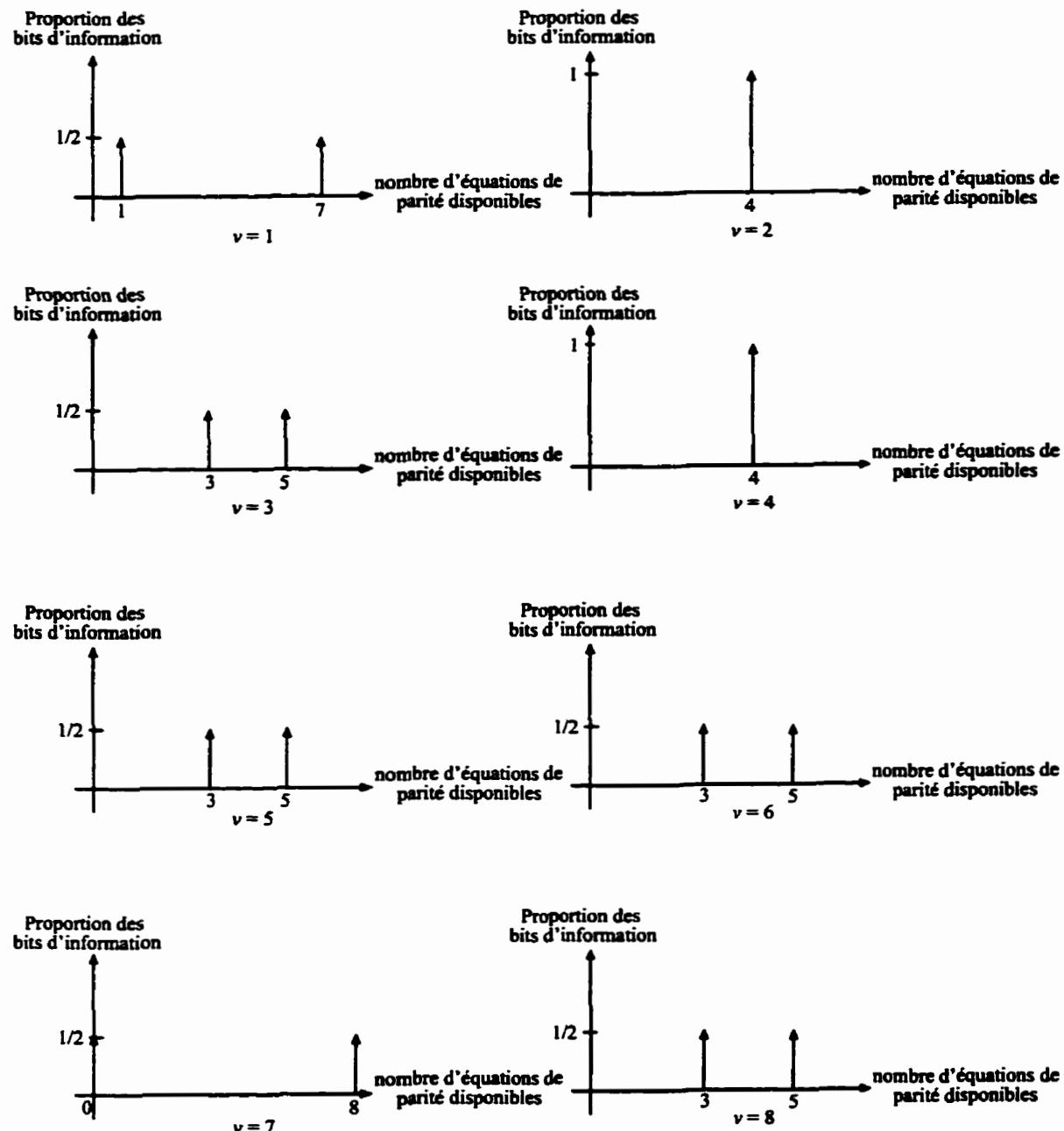


Figure 4.17: Spectres de perforation du PCSO2C-SS de taux $R = \frac{16}{24}$ ayant $J = 8$ registres optimal au sens de [1]

Ainsi, si l'on note par $\{J_{v,i}, v = 1, 2, \dots, J, i = 0, 1, 2, \dots\}$ le nombre d'équations de parité disponibles pour décoder le bit courant $u_{v,i}$, les performances du PCSO2C-SS sont majoritairement dictées par:

$$J_{min}^{SS} = \min_{i=0,1,\dots,k-1} \min_{v=1,2,\dots,J} \{J_{v,i}\} \quad (4.56)$$

On a cherché parmi les CSO2C-SS découverts dans [1], [2] les codes qui présentent de bonnes caractéristiques au niveau de la perforation aux taux $R = \frac{2J}{3J}$ et $R = \frac{3J}{4J}$. Le choix des codes étant limité, on a pu en obtenir qu'un faible nombre.

Pour un taux de codage égal à $R = \frac{2J}{3J}$, les meilleurs codes (c'est-à-dire ceux dont la latence est la plus faible) obtenus sont les suivants:

a) $J_{min}^{SS} = 1$

$$\begin{pmatrix} 13 & 0 & 24 & \underline{25} \\ 0 & 1 & 0 & 24 \\ 8 & \underline{25} & 14 & 0 \\ 15 & 3 & 23 & 0 \end{pmatrix}$$

b) $J_{min}^{SS} = 2$

$$\begin{pmatrix} 173 & 67 & 220 & 0 & 362 & 179 & 89 \\ 126 & 96 & 109 & 0 & 137 & 699 & 20 \\ 0 & 875 & 3 & 262 & 0 & 0 & 111 \\ 16 & 192 & 0 & \underline{879} & 25 & 40 & 0 \\ 270 & 59 & \underline{879} & 0 & 172 & 519 & 61 \\ 565 & 0 & 93 & 79 & 145 & 111 & 878 \\ 219 & 77 & 387 & 0 & 860 & 177 & 64 \end{pmatrix}$$

c) $J_{min}^{SS} = 3$

	0	119	0	<u>4804</u>	48	47	2499	239	952
2001	438	2676	0	1038	453	442	321	615	
397	782	1116	0	3938	529	456	266	403	
715	203	153	0	232	208	4803	1360	175	
2967	0	4803	201	0	0	0	0	0	
366	3124	421	0	580	2285	1217	238	377	
385	465	379	0	1842	875	562	258	4035	
493	1634	691	0	447	4597	440	287	402	
298	326	277	0	329	1227	717	4221	2455	

d) $J_{\min}^{SS} = 4$

22130	0	1592	1883	12941	2522	29211	345	2373	8980	2713	4383	2684
0	52	7362	205	0	15033	0	31175	0	0	0	51999	0
47237	<u>52000</u>	0	7171	2077	0	412	10009	22	20978	429	0	3586
7441	0	15851	3747	44036	1320	1919	21922	1439	1911	2201	1517	4059
5233	341	715	1000	1704	1196	13438	0	32131	1861	2012	6531	1802
1412	15649	74	0	695	162	818	916	4290	37882	879	404	8871
1100	4390	2880	182	808	285	937	0	41814	1447	11954	28252	913
2810	0	45475	7868	33906	12070	5274	405	2270	2920	2747	3291	3072
4035	0	3616	38093	26256	2101	7016	457	2291	16391	2752	2595	3472
40998	0	24239	2125	7262	26463	2728	469	4146	3597	9959	2313	2695
15034	0	1671	2354	3762	3619	2759	380	20591	51999	6305	10271	2867
2756	0	1630	18977	3208	6491	2778	381	2278	4364	35799	2415	16224
2765	0	1801	1923	2671	2099	3066	4420	11364	2746	3652	21611	31026

Les chiffres soulignés correspondent à la longueur de contrainte du plus grand des registres et donc permet de comparer la latence de ces codes par rapport au codes doublement orthogonaux au sens large étudiés auparavant. Dans le cas d'un taux de codage égal à $R = \frac{3J}{4J}$, seul le cas $J_{\min}^{SS} = 1$ a pu être trouvé. Il s'agit du code défini par la matrice de connexions suivante:

$$\left(\begin{array}{cccccc} 110 & 50 & 0 & 301 & 228 & 67 \\ 311 & 42 & 5 & 0 & 79 & 154 \\ 94 & 71 & 0 & 0 & 82 & \underline{323} \\ 0 & 25 & \underline{323} & 18 & 0 & 0 \\ 46 & 0 & 6 & 300 & 36 & 20 \\ 90 & 232 & 0 & 14 & 120 & 69 \end{array} \right)$$

Évidemment, cette liste ne peut être qu'améliorée du fait que la prospection de bons codes ne s'est fait que parmi la liste des codes déjà existante. De plus les codes présentés ici ne sont pas nécessairement optimaux. Ce sont juste les meilleurs que l'on avait à notre disposition. L'annexe II présente les performances des PCSO2C-SS dont les matrices de générateurs sont données ci-dessus, excepté le code pour lequel $J_{\min}^{SS} = 4$ du fait du temps de simulation trop important (supérieur à une semaine) sur l'ordinateur personnel dont on a fait mention en introduction.

4.5 Bilan

Nous avons présenté dans ce chapitre l'influence de la perforation des codes doublement orthogonaux (au sens large et au sens strict) sur les performances de l'algorithme de décodage à seuil itératif.

L'effet principal de la perforation dans l'utilisation de codes doublement orthogonaux réside dans la suppression d'équations de parité disponibles pour le décodage des bits d'information. Nous avons analysé le comportement de ces codes à l'aide d'une représentation spectrale et en avons déduit de nombreuses implications, parmi lesquelles le fait que plus le spectre du code présente des raies rapprochées, meilleures sont ses performances. Nous avons donc dressé la liste des meilleurs codes doublement orthogonaux en notre possession pour la perforation à des taux de codage $R = \frac{k}{k+1}$ et avons présenté de manière exhaustive les meilleures performances que l'on pouvait obtenir à l'aide de cette technique.

Toutefois, il faut noter qu'il y a rarement coïncidence entre l'optimalité en termes de longueur de contrainte (dictant le délai au codage et au décodage) et en terme de représentation spectrale (dictant les performances de probabilité d'erreur). En effet, la recherche de bons codes pour la perforation s'est faite le plus souvent au détriment de γ_J . La mise en œuvre de coefficients de pondération au sein des équations constitutives du décodeur semble être une voie de recherche intéressante pour améliorer les performances des codes à faible longueur de contrainte et qui présentent plusieurs raies spectrales.

Dans le chapitre suivant, nous allons montrer comment on peut utiliser les codes doublement orthogonaux perforés en association avec des modulations à efficacités spectrales élevées telles les modulations *M-QAM* dans le but d'augmenter encore plus le rendement global du système de communications étudié.

CHAPITRE 5

APPLICATION AUX MODULATIONS À EFFICACITÉS SPECTRALES ÉLEVÉES

5.1 Introduction aux modulations numériques

Nous allons exposer en premier lieu quelques notions élémentaires sur les modulations numériques à fortes efficacités spectrales qui sont indispensables à la compréhension de ce chapitre. Pour plus de renseignements sur la représentation des signaux sous forme vectorielle, on se reportera à l'annexe III où l'on introduit ce type de représentation.

Dans un système de communications numériques, le modulateur a pour but d'assigner à toute séquence d'éléments binaires de longueur définie par le type de la modulation un signal distinct qui sera transmis sur le canal de communications. Cette assignation est réalisée en découplant la séquence d'éléments binaires à transmettre en symboles de $m = \log_2(M)$ éléments binaires et en choisissant pour chacun de ces symboles un des $M = 2^m$ signaux déterministes à énergie finie $\{s_q(t), q = 1, 2, \dots, M\}$ qui sera transmis sur le canal. Une modulation sera dite à forte efficacité spectrale si $m \geq 2$, ce qui signifie que plusieurs symboles codés sont regroupés au sein du même symbole de canal. Les différents signaux de la constellation peuvent différer en amplitude, en phase, en fréquence ou en une combinaison de ces facteurs. Nous allons étudier brièvement quelques types de modulations à fortes efficacités spectrales associées à ces différences [27].

5.1.1 Modulation M -PAM

Dans une modulation M -PAM (*Pulse Amplitude Modulation*), les signaux peuvent s'écrire sous la forme suivante:

$$\begin{aligned}s_q(t) &= \operatorname{Re}[A_q g(t) e^{j2\pi f_c t}] \\ &= A_q g(t) \cos(2\pi f_c t), \quad q = 1, 2, \dots, M, \quad 0 \leq t \leq T\end{aligned}\quad (5.1)$$

où $\{A_q, q = 1, 2, \dots, M\}$ représente l'ensemble des amplitudes possibles correspondant aux $M = 2^m$ blocs de m bits. Les amplitudes prennent les valeurs discrètes données par:

$$A_q = (2q - 1 - M)d, \quad q = 1, 2, \dots, M \quad (5.2)$$

où $2d$ est la distance entre deux signaux adjacents. Le signal $g(t)$ est une fonction de mise en forme à valeurs réelles et sa nature influe sur le spectre du signal transmis.

Les M signaux de la constellation ont pour énergie:

$$\mathcal{E}_q = \int_0^T s_q^2(t) dt = \frac{1}{2} A_q^2 \mathcal{E}_g, \quad q = 1, 2, \dots, M \quad (5.3)$$

où \mathcal{E}_g représente l'énergie du signal $g(t)$. Il est de plus évident que ces signaux forment donc un espace vectoriel de dimension 1. Ils peuvent ainsi s'exprimer sous la forme:

$$s_q(t) = s_q f(t) \quad (5.4)$$

où $f(t)$ est la fonction de base:

$$f(t) = \sqrt{\frac{2}{\mathcal{E}_g}} g(t) \cos(2\pi f_c t) \quad (5.5)$$

et:

$$s_q = A_q \sqrt{\frac{\mathcal{E}_g}{2}}, \quad q = 1, 2, \dots, M \quad (5.6)$$

La constellation correspondante dans l'espace des signaux pour $M = 4$ est donnée par la figure 5.1. Cette modulation est également appelée modulation

M-ASK (Amplitude Shift Keying).

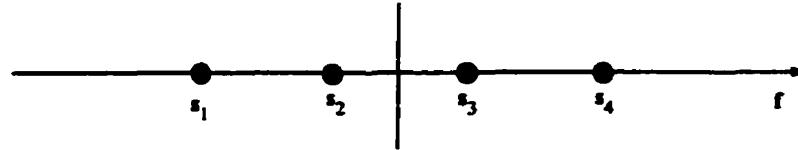


Figure 5.1: Constellation M -PAM pour $M = 4$

5.1.2 Modulation M -PSK

La modulation M -PSK (*Phase Shift Keying*) est une modulation numérique de phase, les M signaux de la constellation étant représentés par:

$$\begin{aligned} s_q(t) &= \operatorname{Re}[g(t)e^{j2\pi\frac{q-1}{M}}e^{j2\pi f_c t}], \quad q = 1, 2, \dots, M, \quad 0 \leq t \leq T \\ &= g(t) \cos(2\pi f_c t + \frac{2\pi}{M}(q-1)) \\ &= g(t) \left[\cos(\frac{2\pi}{M}(q-1)) \cos(2\pi f_c t) - \sin(\frac{2\pi}{M}(q-1)) \sin(2\pi f_c t) \right] \end{aligned} \quad (5.7)$$

Les $\{\theta_q = 2\pi\frac{q-1}{M}, q = 1, 2, \dots, M\}$ sont les M phases possibles de la porteuse. Chaque signal a une énergie constante donnée par:

$$\mathcal{E} = \int_0^T s_q^2(t) dt = \frac{1}{2}\mathcal{E}_g \quad (5.8)$$

De plus, d'après (5.7), on peut exprimer chaque signal de la constellation comme une combinaison linéaire de deux fonctions de base $f_1(t)$ et $f_2(t)$ de la manière suivante:

$$s_q(t) = s_{q,1}f_1(t) + s_{q,2}f_2(t), \quad q = 1, 2, \dots, M \quad (5.9)$$

où:

$$f_1(t) = \sqrt{\frac{2}{\mathcal{E}_g}} g(t) \cos(2\pi f_c t) \quad (5.10)$$

$$f_2(t) = \sqrt{\frac{2}{\mathcal{E}_g}} g(t) \sin(2\pi f_c t) \quad (5.11)$$

et où $\mathbf{s}_q = [s_{q,1} \ s_{q,2}]$ est donné par:

$$\mathbf{s}_q = \left[\sqrt{\frac{\mathcal{E}_g}{2}} \cos\left(\frac{2\pi}{M}(q-1)\right) \ - \sqrt{\frac{\mathcal{E}_g}{2}} \sin\left(\frac{2\pi}{M}(q-1)\right) \right], \quad q = 1, 2, \dots, M \quad (5.12)$$

De la même manière que la modulation M -PAM, la modulation M -PSK permet d'atteindre de fortes efficacités spectrales lorsque M devient grand. La constellation correspondante dans l'espace des signaux est donnée pour $M = 2$ (modulation BPSK), $M = 4$ (modulation QPSK) et $M = 8$ (modulation 8-PSK) par la figure 5.2.

