REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE

Departement d'Electronique

Projet de Fin D'Etudes
En vue de l'obtention du Diplôme
d'Ingénieur d'Etat en Electronique

Thème

Etude des Codes LDPC et application dans un système MIMO

Réalisé par : **M. TEMER Elias**

Soutenu le 06 Septembre 2009 devant le jury composé de :

Président :	M. H. BOUSBIA SALAH	Maître de conférences (ENSP)
Examinateur :	M^{lle} A. MOUSSAOUI	Chargée de cours (ENSP)
Rapporteurs :	M^{me} L. HAMAMI	Professeur (ENSP)
	M. A. KHALIGHI	Maître de conférences (Ecole Centrale Marseille)

Promotion : Septembre 2009

Remerciements

Nous tenons à remercier Dieu le tout Puissant et Maître de l'univers, qui nous a donné la force nécessaire, la forte volonté et la patience, afin d'accomplir ce modeste travail.

Tout d'abord, je remercie particulièrement et très chaleureusement ma promotrice, Pr L. Hamami, avec qui j'ai eu tant de plaisir à travailler et de profiter de son expérience, ses conseils précieux, ses remarques pertinentes et surtout sa patience infinie. je tiens également à exprimer ma profonde reconnaissance au Dr A. Moussaoui pour tous ses efforts consentis pour la réussite de mon stage à l'Institut Fresnel à Marseille.

J'adresse mes plus sincères remerciements au modeste et compétant, Dr M. A. Khalighi avec qui j'ai eu l'occasion de travailler sur un sujet très récent et si passionnant, et avec qui j'ai appris énormément de choses sur les codes correcteurs, les systèmes MIMO, la programmation en C et la rédaction sous LaTeX.

Je n'oublierai pas de remercier le Pr S. Bourennane, le Dr M. Adel et tous les membres de l'équipe Groupe Signaux Multidimensionnels de l'institut Fresnel : R. Khelifi, Yacine, Fang, Dang, J. Sylvain, Damien, N. Drabik et tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce modeste travail.

J'exprime ma gratitude et mes remerciements sincères envers le Dr H. Bousbia qui m'a fait l'honneur de présider le jury. Je remercie encore une deuxième fois le Dr A. Moussaoui d'avoir accepté d'être membre du jury et de juger ce modeste travail.

Que tous mes professeurs qui ont contribué à ma formation trouvent ici ma plus profonde gratitude.

Enfin, je souhaite dédier ce mémoire à mes parents et mes oncles. Rien n'aurait été possible sans leur soutien, confiance et générosité.

Dédicaces

A mes très chers parents

A mes frères et sœur : Sofiane, Mounir et Dahbia

A mes oncles et tante : Kamel, Mourad et Houria

A tous mes proches

A Madame : L. HAMAMI

A Mademoiselle : A. MOUSSAOUI

A Monsieur : M. A. KHALIGHI

A mes amis : Tarek, Adel, Riad, Mehdi, Ahmed ...

A tous mes amis de Hassi Messaoud

A tous mes amis de l'ENP et de GSM

A tous ceux qui ont su croire en moi

A tous ceux qui me sont chers

Je dédie ce modeste travail

"I have not failed. I have just found 10,000 ways that will not work."

Inventor of Electricity

Thomas A. Edison

TEMER Elias

ملخص :

تم تحقيق هذه الدراسة في مخبر فرينل خلال تربص دام ستة أشهر, حيث كان موضوع الدراسة يتمحور حول تقنية التشفير الحديثة LDPC و تطبيقها في النظام MIMO . تم القيام بهذا العمل بهدف تعزيز أبحاث المخبر حول نظام الاتصالات ألاسلكية الجديد WIMAX IEEE 802.16 m الذي هو بصدد إدماج تقنية التشفير الحديثة LDPC و النظام MIMO . قمنا في البداية بتقديم ملخص حول تشفير مجرى الاتصالات. بعد ذلك تم القيام بدراسة مدققة حول تقنية التشفيرLDPC,فقمنا بذالك بتقديم مفاهيم حول فئات الشفرات و تمثيلها بالمصفوفات أو بالرسم البياني. ثم قمنا بعدها بتقديم خوارزميات التشفير و خوارزميات نزع التشفير. بعد الانتهاء من الدراسة النظرية, تطرقنا لتقييم قدرات تقنية التشفير LDPC و ذالك عن طريق التظاهر بواسطة برنامج MATLAB و برنامج C . هذه الدراسة مكنتنا باستنتاج مدى أهمية بعض الإعدادات في اختيار الشفرة المناسبة. استنادا على هذه النتائج, قمنا بتطبيق تقنية التشفير LDPC في النظامMIMO ثم قمنا بمقارنة قدراتها بقدرات تقنية التشفير الكلاسيكية Convolutif . في الأخير بجدر الإشارة إلى أنه تم إنجاز واجهة صورية تمكن من جمع كل برامج الخوارزميات المطورة في كل من برنامج MATLAB و برنامج C تم تطبيق جميع التظاهرات المذكورة بطريقة سهلة و بسيطة.

كلمات مفتاحية : تشفير مجرى الاتصالات, القدرات التقنية, تقنية التشفير LDPC , تقنية التشفير Convolutif , الرسم البياني Tanner , خوارزميات نزع التشفير BP , خوارزميات تشفير الباحث R. Neal , النظام MIMO , تشفير وقت-مكان ,قدرة مجرى الاتصالات.

Résumé :

Les travaux présentés dans ce mémoire sont le fruit de six mois de stage au sein de l'institut Fresnel. Le sujet étudié, portant sur l'étude des codes LDPC et leurs applications dans un système MIMO, rentre dans les travaux de recherche du laboratoire sur la prochaine norme IEEE 802.16 m du standard WIMAX, qui envisage d'introduire les codes LDPC avec les systèmes MIMO. Après une brève introduction sur le codage de canal, une étude théorique des codes LDPC est faite où les différentes classes de codes LDPC ainsi que leurs représentations matricielle et graphique et les différents algorithmes de décodage et d'encodage sont présentés. Une évaluation des performances des codes LDPC à travers des simulations sous MATLAB et le langage C est aussi abordée. Cette étude originale a permis de mettre en évidence l'importance du choix de certains paramètres sur les performances du code. En se basant sur ces résultats pour le choix des paramètres du codeur/décodeur, nous avons implémenté ce dernier dans un système MIMO afin de comparer ses performances avec celles d'un codeur/décodeur convolutif classique, et voir l'apport d'un code correcteur d'erreur puissant sur les performances du système. En dernier lieu, le logiciel LDMO que nous avons conçu dans le but de regrouper, sous forme d'interfaces graphiques, les programmes et fonctions développés sous MATLAB et sous langage C, a été présenté.

Mots-clés : Codage de canal, performances, capacité du canal, codes LDPC, diagramme de Tanner, Algorithme de propagation de croyances BP, Algorithmes de R. Neal, système MIMO, codage spatio-temporel.

Abstract:

The works presented in this report are the fruit of six months of training course within the Fresnel institute. The studied subject, concerning the study of the LDPC codes and their applications in a MIMO system, is one of the research's subjects of the laboratory on the next standard of WIMAX : IEEE 802.16 m, which intends to introduce the LDPC codes and the MIMO systems. After a brief introduction about the channel coding, a theoretical study of the LDPC codes was done, by introducing their different classes, their matrix and graphic representations and the different algorithms of decoding and encoding. An evaluation of the performances of the LDPC codes by making simulations in MATLAB and C language is also done. This original study shows the importance of the choice of certain parameters on the performances of the code. According to the previous study, we chose the right parameters of the LDPC coder/decoder, then we implemented it in a MIMO system, after that, we have compared its performances with these obtained with a simple Convolutional coder/decoder. Finally, we presented the LDMO software, which was conceived in the purpose to gather all the programs and the functions developed under MATLAB and C language, in the form of graphic interfaces.

Keywords: Channel coding, performances, channel capacity, LDPC codes, Convolutional codes, Tanner diagram, belief propagation algorithm BP, R. Neal encoding algorithms, MIMO system, space-time coding.

Table des matières

Résumé **I**

Table des matières **II**

Table des figures **V**

Liste des tableaux **IX**

Liste des abréviations **XI**

Liste des notations **XIII**

Introduction Générale **XV**

1 Présentation du laboratoire **1**

1.1 Introduction 1

1.2 Historique du laboratoire 2

1.3 Organigramme 3

1.4 Le personnel 3

1.5 Les équipes de recherche 4

1.6 les publications 4

1.7 Le Groupe Signaux Multidimensionnels 5

1.7.1 Organisation 5

1.7.2 Les thèmes de recherche 5

1.8 Le plan de travail du stage pratique 6

1.9 Conclusion 6

2 Introduction au codage de canal **7**

2.1 Introduction 7

2.2 Généralités 8

2.2.1 La chaîne de communication numérique 8

2.2.2 Le codage de source . . . 8
2.2.3 Le codage de canal . . . 9
2.2.4 La modulation . . . 9
2.2.5 Le canal de communication . . . 10
2.2.6 La capacité d'un canal . . . 12
2.2.7 Le théorème fondamental du codage de canal . . . 13
2.3 Les codes correcteurs d'erreurs . . . 14
2.3.1 Définitions et notations . . . 14
2.3.2 Mesure des performances d'un code correcteur d'erreurs . . . 15
2.3.3 Concaténation de codes . . . 16
2.3.4 Les classes de codes correcteurs . . . 17
2.4 Les codes en bloc . . . 17
2.4.1 Les codes linéaires en bloc . . . 18
2.4.2 Exemples de codes en bloc . . . 19
2.5 Les codes convolutifs . . . 20
2.5.1 Les codes NSC et RSC . . . 21
2.5.2 Les turbo-codes . . . 22
2.6 Comparaison des performances entre quelques codes correcteurs d'erreurs . . . 23
2.7 Conclusion . . . 25

3 Etude des codes LDPC 26
3.1 Historique . . . 26
3.2 Définitions et notations . . . 27
3.3 Les classes de codes LDPC . . . 29
3.3.1 Les codes réguliers . . . 29
3.3.2 Les codes irréguliers . . . 30
3.4 Représentation graphique des codes LDPC . . . 31
3.4.1 Le profil d'irrégularité des noeuds de données et des noeuds de contrôle . . . 33
3.4.2 La notion de cycle . . . 34
3.5 Décodage itératif des codes LDPC . . . 34
3.5.1 Décodage à décision dure . . . 35
3.5.2 Algorithme de propagation de croyance . . . 37
3.5.3 Algorithme de décodage Log-Domain . . . 40
3.5.4 Algorithme de décodage Min-Sum . . . 43
3.6 Encodage des codes LDPC . . . 44
3.7 Algorithme d'encodage de Radford Neal . . . 46
3.7.1 Algorithme d'encodage dense . . . 47

3.7.2 Algorithme d'encodage mixte 48
3.7.3 Algorithme d'encodage basé sur la décomposition L-U 48
3.8 Etude des performances des codes LDPC 52
3.8.1 Présentation de la chaîne de simulation 52
3.8.2 Comparaison entre les différents algorithmes de décodage 53
3.8.3 Influence de la taille du code sur les performances 54
3.8.4 Influence du nombre d'itérations du processus de décodage sur les performances 57
3.8.5 Influence du rendement de codage sur les performances 58
3.9 Conclusion 61

4 Application des codes LDPC dans un système MIMO 62
4.1 Introduction 62
4.2 Le canal radio mobile 63
4.2.1 Propagation multi-trajet 63
4.2.2 Evanouissement du canal 65
4.3 Le principe des systèmes MIMO 65
4.4 Capacité d'un canal MIMO 67
4.5 Les gains apportés par des systèmes MIMO 69
4.5.1 Gain de diversité spatiale 69
4.5.2 Gain de multiplexage 69
4.6 Le codage spatio-temporel 69
4.6.1 Le code orthogonal d'Alamouti 70
4.6.2 Le multiplexage spatial 71
4.7 Le contexte de l'application 72
4.8 Modélisation du canal radio mobile 72
4.9 Etude des performances des codes LDPC dans une architecture MIMO 73
4.9.1 La chaîne de transmission 74
4.9.2 Conditions et hypothèses de simulations 75
4.9.3 Résultats et discussions 76
4.10 Conclusion 82

5 Présentation du logiciel LDMO 83
5.1 Introduction 83
5.2 Présentation générale 83
5.3 GUI n°I.1 : Performances des codes LDPC 84
5.3.1 GUI n°I.1.1 : Codage et décodage LDPC 85

5.3.2 GUI n°I.1.3 : Simulations 87
5.4 GUI n°I.2 : Application des codes LDPC dans un système MIMO 89
5.4.1 Lancement des simulations sous le langage C 89
5.4.2 Affichage des résultats 90
5.5 GUI n°I.3 : Bibliographie 91
5.6 Help 93
5.7 Conclusion 93

Conclusions et perspectives **94**

Annexes **97**

A Algorithms for LDPC decoder **97**

B Décomposition SVD d'un canal MIMO **100**

C MIMO detection : ST decoding **102**

Bibliographie **105**

Table des figures

1.1 L'institut Fresnel et ses tutelles. 1
1.2 L'organigramme de l'institut Fresnel. 3
1.3 Les équipes de recherche de l'institut Fresnel. 4
1.4 l'organigramme de l'équipe GSM. 5
1.5 Le plan de travail du stage pratique. 6

2.1 Schéma fondamental d'une communication numérique : *le paradigme de Shannon*. 8
2.2 Schéma simplifié d'un codeur de source. 8
2.3 Exemples de modulations numériques. 9
2.4 (a) Le canal binaire avec effacement BEC. (b) Le canal binaire symétrique BSC. . 10
2.5 Canal à entrée binaire perturbé par l'addition d'un bruit gaussien. 11
2.6 Schéma représentatif de l'information mutuelle I(X;Y). 12
2.7 Variation de la capacité du canal AWGN en fonction du SNR. 13
2.8 Schéma simplifié d'un codeur/décodeur de canal. 14
2.9 Illustration des régions caractérisant les performances d'un code correcteur d'erreurs. 16
2.10 Concaténation de deux codes correcteurs d'erreurs. 16
2.11 Classification des codes correcteurs d'erreurs. 17
2.12 Schéma de principe d'un codeur convolutif de rendement R et de mémoire m. . . 20
2.13 Exemple d'un code convolutif de rendement R=1/2. 21
2.14 (a) Un code convolutif de R=1/2. (b) Un diagramme en Treillis. 21
2.15 (a) Un code non-systématique (NSC) (b) Un code récursif systématique (RSC). . 22
2.16 Schéma de principe d'un turbo-codeur. 23
2.17 Schéma de principe d'un turbo-décodeur. 23
2.18 Comparaison entre les performances des codes de parité, de Hamming et de Golay (Courbes reproduites de la référence [1]). 24
2.19 Comparaison entre les performances des codes convolutifs, Reed-Solomon et les techniques de codage avancées (LDPC et Turbo-codes) (Courbes reproduites de la référence [2]). 24

3.1 Exemple d'une matrice de contrôle de parité H de dimension (1000,2000) et de poids $w_c = 3$. 28
3.2 Comparaison de performances entre les codes LDPC réguliers et irréguliers de taille N=16000 et de rendement $R = 1/2$ (Graphes reproduits de la référence [3]) . 31
3.3 Comparaison de performances entre les codes LDPC réguliers et irréguliers de taille N=16000 et de rendement $R = 1/4$ (Graphes reproduits de la référence [3]). 31
3.4 Graphe de Tanner d'un code LDPC irrégulier. 32
3.5 Exemples de cycles de longueur 4, 6 et 8. 34
3.6 Graphe de Tanner d'un code LDPC régulier. 35
3.7 La mise à jour des noeuds de contrôle. 39
3.8 La mise à jour des noeuds de données. 39
3.9 Schéma illustratif de l'algorithme Log-Domain. 42
3.10 L'allure de la fonction $\phi(t)$. 44
3.11 Schéma illustratif de l'algorithme Min-Sum. 45
3.12 Comparaison entre les trois algorithmes de décomposition L-U pour $w_c = 3$ et R=1/2 (Courbes reproduites de la référence [4]). 51
3.13 Comparaison entre les trois algorithmes de décomposition L-U pour $w_c = 4$ et R=1/2 (Courbes reproduites de la référence [4]). 51
3.14 Le modèle de simulation utilisé pour l'évaluation des performances des codes LDPC 53
3.15 Comparaison des performances entre les différents algorithmes de décodage pour N=1024. 54
3.16 Influence de la taille du code sur les performances pour un décodage Hard. 55
3.17 Influence de la taille du code sur les performances pour l'algorithme de décodage Log Domain. 55
3.18 Influence de la taille du code sur les performances pour l'algorithme de décodage Min-Sum. 56
3.19 Influence de la taille du code sur les performances pour l'algorithme de décodage Log-Domain. 57
3.20 Influence du nombre d'itérations sur les performances des codes LDPC pour N=200 et un décodage Min-Sum 58
3.21 Influence du nombre d'itérations sur les performances des codes LDPC pour N=1024 et un décodage Min-Sum 59
3.22 Influence du rendement R sur les performances des codes LDPC (BER =f(SNR)). 60
3.23 Influence du rendement R sur les performances des codes LDPC (BER =f(E_b/N_0)). 60