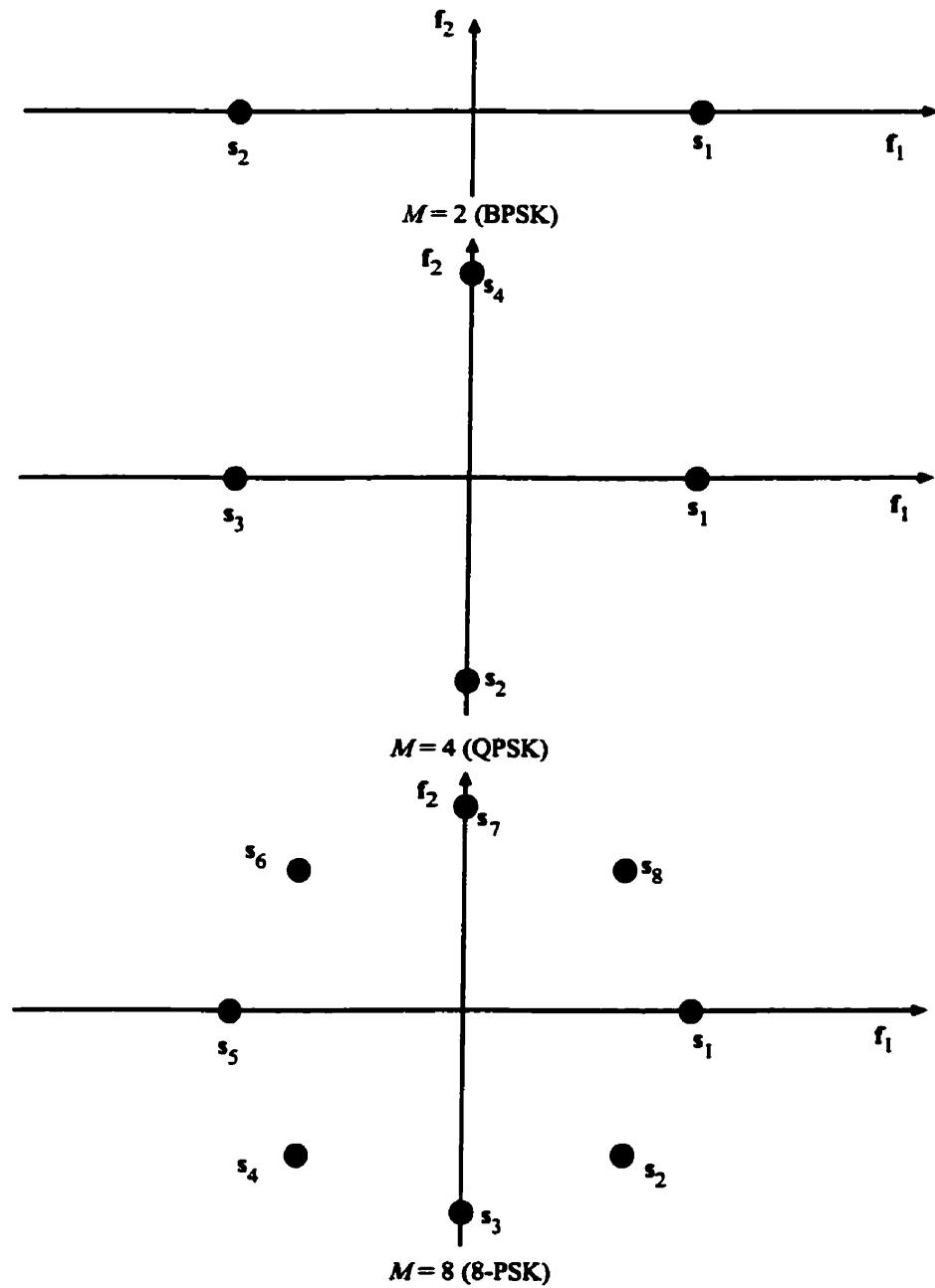


Figure 5.2: Constellations M -PSK pour $M = 2$, $M = 4$ et $M = 8$

5.1.3 Modulation M -QAM

L'efficacité spectrale de la modulation M -PAM peut être obtenue également en utilisant deux porteuses en quadrature ($\cos(2\pi f_c t)$ et $\sin(2\pi f_c t)$), c'est-à-dire de la même manière que pour la modulation M -PSK mais cette fois-ci en occupant plusieurs niveaux d'énergie. On obtient alors la modulation M -QAM (*Quadrature Amplitude Modulation*) où les signaux s'expriment de la manière suivante:

$$\begin{aligned} s_q(t) &= \operatorname{Re}[(A_{q,c} + jA_{q,s})g(t)e^{j2\pi f_c t}], \quad q = 1, 2, \dots, M, \quad 0 \leq t \leq T \\ &= A_{q,c}g(t)\cos(2\pi f_c t) - A_{q,s}g(t)\sin(2\pi f_c t) \end{aligned} \quad (5.13)$$

où ce sont les amplitudes $A_{q,c}$ et $A_{q,s}$ qui portent l'information. En écrivant (5.13) d'une autre manière, on obtient:

$$\begin{aligned} s_q(t) &= \operatorname{Re}[V_q e^{j\theta_q} g(t) e^{j2\pi f_c t}] \\ &= V_q g(t) \cos(2\pi f_c t + \theta_q), \quad q = 1, 2, \dots, M \end{aligned} \quad (5.14)$$

où $V_q = \sqrt{A_{q,c}^2 + A_{q,s}^2}$ et $\theta_q = \arctan(\frac{A_{q,s}}{A_{q,c}})$. Cette écriture permet de mieux se rendre compte que la modulation M -QAM est à la fois une modulation de phase (M -PSK) et une modulation d'amplitude (M -PAM). Un exemple de constellation M -QAM est montré à la figure 5.3.

L'énergie de chaque signal est donnée par:

$$\mathcal{E}_q = \int_0^T s_q^2(t) dt = \frac{1}{2} V_q^2 \mathcal{E}_g = \frac{A_{q,c}^2 + A_{q,s}^2}{2} \mathcal{E}_g, \quad q = 1, 2, \dots, M \quad (5.15)$$

Ainsi, les signaux de la constellation se retrouvent sur un certain nombre de niveaux d'énergie déterminés par les valeurs de $A_{q,c}$ et de $A_{q,s}$.

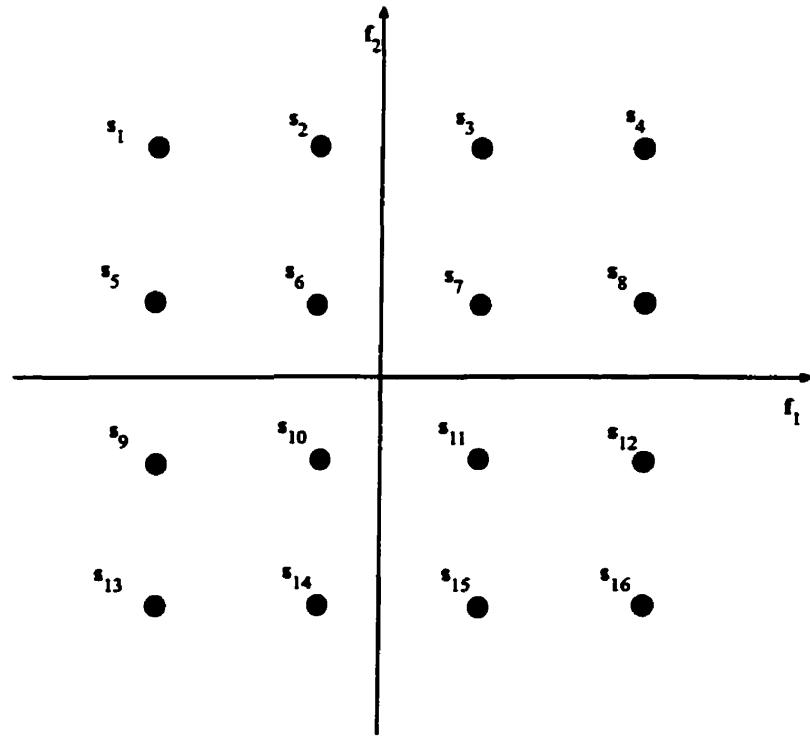


Figure 5.3: Constellation M -QAM pour $M = 16$

Comme dans le cas de la modulation M -PSK, tout signal de la constellation M -QAM peut être écrit comme combinaison linéaire de deux signaux de base, soit:

$$s_q(t) = s_{q,1}f_1(t) + s_{q,2}f_2(t) \quad (5.16)$$

où:

$$f_1(t) = \sqrt{\frac{2}{\mathcal{E}_g}} g(t) \cos(2\pi f_c t) \quad (5.17)$$

$$f_2(t) = \sqrt{\frac{2}{\mathcal{E}_g}} g(t) \sin(2\pi f_c t) \quad (5.18)$$

et où $s_q = [s_{q,1} \ s_{q,2}]$ est donné par:

$$\mathbf{s}_q = \left[A_{q,c} \sqrt{\frac{\mathcal{E}_g}{2}} \quad -A_{q,s} \sqrt{\frac{\mathcal{E}_g}{2}} \right], \quad q = 1, 2, \dots, M \quad (5.19)$$

5.1.4 Autres types de modulations

Il existe d'autres types de modulations que nous allons citer sans plus entrer dans les détails.

En ce qui concerne les types de modulations envisagées jusqu'à présent (modulations sans mémoire), on retiendra les constellations:

- orthogonale (les signaux diffèrent en fréquence et sont donc non corrélés);
- biorthogonale (on ajoute les opposés des signaux de la constellation orthogonale);
- transorthogonale ou simplex qui est une translation de la constellation orthogonale effectuée dans le but de minimiser son énergie moyenne.

Il existe également un autre type de modulations qui sont les modulations avec mémoire pour lesquelles il existe une dépendance entre les signaux transmis correspondant à des symboles de canal successifs, comme par exemple la modulation CPM (*Continuous Phase Modulation*).

5.2 Assignation des symboles de canal aux signaux de la constellation par codage de Gray

Nous avons présenté à la section 5.1 les modulations numériques à fortes efficacités spectrales les plus utilisées dans les systèmes de télécommunications modernes. Autrement dit, nous avons présenté dans l'espace des signaux les positions relatives des signaux au sein des différentes constellations pour les modulations à fortes efficacités spectrales que sont les modulations *M-PAM*,

M-PSK et *M*-QAM. Dans la suite de ce chapitre, nous ne parlerons plus de la modulation *M*-PAM, et presque plus de la modulation *M*-PSK pour nous concentrer sur la modulation *M*-QAM.

Nous allons aborder à présent un point crucial de l'utilisation des modulations multiniveaux, soit le problème de l'assignation de la séquence binaire que l'on désire transmettre aux différents signaux de la constellation. Nous allons nous intéresser en premier lieu à l'assignation par codage de Gray. Une deuxième technique d'assignation (l'assignation par partition d'ensemble) sera présentée à la section 5.4.

Étant donnée une constellation de M signaux, il existe $M! = M(M - 1)(M - 2)\dots 1$ assignations possibles entre les $M = 2^m$ symboles émis par le codeur et les M signaux de la constellation (on exclue le cas où deux symboles sont assignés au même signal). L'assignation utilisée n'est jamais choisie au hasard mais on recherche souvent à minimiser le nombre de bits en erreur par symbole en erreur. Ainsi, si un symbole de m bits erroné est reçu, on recherche à ce que le nombre de bits en erreur correspondant soit minimal.

Comme les erreurs sur le signal reçu sont faites en choisissant l'un des voisins du signal transmis (c'est l'erreur la plus probable statistiquement parlant), l'utilisation du codage de Gray permet d'obtenir une assignation pour laquelle tous les signaux diffèrent par exactement un bit de leurs plus proches voisins. La figure 5.4 présente un exemple d'assignation par codage de Gray d'une constellation 16-QAM.

Décrivons comment l'assignation de la figure 5.4 a été effectuée. Chaque symbole de quatre bits est constitué de deux bits en phase (les premiers dans le cas de la figure 5.4) et de deux bits en quadrature (les deux derniers dans l'exemple choisi). Il existe alors quatre coordonnées possibles sur chaque axe qui sont alors codées par codage de Gray, soit par exemple 00, 01, 11, 10. De manière plus générale, pour une constellation *M*-QAM, chaque symbole de canal est constitué de $\frac{m}{2}$ bits en phase et $\frac{m}{2}$ bits en quadrature. On effectue alors un codage de Gray

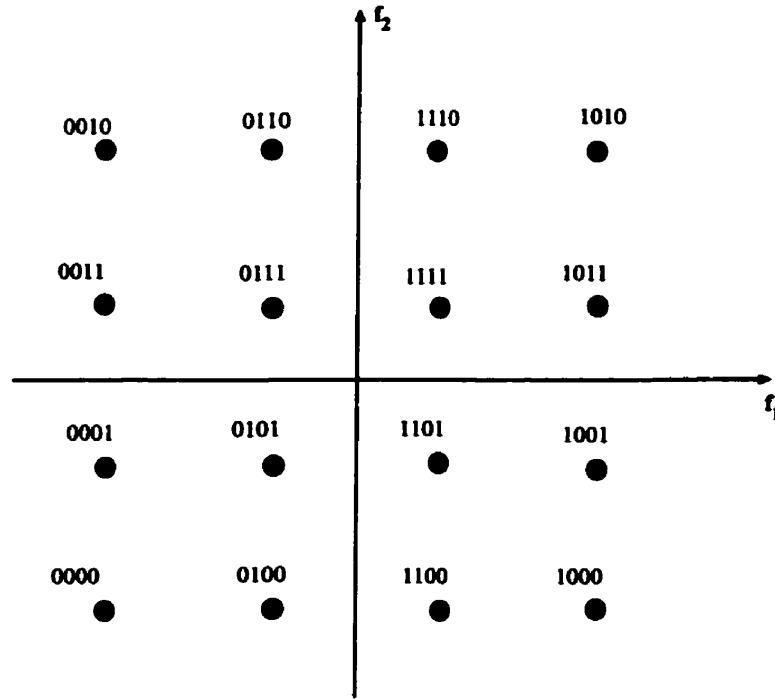


Figure 5.4: Exemple d'assignation par codage de Gray pour la constellation 16-QAM

des $\frac{m}{2}$ coordonnées sur chaque axe afin d'obtenir un couple de coordonnées pour chacun des $M = 2^m$ signaux.

Si l'on suppose alors que l'on a un bit en erreur par symbole, ce qui est statistiquement le plus probable, on peut écrire:

$$P_{e,b} = \frac{P_{e,s}}{m} \quad (5.20)$$

où $P_{e,b}$ et $P_{e,s}$ sont respectivement les probabilités d'erreur par bit et par symbole. Dans la pratique, on supposera que l'équation (5.20) est vérifiée.

On peut remarquer d'après la figure 5.4 que la distance de Hamming entre les plus proches voisins de chaque signal de la constellation (ceux situés le long de chaque axe) est toujours égal à un. La distance de Hamming entre deux signaux

est la différence entre les symboles de canal qui sont assignés à chacun de ces points. Ainsi, les signaux pour lesquels ont été assignés les symboles 0101 et 0111 ont une distance de Hamming égale à un. De même, ceux pour lesquels ont été assignés les symboles 0101 et 0011 ont une distance de Hamming égale à deux. Cette propriété est une caractéristique fondamentale de l'assignation par codage de Gray et assure la minimisation de la probabilité d'erreur par bit.

5.3 Performances du décodage à seuil itératif avec assignation par codage de Gray

Les codes doublement orthogonaux étudiés précédemment ont été perforés dans le but de gagner de l'efficacité spectrale par rapport au système traditionnel utilisant des codes de taux $R = \frac{1}{2}$. Toutefois, il y a moyen de diminuer encore la largeur de bande nécessaire à cette technique de codage en utilisant des modulations à fortes efficacités spectrales telles les modulations M -PSK ou M -QAM.

Dans ce but, nous allons étudier dans cette section les performances du décodage à seuil itératif des codes doublement orthogonaux perforés lorsque la modulation utilisée est la modulation M -QAM avec assignation par codage de Gray. Nous allons analyser les modifications du décodeur qui s'imposent et évaluer les résultats obtenus.

5.3.1 Mise en œuvre d'un calculateur de métrique

Le décodeur à seuil itératif tel que présenté aux chapitres précédents fonctionne à partir de quantifications douces (ou fiabilités) sur les bits d'information et les symboles de parité. Or lors de l'utilisation de modulation M -QAM, le démodulateur délivre quant à lui des fiabilités sur les *symboles* de canal qui représentent plusieurs bits. Pour pouvoir faire fonctionner le décodeur à seuil itératif même avec une modulation M -QAM, il faut mettre en place un mécanisme

permettant de calculer des fiabilités sur les bits à partir des fiabilités sur les symboles de canal. Pour ce faire, nous allons utiliser encore une fois la notion de logarithme de rapport de vraisemblance [20].

Soit \mathbf{y} le symbole de canal reçu, représenté par un point dans la constellation. Nous allons déterminer le vecteur binaire $[y_1 \dots y_m]$ correspondant au symbole reçu \mathbf{y} de la manière suivante: pour $r = 1, 2, \dots, m$, $y_r = 1$ si et seulement si:

$$L(y_r) = \ln \left(\frac{P(y_r = 1 | \mathbf{y})}{P(y_r = 0 | \mathbf{y})} \right) \geq 0 \quad (5.21)$$

Les $L(y_r), r = 1, 2, \dots, m$ seront les fiabilités sur les bits qui seront finalement transmises au décodeur.

Alors, d'après le théorème de Bayes:

$$\begin{aligned} L(y_r) &= \ln \left(\frac{P(y_r = 1 | \mathbf{y})}{P(y_r = 0 | \mathbf{y})} \right) \\ &= \ln \left(\frac{P(y_r = 1) P(\mathbf{y} | y_r = 1)}{P(y_r = 0) P(\mathbf{y} | y_r = 0)} \right) \end{aligned} \quad (5.22)$$

Soit $\mathcal{Q}_{0,r}$ (respectivement $\mathcal{Q}_{1,r}$) l'ensemble des indices q des signaux de la constellation M -QAM considérée pour lesquels $s_{q,r} = 0$ (respectivement $s_{q,r} = 1$), c'est-à-dire l'ensemble des indices des signaux représentant des symboles dont le bit d'ordre r est égal à 0 (respectivement égal à 1):

$$\mathcal{Q}_{0,r} = \{q = 1, 2, \dots, M | s_{q,r} = 0\}, \quad r = 1, 2, \dots, m \quad (5.23)$$

$$\mathcal{Q}_{1,r} = \{q = 1, 2, \dots, M | s_{q,r} = 1\}, \quad r = 1, 2, \dots, m \quad (5.24)$$

Bien sur, on a:

$$\mathcal{Q}_{0,r} \bigcup \mathcal{Q}_{1,r} = \{1, 2, \dots, J\} \quad (5.25)$$

Si l'on suppose que les bits constitutifs du symbole représenté par le signal reçu

\mathbf{y} sont équiprobables, on obtient:

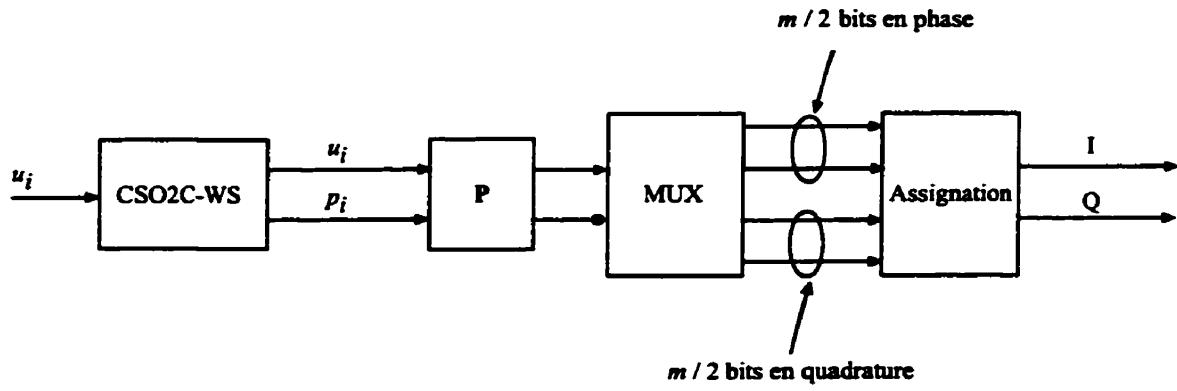
$$L(y_r) = \ln \left(\frac{P(\mathbf{y}|y_r = 1)}{P(\mathbf{y}|y_r = 0)} \right) \quad (5.26)$$

$$= \ln \left(\frac{\sum_{q \in Q_{1,r}} P(\mathbf{y}|\mathbf{s}_q)}{\sum_{q \in Q_{0,r}} P(\mathbf{y}|\mathbf{s}_q)} \right) \quad (5.27)$$

Si l'on considère un canal à bruit blanc additif gaussien (AWGN) dont la variance est égale à $\frac{N_0}{2}$, on aboutit à:

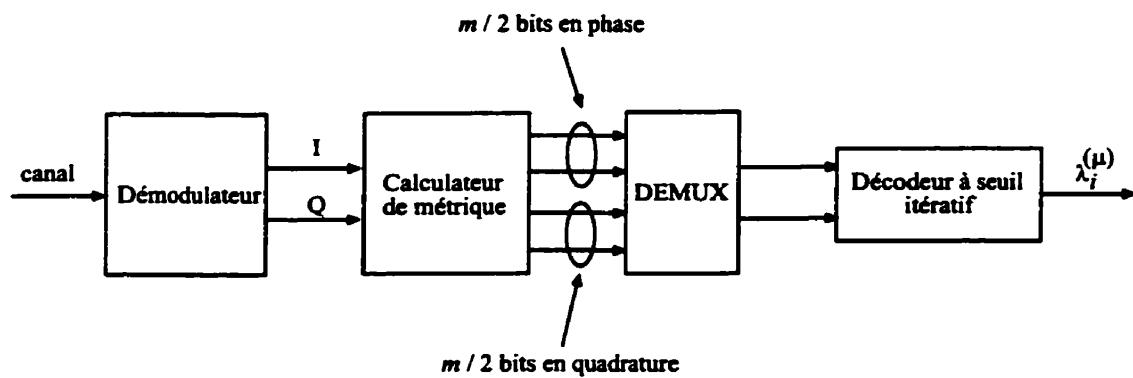
$$L(y_r) = \ln \left(\frac{\sum_{q \in Q_{1,r}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{s}_q\|^2}{N_0}\right)}{\sum_{q \in Q_{0,r}} \exp\left(-\frac{\|\mathbf{y}-\mathbf{s}_q\|^2}{N_0}\right)} \right) \quad (5.28)$$

Ces fiabilités ainsi calculées sont alors directement transmises au décodeur à seuil itératif standard pour le décodage des données reçues. Les figures 5.5 et 5.6 illustrent de manière schématique les architectures de l'émetteur et du récepteur. On pourrait songer à améliorer ce système à partir de la deuxième itération en effectuant une correction *a posteriori* des symboles de canal reçus grâce aux bits d'information qui ont déjà été décodés. Toutefois, nous ne traiterons pas de cette amélioration dans ce mémoire.



MUX: Multiplexeur

Figure 5.5: Schéma de principe d'un émetteur multiniveaux



DEMUX: Démultiplexeur

Figure 5.6: Schéma de principe d'un récepteur multiniveaux

5.3.2 Performances d'erreur et résultats

Une fois le calculateur de métrique décrit précédemment implémenté, on peut alors analyser les performances du décodage à seuil itératif utilisé avec une modulation M -QAM. Nous allons donc étudier l'influence de divers paramètres afin de retrouver divers résultats établis dans le cas de la modulation BPSK.

- **Influence du taux de codage R**

La figure 5.7 permet de voir l'influence du taux de codage après perforation lorsque la modulation 16-QAM est utilisée avec assignation par codage de Gray.

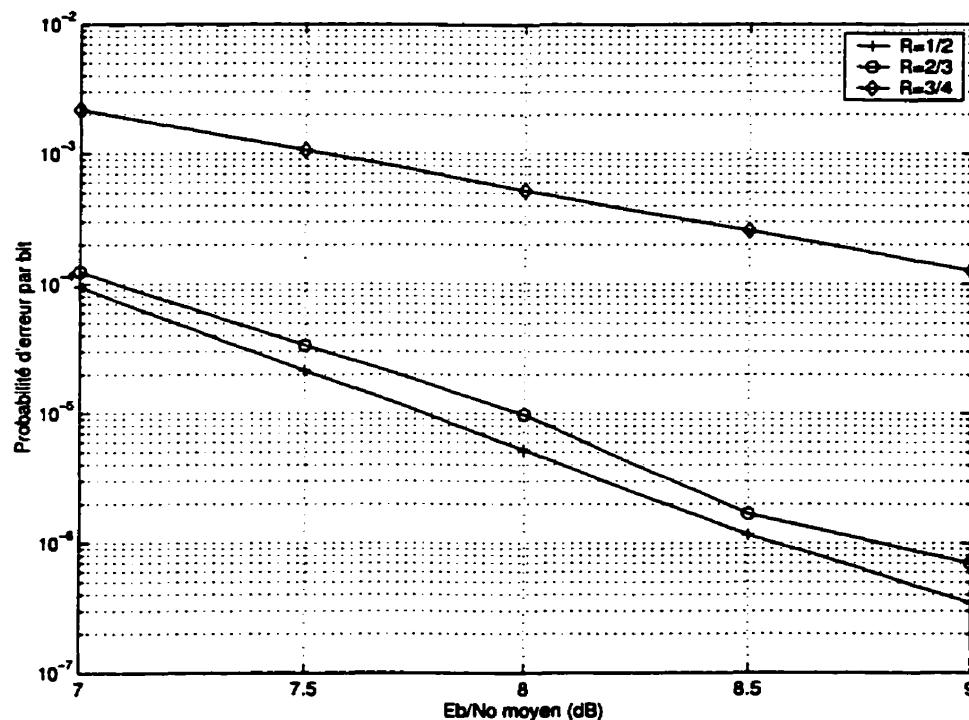


Figure 5.7: Performances à la quatrième itération du décodage des PCSO2C-WS de taux $R = \frac{2}{3}$ et $R = \frac{3}{4}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$, modulation 16-QAM

Comme prévu, on retrouve la même variation des performances en fonction du taux de codage, c'est-à-dire qu'un fort taux de codage dégrade fortement la qualité de l'information décodée. De plus, on peut noter un énorme écart entre les taux de codage $R = \frac{2}{3}$ et $R = \frac{3}{4}$ comparés à celui entre les taux de codage $R = \frac{1}{2}$ et $R = \frac{2}{3}$. Ceci vient du fait que le code choisi est optimal (du point de vue de la perforation) pour $R = \frac{2}{3}$ et très mauvais pour $R = \frac{3}{4}$, et donc il présente des performances catastrophiques pour ce dernier taux de perforation.

- **Influence de la position des connexions** $\gamma_j, j = 1, 2, \dots, J$

La figure 5.8 illustre la relation entre la position des connexions et les performances du code correspondant. On remarque que, comme dans le cas d'une modulation BPSK, les meilleurs codes sont ceux pour lesquels le spectre comporte une seule raie à $J_{\min}^{WS} = \frac{J}{k}$.

On peut donc supposer que l'utilisation de la modulation QAM n'introduit pas de modification notable quant au comportement des codes doublement orthogonaux perforés. Ainsi, toutes les conclusions auxquelles nous étions arrivés dans le cas de la modulation BPSK (et notamment le concept de spectre de perforation) s'appliquent.

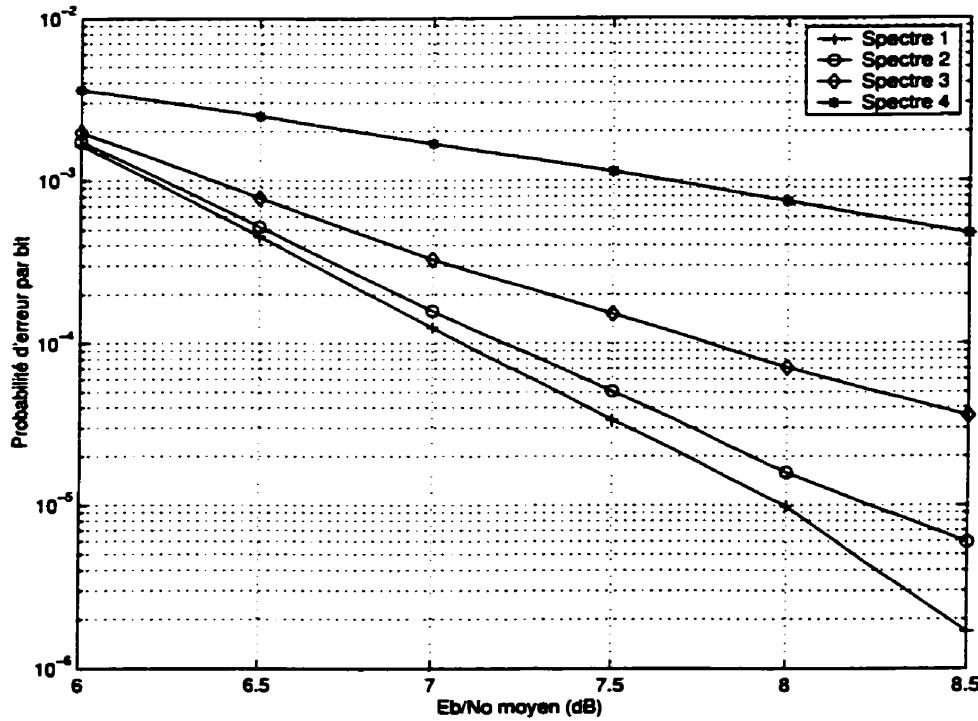


Figure 5.8: Performances à la quatrième itération du décodage de plusieurs PCSO2C-WS de taux $R = \frac{2}{3}$ ayant $J = 8$ connexions, modulation 16-QAM

• Influence de l'ordre de la modulation M

Nous allons regarder ici les performances du décodage à seuil itératif des codes doublement orthogonaux perforés lors de l'utilisation des modulations 16-QAM, 64-QAM et 256-QAM et les comparer aux résultats prévus théoriquement. La figure 5.9 présente les résultats obtenus par notre simulateur.

Considérons le cas d'une information non codée. Dans le cas de la modulation BPSK, la probabilité d'erreur est donnée par [27], où \mathcal{E}_b est l'énergie par bit:

$$P_{e,b} = P_{e,s} = Q\left(\sqrt{\frac{2\mathcal{E}_b}{N_0}}\right) \quad (5.29)$$

Comme mentionné à la section 5.1, dans le cas de constellations M -QAM en carré (ce sont les seules constellations envisagées ici), chaque signal de la constella-

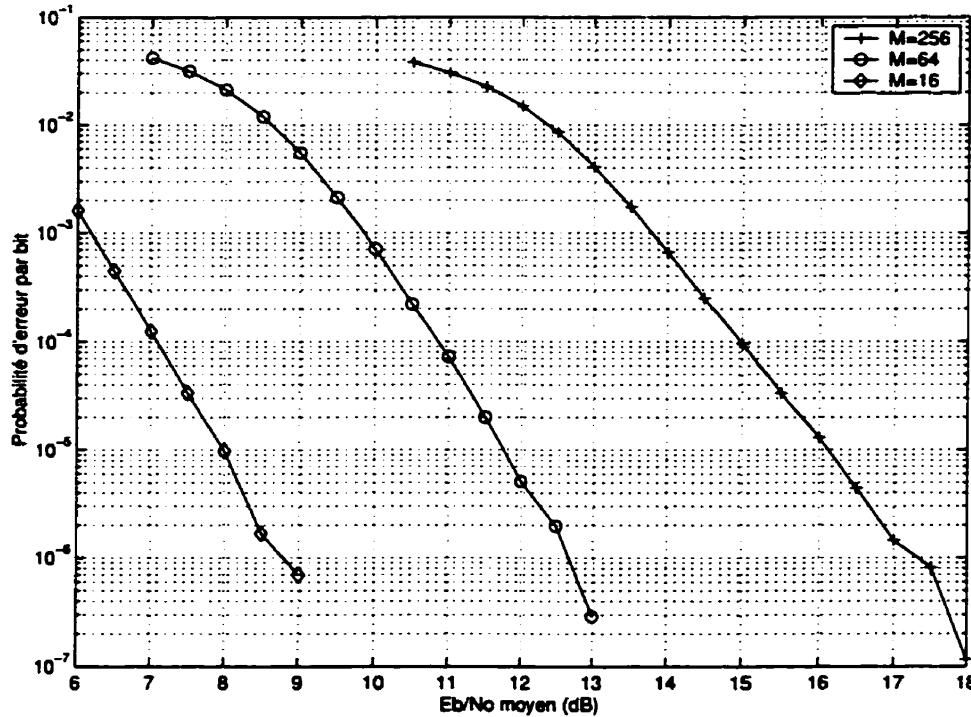


Figure 5.9: Performances à la quatrième itération du décodage du PCSO2C-WS de taux $R = \frac{2}{3}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$, modulation M -QAM

tion est équivalent à deux signaux d'une constellation \sqrt{M} -PAM dont les porteuses sont en quadrature. Du fait que les composantes en phase et en quadrature peuvent être parfaitement séparées par le démodulateur, la probabilité d'erreur d'une modulation M -QAM est facilement déterminée par la probabilité d'erreur de la modulation \sqrt{M} -PAM. Plus spécifiquement, la probabilité d'une décision correcte pour la modulation M -QAM est:

$$P_c = (1 - P_{\sqrt{M}})^2 \quad (5.30)$$

où $P_{\sqrt{M}}$ est la probabilité d'erreur d'une constellation \sqrt{M} -PAM avec la moitié de la puissance moyenne sur chaque composante des signaux de la constellation QAM équivalente (une composante en phase et une composante en quadrature). D'après [27], on a:

$$P_{\sqrt{M}} = 2 \left(1 - \frac{1}{\sqrt{M}} \right) Q \left(\sqrt{\frac{3}{M-1} \frac{\bar{\mathcal{E}}}{N_0}} \right) \quad (5.31)$$

où $\bar{\mathcal{E}}$ est l'énergie moyenne de la constellation et $\frac{\bar{\mathcal{E}}}{N_0}$ est le rapport signal à bruit moyen par symbole. Ainsi:

$$P_{e,s} = 1 - \left(1 - 2 \left(1 - \frac{1}{\sqrt{M}} \right) Q \left(\sqrt{\frac{3}{M-1} \frac{\bar{\mathcal{E}}}{N_0}} \right) \right)^2 \quad (5.32)$$

De plus, on a vu que l'utilisation du codage de Gray implique que $P_{e,b} = \frac{P_{e,s}}{m}$. Les probabilités d'erreurs résultantes ont été tracées dans [27] et on obtient approximativement les valeurs données dans le tableau 5.1.

Tableau 5.1: Rapport signal sur bruit moyen $\frac{\bar{\mathcal{E}}_b}{N_0}$ (en dB) pour diverses probabilités d'erreur dans le cas d'une information non codée transmise avec les modulations BPSK, 16-QAM et 64-QAM

$P_{e,b}$	$\frac{\bar{\mathcal{E}}_b}{N_0}$ (dB)		
	BPSK	16-QAM	64-QAM
10^{-4}	7	12,5	16,5
10^{-5}	8,5	13,5	18

À titre de comparaison le tableau 5.2 donne les rapports signal sur bruit moyens $\frac{\bar{\mathcal{E}}_b}{N_0}$ qui sont observés après simulation du décodage à seuil itératif.

On remarque d'après ces deux tableaux une différence constante (de l'ordre de 1,5 dB) entre les pertes de performances générées par l'utilisation des différentes constellations. On peut supposer que ceci est à mettre sur le compte du calculateur de métrique qui est susceptible de modifier les probabilités d'erreur par bit. Toutefois, et on peut le constater sur la figure 5.9, les écarts de performances restent sensiblement les mêmes quelle que soit la probabilité d'erreur considérée.

Tableau 5.2: Rapport signal sur bruit moyen $\frac{\epsilon_b}{N_0}$ (en dB) pour diverses probabilités d'erreur dans le cas d'une information codée (avec PCSO2C-WS optimal de taux $R = \frac{2}{3}$, $J = 8$, quatrième itération) transmise avec les modulations BPSK, 16-QAM et 64-QAM

$P_{e,b}$	$\frac{\epsilon_b}{N_0}$ (dB)		
	BPSK	16-QAM	64-QAM
10^{-4}	3,8	7,1	10,8
10^{-5}	4,5	8	11,5

5.3.3 Influence de la sous-optimalité du calculateur de métrique

Nous allons aborder dans cette section un aspect un peu moins théorique et qui concerne plus le côté implémentation d'une telle architecture, soit la rapidité du calculateur de métrique que nous avons proposé. En effet, l'un des atouts de la technique du décodage à seuil itératif par rapport à d'autres techniques de décodage itératif étant la rapidité, il est important d'optimiser la vitesse du calculateur de métrique.

On rappelle ici que le rôle de ce calculateur est de calculer la quantité suivante pour tous les bits de chaque symbole:

$$L(y_r) = \ln \left(\frac{\sum_{q \in Q_{1,r}} \exp\left(-\frac{\|y - s_q\|^2}{N_0}\right)}{\sum_{q \in Q_{0,r}} \exp\left(-\frac{\|y - s_q\|^2}{N_0}\right)} \right) \quad (5.33)$$

Cette quantité implique un grand nombre d'additions et de multiplications/divisions ainsi que de nombreux appels à la fonction exponentielle, et ce d'autant plus que la taille de la constellation est importante. Nous allons proposer ici un moyen de diminuer la complexité du calculateur de métrique en simplifiant l'opération effectuée pour chaque bit.