4.1 La propagation multi-trajet dans un milieu urbain. 63

4.2 Variation de la réponse impulsionnelle et de la fonction de transfert dans le temps. 64
4.3 Illustration des deux types de Fading. 65
4.4 Système de transmission sans fil MIMO. 66
4.5 Capacité ergodique pour différentes configurations du canal MIMO (Courbes reproduites de la référence [5]). 68
4.6 Schéma d'un codeur d'Alamouti $N_t = 2$ et $Q = T = 2$. 71
4.7 Synoptique d'un schéma de multiplexage spatial. 71
4.8 Schéma bloc de la chaîne de transmission à l'émission. 74
4.9 Schéma bloc de la chaîne de transmission à la réception. 75
4.10 Comparaison des performances en terme de BER entre le code orthogonal d'Alamouti et le schéma de multiplexage spatial MUX. 77
4.11 Comparaison des performances en terme de FER entre le code orthogonal d'Alamouti et le schéma de multiplexage spatial MUX. 78
4.12 Comparaison des performances en terme de BER entre les codes convolutifs et les codes LDPC pour une diversité temporelle $N_c = 1$. 80
4.13 Comparaison des performances en terme de FER entre les codes convolutifs et les codes LDPC pour une diversité temporelle $N_c = 1$. 80
4.14 Comparaison des performances en terme de BER entre les codes convolutifs et les codes LDPC pour une diversité temporelle $N_c = 8$. 81
4.15 Comparaison des performances en terme de FER entre les codes convolutifs et les codes LDPC pour une diversité temporelle $N_c = 8$. 81

5.1 GUI n°I : L'interface graphique principale du logiciel LDMO. 84
5.2 GUI n°I.1 : Performances des codes LDPC. 85
5.3 GUI n°I.1.1 : Codage et décodage LDPC. 86
5.4 Le diagramme de Tanner sous le logiciel LDMO. 87
5.5 (a) Le message d'attente affiché durant les simulations. (b) Le message d'erreur. . 87
5.6 GUI n°I.3 : Simulations : (a) GUI n°I.3.1 : selon la taille N. (b) GUI n°I.3.2 : selon le nombre d'itérations. (c) GUI n°I.3.3 : selon le rendement de codage. 88
5.7 Exécution d'une des simulation sous le langage C. 90
5.8 GUI n°I.2 :Application des codes LDPC dans un système MIMO. 91
5.9 (a) GUI n°I.3 : Bibliographie. (b) GUI n°I.3.1 : Sur les codes LDPC. (c) GUI n°I.3.2 : Sur les systèmes MIMO. 92
5.10 (a) Le menu Help. (b) About. 92
5.11 Document d'aide donné en PDF. 93

B.1 Décomposition SVD du canal MIMO. 100

C.1 Block diagram of the receiver. 104

Liste des tableaux

2.1 Addition et multiplication dans le corps de Galois F_2. 15

3.1 La dépendance entre les bits de données et les bits de contrôle de parité. 32
3.2 Les liens entre les contraintes de parité et les bits de données. 33
3.3 Illustration de la deuxième étape de l'algorithme BP dans le cas d'un décodage Hard. 36
3.4 Illustration de la troisième étape de l'algorithme BP dans le cas d'un décodage Hard. 37
3.5 Liste des algorithmes de décodage couramment utilisés dans la littérature. 43
3.6 Exemples de techniques de codage utilisées par quelques standards de télécommunication. 46
3.7 Les paramètres de simulation utilisés pour évaluer les performances des codes LDPC. 53

4.1 Les paramètres de simulation utilisés pour évaluer les performances des codes LDPC dans un système MIMO. 76

Liste des abréviations

APP	A posteriori Probability
AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hocquenghem
BEC	Binary Erasure Channel
BER	Bit Error Rate
BLAST	Bell Layered Space Time
BP	Belief Propagation
BPSK	Binary Phase Shift Keying
BSC	Binary Symmetric Channel
CCSDS	The Consultative Committee for Space Data Systems
CNRS	Centre National de Recherche Scientifique
DVB-S2	2nd Generation Digital Video Broadcast Spatial
DVB-T	Digital Video Broadcast Terrestrial
ENSPM	Ecole Nationale Supérieure de Physique de Marseille
FER	Frame Error Rate
FSO	Free Space Optic
GSM	Groupe Signaux Multidimensionnels
GUI	Graphic User Interface
i.i.d	indépendant identiquement distribué
IEEE	Institute of Electrical and Electronics Engineers
LD	Linear Dispersion
LDPC	Low Density Parity Check
LLR	Log Likelihood Ratio
MAP	Maximum A Posteriori
MIMO	Multiple Input Multiple Output
MIT	Massachusetts Institute of Technology
MMSE	Minimum Mean Square Error
MP	Message Passing
MUX	Schéma de multiplexage spatial
NRNSC	Non Recursive Non Systematic Code
NSC	Non Systematic Code
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying

RS	Reed Solomon
RSC	Recursive Systematic Code
SISO	Single input Single Output
SNR	Signal to Noise Ratio
SP	Sum Product
STBC	Space Time Block Code
STC	Space Time Code
SVD	Singular Value Decomposition
TF	Transformée de Fourrier
WiFi	Wireless Fidility
WIMAX	Worldwide Interoperability for Microwave Access
UMR	Unité Mixte de Recherche

Liste des notations

$sign(.)$	La fonction signe
$abs(.)$	La valeur absolue
$(.)^t$	La transposée d'un vecteur ou d'une matrice
$(.)^H$	L'opérateur Hermitien c.à.d le conjugué de la transposée d'une matrice
$(.)^*$	Le conjugué
$L(.)$	Le log-rapport de vraisemblance
Π	La fonction d'entrelacement
Π^{-1}	La fonction de désentrelacement
c	Le vecteur de la séquence d'information
$M_{y\|x}$	Matrice de transition d'un canal
$\mathcal{C}$	La capacité du canal
d_s	Le débit symbole de la source
x	Le vecteur du mot de code
p	Le vecteur de parité
K	La taille de la séquence d'information
N	La taille du mot de code
M	Le nombre de bits de redondance ou de parité
R	Rendement de codage de canal
ν	Longueur de contrainte d'un code convolutif
m	Mémoire d'un code convolutif
F_q	Corps de Galois de q éléments
d_H	La distance de Hamming
w_H	Le poids de Hamming
d_{min}	La distance minimale
Z	L'ensemble des mots de code possibles
G	Matrice génératrice de code
H	Matrice de contrôle de parité
w_r	Le poids des lignes de la matrice H
w_c	Le poids des colonnes de la matrice H
q_{ij}	Le message passé d'un noeud de données x_i à un noeud de contrôle f_j
r_{ji}	Le message passé d'un noeud de contrôle f_j à un noeud de données x_i
h_c	La réponse impulsionnelle d'un canal de transmission
H_c	La fonction de transfert d'un canal de transmission

X	La matrice du code espace-temps
N	La matrice de bruit
Y	La matrice de réception
R_{STC}	Rendement de codage espace-temps
T	Le nombre d'intervalles de temps (Channel use)
N_t	Nombre d'antennes d'émission du système MIMO
N_r	Nombre d'antennes de réception du système MIMO
N_c	La diversité temporelle du canal MIMO
E_b/N_0	Le rapport de l'énergie transmise par bit d'information sur la densité de puissance du bruit
E_s/N_0	Le rapport de l'énergie transmise par symbole d'information sur la densité de puissance du bruit

Introduction générale

En 1948, Claude Shannon a montré qu'il existe une limite au débit d'information transmis en présence de bruit , appelée la capacité du canal, mais il n'a pas explicité les moyens de l'approcher. Même si le caractère asymptotique de cette limite ne laisse aucun espoir de l'atteindre, la communauté de la théorie de l'information ne cesse d'effectuer des recherches afin de pouvoir atteindre cette limite. Après plus de 40 ans de recherche, où "des demi-vérités ont fini par devenir des dogmes", C. Berrou, A. Glavieux et P. Thitimajshima ont montré comment réussir à s'approcher de cette fameuse limite avec une complexité raisonnable, et cela, par l'application d'un principe analogue à celui de la contre-réaction bien connu en électronique. Ce principe, baptisé "Principe Turbo" par analogie aux moteurs en mécanique, consiste à réinjecter à l'entrée du système une partie de l'information de sortie [6].

Cette révolution a ouvert de nombreuses voies de recherche dans le domaine du codage correcteur d'erreurs (codage de canal) et plus globalement dans les systèmes de communications numériques. Cette avancée a eu pour conséquence la redécouverte de la classe des codes linéaires en bloc LDPC (Low Density Parity Check en anglais), introduits initialement par Gallager en 1962. A cette période, les travaux de Gallager sur ces codes n'avaient pas suscité d'engouement, une raison communément admise pour expliquer cet oubli, est la difficulté pour l'époque de concevoir des circuits performants permettant de traiter les algorithmes décrits. En 1995, encouragé par le contexte qui a suivi la découverte du principe turbo, MacKay redécouvre les codes LDPC. Par la suite, de nombreux travaux ont été effectués sur cette famille de codes, appelés aussi *technique avancée de codage*.

En 2004, les codes LDPC ont été introduits pour la première fois dans la norme de télédiffusion numérique par satellite (DVB-S2), et à partir de cette année, les applications de ces codes ne cessent d'augmenter, surtout dans les communications sans fil.

La caractéristique essentielle des futurs systèmes de télécommunications sans-fil, est l'évaluation vers les débits élevés tout en envisageant des mobilités de plus en plus importantes des usagers. Face aux imperfections du canal radio mobile, et en plus d'un code correcteur d'erreur

puissant, l'usage des réseaux d'antennes et des techniques de traitement d'antennes s'avère très efficace pour préserver la qualité de transmission des données. En particulier, une technique récente consiste à utiliser des antennes multiples à l'émission et à la réception, connue sous le nom de systèmes MIMO (Multiple Input Multiple output en anglais).
Introduits par Telatar en 1995, les systèmes MIMO permettent d'atteindre des taux de transmission très élevés, de ce fait, ils sont considérés avec les codes LDPC comme l'une des voies les plus prometteuses pour les nouvelles générations de communications mobiles, à savoir la $4^{ème}$ génération de téléphonie mobile et les prochaines normes du standard WIMAX. L'objet de ce travail est donc d'étudier les performances des codes LDPC, puis les appliquer dans un système MIMO, et enfin, voir l'apport d'un code correcteur d'erreur puissant par rapport à un code classique.

Organisation du document

Ce mémoire est composé de cinq chapitres. Le détail de chacun des chapitres est décrit ci-dessous.

Dans le **chapitre 1**, nous présenterons brièvement l'Institut Fresnel, le laboratoire au sein duquel j'ai effectué mon stage de fin d'études.

Le **chapitre 2** est consacré à l'introduction des concepts généraux liés à la théorie d'information et au codage de canal.

Après avoir introduit brièvement le rôle de chaque partie de la chaîne de communication numérique, le théorème fondamental du codage de canal ainsi que les classes de codes correcteurs seront présentés. Ensuite, nous détaillerons un peu la famille des codes convolutifs et les codes en bloc (la famille à laquelle les codes LDPC appartiennent). A la fin, on conclut ce deuxième chapitre par une comparaison de performances entre les codes correcteurs les plus connus.

Dans le **chapitre 3**, nous entamerons le coeur du sujet qui est l'étude des codes LDPC. Pour cela, nous commencerons par un bref historique sur les codes LDPC, ensuite, nous présenterons les différentes classes de codes LDPC ainsi que leurs représentations matricielle et graphique. Ensuite, une étude détaillée sur les algorithmes de décodage et d'encodage sera présentée. A la fin de ce chapitre, nous illustrons les résultats de simulations ainsi que les conclusions tirées.

L'objectif du **chapitre 4** est d'implémenter un codeur/décodeur LDPC (dont les paramètres des codes sont sélectionnés à partir des résultats de l'étude présentée dans le chapitre 3) dans un système MIMO et voir ce qu'il apporte par rapport au codeur/décodeur convolutif classique. Pour cela, nous allons définir d'abord le canal radio mobile, en précisant ses imperfections, puis,

nous introduirons brièvement les systèmes MIMO et les gains apportés par rapport au cas d'un système mono-antenne à l'émission et à la réception. Ensuite, nous présenterons le contexte de l'application. En fin, nous donnerons quelques résultats de simulations et nous verrons les améliorations apportées par l'introduction des codes LDPC.

Le **chapitre 5** est, quant à lui, consacré à la présentation du logiciel LDMO que nous avons réalisé avec l'outil 'GUIDE' du logiciel MATLAB et l'éditeur/compilateur C Code Blocks.
Ce logiciel permet ainsi de compléter le travail théorique en fournissant un outil de simulation simple à utiliser, et une base pour ceux désirant continuer sur les codes LDPC et les systèmes MIMO. Finalement, nous signalons que la conception reste ouverte à des améliorations afin d'augmenter et d'optimiser les performances de ce petit logiciel.

Nous terminerons ce travail par une conclusion générale qui passera en revue tout ce qui a été abordé dans ce mémoire ainsi que les perspectives des travaux futurs.

Chapitre 1

Présentation du laboratoire

Durant ces dernières années, l'environnement scientifique français a connu plusieurs développements et progressions dans le domaine des télécommunications et du traitement du signal, et cela, par l'ouverture de nouveaux pôles de recherche comme *l'Institut Fresnel*.

Dans ce premier chapitre, nous allons présenter brièvement ce laboratoire. Puis, nous présenterons l'équipe GSM (Groupe Signaux Multidimensionnels), l'équipe au sein de laquelle j'ai effectué mon stage de fin d'études.

1.1 Introduction

A l'origine, Marseille comptait des laboratoires d'excellence, dont le laboratoire d'Optique des Surfaces et des Couches Minces, Le laboratoire Signal et Image, et le laboratoire d'Optique Électromagnétique. C'est le regroupement de ces trois entités qui a conduit à la création de l'institut Fresnel [7].

FIGURE 1.1 – L'institut Fresnel et ses tutelles.

Le regroupement de ces laboratoires nécessitait, au préalable, la conciliation de différentes cultures pour préserver les passerelles entre recherches fondamentales et applications industrielles, sur la base d'un équilibre Théorie - Expérience - Applications. Concrètement, la mission première de l'institut est d'inventer et de réaliser des micro ou macro-objets ou distributions d'objets capables de contrôler la lumière et les ondes. Et l'autre mission concerne le traitement des images et des signaux [8].

1.2 Historique du laboratoire

L'historique de l'institut peut se résumer sur deux périodes marquantes :

i. Le premier quadriennal 1999-2002 : Dans cette période, le laboratoire a eu l'occasion de :

- S'ancrer dans l'environnement de la recherche académique, national et européen.
- Dynamiser la recherche partenariale en Région (4 start-up, transferts de technologie).
- Avoir une forte production scientifique (272 articles dans des revues internationales).
- Favoriser l'émergence de jeunes talents (médaille de bronze CNRS, prix du doctorant de la ville de Marseille).
- Participer à la stratégie scientifique, politique et économique des établissements et institutionnels d'Aix-Marseille.

A l'issue de cette période, l'UMR (Unité Mixte de Recherche) était ainsi positionnée comme l'un des barycentres naissants de l'optique, l'imagerie, traitement du signal et la photonique françaises, avec une production scientifique très solide.

ii. Le second quadriennal 2003-2006 : C'est dans un contexte différent que le deuxième quadriennal a été abordé, avec comme priorités, la confirmation et la stabilisation de la place de l'UMR dans la compétition internationale, et la professionnalisation du mode de fonctionnement interne. Il s'agissait en effet de positionner définitivement l'institut dans le cercle des grands laboratoires français de la Photonique, de l'Image et des télécommunications, tout en faisant face à une croissance forte en effectifs et en flux financiers.
L'objectif majeur a donc été le site unique, qui faisait défaut à l'institut depuis sa création, et dont l'absence constituait un frein aux synergies scientifiques internes et à l'optimisation des moyens logistiques. Ce dossier a été concrétisé en Juillet 2005, grâce au soutien des Universités et du Rectorat, qui ont conféré au laboratoire le bénéfice d'un site unique dans l'ex-bâtiment ENSPM [1].