En effet, l'expression (5.33) peut s'écrire en ne prenant en compte que l'exponentielle dont l'argument est maximal:

$$L(y_r) \approx \ln \left(\frac{\max_{q \in Q_{1,r}} \exp(-\frac{\|y - s_q\|^2}{N_0})}{\max_{q \in Q_{0,r}} \exp(-\frac{\|y - s_q\|^2}{N_0})} \right) \quad (5.34)$$

$$\approx \ln \left(\frac{\exp(-\min_{q \in Q_{1,r}} \frac{\|y - s_q\|^2}{N_0})}{\exp(-\min_{q \in Q_{0,r}} \frac{\|y - s_q\|^2}{N_0})} \right) \quad (5.35)$$

soit encore:

$$L(y_r) \approx \min_{q \in Q_{0,r}} \left(\frac{\|y - s_q\|^2}{N_0} \right) - \min_{q \in Q_{1,r}} \left(\frac{\|y - s_q\|^2}{N_0} \right) \quad (5.36)$$

Par cette approximation, le calculateur de métrique se trouve grandement simplifié. Pour fournir une décision dure sur les y_r , il suffit juste de trouver le signal le plus proche du signal reçu, de voir si ce signal correspond à un symbole dont le bit à la position r est égal à 1 ou 0 et d'en déduire la valeur de y_r . Dans le cas d'une décision douce, la fiabilité de ce bit est alors extrêmement simple du fait qu'elle ne fait intervenir qu'une différence de distances normalisées.

La figure 5.10 présente un exemple de performances de notre système sous-optimal par rapport au système optimal considéré jusqu'à présent.

On remarque un très faible écart des performances (inférieur à 0.1 dB) entre les deux calculateurs. Le tableau 5.3 présente les différents temps de simulation qui ont été compilés pour différentes constellations.

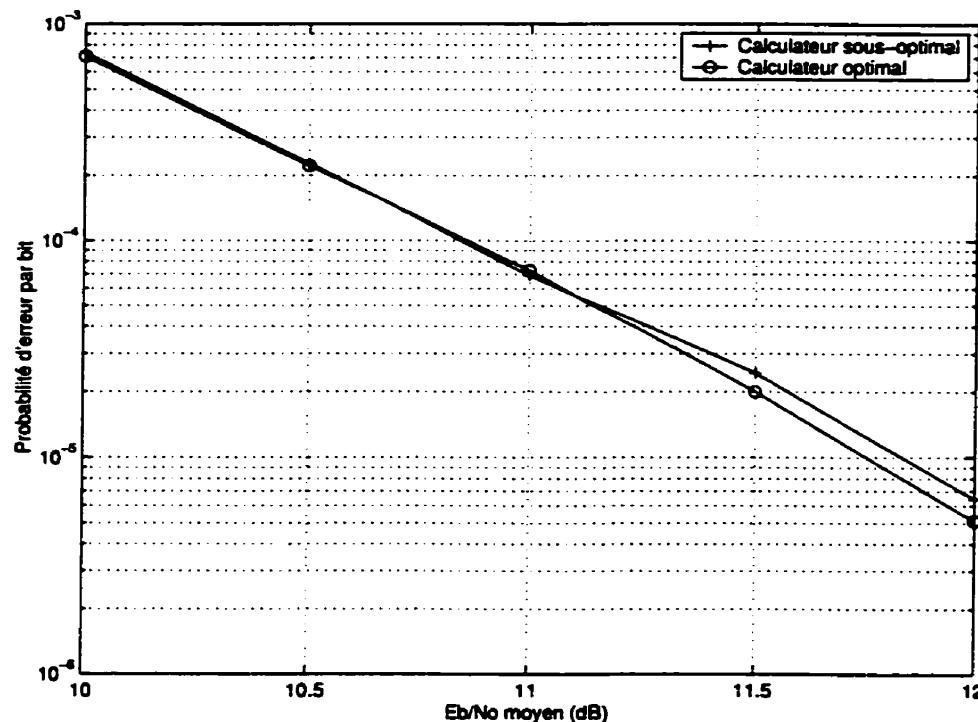


Figure 5.10: Performances à la quatrième itération du calculateur de métrique sous-optimal en utilisant le PCSO2C-WS de taux $R = \frac{2}{3}$ ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$, modulation 64-QAM

Tableau 5.3: Temps de simulation en secondes pour les deux calculateurs de métriques dans le cas de la figure 5.10

Modulation	Temps de simulation		Temps économisé
	calc. optimal	calc. sous-optimal	
16-QAM	5933	5521	7 %
64-QAM	11558	9716	16 %
256-QAM	33434	25924	22 %

On remarque d'après le tableau 5.3 que l'on peut gagner un temps important en utilisant le calculateur de métrique simplifié, et ce d'autant plus que la constellation possède un nombre de signaux important. En effet, dans le cas de la modulation 16-QAM on gagne 7% du temps de simulation alors que dans le cas

de la modulation 256-QAM, on peut gagner jusqu'à 22 % du temps de simulation. Ceci s'explique par le fait que dans le cas de grandes constellations, le calculateur simplifié permet d'économiser beaucoup plus de calculs que dans le cas d'une petite constellation, notamment l'appel à la fonction exponentielle.

Ainsi, étant donné le très faible niveau de pertes engendrées par l'utilisation du calculateur de métrique simplifié et le gain de temps substantiel que l'on peut en retirer, il peut être très avantageux d'utiliser cette simplification.

5.3.4 Entrelacement des bits au sein des trames

Nous allons ici préciser une propriété des codes doublement orthogonaux qui permet d'avoir de bons résultats tout en utilisant des modulations à fortes efficacités spectrales, alors que l'emploi de codes simplement orthogonaux et du décodeur à seuil classique aurait nécessité des modifications à la structure de notre système.

Lorsque nous avons développé les calculs qui nous ont permis d'aboutir à l'expression de λ_i donnée par (3.34), nous avons supposé que les équations $\{B_{i,j}, j = 1, 2, \dots, J\}$ étaient indépendantes afin de pouvoir appliquer le théorème de Bayes et ainsi en déduire l'équation fondamentale donnée par (3.27). Nous allons démontrer que dans le cas des modulations multiniveaux cette affirmation n'est plus valable.

Nous rappelons en (5.37) l'expression des $\{B_{i,j}\}$ obtenue en (3.22), où \tilde{u}_i est la décision dure sur les symboles de canal BPSK reçus et \hat{e}_i^u est l'estimation dure de l'erreur sur u_i .

$$\begin{aligned} B_{i,0} &= \tilde{u}_i = u_i \oplus e_i^u & (5.37) \\ B_{i,j} &= u_i \oplus \sum_{k=1}^{j-1} e_{i+\gamma_j-\gamma_k}^u \oplus e_{i+\gamma_j}^p \oplus \sum_{k=j+1}^J (e_{i+\gamma_j-\gamma_k}^u \oplus \hat{e}_{i+\gamma_j-\gamma_k}^u), \quad j = 1, 2, \dots, J \end{aligned}$$

On remarque que chaque équation $B_{i,j}$ utilise un certain nombre de symboles d'erreur affectant des bits d'information et des symboles de parité. Dans le cas d'une modulation BPSK et d'un canal à bruit blanc, chaque symbole de bruit est indépendant des autres étant donné qu'à chaque bit transmis est associé un symbole de bruit, et les symboles de bruit affectant des symboles de canal successifs sont statistiquement indépendants.

Toutefois, dans le cas des modulations multiniveaux, un symbole de canal représente plusieurs symboles codés (à la fois d'information et de parité). La propriété d'indépendance des symboles de bruit affectant des symboles de canal successifs est toujours valide mais cette fois les symboles codés faisant partie du même symbole de canal sont affectés du *même* bruit. Ainsi, après calcul des métriques associées aux différents bits d'information, les symboles de bruit affectant des bits successifs ne seront plus indépendants et par conséquent les équations $B_{i,j}$ ne le seront plus également. Ainsi, la valeur MAP approximative λ_i calculée par le décodeur à seuil se trouve être encore plus éloignée de sa vraie valeur $L(u_i | \{B_{i,j}\})$.

L'emploi d'un entrelaceur semble être une solution pour résoudre ce problème d'indépendance. En effet, celui-ci pourrait séparer les bits intervenant dans le même ensemble d'équations $B_{i,j}$ dans des symboles de canal différents et ainsi conserver l'indépendance des symboles de bruit appartenant au même ensemble d'équations de parité. Toutefois, nous allons montrer que l'utilisation d'un entrelaceur est totalement inutile si des codes doublement orthogonaux sont utilisés.

Prenons l'exemple du PCSO2C-WS ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$ de taux $R = \frac{2}{3}$ et regardons quels symboles d'information sont utilisés par chaque équation du décodeur à seuil. Les résultats sont donnés au tableau 5.4.

On remarque que les positions des bits d'information utilisés au sein des équations sont très espacées les unes des autres. Ceci vient du fait que le décodeur à seuil utilise les informations placées aux différences simples et comme les codes

Tableau 5.4: Position des bits d'information utilisés à chaque itération au temps $i = 0$ par le décodeur à seuil correspondant au CSO2C-WS ayant $J = 8$ connexions aux positions $\{0, 43, 139, 322, 422, 430, 441, 459\}$

Équation	Position des bits d'information utilisés au temps $i = 0$
1	$\{\}$
2	$\{43\}$
3	$\{139, 96\}$
4	$\{322, 279, 183\}$
5	$\{422, 379, 283, 100\}$
6	$\{430, 387, 291, 108, 8\}$
7	$\{441, 398, 302, 119, 19, 11\}$
8	$\{459, 416, 320, 137, 37, 29, 18\}$

doublement orthogonaux vérifient la propriété d'orthogonalité à un rang supérieur, les différences simples sont nécessairement très espacées les unes des autres [1]. Il existe une très faible probabilité pour que les équations soient corrélées étant donné qu'un symbole de canal ne représentera presque jamais plusieurs bits entrant dans le calcul des équations $B_{i,j}$. Cela revient à dire que la double orthogonalité du code joue le rôle de l'entrelaceur de bits. Cette propriété peut s'avérer particulièrement intéressante dans le cas d'une utilisation sur des canaux à évanouissements.

5.4 Assignation par partition d'ensemble

Nous avons parlé dans la section précédente des performances du décodage à seuil itératif lors de l'utilisation de modulations multiniveaux avec assignation par codage de Gray. Cette technique permet d'augmenter encore davantage l'efficacité spectrale du système de télécommunications utilisé. Nous allons présenter dans cette section les propriétés d'un autre type d'assignation, soit l'assignation par partition d'ensemble et nous présenterons comment elle pourrait être utilisée avantageusement avec des PCSO2C-WS dont le spectre comporte plusieurs raies spectrales.

L'assignation par partition d'ensemble est apparue lors de l'invention de la modulation codée (en anglais TCM pour *Trellis Coded Modulation*) [36]. Cette technique, dont le but est de réduire l'expansion de largeur de bande liée au codage de canal, propose d'associer de manière harmonieuse la modulation et le codage de manière à compenser la redondance introduite par le code (i.e. l'expansion de largeur de bande) en augmentant le nombre de signaux de la constellation utilisée.

Si la modulation est traitée séparément du codage, l'augmentation de l'efficacité spectrale liée à l'emploi de modulations telles que M -PSK ou M -QAM ne peut être compensée que par l'utilisation de codes très puissants (c'est-à-dire de codes possédant une grande longueur de contrainte) pour avoir un gain de codage satisfaisant. Au contraire, si la modulation est intégrée au codage et si l'assignation des symboles codés aux signaux de la constellation est conçue en fonction du code utilisé de façon à augmenter la distance euclidienne minimale entre les paires de signaux, la perte induite par l'augmentation du nombre de signaux au sein de la constellation est facilement compensée par un gain de codage important tout en utilisant des codes présentant une faible complexité au décodage.

La clé de la modulation codée réside donc dans la construction d'une assignation adéquate qui permet de maximiser la distance euclidienne entre certains signaux de la constellation. Ce type d'assignation a été développé par Ungerboek en 1982 [36] et a été baptisé *assignation par partition d'ensemble* (*mapping by set partitioning* en anglais). Nous allons présenter deux exemples pour décrire le principe de cette assignation.

Nous avons représenté à la figure 5.11 une partition de la constellation 8-PSK en sous-ensembles pour lesquels la distance euclidienne minimale entre les signaux augmente d'une partition à la suivante. Dans la constellation 8-PSK originelle, les signaux sont situés sur un cercle de rayon $\sqrt{\mathcal{E}}$ et la distance minimale qui sépare les signaux est égale à:

$$\begin{aligned}
 d_0 &= 2\sqrt{\mathcal{E}} \sin\left(\frac{\pi}{8}\right) \\
 &= \sqrt{(2 - \sqrt{2})\mathcal{E}} \\
 &= 0,765\sqrt{\mathcal{E}}
 \end{aligned} \tag{5.38}$$

Au cours de la première partition, l'ensemble de huit signaux est subdivisé en deux sous-ensembles de quatre signaux chacun, de telle manière que la distance minimale entre les signaux augmente à:

$$d_1 = \sqrt{2\mathcal{E}} \tag{5.39}$$

Dans le second niveau de partition, chacun des deux sous-ensembles est divisé en deux sous-ensembles de deux signaux, de telle manière que la distance minimale entre les signaux augmente à:

$$d_2 = 2\sqrt{\mathcal{E}} \tag{5.40}$$

Ce processus aboutit à quatre sous-ensembles de deux signaux chacun. Finalement, le dernier niveau de partition génère huit sous-ensembles contenant chacun un seul signal. On remarque donc que chaque niveau de partition augmente la distance euclidienne minimale entre les signaux. Le résultat de ces trois partitions est représenté à la figure 5.11.

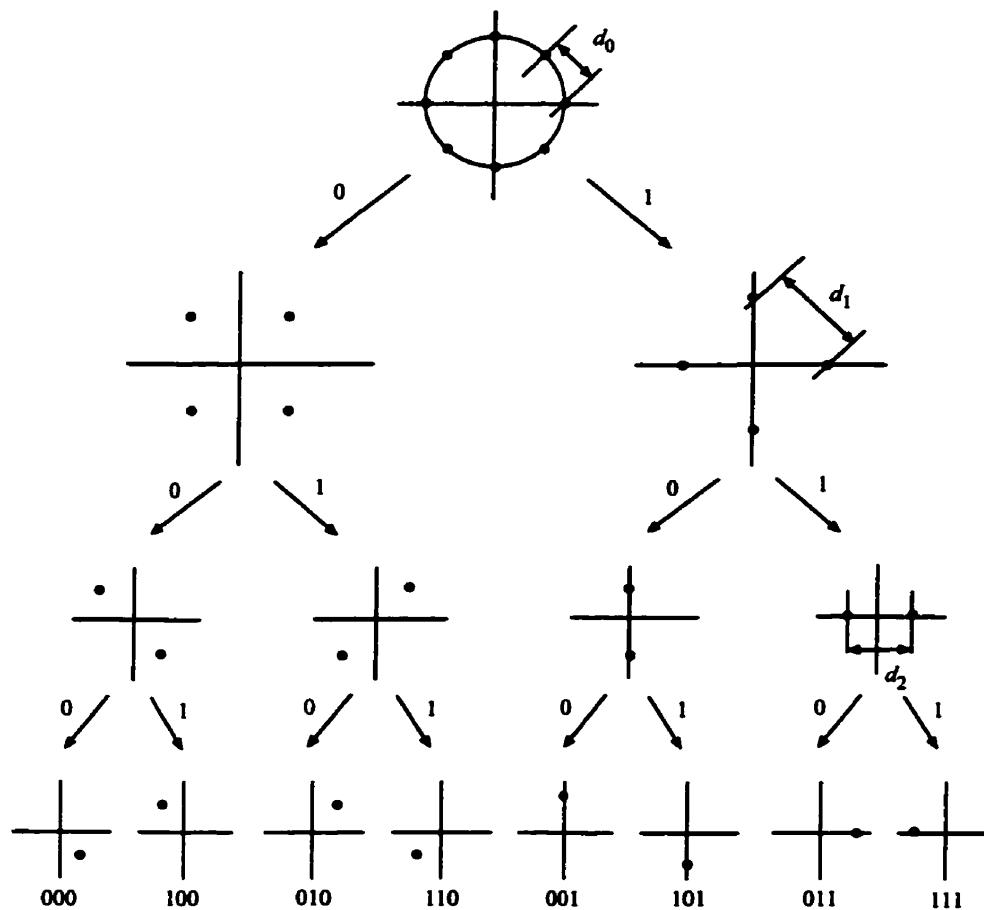


Figure 5.11: Exemple d'assignation par partition d'ensemble pour la constellation 8-PSK

Examinons à présent l'assignation par partition d'ensemble de la constellation 16-QAM. Comme l'illustre la figure 5.12, la constellation de seize signaux est d'abord divisée en deux sous-ensembles en assignant des signaux successifs à chacun des sous-ensembles. Ainsi, la distance entre les signaux passe de $2\sqrt{\mathcal{E}}$ à $2\sqrt{2\mathcal{E}}$ par la première partition. En poursuivant la partition de la constellation, on augmente progressivement la distance euclidienne entre les signaux. Il est intéressant de noter que pour la constellation 16-QAM, chaque niveau de partition augmente la distance euclidienne minimale de $\sqrt{2}$, soit:

$$\frac{d_{i+1}}{d_i} = \sqrt{2}, \quad i = 0, 1, \dots, \log_2 M \quad (5.41)$$

Maintenant que l'on a vu à travers ces deux exemples comment l'assignation peut être effectuée en effectuant une partition de la constellation considérée, reste à savoir comment tirer parti de cette technique dans le cas qui nous concerne.

L'avantage majeur de ce type d'assignation est qu'il permet une protection accrue de certains bits par rapport à d'autres. En effet, si l'on considère par exemple le cas de l'assignation par partition d'ensemble de la constellation 16-QAM donnée à la figure 5.12, on remarque que la détection du symbole 0110 à la réception sachant que le symbole 1110 a été transmis est très peu probable car les signaux correspondants sont très éloignés. Au contraire, la détection de la séquence 0110 sachant que la séquence transmise est 0111 est beaucoup plus probable sachant que les deux signaux correspondants sont voisins. Ainsi, on peut compenser une capacité de correction inégale des bits transmis au niveau du décodage par une protection elle aussi inégale au sein de la constellation: les bits faiblement protégés par le codage de canal se verront assigner des signaux pour lesquels la détection des signaux correspondant à des symboles de canal pour lesquels ces mêmes bits sont erronés est très peu probable et inversement. C'est cette assignation qui est à la base de la technique TCM.

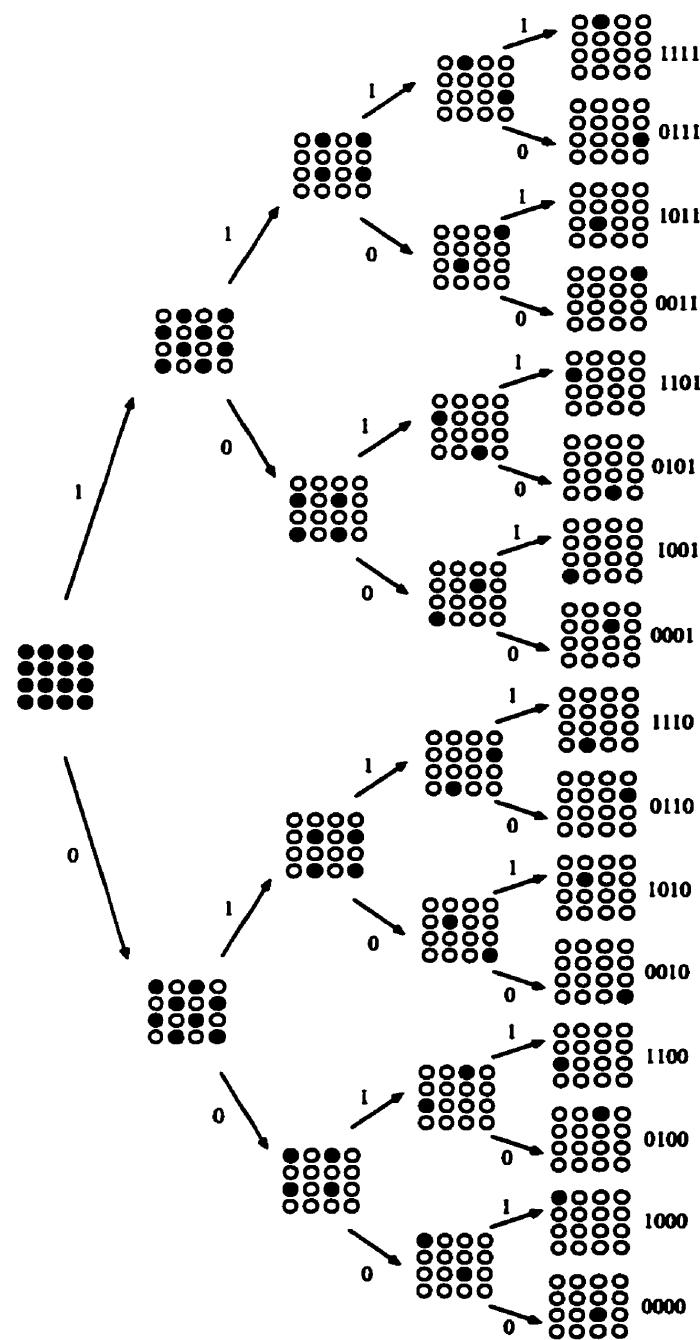


Figure 5.12: Exemple d'assignation par partition d'ensemble pour la constellation 16-QAM

5.5 Utilisation de l'assignation par partition d'ensemble dans le cas de PCSO2C non optimaux

5.5.1 Protection inégale des symboles d'information

Nous avons présenté à la section précédente comment on pouvait combiner les modulations à fortes efficacités spectrales comme la modulation M -QAM avec les codes doublement orthogonaux perforés pour aboutir à un système de communications optimisé en largeur de bande. La technique présentée utilise une assignation des signaux par codage de Gray et un calculateur de métrique.

Toutefois, on a montré que pour tous les types de constellations envisagés jusqu'à présent, les meilleurs codes doublement orthogonaux perforés que l'on avait à notre disposition n'avaient pas nécessairement une faible longueur de contrainte. Ainsi, une utilisation efficace de codes doublement orthogonaux perforés mais non optimaux pourrait conduire à une meilleure gestion du délai au codage et au décodage.

C'est cette idée qui nous a poussé à présenter l'assignation par partition d'ensemble. Nous allons montrer comment *a priori* on pourrait utiliser avantageusement cette technique pour compenser la dissymétrie des performances du décodage des symboles d'information émis.

Considérons pour cela un PCSO2C-WS dont le spectre de perforation à un taux R fixé comporte plusieurs raies. Ce code n'est donc pas optimal au niveau de la perforation. Certains symboles d'information seront donc décodés avec un grand nombre d'équations de parité et d'autres avec moins d'équations de parité. L'idée est donc de faire correspondre les symboles qui sont décodés avec peu d'équations de parité avec ceux qui sont fortement protégés de par la position des signaux correspondants au sein de la constellation.

Plus précisément, on rappelle que la valeur MAP approximative λ_i délivrée par

le décodeur à seuil est donnée par:

$$\lambda_i = y_i^u + L_e(\hat{u}_i) \quad (5.42)$$

Dans le cas des bits d'information u_i qui sont décodés avec un faible nombre d'équations de parité, l'information extrinsèque $L_e(\hat{u}_i) = \sum_{j=1}^J \psi_{i,j}$ ne sera pas très fiable. On va donc faire en sorte que l'information délivrée par le canal y_i^u soit elle assez fiable pour compenser la faiblesse de l'information extrinsèque.

En pratique, il suffit d'implémenter la table d'assignation par partition d'ensemble de la constellation choisie et de réordonner les bits à l'intérieur du symbole de canal de telle manière qu'ils soient classés selon le nombre d'équations de décodage avec lesquels ils vont être décodés du fait que la protection au sein de la constellation dépend de la position du bit au sein du symbole comme on peut le voir à la figure 5.12.

5.5.2 Étude des performances d'erreur

Dans cette section nous allons présenter les performances de ce type d'assignation et les comparer à ceux de l'assignation par codage de Gray. Certains résultats sont présentés à la figure 5.13. Ils correspondent aux performances du code dont le spectre est le deuxième de ceux représentés à la figure 4.10.

On remarque donc que les performances de ce type d'assignation sont très décevantes étant donné qu'aucune amélioration n'est apportée par rapport à l'assignation par codage de Gray et que pire encore, les performances sont dégradées (perte de 2 dB environ).

L'explication que l'on peut fournir est la suivante. L'utilisation de l'assignation par partition d'ensemble permet de protéger certains bits d'information par rapport à d'autres lors de la transmission. Toutefois, cette différence de protection se fait au détriment de ce qui fait la force de l'assignation par codage de Gray, soit

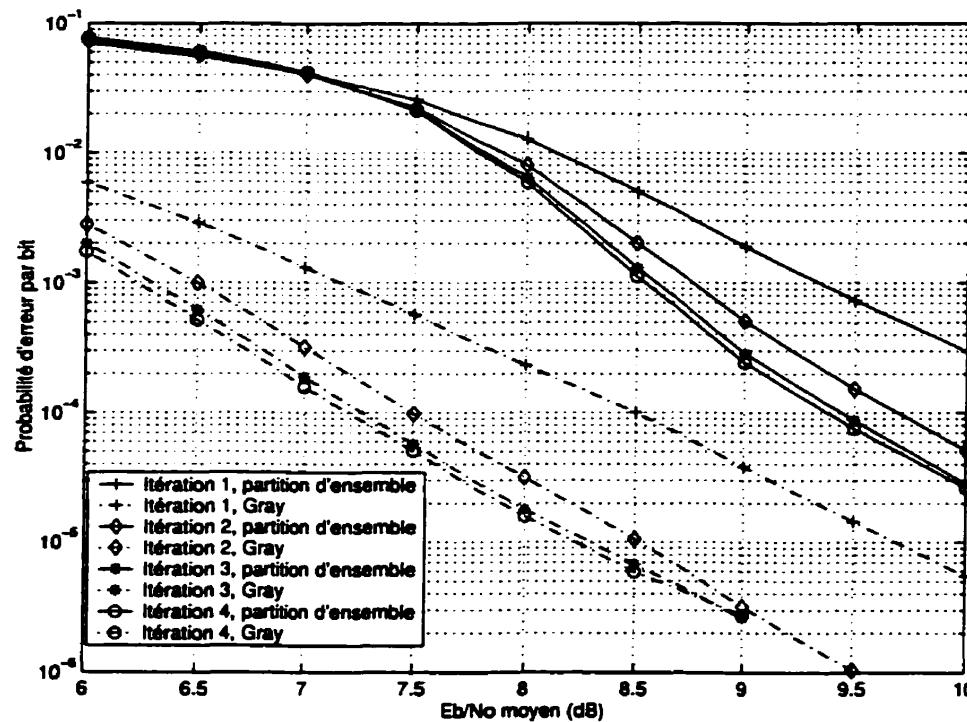


Figure 5.13: Performances du décodage du PCSO2C-WS de taux $R = \frac{2}{3}$ ayant $J = 8$ connexions aux positions $\{0, 9, 21, 395, 584, 767, 871, 912\}$, modulation 16-QAM

que les plus proches voisins de chaque signal ne diffèrent de lui que par un seul bit. Or le décodeur à seuil (et *a fortiori* le décodeur à seuil itératif) est un décodeur de bits et non un décodeur de symboles. Ainsi, l'avantage de protéger certains bits au sein de la constellation ne compense pas l'inconvénient que représente le fait d'avoir les symboles correspondant à ses plus proches voisins qui diffèrent de plus d'un bit.

Par conséquent, il est probable que les performances de l'assignation par partition d'ensemble aient été meilleures et plus proches des performances atteintes par codage de Gray si l'on avait considéré la probabilité d'erreur par symbole et non par bit. Toutefois, la façon dont nous avons conçu notre modulateur ne permet pas de tracer ce type de probabilité d'erreur.

Nous pouvons donc conclure que l'assignation par codage de Gray est la meilleure envisageable pour le décodeur à seuil itératif par rapport à d'autres types de modulations telles la modulation par partition d'ensemble.

5.6 Bilan

Nous avons montré dans ce chapitre comment nous pouvions adapter le décodeur à seuil à des techniques de modulations multiniveaux par l'ajout d'un calculateur de métriques sur les bits à partir de métrique sur les symboles.

Le système ainsi conçu permet donc l'obtention d'efficacités spectrales élevées, jusqu'à au moins 4 bits/s/Hz (dans le cas d'un PCSO2C-WS de taux $R = \frac{2}{3}$ utilisé avec une modulation 64-QAM).

Toutefois, ces bons résultats ne doivent pas occulter le fait que l'on n'a pas pu mettre au point une technique d'assignation permettant de compenser les faibles performances des PCSO2C présentant plusieurs raies spectrales, et donc tenter d'optimiser le délai au niveau du codage et du décodage.

CHAPITRE 6

CONCLUSION

6.1 Bilan de la recherche

Ce mémoire, qui s'inscrit dans la continuité des travaux précédemment entrepris, a permis de mieux cerner certaines propriétés du décodage à seuil itératif.

En premier lieu, nous avons présenté une étude théorique du comportement du décodeur à seuil itératif face à la perforation des codes doublement orthogonaux utilisés, qu'ils soient définis au sens large ou au sens strict. Cette étude, menée à l'aide d'une nouvelle représentation spectrale, a permis de dégager des critères d'optimalité importants concernant les performances des codes doublement orthogonaux au sens large perforés. Les performances d'erreur optimales obtenues sont certes inférieures à celles présentées par l'algorithme de décodage itératif turbo à faible rapport signal sur bruit mais deviennent très compétitives de manière asymptotique, et ce tout en ayant une complexité très réduite.

Par la suite, dans une deuxième partie, nous avons étudié de quelle manière ces codes doublement orthogonaux de forts taux pouvaient être combinés avec des modulations multiniveaux dans le but d'obtenir un système à forte efficacité spectrale. La mise en œuvre d'un calculateur de métrique a été effectuée permettant l'utilisation du décodeur à seuil itératif avec des modulations multiniveaux. L'étude des performances d'erreur lorsqu'une technique d'assignation par codage de Gray est utilisée a été entreprise et permet d'obtenir des performances conformes à nos attentes.

Ainsi, ce mémoire permet de faire les compromis nécessaires entre les performances d'erreur que l'on recherche et l'efficacité spectrale que l'on désire atteindre. Par conséquent, il permet d'atteindre les objectifs que l'on s'était fixé initialement. Toutefois, il existe des pistes pour améliorer les performances de cette

technique de codage et nous allons les présenter brièvement dans la section suivante.

6.2 Améliorations envisageables

L'évolution majeure que l'on puisse apporter à cette étude est l'amélioration des performances des codes à faible longueur de contrainte. Ceci peut se faire principalement suivant deux axes:

- **Recherche de codes optimaux**

Une recherche du type de celle entreprise dans [1] pourrait à nouveau être effectuée dans le but de trouver des codes doublement orthogonaux définis au sens large et au sens strict dont le spectre de perforation ne comporte qu'une seule raie et qui possèdent une longueur de contrainte réduite au maximum. En effet, l'un des avantages de la technique de décodage à seuil itératif étant sa rapidité, il serait dommage qu'elle soit compromise lors de l'utilisation de codes perforés.

- **Amélioration des performances des codes non optimaux**

La recherche de codes dont le spectre ne comporte qu'une seule raie et dont la longueur de contrainte est réduite peut s'avérer difficile à la fois d'un point de vue théorique et pratique. Il pourrait donc être intéressant d'améliorer les performances des codes pour lesquels le spectre de perforation comporte plusieurs raies. Ceci pourrait se faire de deux manières.

Tout d'abord, nous avons abordé dans ce mémoire la problématique de l'introduction de coefficients de pondération au sein des équations constitutives du décodeur à seuil. Par cette méthode, prolongeant celle effectuée dans [11], on pourrait rééquilibrer la fiabilité des informations d'une itération à l'autre et ainsi présenter de meilleures performances d'erreur.

En second lieu, l'utilisation d'une technique d'assignation fondée sur le codage de Gray et permettant la compensation d'un faible nombre d'équations de décodage par une protection accrue au sein de la constellation pourrait permettre d'améliorer sensiblement les performances des codes non optimaux lorsque des modulations multiniveaux sont utilisées.

6.3 Ouverture

Il pourrait être fructueux, dans le cadre de recherches futures, d'utiliser d'autres schémas permettant l'obtention de codes de taux élevés, par exemple en utilisant une structure pragmatique. Toutefois, les codes perforés ne peuvent pas être utilisés dans un tel schéma. Il faudrait plutôt mettre en œuvre des codes convolutionnels doublement orthogonaux de taux élevés comme par exemple des codes récursifs de forts taux de codage [11].

Dans un contexte plus général, il pourrait être très intéressant d'étudier les performances du décodeur à seuil itératif dans un contexte simulant plus fidèlement le comportement des canaux réels, tels les canaux de Rayleigh. Ceci permettrait de développer un prototype du décodeur et de se rendre compte de son fonctionnement dans des conditions réelles d'utilisation.

RÉFÉRENCES

- [1] Baechler, B. (2000), "Analyse et détermination de codes doublement orthogonaux pour décodage à seuil itératif", *Mémoire de Maîtrise de l'École Polytechnique de Montréal*
- [2] Baechler, B., Haccoun, D. et Gagnon, F. (2000), "On the Search for Self Doubly Orthogonal Codes", *Proceedings of ISIT'00*, p.292
- [3] Bahl, L., Cocke, J., Jelinek, F. et Raviv, J. (1979), "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284-287
- [4] Bégin, G. et Haccoun, H. (1989), "High-Rate Punctured Convolutional Codes: Structure Properties and Construction Technique", *IEEE Trans. on Comm.*, vol. 37, pp. 1381-1385
- [5] Bégin, G., Haccoun, H. et Paquin, C. (1990), "Further Results on High-Rate Punctured Convolutional Codes for Viterbi and Sequential Decoding", *IEEE Trans. on Comm.*, vol. 38, pp. 1922-1927
- [6] Benedetto, S., Montorsi, G., Divsalar, D. et Pollara, F. (1996), "Soft-Output Decoding Algorithms in Iterative Decoding of Turbo Codes", *JPL TDA Progress Report*, vol. 42-124, pp. 63-87
- [7] Berrou, C., Glavieux, A. et Thitimajshima P. (1993), "Near Shannon Limit Error Correcting Coding and Decoding: Turbo Codes", *Proceedings of ICC'93*, pp. 1064-1070
- [8] Bose, R. C. et Ray-Chaudhuri, D. K. (1960), "On a Class of Error Correcting Binary Group Codes", *Inform. Control*, vol. 3, pp. 68-79
- [9] Bose, R. C. et Ray-Chaudhuri, D. K. (1960), "Further Results in Error Correcting Binary Group Codes ", *Inform. Control*, vol. 3, pp. 279-290

- [10] Cain, J., Clark, G. J. et Geist J. (1979), "Punctured Convolutional Codes of Rate $\frac{n-1}{n}$ and Simplified Maximum Likelihood Decoding", *IEEE Trans. Inform. Theory*, vol. IT-25, pp.97-100
- [11] Cardinal, C. (2001), "Décodage à seuil itératif sans entrelacement des codes convolutionnels doublement orthogonaux", *Thèse de Doctorat de l'École Polytechnique de Montréal*
- [12] Cardinal, C., Haccoun, D., Gagnon, F. et Batani, N. (1999), "Turbo Decoding Using Convolutional Self Doubly Orthogonal Codes", *Proceedings of ICC'99*, pp. 113-117
- [13] Elias, P. (1954), "Error-free Coding", *IRE Trans. Inform. Theory*, vol. IT-4, pp.29-37
- [14] Fano, R. M. (1963), "A Heuristic Discussion of Probabilistic Coding", *IEEE Trans. Inform. Theory*, vol. IT-9, pp. 64-74
- [15] Forney, G. D. Jr (1966), *Concatenated Codes*, MIT Press, Cambridge, MA
- [16] Gagnon, F., Haccoun, D., Batani, N. et Cardinal, C. (2000), "Apparatus for Convolutional Self Doubly Orthogonal Codes for Encoding and Decoding", *US Patent No 6167552*
- [17] Hagenauer, J. et Hoeher, P. (1989), "A Viterbi Algorithm with Soft Decision Outputs and its Applications", *Proceedings of Globecom'89*, pp. 1680-1686
- [18] Hamming, R. W. (1950), "Error Detecting and Error Correcting Codes", *Bell Syst. Tech. J.*, vol. 29, pp. 147-160
- [19] Jelinek, F. (1969), "Fast Sequential Decoding Algorithm Using a Stack", *IBM J. Res. Dev.*, vol. 13, pp. 675-685
- [20] LeGoff, S., Glavieux, A. et Berrou, C. (1994), "Turbo-Codes and High Efficiency Modulation", *Proceedings of ICC'94*, pp. 645-649