1. ENSPM : Ecole Nationale Supérieure de Physique de Marseille, aujourd'hui intégrée à l'école Centrale de Marseille.

1.3 Organigramme

L'organisation de l'institut Fresnel est donnée par l'organigramme suivant :

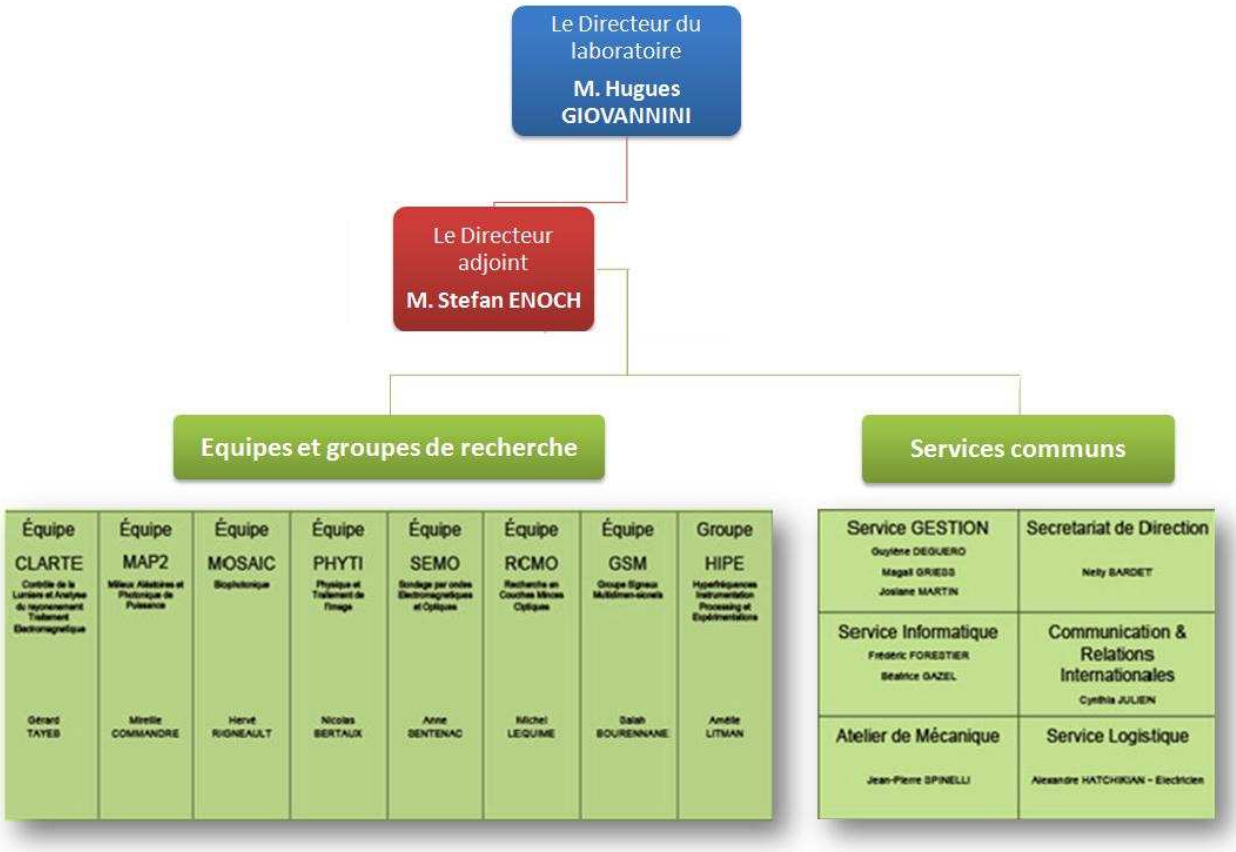


FIGURE 1.2 – L'organigramme de l'institut Fresnel.

1.4 Le personnel

L'Institut Fresnel accueille au quotidien 140 personnels, qui se consacrent à des travaux de recherche dans les domaines de la Photonique, de l'Electromagnétisme, du Traitement des Signaux et des Images. Une majorité de ces personnels est impliquée dans les enseignements dans les établissements marseillais d'enseignement supérieur : Université Paul Cézanne, Ecole Centrale et l'Université de Provence (les tutelles de l'institut Fresnel déjà mentionnés dans la Figure 1.1).

1.5 Les équipes de recherche

L'Institut comporte sept équipes et un groupe de recherche, dont les noms sont indiqués dans la figure ci-dessous :

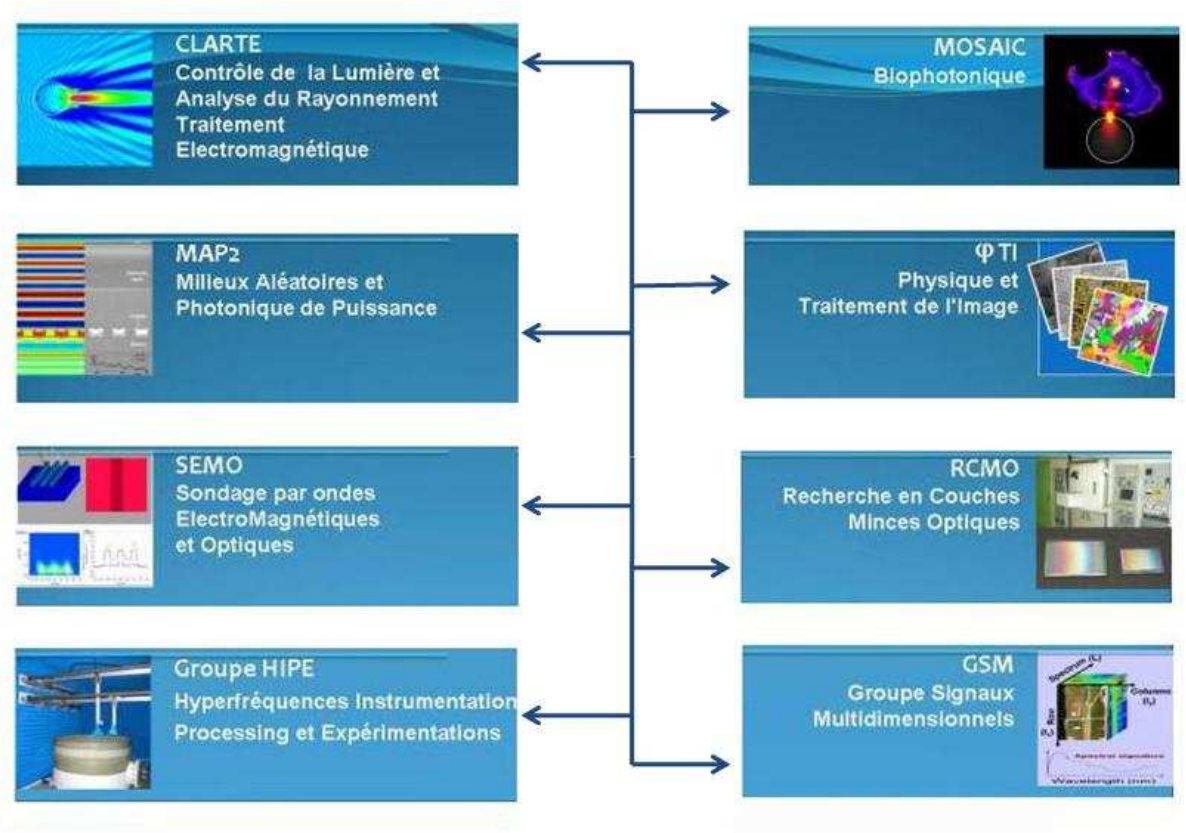


FIGURE 1.3 – Les équipes de recherche de l'institut Fresnel.

1.6 les publications

Selon les données présentées dans la référence [8], l'Institut Fresnel a publié :

- 604 articles dans des revues avec comité de lecture.
- 6 articles dans des revues sans comité de lecture.
- 151 conférences invitées dans des congrès internationaux.
- 10 ouvrages scientifiques.
- 25 thèses.
- 19 brevets.
- 4 Start-up : SIMAG, Silios Technologies, High Wave Marseille , PHLOX.

1.7 Le Groupe Signaux Multidimensionnels

Le traitement du signal multidimensionnel est essentiel pour l'avancée d'un grand nombre de domaines scientifiques et techniques, relevant aussi bien de l'ingénierie des systèmes, que de l'observation des phénomènes naturels. L'équipe GSM de l'institut Fresnel développe des thèmes relevant directement du traitement du signal multidimensionnel [9]. Ses activités de recherche ont pour principaux objectifs les développements théoriques et applications de méthodes de traitement de données multidimensionnelles, favorisées par l'émergence de nouveaux capteurs (multi-spectraux, multitemporels...). Ces données multidimensionnelles mettent en exergue certaines carences méthodologiques des techniques actuelles, que l'équipe participe à résoudre.

1.7.1 Organisation

L'organisation de l'équuipe GSM est donnée par l'organigramme suivant :

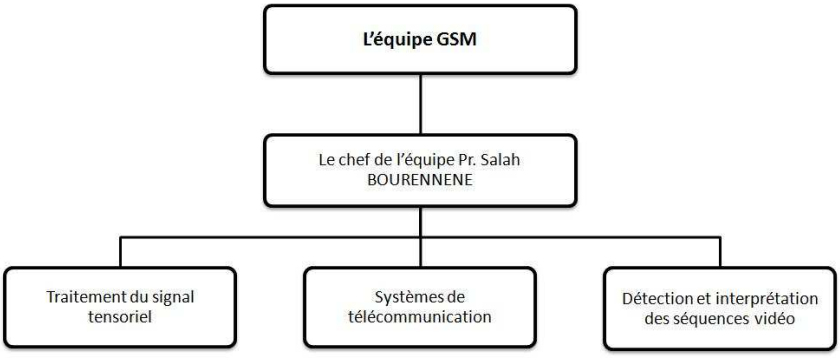


FIGURE 1.4 – l'organigramme de l'équipe GSM.

1.7.2 Les thèmes de recherche

L'équipe GSM s'intéresse à quatre thèmes de recherche, qui ont volontairement des intersections importantes tant sur les plans de la méthodologie que de la mise en œuvre :

- Traitement du signal tensoriel.
- Traitement statistique du signal.
- Systèmes de télécommunications : Optique en espace libre (FSO) et les systèmes MIMO envisagés pour la prochaine norme WIMAX et la 4^{eme} génération de téléphonie mobile.
- Traitement du signal multidimensionnel : Imagerie médicale, imagerie hyper-spectrale, imagerie satellitaire, astrophysique.

1.8 Le plan de travail du stage pratique

Le plan de travail suivi pendant les six mois de stage de fin d'études au sein de l'équipe GSM, est résumé dans le schéma suivant :

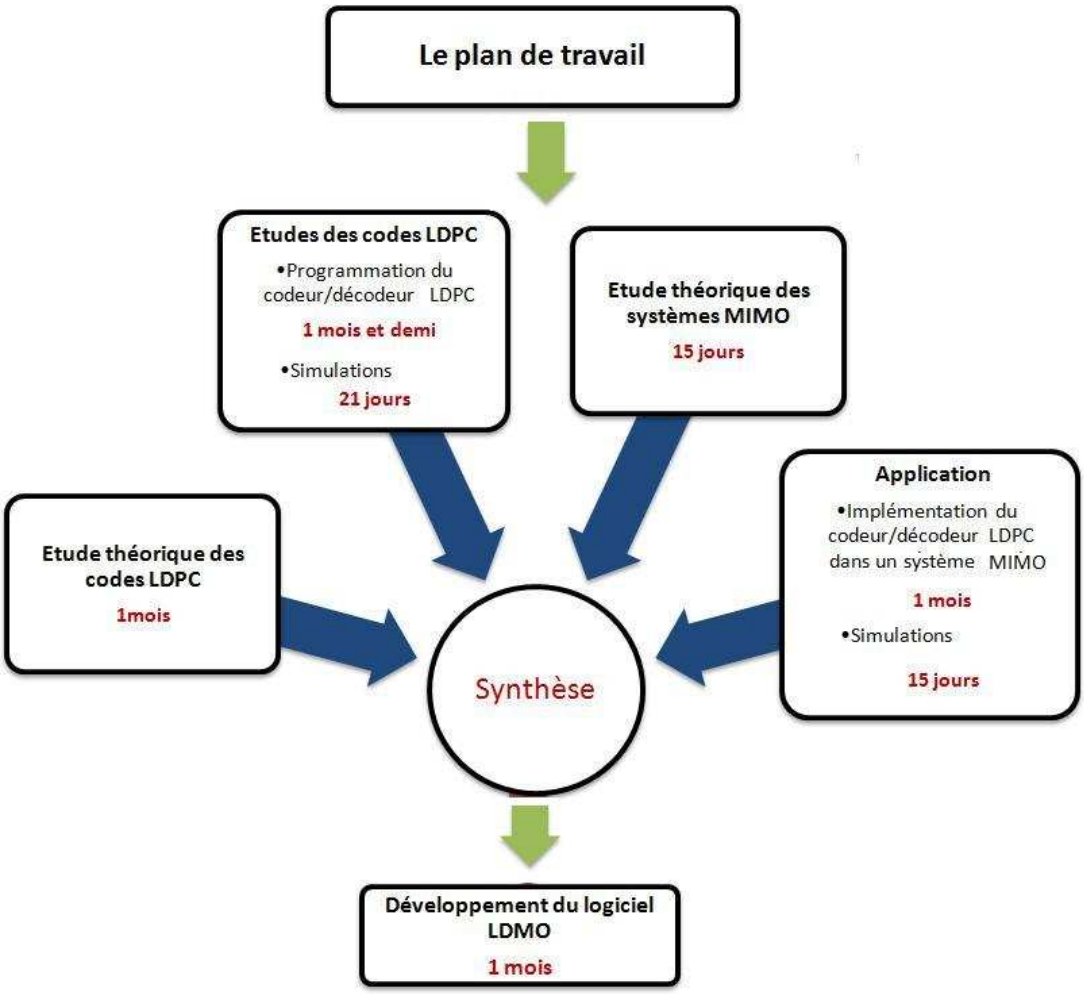


FIGURE 1.5 – Le plan de travail du stage pratique.

1.9 Conclusion

Pour conclure ce premier chapitre, on peut dire que l'institut Fresnel représente un pôle de recherche important dans l'environnement scientifique français, et cela, à cause du travail sérieux de toutes ses équipes de recherche, spécialement, l'équipe GSM qui travaille sur des sujets d'actualité dans le domaine des signaux multidimensionnels.

Chapitre 2

Introduction au codage de canal

Ce chapitre a pour objectif de présenter des principes et des concepts fondamentaux qui nous seront utiles pour la suite.

Dans un premier temps, nous donnerons quelques rappels sur la théorie d'information et la théorie de codage. Puis, nous citerons les différentes catégories de codes correcteurs. Notre étude sera focalisée par la suite, sur la famille des codes convolutifs et la famille des codes en bloc. En fin, on conclut ce chapitre par une comparaison des performances entre les codes correcteurs d'erreurs les plus connus.

2.1 Introduction

L'objectif fondamental d'un système de communication, est de reproduire en un point de la chaîne de communication un message transmis à partir d'un autre point, mais le problème qui se pose, est que le canal est généralement soumis à des perturbations, telles que le bruit et les interférences. Alors, comment faire pour établir une communication fiable dans ces conditions ?

Les premières tentatives d'évaluations de performance datent des contributions de Nyquist en 1924 [10] et de l'ingénieur Américain R.Hartetly en 1928 [11], mais l'étape décisive fut franchie en 1948 par Claude E. Shannon, lorsque parut son article fondateur de la théorie de l'information, intitulé « *The Mathematical Theory of Communication* » [12].

À la croisée de la théorie de l'information, des mathématiques et de l'électronique, le codage correcteur d'erreurs (codage de canal) a connu de nombreux développements depuis les travaux fondateurs de Shannon. Du simple code de Hamming (1950) aux récents turbocodes (1993) en passant par les codes LDPC (1962), le codage de canal a considérablement évolué et a intégré des concepts de plus en plus sophistiqués, en particulier le traitement probabiliste de l'information.

2.2 Généralités

2.2.1 La chaîne de communication numérique

On désigne par *le paradigme de Shannon* (voir Figure 2.1) le schéma fondamental d'une communication numérique. Une *source* engendre un *message* à l'intention d'un *destinataire*. La source et le destinataire sont deux entités séparées, éventuellement distantes, qui sont reliées par un canal qui est le support de communication d'une part, mais qui d'autre part est le siège des perturbations [13].

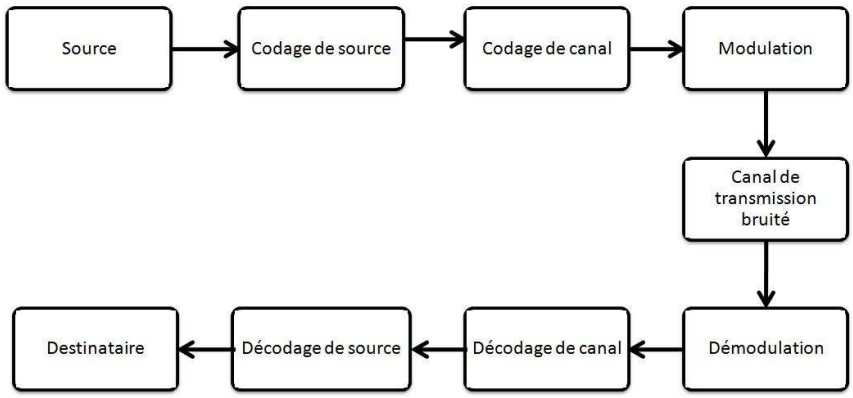


FIGURE 2.1 – Schéma fondamental d'une communication numérique : *le paradigme de Shannon*.

2.2.2 Le codage de source

«...Le message codé est dépourvu de redondance, bien que le message initial provienne d'une source redondante...» Gérard Bettail [14].

Le codage de source vise à la concision maximale. L'usage d'un canal coûte d'autant plus cher

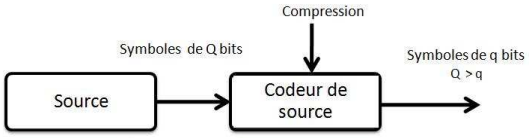


FIGURE 2.2 – Schéma simplifié d'un codeur de source.

que le message est long, le verbe "coûter" devant s'étendre ici en un sens très général, celui d'exiger l'emploi de ressources limitées telles que le temps, la puissance et la bande passante. Pour diminuer ce coût, on cherche à représenter le message avec le moins de bits possibles, c'est-à-dire le compresser. Pour ce faire, on essaye d'éliminer la redondance contenue dans le message transmis par la source [15].

Un point essentiel dans le codage de source est le critère de fidélité. Ce critère varie selon l'application. Les applications où la compression de données doit se faire sans perte, utilisent un codage de source appelé *réversible*[1], tandis que, les applications où les pertes sont tolérables, utilisent un codage dit *non-réversible*[2] [16].

2.2.3 Le codage de canal

«...Le message est restitué sans erreurs après le décodage, bien que le message codé soit reçu à travers un canal bruité...» Gérard Bettail.