- [21] Li, Y., Vucetic, B. et Sato, Y. (1995), "Optimum Soft Output Detection for Channels with Inter-Symbol Interference", *IEEE Trans. Inform. Theory*, vol. 41, No. 3, pp. 704-713
- [22] Lin, S. et Costello, D. J. Jr (1983), *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Englewood Cliffs, N.J.
- [23] Massey, J. L. (1963), *Threshold Decoding*, MIT Press, Cambridge, MA
- [24] Muller, D. E. (1954), "Application of Boolean Algebra to Switching Circuit Design and to Error Detection", *IRE Trans. Electronic Comput.*, vol. EC-3, pp. 6-12
- [25] Nyquist, H. (1924), "Certain Factors Affecting Telegraph Speed", *Bell Syst. Tech. J.*, vol. 3, p. 324
- [26] Pietrobon, S. S. et Barbulescu, S. A. (1994), "A Simplification of the Modified Bahl Decoding Algorithm for Systematic Convolutional Codes", *Int. Symp. Inform. Theory and its applications*, pp. 1073-1077
- [27] Proakis, J. (1995), *Digital Communications*, Third Edition, McGraw-Hill, New York
- [28] Reed, I. S. (1954), "A Class of Multiple-Error Correcting Codes and the Decoding Scheme", *IRE Trans. Inform.*, vol. IT-4, pp. 38-49
- [29] Reed, I. S. et Solomon, G. (1960), "Polynomial Codes Over Certain Finite Fields", *SIAM J.*, vol. 8, pp. 300-304
- [30] Robertson, P. (1994), "Improving Decoder and Code Structure of Parallel Concatenated Recursive Systematic (Turbo) Codes", *Int. Conf. Universal Personal Commun.*, pp. 183-187
- [31] Robertson, P., Villebrun, E. et Hoeher, P. (1995), "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain", *Proceedings of ICC'95*, pp. 1009-1013

- [32] Royer, G. (2000), "Évaluation des entrelaceurs au sein des codes turbo par simulations", *Mémoire de Maîtrise de l'École Polytechnique de Montréal*
- [33] Shannon, C. E. (1948), "A Mathematical Theory of Communication", *Bell Syst. Tech. J.*, vol. 27, pp. 379-423
- [34] Shannon, C. E. (1948), "A Mathematical Theory of Communication", *Bell Syst. Tech. J.*, vol. 27, pp. 623-656
- [35] Tanaka, H., Furusawa, K. et Kaneku, S. (1980), "A Novel Approach to Soft Decision Decoding of Threshold Decodable Codes", *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 244-246
- [36] Ungerboeck, G. (1982), "Channel Coding with Multilevel/Phase Signals", *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55-67
- [37] Viterbi, A. J. (1967), "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260-269
- [38] Viterbi, A. J. (1971), "Convolutional Codes and their Performance in Communication Systems", *IEEE Trans. Commun. Tech.*, vol. COM-19, pp. 751-772
- [39] Vucetic, B. et Yuan, J. (2000), *Turbo Codes: Principles and Applications*, Kluwer Academic Publisher, Boston
- [40] Wozencraft, J. M. et Jacobs, I. M. (1965), *Principles of Communication Engineering*, Wiley, New York
- [41] Wozencraft, J. M. et Rieffen, B. (1961), *Sequential Decoding*, MIT Press, Cambridge, MA
- [42] Yasuda, Y., Hirata, Y., Nakamura, K. et Otani, S. (1983), "Development of Variable-Rate Viterbi Decoder and its Performance Characteristics", *Sixth International Conference on Digital Satellite Communications*

Annexe I

Performances de PCSO2C-WS optimaux

Cette annexe présente les meilleures performances en termes de probabilité d'erreur que l'on peut obtenir avec des PCSO2C-WS de taux de codage $R = \frac{2}{3}$, $R = \frac{3}{4}$ et $R = \frac{4}{5}$. Ces résultats ont été obtenus avec les codes dont les positions des connexions sont données par les tableaux 4.1, 4.2 et 4.3.

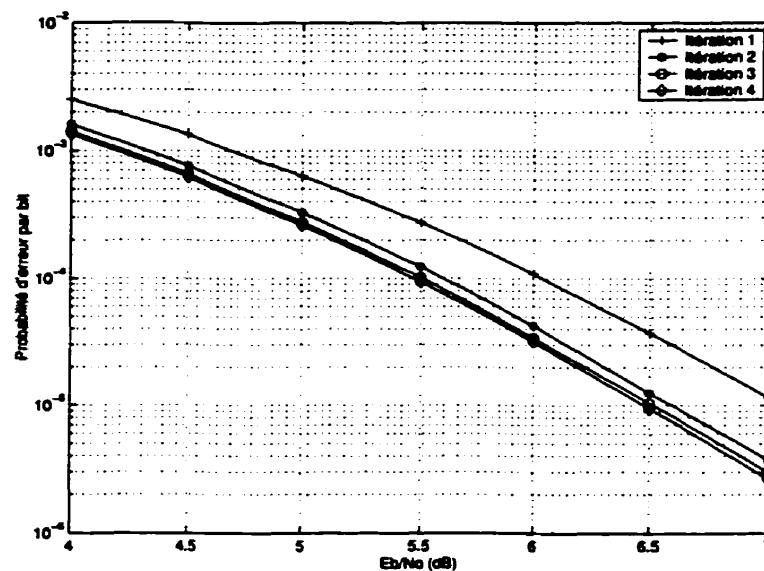


Figure I.1: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 4$, $R = \frac{2}{3}$

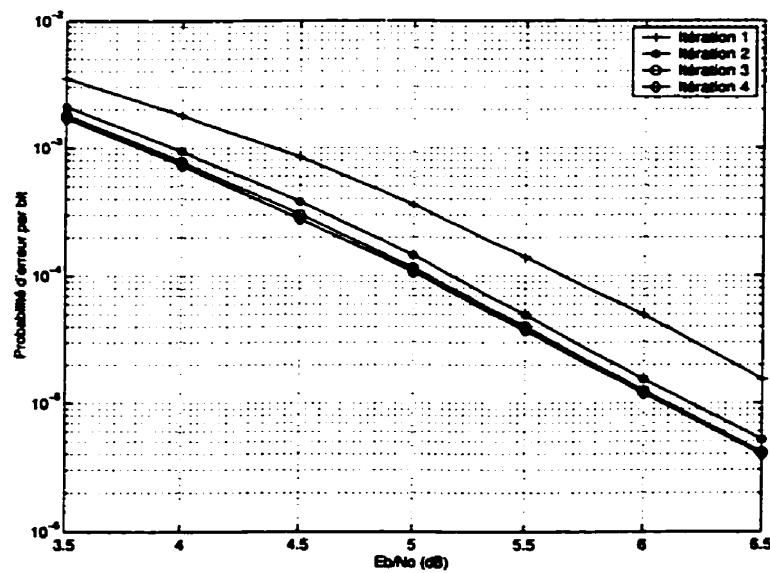


Figure I.2: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 5, R = \frac{2}{3}$

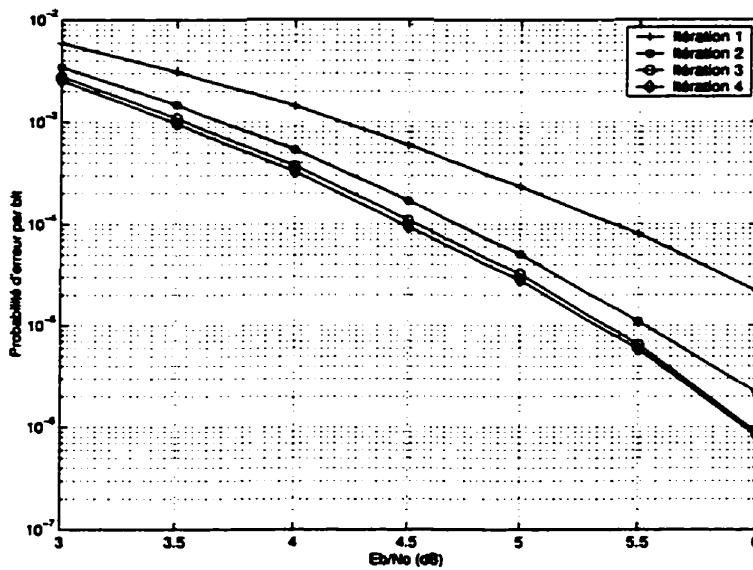


Figure I.3: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 6, R = \frac{2}{3}$

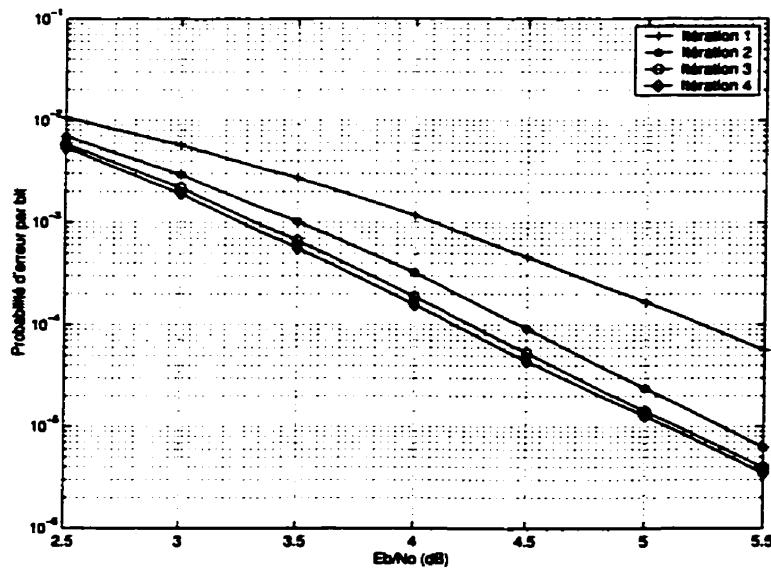


Figure I.4: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 7$, $R = \frac{2}{3}$

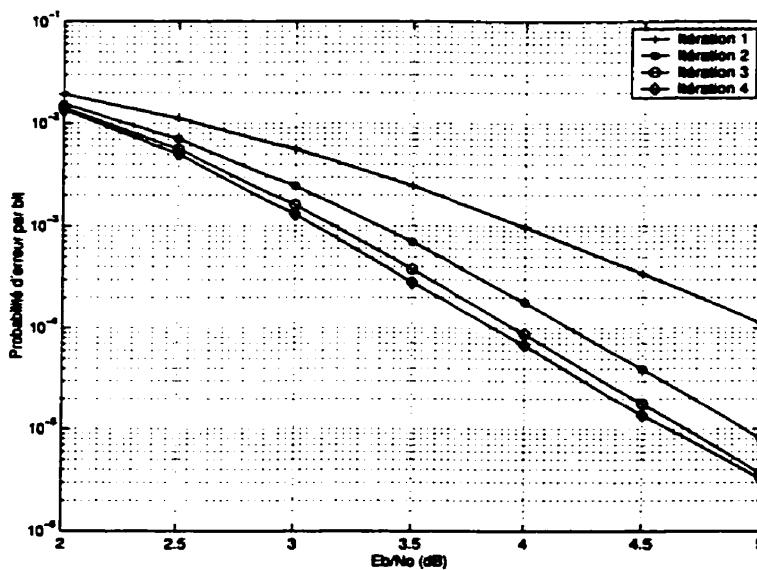


Figure I.5: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 8$, $R = \frac{2}{3}$

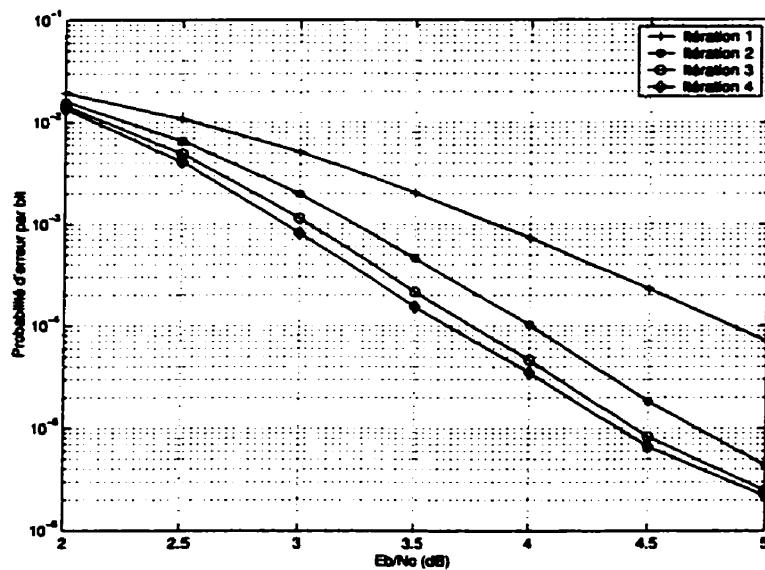


Figure I.6: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 9$, $R = \frac{2}{3}$

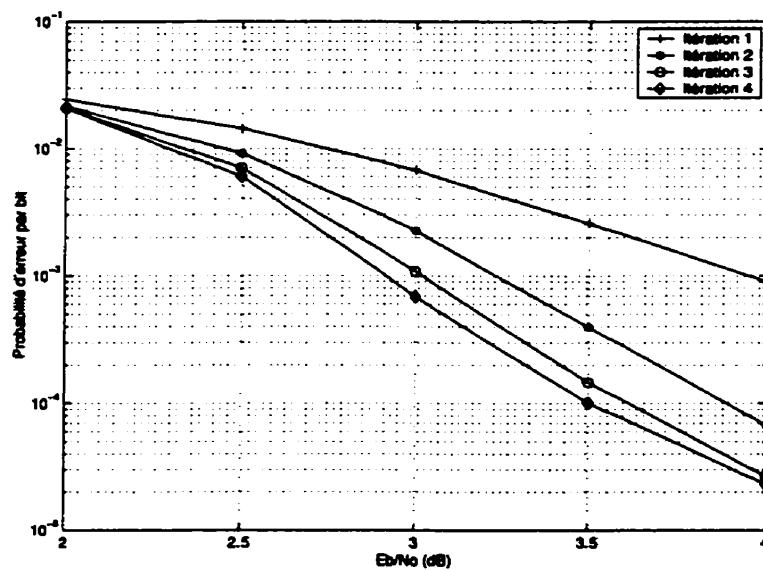


Figure I.7: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 10$, $R = \frac{2}{3}$

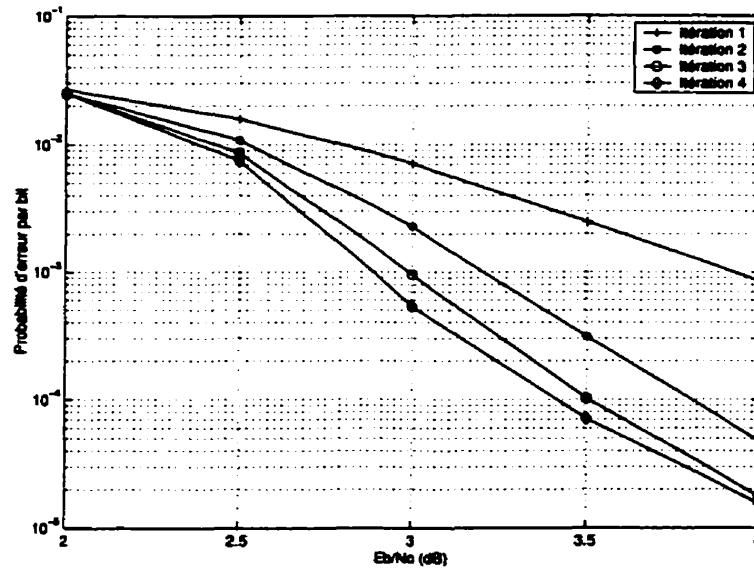


Figure I.8: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 11$, $R = \frac{2}{3}$

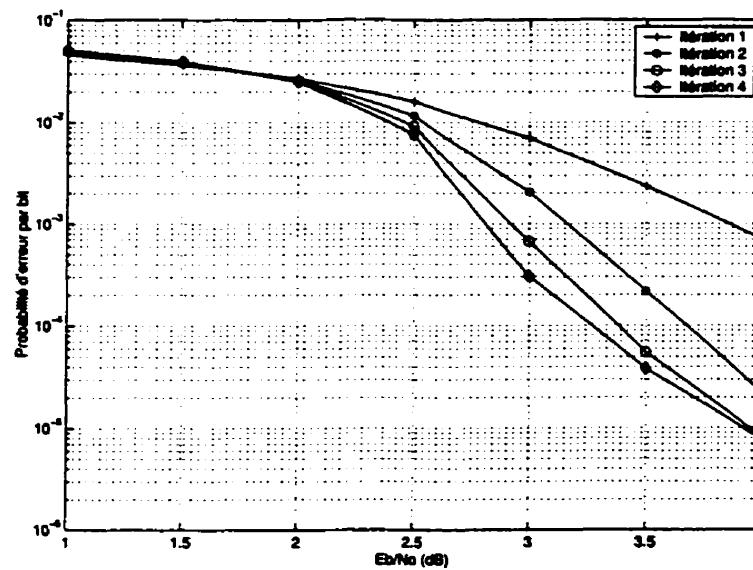


Figure I.9: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 12$, $R = \frac{2}{3}$