La finalité du codage de canal est de protéger le message contre les perturbations du canal, et cela, en introduisant une redondance à l'information utile dans le message à transmettre. La redondance et l'information utile sont liées par une loi donnée. A la réception, le décodeur de canal exploite la redondance produite par le codeur dans le but de détecter, puis de corriger si c'est possible les erreurs introduites lors de la transmission [14]. Ce point sera détaillé encore plus dans les sections suivantes.

2.2.4 La modulation

La modulation consiste à effectuer un codage dans l'espace euclidien, espace généralement adapté aux canaux rencontrés en pratique [1]. Pour une modulation M-aire, on associe à chaque mot de L bits un signal x_i(t), i= 1,..., M de durée T choisi parmi les M= 2^L signaux. Quant à la démodulation, son rôle est d'extraire les échantillons et de décider en faveur des symboles les plus probablement émis [17]. Les données fournies par l'unité de démodulation seront traitées

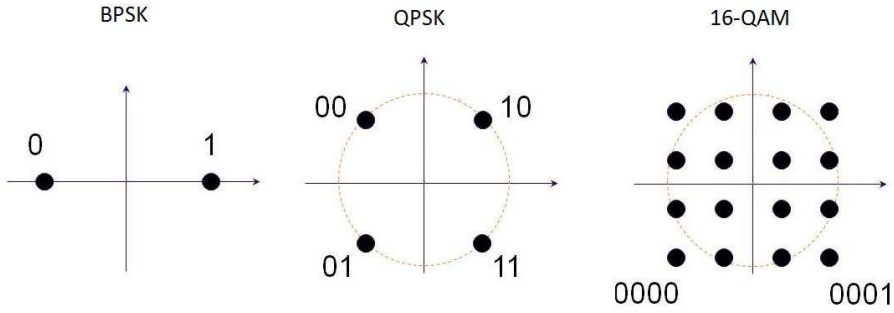


FIGURE 2.3 – Exemples de modulations numériques.

par ce que l'on appelle *le décodeur*. On distingue deux types de décodeurs, le premier est appelé

1. Réversible : le message envoyé sera exactement restitué au niveau du récepteur (sans perte ou distorsion).
2. Par exemple, dans la norme MPEG4, la compression des données multimédia est faite avec une distorsion tolérable par l'observateur (critère de fidélité), et cela à cause des défauts de l'oeil (persistance rétinienne) et de l'oreille humaine (l'effet de masquage).

décodeur à décision dure (Hard decision), car il fonctionne à partir des données fermes ('0' ou '1'). Le second type est appelé *décodeur à décision pondérée (Soft decision)*, car le démodulateur fournie au décodeur une valeur ferme accompagnée d'une mesure de fiabilité [18].

2.2.5 Le canal de communication

Le canal de communication est le support physique permettant d'acheminer un message entre une source et un ou plusieurs destinataires. Il existe plusieurs type de canaux, mais en théorie d'information, les canaux les plus utilisés sont appelés *canaux discrets*[3].

Un canal discret est un système stochastique acceptant en entrée des suites de symboles définies sur un alphabet $\mathcal{X}$, et émettant en sortie des suites de symboles définies sur un alphabet de sortie $\mathcal{Y}$, reliés par une loi de transition $P_{Y|X}$ i.e. une matrice stochastique $M_{Y|X}$.

$$M_{Y|X} = \begin{bmatrix} P(y_1|x_1) & \cdots & P(y_j|x_1) \\ \vdots & \ddots & \vdots \\ P(y_1|x_k) & \cdots & P(y_j|x_k) \end{bmatrix} \tag{2.1}$$

Le canal sera donc défini par :

$$\tau = (\mathcal{X}, \mathcal{Y}, M_{Y|X}) \tag{2.2}$$

Le canal est dit *sans mémoire* si le symbole courant de sortie ne dépend que du symbole courant d'entrée et il ne dépend pas des précédents ni des suivants.

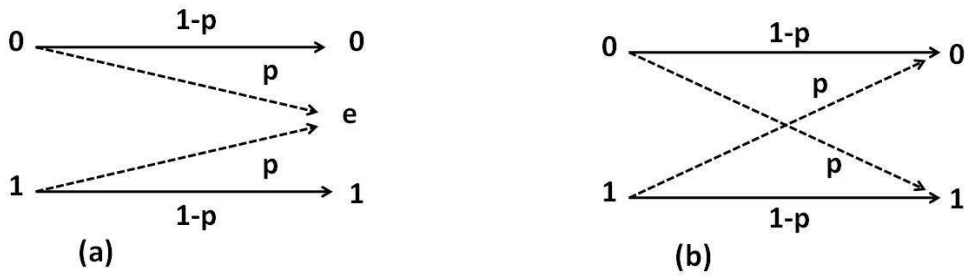


FIGURE 2.4 – (a) Le canal binaire avec effacement BEC. (b) Le canal binaire symétrique BSC.

3. Pour plus de détails sur les autres types de canaux, le lecteur peut se reporter aux références suivantes : [1], [17], [18] et [19].

Les canaux binaires BSC et BEC

Le canal *binaire symétrique* (BSC), présenté par le schéma de droite de la Figure 2.4, est le canal le plus simple qu'on puisse imaginer. Ce canal est caractérisé par des alphabets d'entrée et de sortie binaires, une probabilité d'erreur p et une matrice de transition $M_{Y|X}$ donnée par la relation (2.3). L'autre canal, présenté par le schéma de gauche de la Figure 2.4, est appelé *canal binaire avec effacement* (BEC). Ce canal est binaire à l'entrée, mais ternaire en sortie (aux deux symboles notés '0' et '1' est adjoint un troisième, noté e). Ce canal est caractérisé par une probabilité d'erreur p et une matrice de transition $M_{Y|X}$ donnée par la relation (2.4).

$$M_{Y|X(BSC)} = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix} \tag{2.3}$$

$$M_{Y|X(BEC)} = \begin{bmatrix} 1-p & p & 0 \\ 0 & p & 1-p \end{bmatrix} \tag{2.4}$$

Le canal à bruit Blanc Additif et Gaussien

Parmi les canaux stationnaires le plus utilisé, celui sur lequel l'évaluation des performances des systèmes de communications est aussi la plus simple, est le canal à bruit Blanc Additif et Gaussien ou AWGN (Additive White Gaussian Noise en anglais). Ce canal de transmission est rencontré dans les transmissions par faisceaux hertziens à faible débit ou dans les liaisons entre des satellites ou des sondes spatiales et des stations terriennes, c'est l'ensemble des transmissions radio électriques en espace libre.

Le bruit introduit dans ce canal est modélisé par un signal aléatoire $n(t)$, dont la distribution de probabilité suit la loi Gaussienne :

$$f_N(n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(n-\mu)^2}{2\sigma^2} \tag{2.5}$$

où $\mu = E\{n(t)\} = 0$ et $\sigma^2 = E\{[n(t)-\mu]^2\} = E\{n(t)^2\}$ représentent respectivement la moyenne et la variance.

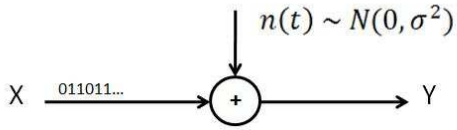


FIGURE 2.5 – Canal à entrée binaire perturbé par l'addition d'un bruit gaussien.

2.2.6 La capacité d'un canal

On définit la capacité d'un canal comme le *maximum de l'information mutuelle moyenne* $I(X;Y)$, elle représente donc la plus grande quantité d'information pouvant transiter entre l'émetteur et le récepteur [17], comme le montre la Figure 2.6.

$$C \triangleq \max_{P(X)} I(X;Y) \tag{2.6}$$

L'information mutuelle $I(X;Y)$ est définie par la relation suivante :

$$I(X;Y) = (H(Y) - H(Y|X)) \tag{2.7}$$

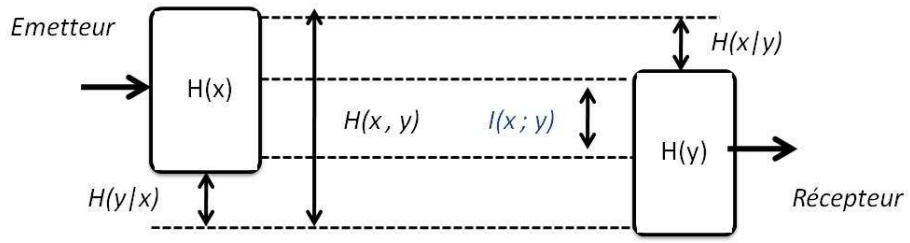


FIGURE 2.6 – Schéma représentatif de l'information mutuelle I(X ;Y).

On désigne par $H(X)$ l'entropie de la source qui représente la surprise moyenne ou la quantité d'information délivrée par une source d'information. Par contre, $H(X|Y)$ représente l'information requise pour supprimer l'ambiguïté sur l'entrée.

$$H(X) = -\sum_{i}^{N} p_i \log_2 p_i \tag{2.8}$$

p_i : la probabilité d'apparition des lettres de l'alphabet de la source.

La capacité $\mathcal{C}$ s'exprime en *Shannon par symbole* ou *bit par symbole* . Il est également possible de l'exprimer en *Shannon par seconde*, on parle alors de capacité par unité de temps [1]. Pour la distinguer de la capacité par symbole, on trouve généralement dans la littérature la notation suivante :

$$\mathcal{C}_s = \mathcal{C}.d_s \tag{2.9}$$

où d_s représente le débit symbole de la source.

Si on considère le cas d'un canal à entrée binaire perturbé par l'addition d'un bruit balnc et gaussien (AWGN) de densité spectrale N_0, et on suppose que les signaux occupent une bande

passante B et que leur puissance reçue est P. La capacité de ce canal en Shannon par symbole est donnée par la relation suivante :

$$\mathcal{C}_{AWGN} = \frac{1}{2}\log_2(1 + \frac{P}{N}) \tag{2.10}$$

où $N = N_0 B$ est la puissance du bruit [4] . L'allure de la capacité C en fonction du rapport signal sur bruit SNR est illustré dans la Figure 2.7.

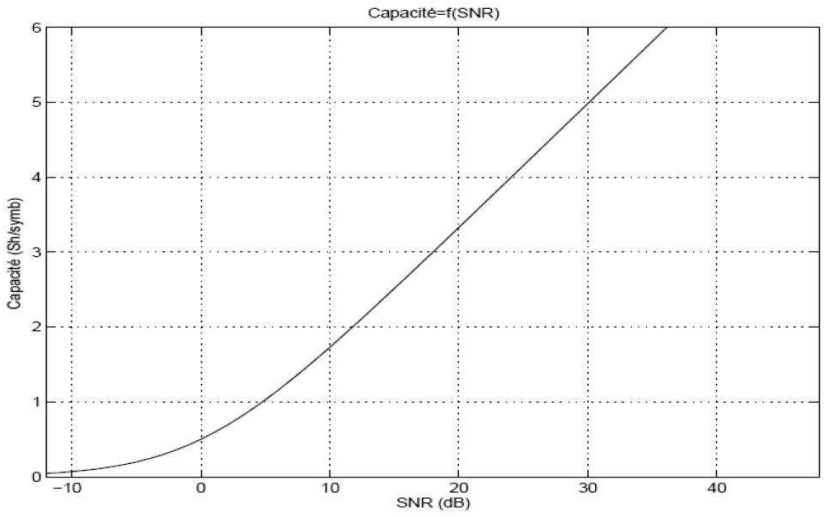


FIGURE 2.7 – Variation de la capacité du canal AWGN en fonction du SNR.

2.2.7 Le théorème fondamental du codage de canal

En 1948, Shannon a annoncé dans son article intitulé «*A Mathematical Theory of Communication*» [12], le théorème fondamental de la théorie de l'information :

«*Pour une source à débit d'information R_D et un canal de capacité $\mathcal{C}$, si $R_D < \mathcal{C}$, il existe un code ayant des mots d'une longueur N tel que sa probabilité d'erreur soit arbitrairement petite* ».

Ce théorème affirme l'existante de codes dont la probabilité de décodage erroné est arbitrairement petite, mais ne montre pas comment ces codes peuvent être construits. Ce théorème affirme une chose tout à fait surprenante, à savoir que, quelque soit le niveau des perturbations d'un canal, on peut toujours y passer des messages avec une probabilité d'erreur aussi faible que l'on veut. Ce théorème a causé un énorme développement dans la théorie de codage de canal [19].

4. La démonstration complète de la relation (2.10) est disponible dans la référence [17].

2.3 Les codes correcteurs d'erreurs

Il existe une grande variété de codes correcteurs d'erreurs, dont les performances et les applications sont variables. Mais, le principe de base reste le même : ajouter de la redondance intelligemment et utiliser cette surinformation pour déterminer la fiabilité du message (détection d'erreur), puis, si c'est possible reconstruire le message d'origine au mieux (correction d'erreur). Mais en revanche, l'ajout de la redondance dans le message à transmettre, entraîne une perte d'efficacité du système. En effet, les bits de redondance introduits ne véhiculent pas de l'information utile. Cependant, cette perte est à mettre en balance avec le gain de qualité obtenu par l'utilisation du codage [20].

2.3.1 Définitions et notations

• Le codeur de canal permet de générer un *mot de code* x de N bits à partir d'un mot d'information c de K bits. Ce code engendre donc M bits de redondance, avec $M = N - K$, appelés *bits de parité*, que nous noterons par le vecteur p (voir la Figure 2.8)

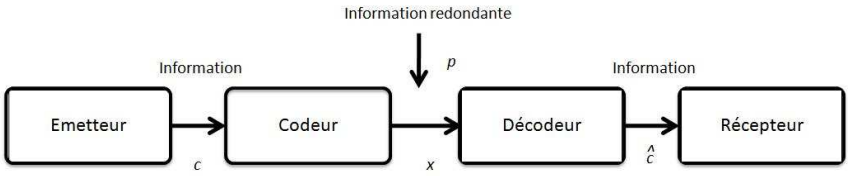


FIGURE 2.8 – Schéma simplifié d'un codeur/décodeur de canal.

• Un code est dit *systématique* si les symboles de c apparaissent explicitement dans x. On appelle aussi *rendement de codage* R, le rapport entre le nombre de bits d'information et le nombre de bits du mot de code transmis :

$$R = \frac{K}{N} \tag{2.11}$$

• Les symboles du message d'information c et du mot de code x prennent leurs valeurs dans un corps fini F_q à q éléments, appelé *corps de Galois* et dont les principales propriétés sont illustrées dans la référence [21]. Pour la majorité des codes, les symboles sont binaires et prennent leur valeur dans le corps F_2 à deux éléments.

Les opérations élémentaires d'addition et de multiplication dans le corps F_2 sont données dans le Tableau 2.1.

• On appelle *distance de Hamming* entre deux codes x_i et x_j, le nombre de composantes où les deux codes sont différents, on la note par $d_H(x_i, x_j)$. On appelle *poids de Hamming*, noté

a	b	a $\oplus$ b	a $\otimes$ b
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

TABLE 2.1 – Addition et multiplication dans le corps de Galois F_2.

$w_H(x)$, le nombre d'éléments non nuls présents dans un mot de code.

• On appelle *distance minimale* d_{min} la plus petite distance de Hamming entre deux mots de code x_i et x_j.

$$d_{min} = \min_{\{x_i, x_j \in \mathcal{Z}\}} d_H(x_i, x_j) \tag{2.12}$$

où $\mathcal{Z}$ représente l'ensemble des mots de code possibles.

• Le pouvoir de détection et de correction d'un code est déterminé par sa distance minimale d_{min}. Pour détecter e erreurs pouvant intervenir dans le mot de code, il faut que :

$$d_{min} = e + 1 \tag{2.13}$$

Pour la correction de e erreurs pouvant intervenir dans le mot de code, il faut que :

$$d_{min} = 2e + 1 \tag{2.14}$$

2.3.2 Mesure des performances d'un code correcteur d'erreurs

On appelle *gain du codage*, l'écart d'énergie par bit utile entre deux systèmes pour un taux d'erreur donné (voir Figure 2.9).

Dans le cas de l'utilisation de techniques de codage avancées, l'évolution de la performance du code peut se diviser en trois régions comme l'illustre la Figure 2.9. La première région correspond à un comportement où le décodage ne converge pas pour des SNR faibles, le décodage dégrade les performances par rapport à un système non codé, on parle alors de la *région de non convergence*. A partir d'un certain rapport signal sur bruit SNR, appelé *seuil de convergence*, le décodage rentre dans une phase où la probabilité d'erreur diminue très rapidement avec le SNR, on parle de la région du *waterfall*. Enfin, il existe une région où la probabilité d'erreur diminue de manière moins rapide que la région du waterfall. Ce comportement est spécifique de la région du *plancher d'erreur* ou *error floor*.

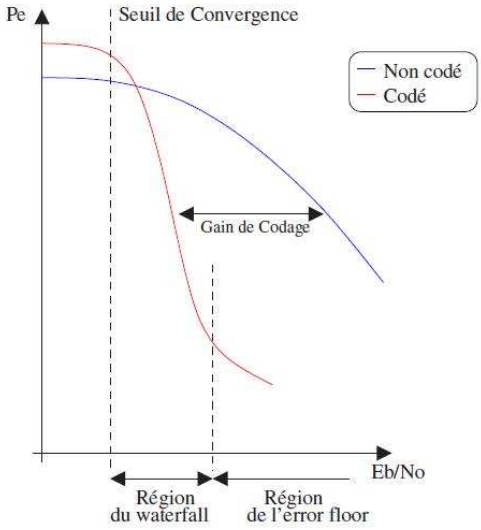


FIGURE 2.9 – Illustration des régions caractérisant les performances d'un code correcteur d'erreurs.