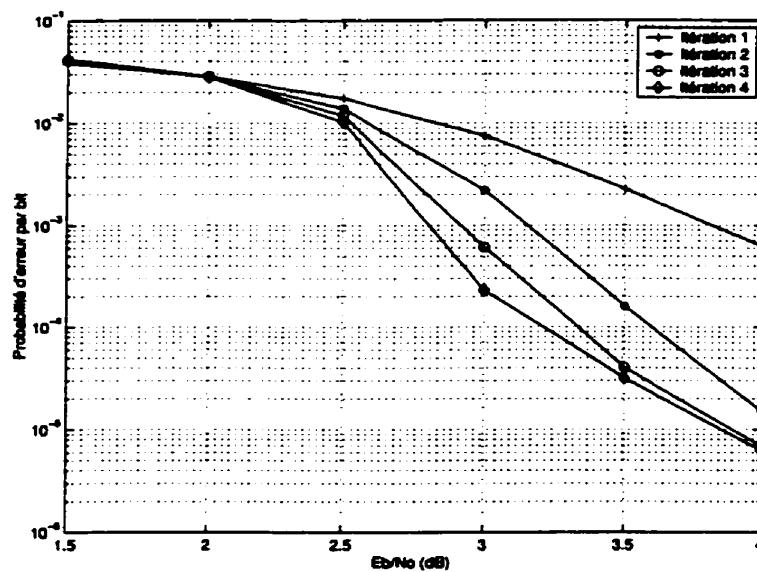


Figure I.10: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 13$, $R = \frac{2}{3}$

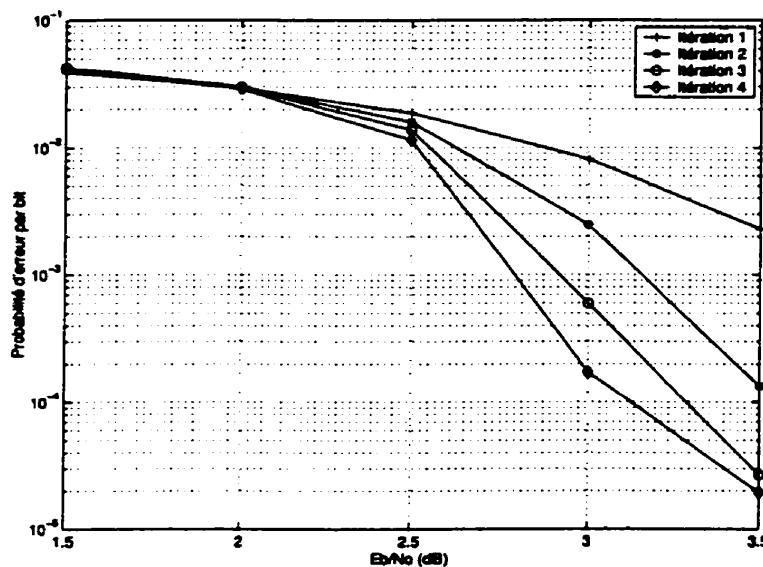


Figure I.11: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 14$, $R = \frac{2}{3}$

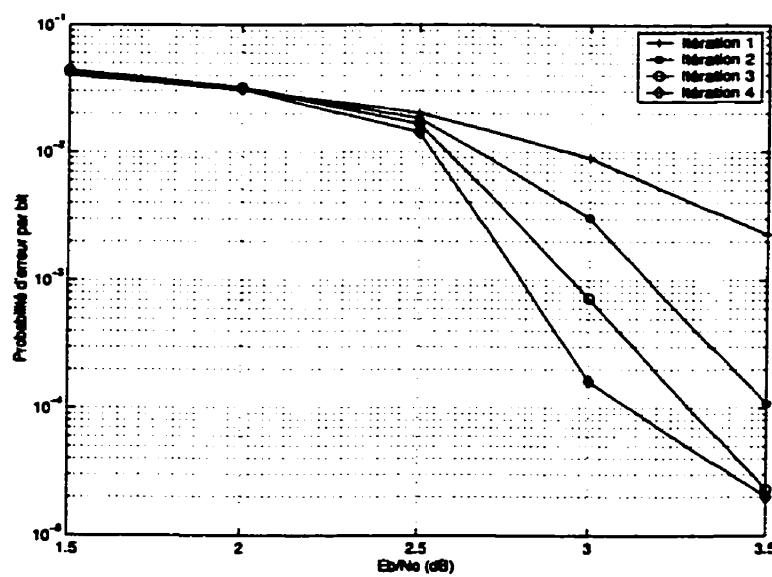


Figure I.12: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 15$, $R = \frac{2}{3}$

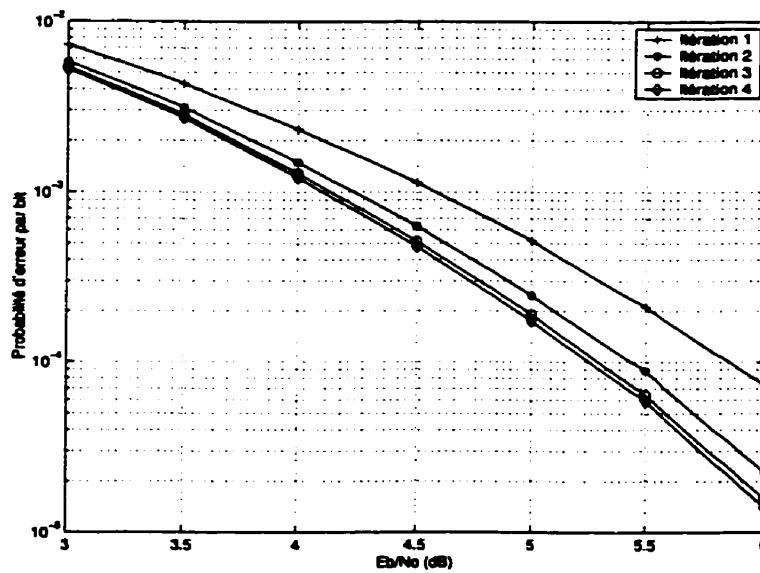


Figure I.13: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 6$, $R = \frac{3}{4}$

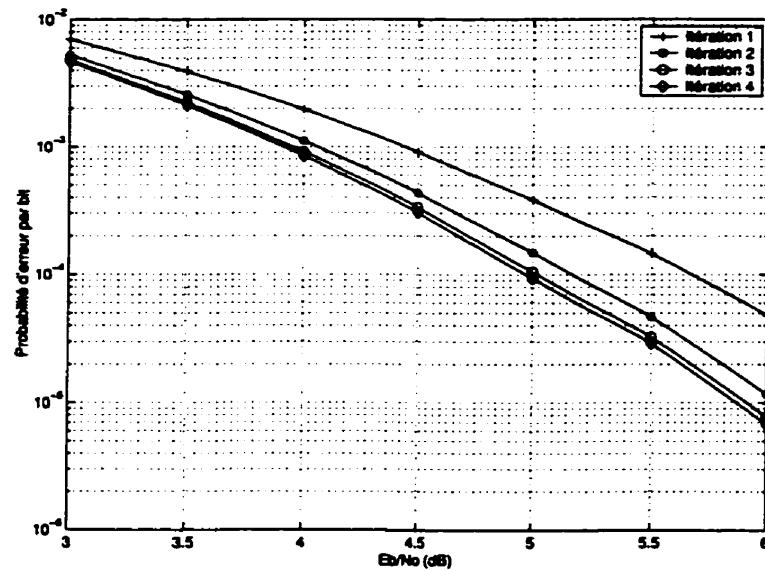


Figure I.14: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 7$, $R = \frac{3}{4}$

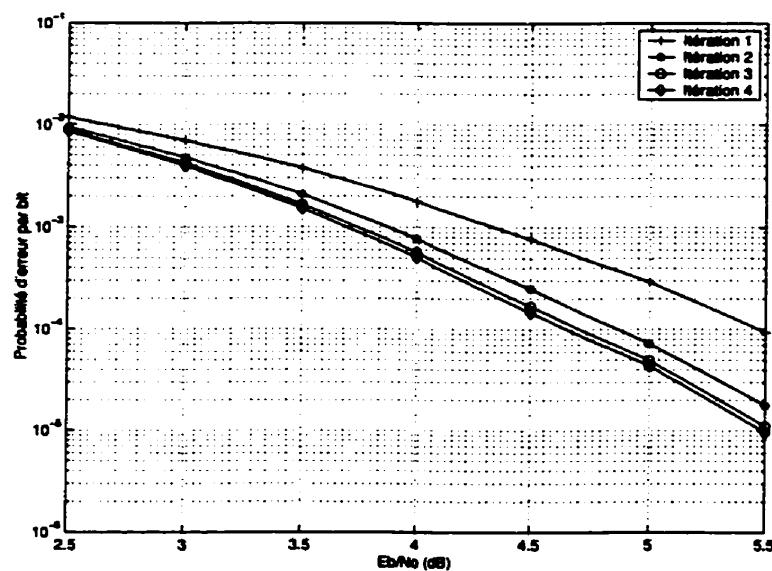


Figure I.15: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 8$, $R = \frac{3}{4}$

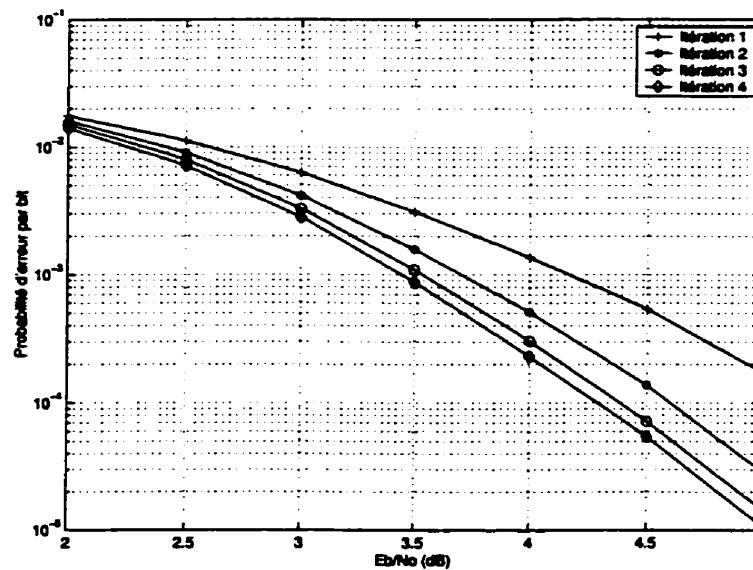


Figure I.16: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 9, R = \frac{3}{4}$

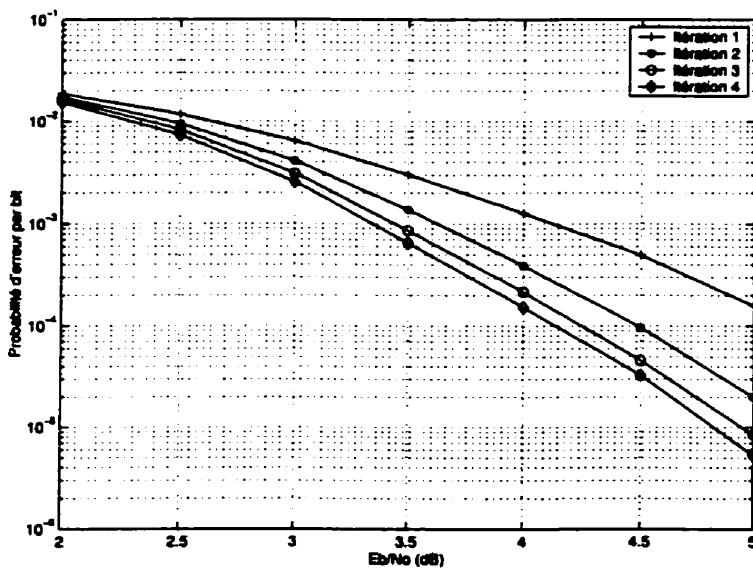


Figure I.17: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 10, R = \frac{3}{4}$

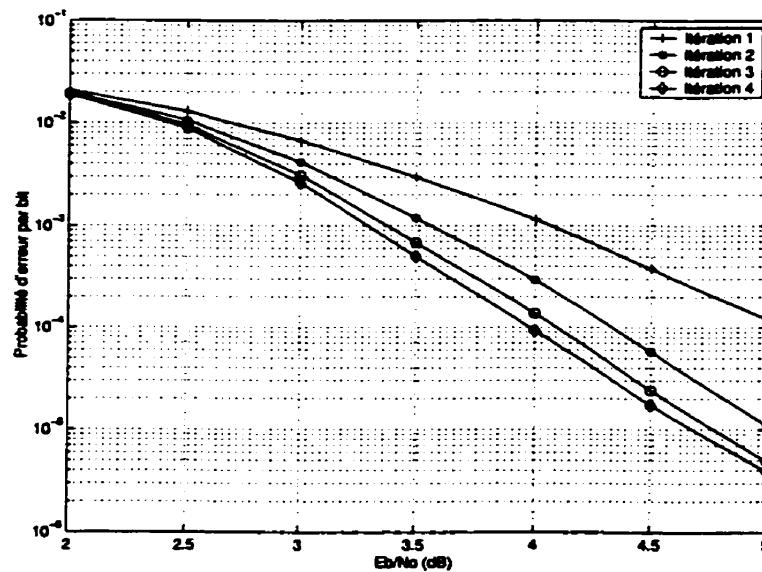


Figure I.18: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 11$, $R = \frac{3}{4}$

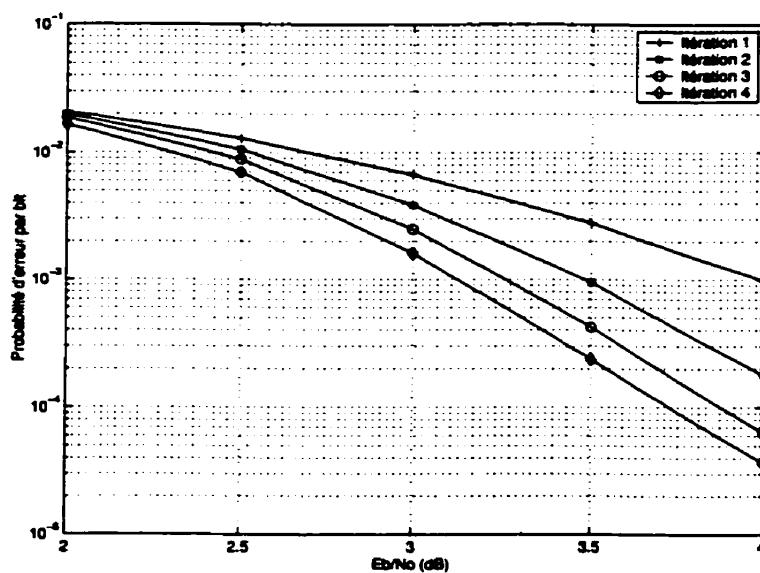


Figure I.19: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 12$, $R = \frac{3}{4}$

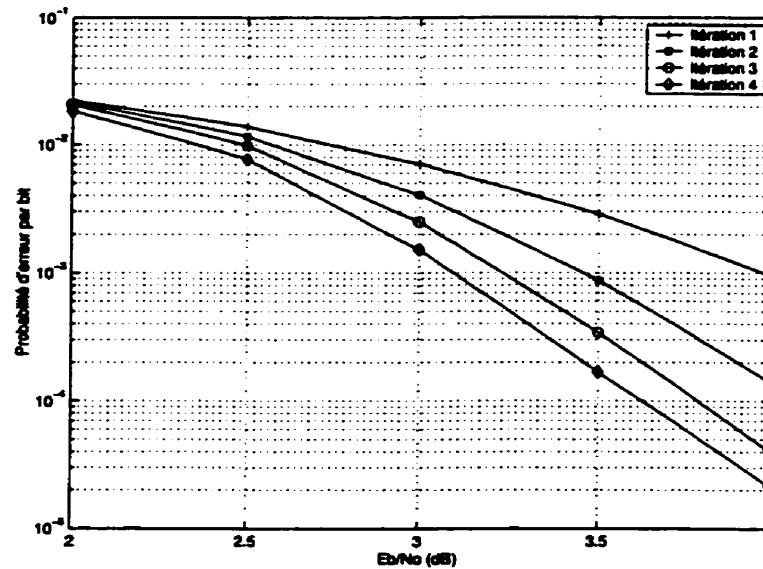


Figure I.20: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 13, R = \frac{3}{4}$

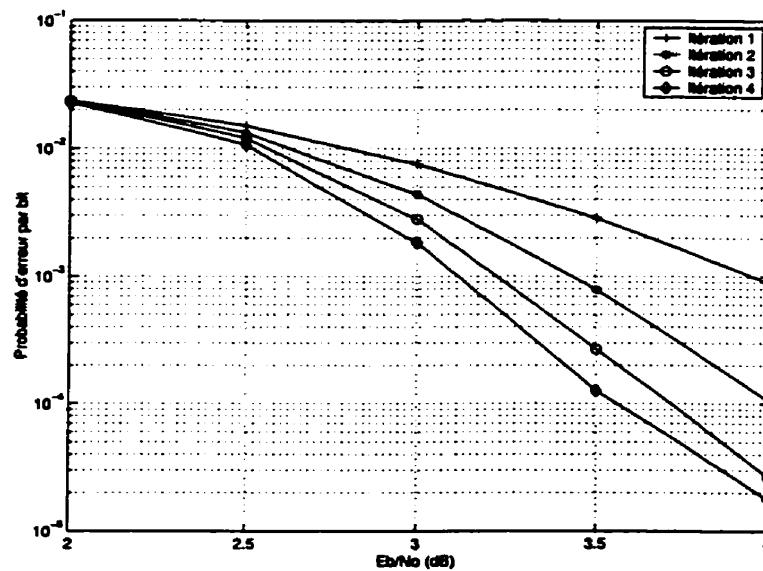


Figure I.21: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 14, R = \frac{3}{4}$

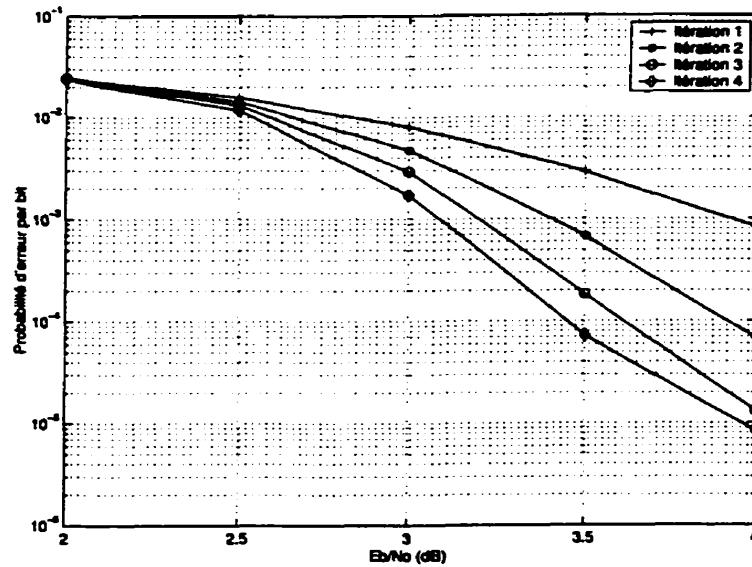


Figure I.22: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 15$, $R = \frac{3}{4}$