Remarque : Dans les systèmes de communications, le critère utilisé pour évaluer les performance est donné en taux d'erreur binaire BER (Bit Error Rate en anglais) en fonction du SNR, mais pour avoir des résultats plus exacts, surtout quand on compare des codes de rendements différents, on utilise un autre critère qui donne le BER en fonction de E_b/N_0 [1].
avec :
E_b : L'énergie transmise dans un bit d'information.
N_0 : La densité spectrale du bruit.

2.3.3 Concaténation de codes

La concaténation de codes consiste à combiner plusieurs codes élémentaires de taille raisonnable, de telle sorte que le code global (résultant) possède un pouvoir de correction élevé (d_{min} élevée) et qu'il soit aisément décodable.

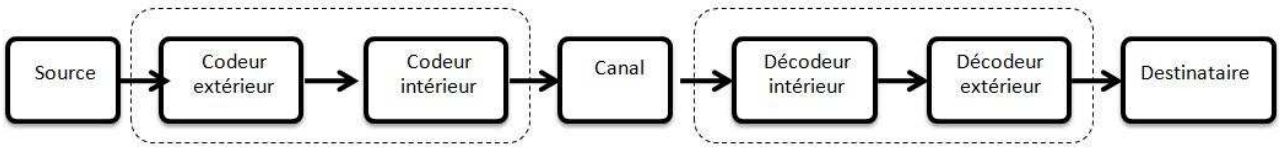


FIGURE 2.10 – Concaténation de deux codes correcteurs d'erreurs.

Le premier schéma de codage composite, appelé *concaténation de codes*, à été introduit par Forney dans son travail de thèse en 1965 (voir Figure 2.10) [22]. Ce schéma est constitué d'un premier codeur, dit *codeur extérieur*, permettant de fournir un mot de code à un deuxième codeur, dit *codeur intérieur*, pour générer un code concaténé.
Si les deux codes sont systématiques, le code concaténé est lui-même systèmatique.

2.3.4 Les classes de codes correcteurs

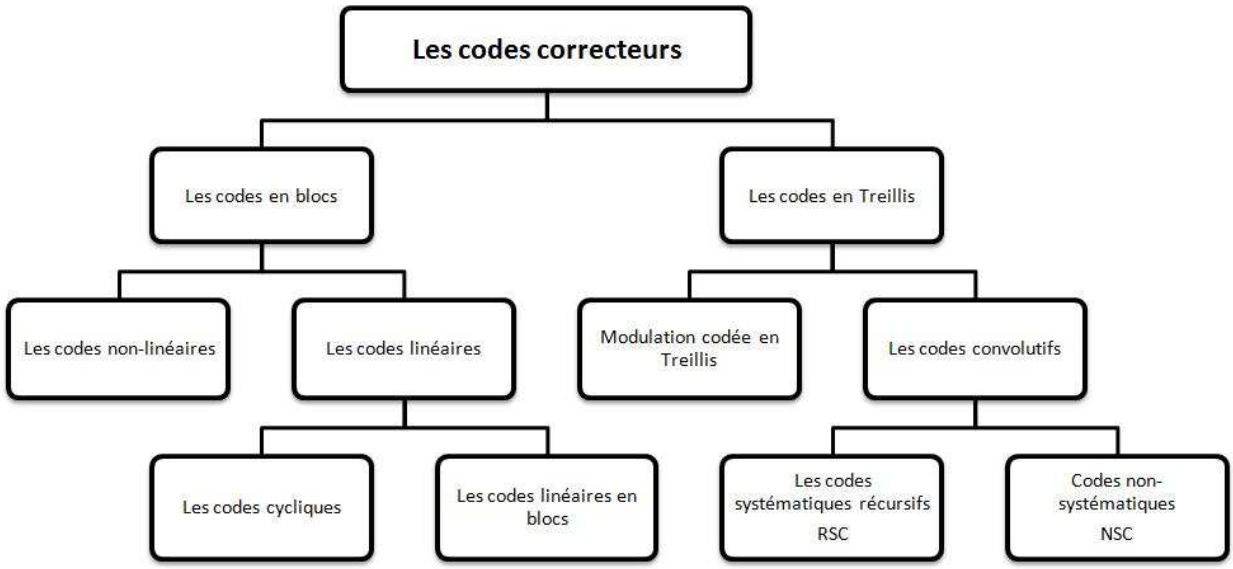


FIGURE 2.11 – Classification des codes correcteurs d'erreurs.

Un code est dit *linéaire* si la fonction de codage est une application linéaire, sinon il est dit *non-linéaire*. Lorsque les traitements requis pour obtenir les propriétés de détection ou de correction ont lieu par bloc de N symboles, on dit qu'on a affaire à un *code en bloc*. Lorsque les symboles générés par la source ne sont pas traités par des blocs, mais de manière continue, on dit qu'on a affaire à un *code convolutif*. Pour la suite de ce chapitre, nous allons parler uniquement des codes en bloc et des codes convolutifs.

2.4 Les codes en bloc

Le but de l'opération de codage en bloc est d'associer à chaque mot d'information composé de K symboles q-aire un mot de code composé de N symboles q-aire. Cette opération peut être représentée par une application g de l'ensemble F_K^q vers l'ensemble F_N^q

$$g : F_K^q \rightarrow F_N^q \qquad (2.15)$$
$$c \rightarrow x = g(c)$$

Selon le classement donné par la Figure 2.11, Les codes linéaires en bloc se divisent en deux grandes parties :

1. **Les codes linéaires en bloc** : sont ceux où les mots de code sont considérés comme étant des éléments dans un espace vectoriel.
2. **Les codes cycliques** : sont ceux où les mots de code sont considérés comme étant des éléments dans une algèbre, à savoir des polynômes [23].

Pour la suite, nous ne nous intéresserons qu'aux codes en bloc linéaires binaires ($q = 2$).

2.4.1 Les codes linéaires en bloc

Les codes linéaires en bloc sont caractérisés par une matrice G de taille (K,N) appelée *la matrice génératrice* [17]. Cette matrice transforme un message d'information c de K bits en un mot de code x de taille N ($N > K$) par l'opération matricielle suivante :

$$x = c.G \qquad (2.16)$$

avec :

$$G = \begin{bmatrix} G_1 \\ G_2 \\ \vdots \\ G_K \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ g_{K1} & g_{K2} & \cdots & g_{KN} \end{bmatrix} \qquad (2.17)$$

Chaque mot de code est une combinaison linéaire des vecteurs G_i de G. Ainsi donc, un code en bloc linéaire peut être défini comme un sous espace vectoriel à $K<N$ dimensions construit suivant la relation (2.16) [1].

Pour faciliter l'opération de codage, il est toujours possible de mettre la matrice G sous la forme systématique, en combinant les lignes entre elles [5].

$$G_{syst} = [R^t | I_K] \qquad (2.18)$$

$$I_K = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \qquad et \qquad R^t = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1M} \\ r_{21} & r_{22} & \cdots & r_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ r_{K1} & r_{K1} & \cdots & r_{KM} \end{bmatrix}$$

5. On désigne par $(.)^t$ la transposé d'une matrice ou d'un vecteur.

Les codes linéaires en bloc sont aussi caractérisés par une autre matrice H de taille (N,M) appelée *matrice de contrôle de parité* [17]. La propriété principale[6] de cette matrice est :

$$H.x^t = 0 \tag{2.19}$$

$$H.G^t = 0 \tag{2.20}$$

Dans le cas symétrique la matrice H devient comme suit :

$$H_{syst} = [I_M|R] \tag{2.21}$$

$$I_M = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \qquad et \qquad R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1K} \\ r_{21} & r_{22} & \cdots & r_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ r_{M1} & r_{M2} & \cdots & r_{MK} \end{bmatrix}$$

Une fois l'opération de codage est terminée, le message x sera transmis à travers un canal qui est généralement bruité, un bruit n s'ajoute à ce dernier. A la réception, le message reçu r sera donné par la relation suivante :

$$r = x + n = c.G + n \tag{2.22}$$

A partir de (2.22) et (2.19), on aura :

$$H.r^t = H.x^t + H.n^t = H.n^t \tag{2.23}$$

On appelle le produit $H.r^t$ *un syndrome*. Si le résultat de ce produit est un vecteur nul, alors r est normalement un mot de code, sinon le vecteur r contient des bits erronés.

Le calcul de syndrome est la méthode utilisée par la plupart des codes en bloc pour détecter la présence d'erreurs, puis en fonction de l'algorithme de décodage, corriger si c'est possible ces erreurs [23].

2.4.2 Exemples de codes en bloc

Les premiers codes en bloc sont les codes de *Hamming*, introduits en 1950 par Richard Hamming [24]. Ces codes donnaient des résultats médiocres par rapport aux critères de Varshamov et Gilbert [25], c'est la raison pour laquelle de nouveaux codes correcteurs ont été développés. On peut citer par exemple : les codes Reed-Solomon qui sont une classe particulière des codes

6. En remplaçant la relation (2.16) dans la la relation (2.19) on obtient $H.(cG)^t = H.G^t.c^t = 0$, et comme cette relation est valable pour n'importe quelle séquence d'information c, donc $H.G^t = 0$.

cycliques BCH. Ces codes, développés par I. S. Reed et G. Solomon [26], sont largement utilisés pour la correction d'erreurs groupées dans la plupart des supports de données numériques comme les CD, DVD, blu-ray Discs, et dans de nombreux standards comme DVB-T [27]. Il existe beaucoup d'autres classes de codes en bloc, qu'on ne va pas détailler dans ce manuscrit comme : les codes de Goppa [28] qui sont très utilisés dans les crypto-systèmes de McEliece et Niederreiter, les codes Reed-Muller [29], les codes Golay [30] ...etc.

La classe de codes en bloc la plus puissante jusqu'à présent, est appelée les codes LDPC (Low Density Parity Check). Cette famille représente l'objet de notre étude, qu'on présentera dans le chapitre 3.

2.5 Les codes convolutifs

Les codes convolutifs, inventés en 1954 par Peter Elias [31], constituent une famille de codes correcteurs d'erreurs, dont la simplicité de codage et de décodage sont à l'origine de leur succès. Le principe est non plus de découper le message en blocs finis, mais de le considérer comme une séquence semi-infinie $a_0a_1a_2\ldots a_n$ de symboles qui passe à travers une succession de registres à décalage, dont le nombre d'étages m est appelé *mémoire du code* et 2^m le nombre d'états possibles. La quantité $\nu = m + 1$ est appelée *longueur de contrainte du code* et le rapport $R = K/N$ est appelé *le rendement de codage*.

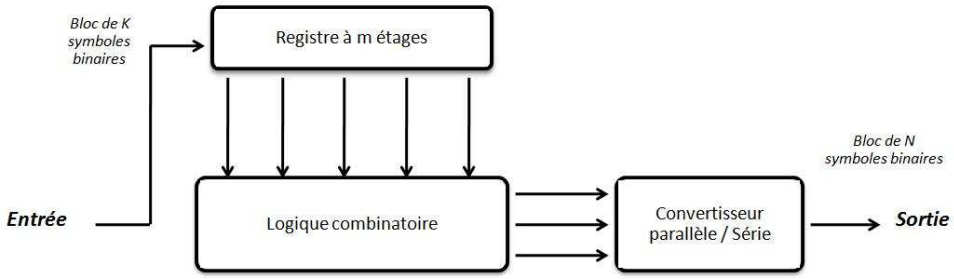


FIGURE 2.12 – Schéma de principe d'un codeur convolutif de rendement R et de mémoire m.

Pour illustrer le principe des codes convolutifs, voici un exemple présenté par la Figure 2.13, pour $K = 1$, $m = 2$ et $N = 2$. a_t parvient au codeur à l'instant t, les bits de sortie X et Y sont calculés par les relations suivantes :

$$\begin{cases} X = a_t + a_{t-1} + a_{t-2} \\ Y = a_t + a_{t-2} \end{cases}$$

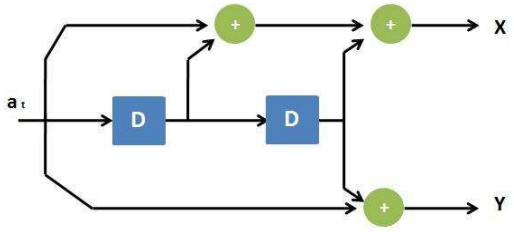


FIGURE 2.13 – Exemple d'un code convolutif de rendement R=1/2.

Supposons que le codeur reçoive le message 1011, les registres étant initialement tous les deux à 0. A la sortie on obtient la séquence codée suivante 11100001, et les registres seront finalement à l'état 11.

Le diagramme en Treillis (voir Figure 2.14 (b)) est une représentation utile pour l'algorithme de *Viterbi*, l'algorithme de décodage le plus utilisé pour les codes convolutifs [32], [33].

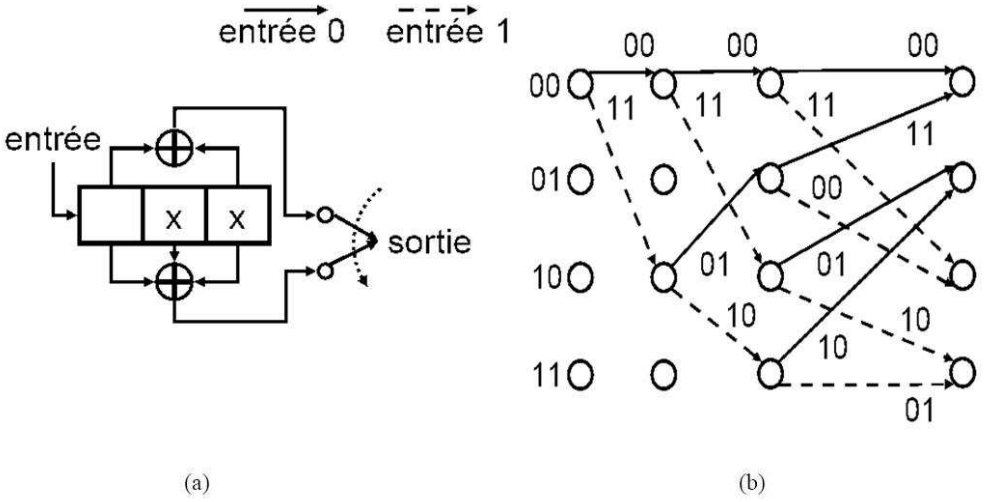


FIGURE 2.14 – (a) Un code convolutif de R=1/2. (b) Un diagramme en Treillis.

2.5.1 Les codes NSC et RSC

On désigne par NSC les codes non-systématiques (Non-Systematic Code en anglais) et par RSC les codes récursifs et systématiques (Recursive Systematic Code en anglais).

Un code convolutif est dit *récursif* si la séquence passant dans les registres à décalages est alimentée par le contenu de ses registres (voir la Figure 2.15 (b)). Si les K symboles d'information à l'entrée du codeur se retrouvent explicitement dans le code, alors le code est dit *systématique*, sinon il est dit *non-systématique* [18] (voir la Figure 2.15 (a)). Les codes non-systématiques et non-

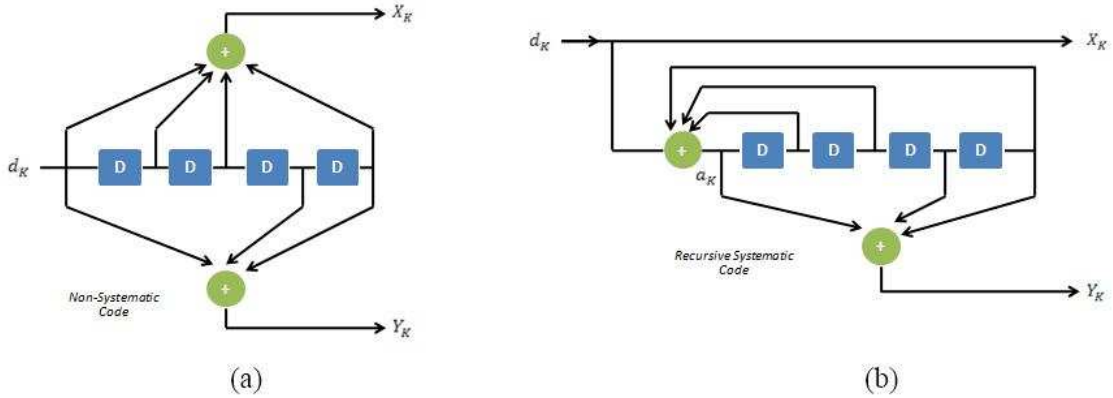


FIGURE 2.15 – (a) Un code non-systématique (NSC) (b) Un code récursif systématique (RSC).

récursifs présentent, pour des SNR élevés, des performances meilleures qu'un code systématique et non-récursive et l'inverse pour les SNR faibles [6], [34]. Pour cette raison, les codes NSC ont été principalement étudiés et utilisés jusqu'au début des années 1990. On rappelle que la puissance d'un code (capacité de correction d'erreurs) et la complexité du décodeur augmentent avec l'augmentation de la mémoire m de ce code.
Quant aux codes récursifs systématiques (RSC), ils sont utilisés par les turbo-codes, car ils sont les seuls susceptibles d'atteindre la limite de Shannon [6].

2.5.2 Les turbo-codes

Le plus célèbre des codes convolutifs est sans doute le turbo-code[7] inventé par C.Berro, A.Glavieux et P.Thitimajshima en 1993 [6]. Ces codes et les codes LDPC forment ce que l'on appelle *les techniques de codage avancées*. Le turbo-code utilise deux (ou plusieurs) encodeurs de type convolutifs. La Figure 2.16 montre le cas d'une concaténation parallèle, constituée de deux codes convolutifs récursifs systématiques identiques et un entrelaceur pseudo-aléatoire. A chaque mot d'information c, on associe une redondance p, qui peut être divisée en une redondance p_0 issue du premier encodeur et une redondance p_1 issue du deuxième encodeur.

Pour la première fois, un *décodeur itératif* est introduit. L'idée, très simple en soi, consiste en un décodeur comportant deux sous-ensembles de décodage s'échangeant de l'information. Le principe de ce récepteur est illustré dans la Figure 2.17. Pour expliquer le fonctionnement d'un tel décodeur, la notion d'*information extrinsèque* fut introduite. C'est cette information qui est échangée entre les décodeurs au cours des itérations. Après un certain nombre d'itérations, la décision ferme est prise sur l'information *a posteriori*. Cette information regroupe à la fois

7. Appelé turbo-code, car il utilise un principe analogue à celui du moteur turbo, c.à.d. que l'information issue du deuxième décodeur sera réinjecté dans le premier décodeur.

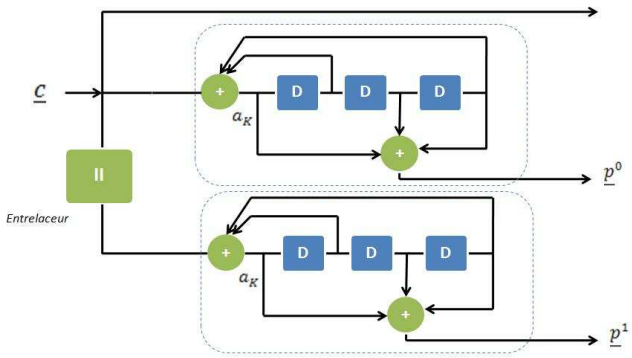


FIGURE 2.16 – Schéma de principe d'un turbo-codeur.

l'information issue de l'observation du canal et les informations extrinsèques issues des différents décodeurs [20].

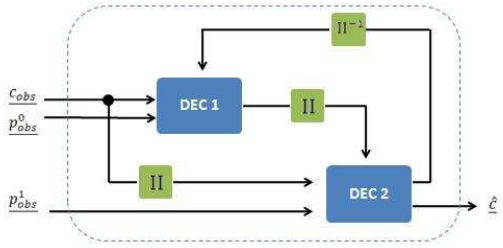


FIGURE 2.17 – Schéma de principe d'un turbo-décodeur.

2.6 Comparaison des performances entre quelques codes correcteurs d'erreurs

Avant de clôturer ce chapitre, nous allons donner une comparaison des performances entre les codes correcteurs les plus utilisés.

Pour un canal à bruit blanc additif et gaussien AWGN, la Figure 2.18 montre que le code de *Golay* donne des performances meilleures par rapport au code de *Hamming* et le code de *parité*. Pour une probabilité d'erreur égal à 10^{-5} Le code Golay (23,12) apporte un gain de codage de 3.8 dB, et le code de Hamming (7,4) apporte 1.8 dB, tandis que le code de parité apporte uniquement 1 dB par rapport au cas sans codage. Malgré cela, les performances de ces codes restent toujours médiocres par rapport aux critères de Varshamov et Gilbert. C'est pour cette raison que ces codes sont généralement utilisés dans des applications simples, par exemple dans le télétexte : on utilise le code de Hamming étendu (8,4) [1].

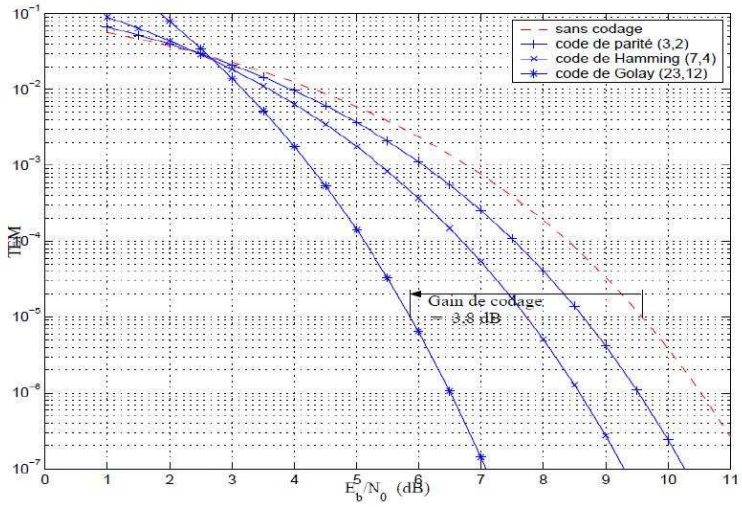


FIGURE 2.18 – Comparaison entre les performances des codes de parité, de Hamming et de Golay (Courbes reproduites de la référence [1]).

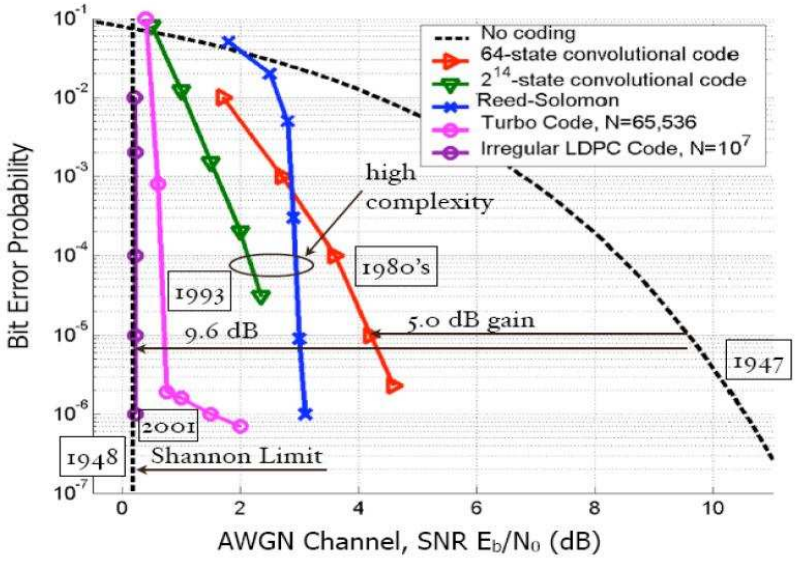


FIGURE 2.19 – Comparaison entre les performances des codes convolutifs, Reed-Solomon et les techniques de codage avancées (LDPC et Turbo-codes) (Courbes reproduites de la référence [2]).

Par contre, la Figure 2.19 montre dans un ordre chronologique, l'évolution des performances des codes correcteurs. A première vue, on constate que seules les techniques de codage avancées peuvent approcher la limite de Shannon ; par exemple pour une probabilité d'erreur égal à 10^{-5}, un code LDPC irrégulier de taille $N = 10^7$ et de rendement R=1/2 apporte un gain de 9.6 dB et il approche la limite de Shannon de 0.04 dB [35] (Les détails sur les codes LDPC seront

illustrées dans le chapitre suivant). Quant au Turbo-code avec un entrelaceur de taille N=65536, le gain apporté est de 9 dB et il fonctionne à moins de 0.5 dB de la limite de Shannon. Pour les autres codes : le code convolutif avec 2^{14} états, le code R-S et le code convolutif avec 64 états, ils apportent respectivement des gains de 6.5, 6 et 5 dB pour un taux d'erreur égal à 10^{-5}.

2.7 Conclusion

Dans ce chapitre, nous avons introduit, dans un premier temps, le schéma fondamental d'une communication numérique, en expliquant brièvement chacune de ses parties. Ensuite, nous avons présenté quelques notions fondamentales sur le codage de canal, en introduisant le théorème fondamental de codage de canal et les différentes classes de code correcteur d'erreurs.

A la fin de ce chapitre, nous avons présenté quelques résultats sur les performances des codes correcteurs les plus connus. En analysant ces résultats, nous avons constaté que seules les techniques de codage avancées (turbo-codes et codes LDPC) peuvent atteindre la limite de Shannon. Pour ce qui suit, notre étude sera focalisée sur des codes LDPC.

Chapitre 3

Etude des codes LDPC

Dans ce chapitre, nous entamerons la première partie du projet qui est l'étude des codes LDPC. Pour cela, nous commencerons d'abord par un bref historique sur les codes LDPC, ensuite, nous présenterons les différentes classes des codes LDPC ainsi que leurs présentations matricielle et graphique. Une fois terminé avec cette partie, nous passerons au décodage des codes LDPC, dans cette partie, nous présenterons l'algorithme de propagation de croyance et les différents algorithmes dérivés. Et pour la partie encodage, nous présenterons, dans un premier temps, les différentes techniques d'encodage, puis on en choisira, en précisant les raisons de notre choix.

Après l'étude théorique, nous aborderons l'étude de performances des codes LDPC, pour cela, on commence par présenter la chaîne de transmission ainsi que les conditions de simulations, ensuite, nous illustrerons nos résultats de simulations et nos conclusions.

3.1 Historique

Les codes LDPC (Low Density Parity Check) ont été inventés par Robert Gallager en 1962 dans le cadre de son PhD au Massachusetts Institute of Technology (MIT) [36], [37]. Mais à cause de l'introduction des codes Reed-Solomon et des codes convolutifs ainsi que la complexité d'encodage et de décodage vis-à-vis des moyens matériels de l'époque, les codes LDPC furent rapidement abandonnés et oubliés pendant 30 ans. Durant cette période de sommeil, seules quelques études y font référence, notamment, celle de Tanner en 1981, qui proposa une généralisation des codes de Gallager et une représentation par graphe bipartite [38]. L'introduction des turbo-codes et leur décodage itératif en 1993, relancèrent l'intérêt porté aux codes LDPC. D'ailleurs, en 1995, D. Mackay et R. Neal [39] ont montré que le décodage des codes de Gallager peut se réaliser à l'aide de l'algorithme itératif de Pearl [40], appelé *propagation de croyance*. Trois ans plus tard, M. Luby a introduit un nouveau type de codes plus performants que les codes de Gallager appelé *code LDPC irrégulier* [3].

Plus récemment, Chung et al. [41] ont montré que l'optimisation des codes LDPC irréguliers

pour un modèle de canal particulier était possible et que les performances asymptotiques des codes obtenues dépendaient bien des paramètres définissant le code.

La relative simplicité d'encodage et de décodage des LDPC par rapport aux turbo-codes, ainsi que leur grande efficacité dans les transmissions haut débit[42] ont motivé, d'un coté, la communauté scientifique pour poursuivre les travaux de recherche, afin d'optimiser encore plus leurs performances, et les industriels de l'autre coté, pour les introduire dans les nouvelles générations de téléphonie mobile, et dans les nouvelles normes de communications sans fil, optiques et spatiales.

La première apparition des codes LDPC était en 2004 dans la norme de télédiffusion numérique par satellite (DVB-S2) qui utilise un code LDPC irrégulier (voir section 3.3.2) pour la protection de la transmission des données descendantes [43], [44]. Récemment, les codes LDPC ont été introduits dans les standards IEEE 802.16e (WIMAX) , IEEE 802.11n (WiFi) [45] et dans le standard des transmissions spatiales CCSDS (The Consultative Committee for Space Data Systems) [46].

3.2 Définitions et notations

Un code LDPC (N,K) est un code linéaire en bloc, dont les N bits du mot de code doivent satisfaire $M = N - K$ équations de parité déduites à partir de la relation (2.19) présentée dans la section 2.4.1. La particularité de ce type de code est que sa matrice de contrôle de parité H est de faible densité, c'est-à-dire que le rapport entre le nombre d'éléments non-nuls "1" et le nombre d'éléments nuls "0" tend vers zéro. De ce fait, le processus de décodage et de calcul de syndromes (décrit dans la section 2.4.1) peuvent être effectués plus rapidement si l'on exploite la propriété de la matrice creuse. Par contre, la matrice de codage G devient dense et entraîne une complexité pour le codeur [47].

Pour illustrer comment obtenir les équations de parité, voici l'exemple d'un code $x = [x_0, \cdots, x_6]$ de rendement R= 3/7, caractérisé par sa matrice de contrôle de parité H (4,7).

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Les équations de parité associées à cette matrice et au mot de code x sont :

$$\begin{cases} x_0 \oplus x_2 \oplus x_4 \oplus x_5 = 0 \\ x_1 \oplus x_3 \oplus x_5 = 0 \\ x_1 \oplus x_2 \oplus x_6 = 0 \\ x_0 \oplus x_3 \oplus x_4 \oplus x_6 = 0 \end{cases}$$

Un code LDPC est aussi caractérisé par le nombre d'éléments non nuls présents dans les lignes et les colonnes de la matrice H. On note par w_r le poids[1] d'une ligne et par w_c le poids d'une colonne de la matrice H.

Pour que la matrice H soit de faible densité, il faut que :

$$w_r << N \tag{3.1}$$

$$w_c << M \tag{3.2}$$

Selon les résultats présentés dans la référence [48], il faut que $w_c \geq 3$ pour avoir de bonnes performances du code LDPC.

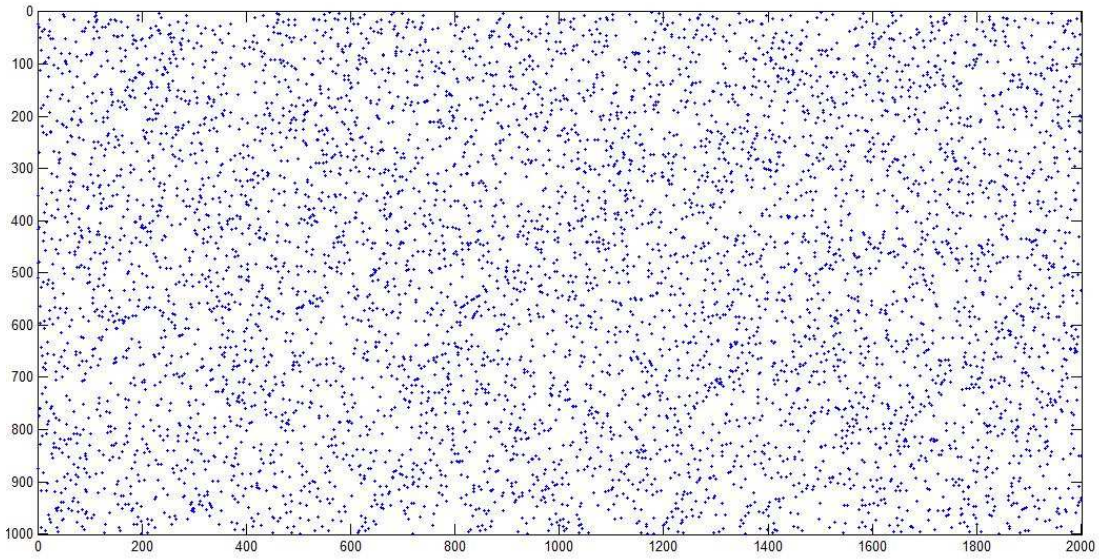


FIGURE 3.1 – Exemple d'une matrice de contrôle de parité H de dimension (1000,2000) et de poids $w_c = 3$.

N.B : La figure présentée ci-dessus est tirée du logiciel LDMO qu'on présentera dans le chapitre 5. Les points donnés en bleu représentent les éléments non-nuls ("1").

1. On rappelle que le poids d'un vecteur est égal au nombre d'éléments non nuls.

3.3 Les classes de codes LDPC

Les codes LDPC se divisent en deux grandes familles, à savoir les *codes réguliers* et les *codes irréguliers*.

3.3.1 Les codes réguliers

Les codes réguliers ont été introduits par R. Gallager en 1962 [36]. La régularité de ces codes est spécifiée par le nombre constant de "1" dans les lignes et les colonnes de la matrice H, c'est-à-dire que w_r et w_c sont constants et reliés par la relation suivante :

$$w_r = w_c \frac{N}{M} \tag{3.3}$$

Les codes LDPC réguliers sont alors paramétrés par (N,w_r,w_c) représentant respectivement, la longueur du mot de code, le poids des lignes et le poids des colonnes. Il est clair que w_r (respectivement w_c) sont des entiers très petits devant N (respectivement M) de telle sorte que H soit clairsemée.

Pour construire les matrices de contrôle de parité décrites en haut, Gallager [36], [2] a proposé la technique suivante :

1. Définition des paramètres du code : N, K, w_c et w_r.
2. Construction d'une sous-matrice H_1 de $N - K/w_c$ lignes et N colonnes.
3. Construction des autres sous-matrices par une permutation pseudo-aléatoire des colonnes de H_1, qu'on note $\pi_i(H_1)$.
4. Entassement des w_c sous matrices pour construire une matrice H creuse et régulière.

$$H = \begin{pmatrix} \pi_1(H_1) \\ \pi_2(H_1) \\ \vdots \\ \pi_{w_c}(H_1) \end{pmatrix} \tag{3.4}$$

Dans le cas des codes LDPC réguliers, le rendement R (qu'on a défini dans la section 2.3.1) peut s'écrire de la manière suivante :

$$R = 1 - \frac{w_c}{w_r} \tag{3.5}$$

Exemple : Supposons que N=20, K=5, $w_c = 2$ et $w_r = 4$, il s'ensuit que :

$$H_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Après la réalisation des étapes 3 et 4, on obtient une matrice de contrôle de parité H (voir (3.6)) avec un rendement de codage $R = 1 - \frac{2}{4} = \frac{1}{2}$.

$$H = \left(\begin{array}{cccccccccccccccccccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \tag{3.6}$$

3.3.2 Les codes irréguliers

Dans le cas où la distribution des éléments non nuls est non uniforme, ces codes LDPC seront appelés codes irréguliers. L'irrégularité de ces codes n'est pas paramétré par w_c et w_r, mais plutôt par deux polynômes qu'on illustrera dans la section 3.4.