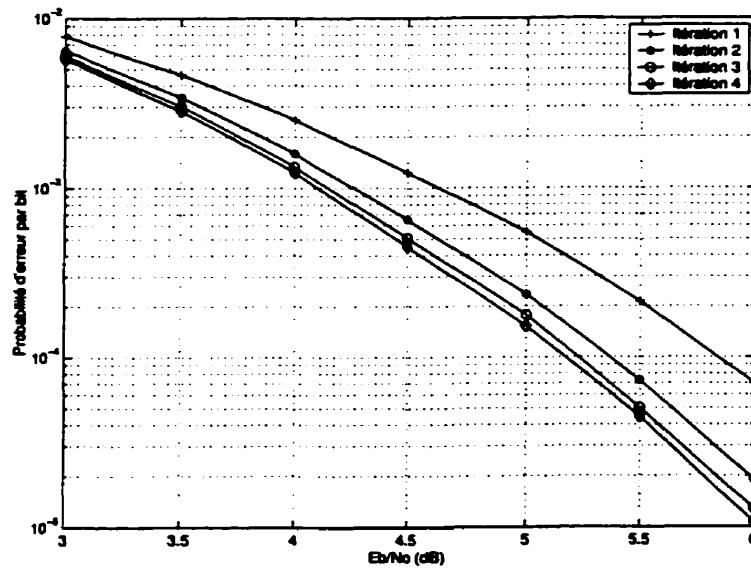


Figure I.23: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 8$, $R = \frac{4}{5}$

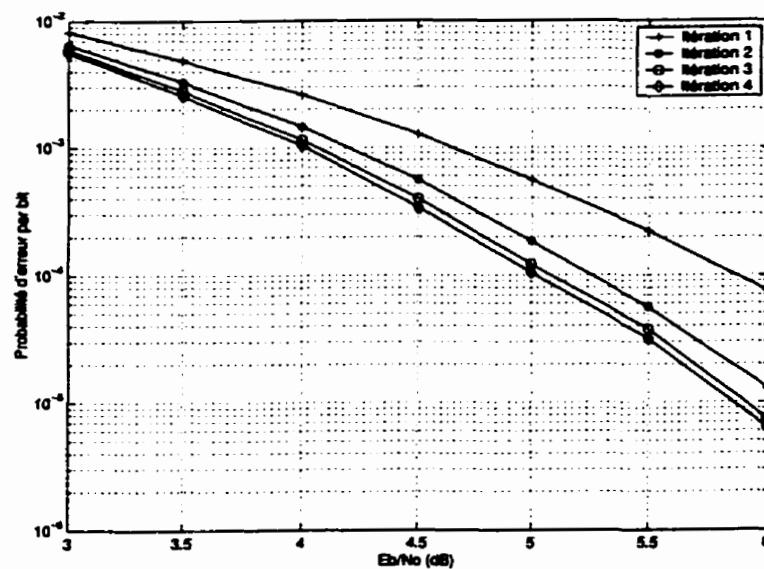


Figure I.24: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 9$, $R = \frac{4}{5}$

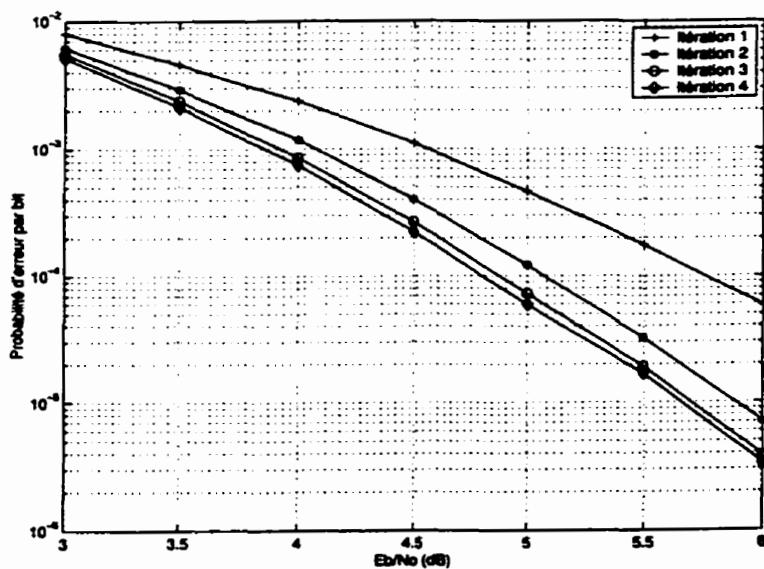


Figure I.25: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 10$, $R = \frac{4}{5}$

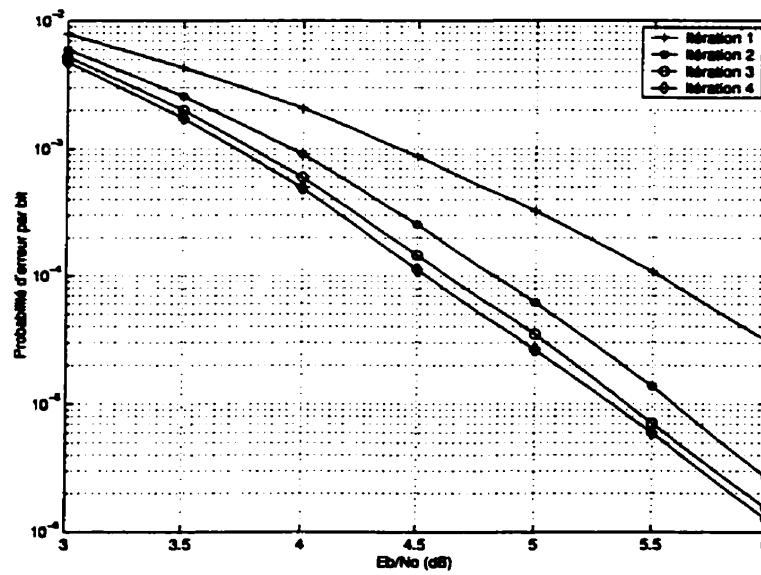


Figure I.26: Probabilité d'erreur par bit pour le PCSO2C-WS $J = 11$, $R = \frac{4}{5}$

Annexe II

Performances de certains PCSO2C-SS

Cette annexe présente les performances que l'on peut obtenir avec des PCSO2C-SS de taux de codage $R = \frac{2J}{3J}$ et $R = \frac{3J}{4J}$ dont les matrices de positions des connexions sont données à la section 4.4.

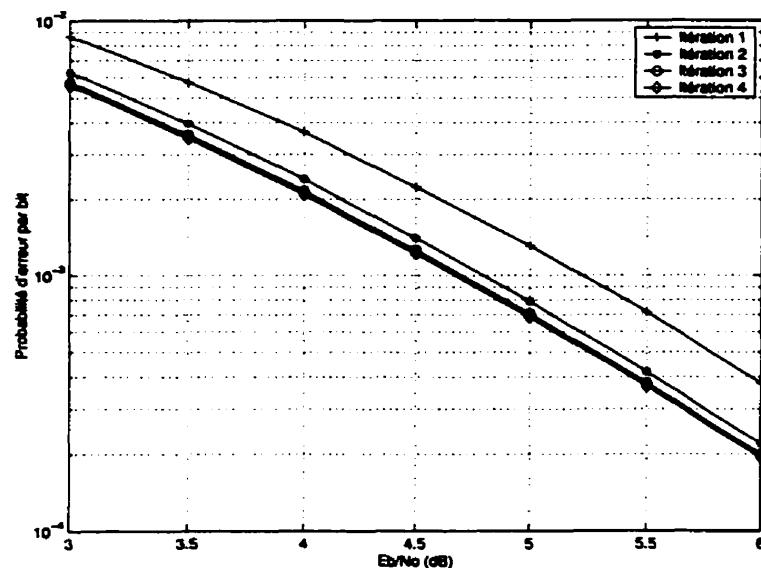


Figure II.1: Probabilité d'erreur par bit pour le PCSO2C-SS $J = 4$, $R = \frac{8}{12}$

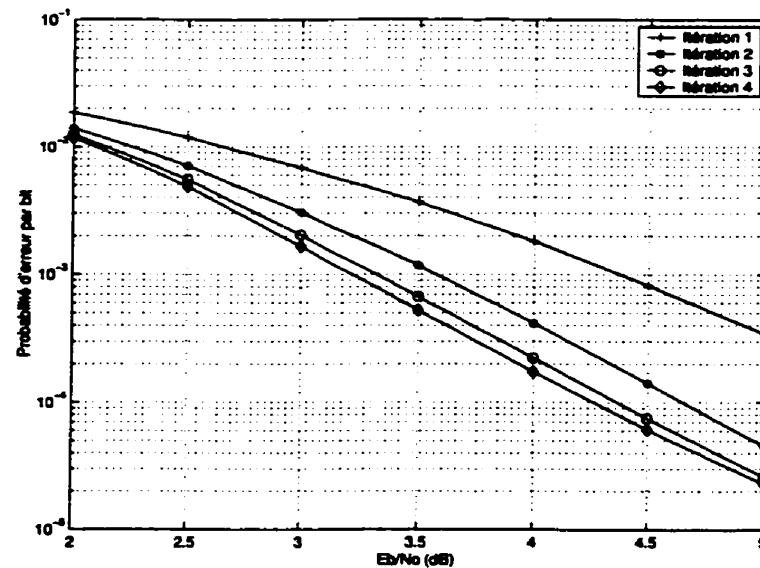


Figure II.2: Probabilité d'erreur par bit pour le PCSO2C-SS $J = 7$, $R = \frac{14}{21}$

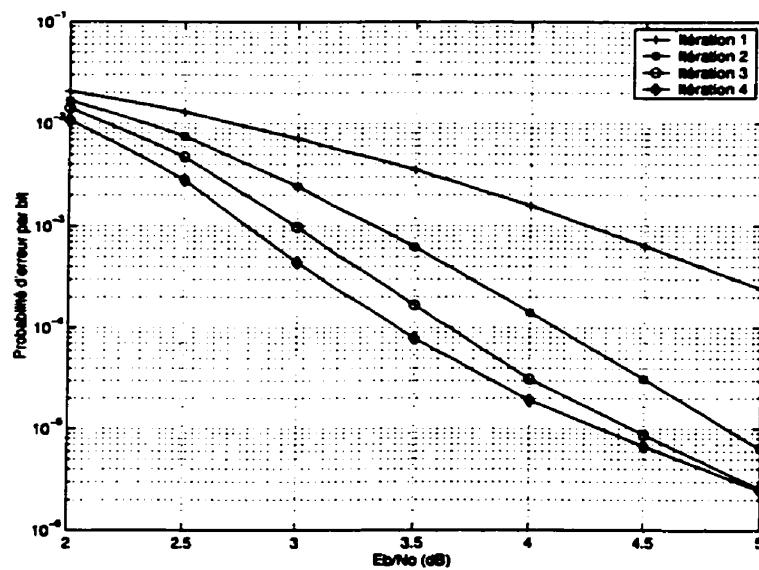


Figure II.3: Probabilité d'erreur par bit pour le PCSO2C-SS $J = 9$, $R = \frac{18}{27}$

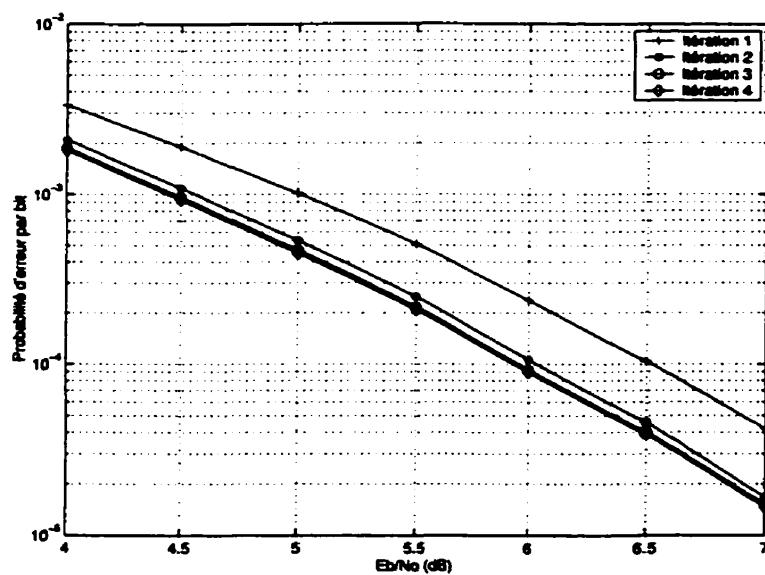


Figure II.4: Probabilité d'erreur par bit pour le PCSO2C-SS $J = 6$, $R = \frac{18}{24}$

Annexe III

Représentation vectorielle des signaux

Considérons un signal $s(t)$ déterministe, réel et à énergie finie donnée par:

$$\mathcal{E}_s = \int_{-\infty}^{\infty} s^2(t) dt \quad (\text{III.1})$$

En outre, supposons que nous possédons un ensemble de N fonctions $\{f_n(t), n = 1, 2, \dots, N\}$ orthonormales, c'est-à-dire vérifiant:

$$\int_{-\infty}^{\infty} f_n(t) f_m(t) dt = \delta_{m,n} \quad (\text{III.2})$$

On peut alors approximer le signal $s(t)$ par une combinaison linéaire des ces fonctions, soit:

$$\hat{s}(t) = \sum_{n=1}^{N} s_n f_n(t) \quad (\text{III.3})$$

où $\{s_n, n = 1, \dots, N\}$ sont les coefficients de l'approximation de $s(t)$. L'erreur induite par cette approximation est alors:

$$e(t) = s(t) - \hat{s}(t) \quad (\text{III.4})$$

et:

$$\mathcal{E}_e = \int_{-\infty}^{\infty} \left[s(t) - \sum_{n=1}^{N} s_n f_n(t) \right]^2 dt \quad (\text{III.5})$$

On veut donc choisir les coefficients de l'approximation de $s(t)$ tels que l'énergie de l'erreur exprimée par (III.5) soit minimale. D'après le critère des moindres carrés, ce minimum est obtenu pour une erreur orthogonale aux fonctions qui constituent la base, c'est-à-dire:

$$\int_{-\infty}^{\infty} \left[s(t) - \sum_{n=1}^{N} s_n f_n(t) \right] f_m(t) dt = 0, \quad m = 1, 2, \dots, N \quad (\text{III.6})$$

soit:

$$s_n = \int_{-\infty}^{\infty} s(t) f_n(t) dt, \quad n = 1, 2, \dots, N \quad (\text{III.7})$$

Suivant ce critère, les coefficients de l'approximation de $s(t)$ sont obtenus en projetant ce signal sur les fonctions constituant la base, ce qui revient à dire que $\hat{s}(t)$ est la projection de $s(t)$ sur l'espace vectoriel engendré par les N fonctions $f_n(t)$.

Alors on a:

$$\begin{aligned} \mathcal{E}_{\min} &= \int_{-\infty}^{\infty} s(t) e(t) dt \\ &= \mathcal{E}_s - \sum_{n=1}^N s_n^2 \end{aligned} \quad (\text{III.8})$$

Lorsque $\mathcal{E}_{\min} = 0$, on a:

$$\mathcal{E}_s = \sum_{n=1}^N s_n^2 = \int_{-\infty}^{\infty} s^2(t) dt \quad (\text{III.9})$$

À ce moment-là:

$$s(t) = \sum_{n=1}^N s_n f_n(t) \quad (\text{III.10})$$

On notera bien que $s(t)$ n'est égal à son approximation que si l'énergie minimale de l'erreur est nulle.

La question qui reste en suspens est donc de savoir quel ensemble de fonctions de base choisir pour projeter le signal désirer. Supposons que l'on possède un ensemble de signaux à énergie finie $\{s_q(t), q = 1, 2, \dots, M\}$. La procédure d'orthogonalisation de Gram-Schmidt [27], [40] permet de construire un ensemble tel que $\mathcal{E}_{\min} = 0$. On remarque que la dimension de la base, N , est inférieure ou égale à M ($N \leq M$) et il y a égalité dans le cas où les M signaux dont on dispose sont linéairement indépendants.

Une fois construit cet ensemble de fonctions de base orthonormales $f_n(t)$, on peut exprimer les M signaux $s_q(t)$ comme des combinaisons linéaires des $f_n(t)$. Ainsi:

$$s_q(t) = \sum_{n=1}^N s_{q,n} f_n(t), \quad q = 1, 2, \dots, M \quad (\text{III.11})$$

et:

$$\mathcal{E}_q = \int_{-\infty}^{\infty} s_q^2(t) dt = \sum_{n=1}^N s_{q,n}^2 = \| \mathbf{s}_q \|^2 \quad (\text{III.12})$$

À partir de (III.11), chaque signal peut être représenté par un vecteur:

$$\mathbf{s}_q = [s_{q,1} \ s_{q,2} \ \dots \ s_{q,N}] \quad (\text{III.13})$$

ce qui peut se représenter par un point dans un N -espace vectoriel dont les coordonnées sont $\{s_{q,i}, i = 1, 2, \dots, N\}$. L'énergie du q -ème signal est tout simplement le carré de la longueur de ce vecteur. De cette manière, tout signal peut être représenté géométriquement dans ce que l'on appelle l'espace des signaux qui est généré par une suite de fonctions orthonormales.

Ce genre de représentation s'avère particulièrement utile pour les modulations numériques. En effet, les M signaux considérés sont ceux susceptibles d'être émis par le modulateur numérique sur le canal de transmission. À partir de cette famille de signaux, on peut construire une suite de fonctions de base par la procédure de Gram-Schmidt et représenter alors tout signal de cette famille dans l'espace vectoriel de dimension N généré par cette famille (on parle alors de *constellation*). Les notions d'énergie et de distance sont alors très liées et on peut effectuer des analyses d'erreur directement à partir de cette représentation.