La matrice présentée ci-dessous, représente le cas d'une matrice H pour un code LDPC irrégulier.

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Selon les travaux de Ludy et al. présentés dans [3] et [49], la performance d'un code LDPC irrégulier bien construit dépasse celle d'un code régulier. Les Figures 3.2 et 3.3 montrent une comparaison de performances entre un code régulier et un code irrégulier de taille N=16000 et de rendement $R = 1/2$ et $R = 1/4$. Le canal considéré est un canal à bruit blanc additif et gaussien (AWGN).

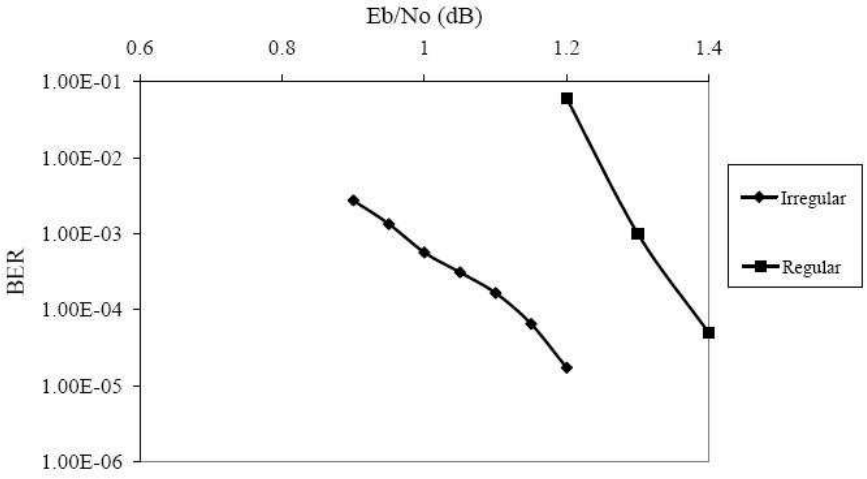


FIGURE 3.2 – Comparaison de performances entre les codes LDPC réguliers et irréguliers de taille N=16000 et de rendement $R = 1/2$ (Graphes reproduits de la référence [3]) .

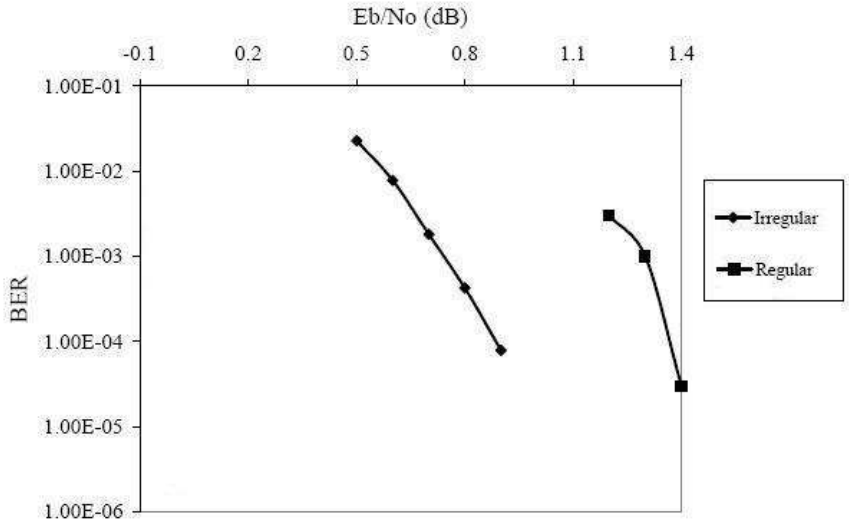


FIGURE 3.3 – Comparaison de performances entre les codes LDPC réguliers et irréguliers de taille N=16000 et de rendement $R = 1/4$ (Graphes reproduits de la référence [3]).

3.4 Représentation graphique des codes LDPC

Un code LDPC peut également être représenté, en plus de sa matrice de contrôle de parité, par un graphe bipartite appelé *graphe de Tanner* [38], ou plus généralement *graphe factoriel* [50]. Ce graphe contient deux types de noeuds, les *noeuds de données* (*variable-nodes*) représentant le mot de code et les *noeuds fonctionnels* ou *noeuds de contrôle* (*check-nodes*) correspondant aux

contraintes de parité. Un noeud de données i est relié à un noeud fonctionnel j par une branche, si et seulement si, l'élément correspondant à la $i^{ème}$ ligne et la $j^{ème}$ colonne de la matrice de contrôle de parité est non nul ($H_{ij} = 1$). Par convention, les noeuds de données seront représentés par des cercles et les noeuds de contrôle par des carrés.

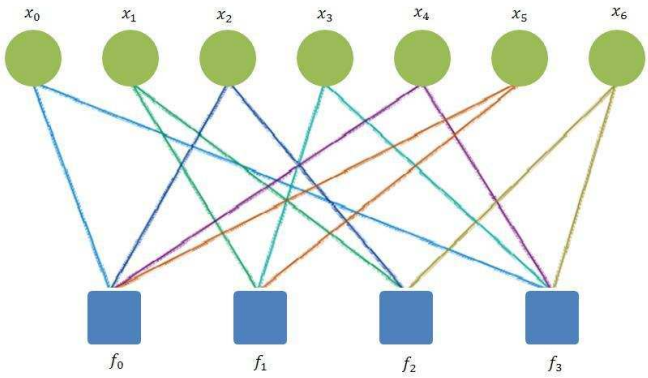


FIGURE 3.4 – Graphe de Tanner d'un code LDPC irrégulier.

La Figure 3.4 représente le graphe de Tanner d'un code LDPC irrégulier caractérisé par sa matrice H présentée ci-dessous :

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

On note par f_i $(i = 0, \cdots, 3)$ les contraintes de parité et par $x = [x_0, \cdots, x_6]$ le message reçu. On peut voir que pour chaque contrainte f_i, il n'y a que certains bits de x qui y participent. Dans cet exemple, on voit facilement la dépendance entre les bits de données et les bits de contrôle de parité dans le Tableau 3.1.

$(x_0, x_2, x_4, x_5) \rightarrow f_0$	$(x_1, x_3, x_5) \rightarrow f_1$
$(x_1, x_2, x_6) \rightarrow f_2$	$(x_0, x_3, x_4, x_6) \rightarrow f_3$

TABLE 3.1 – La dépendance entre les bits de données et les bits de contrôle de parité.

On peut voir que la valeur de chaque bit de x influence au moins un bit de contrôle de parité f_i. On remarque aussi qu'il y a deux contraintes qui contrôlent chaque bit de donnée. On peut supposer que les N bits sont corrects si toutes les contraintes sont vérifiées. On peut également voir le résumé de ces liens dans le Tableau 3.2.

$(f_0, f_3) \rightarrow x_0$	$(f_1, f_2) \rightarrow x_1$	$(f_0, f_2) \rightarrow x_2$	$(f_1, f_3) \rightarrow x_3$
$(f_0, f_4) \rightarrow x_4$	$(f_0, f_1) \rightarrow x_5$	$(f_2, f_3) \rightarrow x_6$	

TABLE 3.2 – Les liens entre les contraintes de parité et les bits de données.

Si l'état des contraintes était utilisé pour modifier (dans le but de corriger les erreurs introduites dans le message) les bits qui lui sont directement associés, alors, l'état d'une contrainte et l'action prise en conséquence influenceraient les autres contraintes via les bits modifiés, et ces contraintes modifiées vont à leur tour influencer l'état des bits qui leur sont associés. En résumé, on peut dire que la décision prise pour un bit de données influencera plusieurs contraintes et donc influencera indirectement les autres bits de données.

3.4.1 Le profil d'irrégularité des noeuds de données et des noeuds de contrôle

Quand le nombre de branches connectées aux différents types de noeuds est constant, on parle alors d'un code LDPC *régulier*. Par conséquent chaque bit du mot de code participe à un même nombre d'équations de parité. De même chacune des équations de parité utilise le même nombre de bits. Par extension, les codes LDPC *irréguliers* sont les codes dont le nombre de branches connectées aux différents types de noeuds varie de façon irrégulière. Pour décrire ces codes, il est d'usage de spécifier l'irrégularité d'un code à travers deux polynômes $\lambda(x)$ et $\rho(x)$ [20] :

$$\lambda(x) = \sum_{i \geq 1} \lambda_i x^{i-1} \tag{3.7}$$

$$\rho(x) = \sum_{i \geq 2} \rho_i x^{i-1} \tag{3.8}$$

où λ_i (respectivement ρ_i) caractérise la proportion du nombre de branches connectées aux noeuds de données (respectivement aux noeuds de contrôle) de degré i par rapport au nombre total de branches. Le degré est défini comme le nombre de branches connectées à un noeud.
On peut relier le profil d'irrégularité du code au rendement de codage R de la façon suivante :

$$R \geq 1 - \frac{\sum_{i \geq 1} \frac{\lambda_i}{i}}{\sum_{i \geq 2} \frac{\rho_i}{i}} \tag{3.9}$$

Si on prend le code représenté par la Figure 3.4, la distribution des branches est donnée par :

$$\lambda(x) = \frac{14}{14} x$$

$$\rho(x) = \frac{6}{14} x^2 + \frac{8}{14} x^3$$

3.4.2 La notion de cycle

On dit qu'un graphe de Tanner contient un *cycle*, s'il existe un chemin pour quitter et revenir à un noeud sans passer par les mêmes branches. Le nombre de branches traversées détermine la longueur du cycle. Un graphe sans cycle est dit graphe en *arbre* [20].

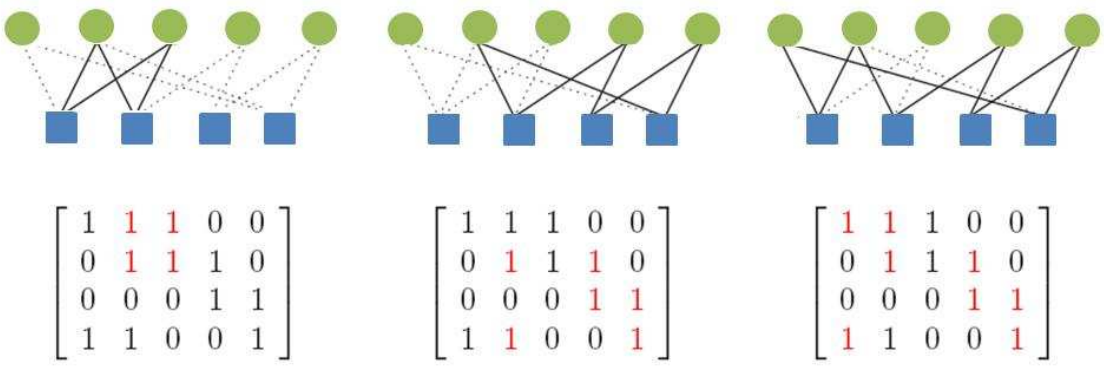


FIGURE 3.5 – Exemples de cycles de longueur 4, 6 et 8.

Le graphe de Tanner est une représentation graphique simple des codes LDPC. Ce graphe permet notamment d'illustrer les algorithmes de décodage, que nous présenterons dans la section suivante.

3.5 Décodage itératif des codes LDPC

L'algorithme de décodage des codes LDPC a été introduit initialement par Gallager [36], revu ensuite par Mackay et Neal [39], Richardson et Urbanke [51], Kschischang et al. [52] dans le cadre de la théorie des graphes. Cet algorithme contient plusieurs appellations : algorithme de *propagation de croyances* BP (Belief Propagation en anglais) [2], *somme-produit* SP (Sum-Product en anglais) et enfin *passage de messages* MP (message-passing en anglais). Toutes ces appellations viennent du fait qu'il y a un échange d'information entre les noeuds du graphe factoriel à travers les branches. Ces messages transitant de noeud en noeud portent une information probabiliste sur l'état des noeuds.

Le principe de la propagation de croyance est une application directe de la règle de Bayes [53] sur chaque bit d'une équation de parité. La vérification de parité permet de calculer une estimation de chaque bit. Ces estimations, formant des messages se propageant sur les branches du graphe, sont alors échangées itérativement afin de calculer une information *a posteriori* sur chaque bit. Dans le cas d'une propagation de croyance sur un graphe sans cycle, les messages

2. Ce terme est issue de la communauté de l'intelligence artificielle et plus particulièrement les travaux de Pearl [40].

échangés sont indépendants, ce qui conduit au calcul simple et exact des probabilités a posteriori, l'algorithme dans ce cas est dit optimal [54]. Dans le cas des codes LDPC où le graphe factoriel présente des cycles, l'hypothèse de messages indépendants n'est plus valide. Cependant, plus le graphe est creux (c'est à dire moins la matrice de contrôle de parité est dense), plus l'approximation d'un graphe sans cycle devient valide. C'est donc sous cette hypothèse que l'algorithme de décodage est décrit.

3.5.1 Décodage à décision dure

Pour comprendre le principe de propagation de croyance dans la cas d'un décodage Soft (à décision pondérée), que nous présenterons dans la partie 3.5.2. Nous allons donner l'algorithme de propagation de croyance dans le cas d'un décodage Hard (à décision ferme), appelé aussi Bit Flip, sous forme d'un exemple tiré de la référence [35].

Pour une matrice de contrôle de parité H, représentée ci-dessous, et dont le graphe de Tanner est donné en Figure 3.6.

$$H = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

On génère un mot de code $x = [10010101]$, puis, on le transmet à travers un canal BSC. A la réception, on a récupéré la séquence suivante $y = [11010101]$. On voit bien qu'il y a eu un changement au niveau du bit x_1.

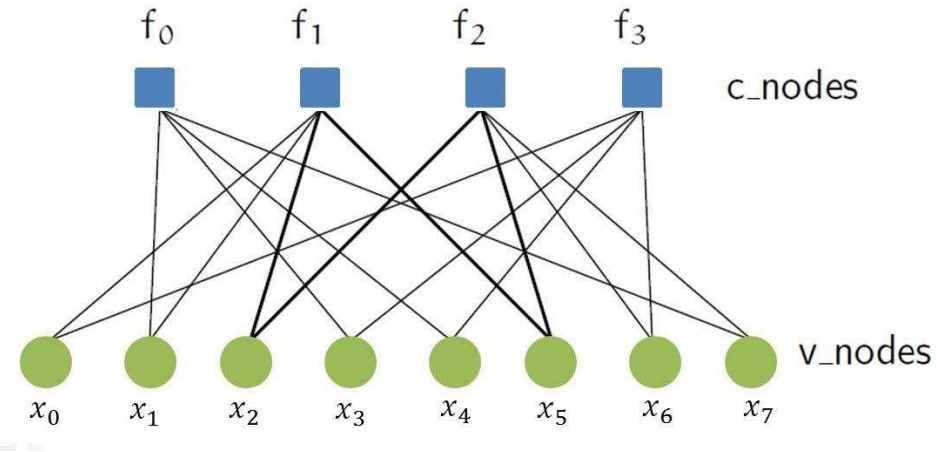


FIGURE 3.6 – Graphe de Tanner d'un code LDPC régulier.

L'algorithme de décodage se divise en 4 étapes :

– **Etape 1 :**

Les noeuds de données x_i (v-nodes) contiennent initialement l'information a priori, c'est-à-dire le mot reçu y. Chacun de ces noeuds de données x_i va envoyer un "message" aux noeuds fonctionnels connectés (c-nodes) f_j (dans notre exemple, chaque noeud de données est connecté à 2 noeuds fonctionnels) dont le contenu correspond au bit que ces v-nodes "croient" juste, d'où vient l'appellation "propagation de croyances". par exemple x_0 envoie un "1" à f_1 et f_3, x_1 envoie le contenu du bit erroné, c'est-à-dire "1" au lieu de "0", à f_0 et f_1 et ainsi de suite.

– **Etape 2 :**

Chaque noeud fonctionnel f_j calcule puis renvoie une réponse aux noeuds de données connectés. La réponse contient le bit que f_j croit juste pour un noeud de données connecté x_i, en supposant que les autres noeuds de données connectés sont justes. Si on prend toujours le cas de notre exemple, chaque noeud fonctionnel f_j est connecté à 4 noeuds de données. Alors, f_j prend les messages reçus à partir de 3 v-nodes et calcule le bit que le $4^{ème}$ v-node devrait avoir pour que l'équation de parité soit vérifiée. Le tableau 3.3 montre les messages envoyés par les différents v-nodes x_i et les messages calculés et renvoyés par les différents c-nodes f_j.

c-node	bit reçu/bit envoyé				
f_0	reçu :	$x_1 \to 1$	$x_3 \to 1$	$x_4 \to 0$	$x_7 \to 1$
	envoyé :	$0 \to x_1$	$0 \to x_3$	$1 \to x_4$	$0 \to x_7$
f_1	reçu :	$x_0 \to 1$	$x_1 \to 1$	$x_2 \to 0$	$x_5 \to 1$
	envoyé :	$0 \to x_0$	$0 \to x_1$	$1 \to x_2$	$0 \to x_5$
f_2	reçu :	$x_2 \to 0$	$x_5 \to 1$	$x_6 \to 0$	$x_7 \to 1$
	envoyé :	$0 \to x_2$	$1 \to x_5$	$0 \to x_6$	$1 \to x_7$
f_3	reçu :	$x_0 \to 1$	$x_3 \to 1$	$x_4 \to 0$	$x_6 \to 0$
	envoyé :	$1 \to x_0$	$1 \to x_3$	$0 \to x_4$	$0 \to x_6$

TABLE 3.3 – Illustration de la deuxième étape de l'algorithme BP dans le cas d'un décodage Hard.

– **Etape 3 :**

Quand un v-node reçoit une information additionnelle (extrinsèque) de la part des c-nodes connectés. ce v-node va utiliser cette information supplémentaire pour vérifier si le bit reçu initialement est juste ou pas. La manière la plus simple de faire cela, est d'effectuer un vote majoritaire entre l'information a priori (y_i) et les suggestions des c-nodes connectés. Pour notre exemple, l'étape 3 de l'algorithme est illustrée dans le Tableau 3.4.

v-node	y_i reçu	suggestions des c-nodes connectés		décision
x_0	1	$f_1 \to 0$	$f_3 \to 1$	1
x_1	1	$f_0 \to 0$	$f_1 \to 0$	0
x_2	0	$f_1 \to 1$	$f_2 \to 0$	0
x_3	1	$f_0 \to 0$	$f_3 \to 1$	1
x_4	0	$f_0 \to 1$	$f_3 \to 0$	0
x_5	1	$f_1 \to 0$	$f_2 \to 1$	1
x_6	0	$f_2 \to 0$	$f_3 \to 0$	0
x_7	1	$f_0 \to 1$	$f_2 \to 1$	1

TABLE 3.4 – Illustration de la troisième étape de l'algorithme BP dans le cas d'un décodage Hard.

- **Etape 4 :**
  Arrêter l'algorithme si toutes les équations de parité sont vérifiées ou bien le nombre d'itérations est terminé. Sinon aller à l'étape 2.

3.5.2 Algorithme de propagation de croyance

L'algorithme de propagation de croyance (BP) est un algorithme basé sur l'échange d'informations entre les différents noeuds du graphe bipartite. Cet algorithme se divise en trois parties : *l'initialisation*, *l'étape horizontale* et *l'étape verticale*[3]. Ce sont les deux dernières parties qui itèrent jusqu'au décodage sans erreur (vérification de toutes les équations de parité) ou jusqu'au nombre d'itérations maximal initialement défini[4].

Pour faciliter la compréhension de l'algorithme, nous allons donner toutes les notations qui vont nous servir pour la suite.

- q_{ij} : Le message passé d'un noeud de données x_i à un noeud de contrôle f_j.
- r_{ji} : Le message passé d'un noeud de contrôle f_j à un noeud de données x_i.
- $R_j = \{i : h_{ji} = 1\}$: L'ensemble des noeuds de données x_i participant au $j^{ème}$ noeud de contrôle.
- $R_{j/i} = \{i' : h_{ji'} = 1\}/\{i\}$: L'ensemble des noeuds de données $x_{i'}$ participant au $j^{ème}$ noeud de contrôle, à l'exception du noeud de données x_i.

3. L'étape 2 est dite horizontale, car elle concerne les noeuds de contrôle qui représentent les lignes de la matrice H, et l'étape 3 est dite verticale, car elle concerne les noeuds de données qui représentent les colonnes.

4. La démonstration de l'algorithme BP est disponible dans l'annexe A.

- $C_i = \{j : h_{ji} = 1\}$: L'ensemble des noeuds de contrôle f_j auquel le noeud de données x_i participe.
- $C_{i/j} = \{j' : h_{j'i} = 1\}/\{j\}$: L'ensemble des noeuds de contrôle $f_{j'}$ auquel le noeud de données x_i participe, à l'exception du noeud de contrôle f_j.

1- Initialisation

Soit $Pr(x_i = 1|y_i) = p_i$ la probabilité *a priori* que le noeud de données x_i est égal à 1 sachant que le bit reçu est y_i et $Pr(x_i = 0|y_i) = 1-p_i$. Si la distribution de bruit est connue, on initialise p_i à la valeur de vraisemblance normalisée. L'initialisation de q_{ij}^0 et q_{ij}^1 est donc donnée comme suit :

$$q_{ij}^1 = Pr(x_i = 1|y_i) = p_i \tag{3.10}$$

$$q_{ij}^0 = Pr(x_i = 0|y_i) = 1 - p_i \tag{3.11}$$

Dans le cas d'un canal à bruit blanc additif et gaussien AWGN, l'initialisation de q_{ij}^0 et q_{ij}^1 est donnée par les relations suivantes :

$$q_{ij}^0 = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{(y_i + 1)^2}{2\sigma^2} \tag{3.12}$$

$$q_{ij}^1 = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{(y_i - 1)^2}{2\sigma^2} \tag{3.13}$$

2- Etape horizontale : Mise à jour des noeuds de contrôle

La deuxième étape de l'algorithme de propagation de croyance consiste à mettre à jour les messages issus des noeuds de contrôle (voir Figure 3.7). Les messages r_{ji}^0 et r_{ji}^1 sont calculés de la façon suivante :

$$r_{ji}^0 = \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j/i}} (1 - 2q_{i'j}^1) \tag{3.14}$$

$$r_{ji}^1 = 1 - r_{ji}^0 = \frac{1}{2} - \frac{1}{2} \prod_{i' \in R_{j/i}} (1 - 2q_{i'j}^1) \tag{3.15}$$

3- Etape verticale : Mise à jour des noeuds de données

A partir des valeurs de r_{ji}^0 et r_{ji}^1 , le décodeur met à jour les messages q_{ij}^0 et q_{ij}^1 (voir Figure 3.8), à l'aide des relations suivantes :

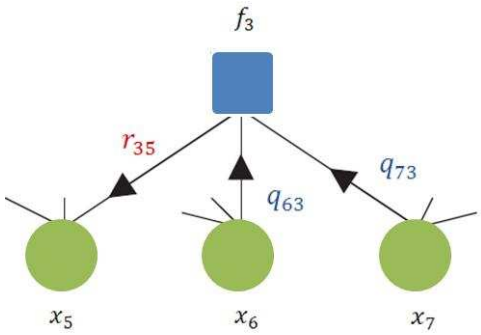


FIGURE 3.7 – La mise à jour des noeuds de contrôle.

$$q_{ij}^0 = K_{ij}(1-p_i)\prod_{j'\in C_{i/j}} r_{j'i}^0 \tag{3.16}$$

$$q_{ij}^1 = K_{ij}p_i\prod_{j'\in C_{i/j}} r_{j'i}^1 \tag{3.17}$$

où K_{ij} sont des constantes fixées de façon à obtenir :

$$q_{ij}^0 + q_{ij}^1 = 1 \tag{3.18}$$

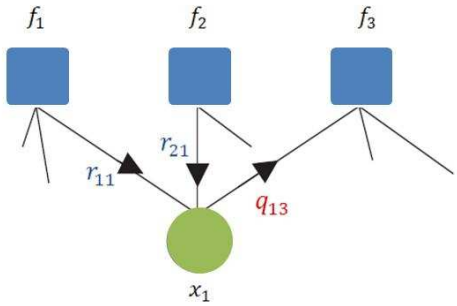


FIGURE 3.8 – La mise à jour des noeuds de données.

Une itération de l'algorithme de propagation de croyance est réalisée lorsque tous les messages se propageant le long des branches ont été calculés par les relations décrites dans les étapes 2 et 3. Après chaque itération, une décision Hard peut être prise sur les informations *a posteriori* Q_i^0 et Q_i^1 associées au noeud de données i, tel que :

$$Q_i^0 = K_i(1-p_i)\prod_{j\in C_i} r_{ji}^0 \tag{3.19}$$

$$Q_i^1 = K_i p_i \prod_{j \in C_i} r_{ji}^1 \tag{3.20}$$

où K_i sont des constantes fixées de façon à obtenir :

$$Q_i^0 + Q_i^1 = 1 \tag{3.21}$$

La décision sur la valeur binaire de chaque noeud de données est donnée par la relation suivante :

$$\hat{x}_i = \begin{cases} 1 & \text{si} \quad Q_i^1 > \frac{1}{2} \\ 0 & \text{sinon} \end{cases} \tag{3.22}$$

Le processus itératif est arrêté au bout d'un nombre maximum d'itérations. On peut également arrêter le processus avant le nombre maximum d'itérations en calculant à chaque itération le syndrome $H.\hat{x}$. Si celui-ci est nul, alors le décodage itératif a convergé vers un mot de code et le processus peut être arrêté .

Avec l'augmentation de la taille du code N et le nombre d'itérations, le temps de calcul devient très grand et les multiplications peuvent causer des dépassements "overflow". Pour contourner ce problème un nouvel algorithme, appelé *Log-Domain Algorithm*, a été introduit. Cet algorithme utilise la propriété de la fonction "Log" qui permet de transformer les multiplications en des sommations, ce qui entraîne une diminution considérable du temps de calcul et de complexité de décodage.

3.5.3 Algorithme de décodage Log-Domain

Cette fois-ci, les messages échangés seront des log-rapports de vraisemblance LLR (Log Likelihood Ratio en anglais), que l'on définit par les relations suivantes :

$$L(q_{ij}) = \log \frac{q_{ij}^0}{q_{ij}^1} \tag{3.23}$$

$$L(r_{ji}) = \log \frac{r_{ji}^0}{r_{ji}^1} \tag{3.24}$$

Le message envoyé par un noeud sur une branche b est le log-rapport de vraisemblance (LLR) de ce noeud, sachant que les log-rapports de vraisemblance proviennent de toutes les branches connectées, excepté b. On rappelle que cette restriction est introduite pour éviter de fortes dépendances entre les messages échangés d'une itération à la suivante [20].

L'algorithme Log-Domain est constitué aussi de trois étapes : initialisation, étape horizontale et étape verticale.

1- Initialisation

Dans l'étape d'initialisation, on calcule le log-rapport de vraisemblance $L(y_i)$ de l'information a priori y_i issue du canal de transmission.

$$L(y_i) = \log \frac{1-p_i}{p_i} \tag{3.25}$$

Dans le cas d'un canal à bruit blanc additif gaussien AWGN de variance σ^2 et à partir des relations (3.12) et (3.13), la relation (3.25) devient comme suit :

$$\begin{aligned} L(y_i) &= \log\{\frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{(y_i+1)^2}{2\sigma^2}\}/\{\frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{(y_i-1)^2}{2\sigma^2}\} \\ &= \log(\exp \frac{(y_i+1)^2}{2\sigma^2}) - \log(\exp \frac{(y_i-1)^2}{2\sigma^2}) \end{aligned} \tag{3.26}$$

après simplification ;

$$L(y_i) = 2\frac{y_i}{\sigma^2} \tag{3.27}$$

Après l'initialisation, l'étape horizontale consiste de mettre à jour les noeuds de contrôle et cela par le calcul des log-rapports de vraisemblance $L(r_{ji})$. Avant de donner la relation qui permet de calculer les $L(r_{ji})$, nous allons introduire la fonction $\phi(t)$ définie par la relation suivante :

$$\phi(t) = -\log \tanh(\frac{t}{2}) = \log \frac{e^t+1}{e^t-1} \tag{3.28}$$

La propriété mathématique de cette fonction est que :

$$\phi^{-1}(t) = \phi(t) \quad pour \quad t > 0 \tag{3.29}$$

On définit aussi "*la règle de la tanh*" qui est donnée par la relation (3.30) et dont la démonstration est présentée dans la référence [54].

$$\tanh(\frac{r_{ji}}{2}) = \prod_{i' \in R_{j/i}} \tanh(\frac{q_{i'j}}{2}) \tag{3.30}$$

Pour la suite de l'algorithme, on pose :

$$L(q_{ij}) = \alpha_{ij}.\beta_{ij} \tag{3.31}$$

avec : $\alpha_{ij} = sign(L(q_{ij}))$ et $\beta_{ij} = abs(L(q_{ij}))$

2- Etape horizontale : Mise à jour des noeuds de contrôle

A partir des équations (3.29), (3.30) et (3.31), on obtient la relation qui permet de calculer les log-rapports de vraisemblance $L(r_{ji})$:

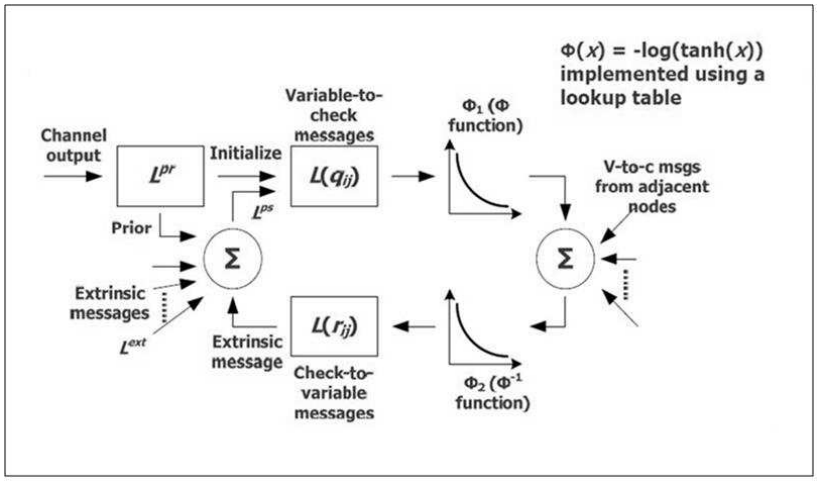


FIGURE 3.9 – Schéma illustratif de l'algorithme Log-Domain.

$$L(r_{ji}) = \prod_{i' \in R_{j/i}} \alpha_{i'j} \phi[\sum_{i' \in R_{j/i}} \phi(\beta_{i'j})] \tag{3.32}$$

3- Etape verticale : Mise à jour des noeuds de données

A partir des LLR $L(r_{ji})$ calculés dans l'étape horizontale, la mise à jour des log-rapports de vraisemblance $L(q_{ij})$ se fait à l'aide de la relation suivante :

$$L(q_{ij}) = L(y_i) + \sum_{j' \in C_{i/j}} L(r_{j'i}) \tag{3.33}$$

Après chaque itération, une décision Hard peut être prise sur l'information *a posteriori* $L(Q_i)$ associée au noeud de donnée i, tel que :

$$L(Q_i) = L(y_i) + \sum_{j \in C_i} L(r_{ji}) \tag{3.34}$$

La décision sur la valeur binaire de chaque noeud de données est donnée par la relation suivante :

$$\hat{x}_i = \begin{cases} 1 & \text{si} \quad L(Q_i) < 0 \\ 0 & \text{sinon} \end{cases} \tag{3.35}$$

En pratique, l'implémentation de l'algorithme de propagation de croyance (BP) engendre une grande complexité de l'organe de décodage. Pour réduire cette complexité plusieurs algorithmes

dérivés de l'algorithme BP ont été introduits. Ces algorithmes peuvent être vus comme une simplification de l'algorithme BP, et donc *sous-optimaux*. La diminution de performance engendrée par la sous-optimalité est à pondérer par le gain obtenu sur la complexité de l'organe de décodage. La majorité des algorithmes issus du BP reposent sur des opérations simplifiées de mise à jour des fiabilités en sortie des noeuds de contrôle.

Le Tableau 3.5 résume les algorithmes fréquemment rencontrés dans la littérature. Les différentes simplifications utilisées peuvent être combinées pour produire d'autres algorithmes de décodage et atteindre un objectif de compromis performance - complexité.

Nom de l'algorithme	Mise à jour des noeuds de données	Mise à jour des noeuds de contrôle
BP	$L(q_{ij}) = L(y_i) + \sum_{j' \in C_{i/j}} L(r_{j'i})$	$\lvert L(r_{ji})\rvert = \phi[\sum_{i' \in R_{j/i}} \phi(\beta_{i'j})]$
BP-Based /Min Sum [55]	idem BP	$\lvert L(r_{ji})\rvert = \min_{i' \in R_{j/i}}(\beta_{i'j})$
Offset Min-Sum [56]	idem BP	$\lvert L(r_{ji})\rvert = \min_{i' \in R_{j/i}}(\beta_{i'j}) + \gamma$
Normalised Min-Sum [43]	idem BP	$\lvert L(r_{ji})\rvert = \tau \min_{i' \in R_{j/i}}(\beta_{i'j})$
$\lambda - min$ [57]	idem BP	$\lvert L(r_{ji})\rvert = \phi[\sum_{i' \in R^{\lambda}_{j/i}} \phi(\beta_{i'j})]$
APP-check [55]	idem BP	$\lvert L(r_{ji})\rvert = \phi[\sum_{i' \in R_j} \phi(\beta_{i'j})]$
APP-variable	$L(q_{ij}) = L(y_i) + \sum_{j' \in C_i} L(r_{j'i})$	$\lvert L(r_{ji})\rvert = \phi[\sum_{i' \in R_{j/i}} \phi(\beta_{i'j})]$

TABLE 3.5 – Liste des algorithmes de décodage couramment utilisés dans la littérature.

Pour la suite de ce manuscrit, on s'intéressera uniquement à l'algorithme Min-Sum.

3.5.4 Algorithme de décodage Min-Sum

La simplification la plus couramment utilisée est celle de l'algorithme *Min Sum*, connu aussi sous le nom *BP-Based* [55]. Cette approximation repose sur le fait que le message calculé en sortie du noeud de contrôle est fortement dépendant du plus petit message entrant en valeur absolue. La fonction $\phi(t)$ étant une fonction décroissante positive (voir Figure 3.10), on peut écrire :

$$\sum_i \phi(|t_i|) \geq \phi(\min_i |t_i|) \tag{3.36}$$

$$\phi(\sum_i \phi(|t_i|)) \leq \min_i |t_i| \tag{3.37}$$

La fiabilité du message en sortie du noeud de contrôle peut être approximée par celle du message le moins sûr. Cette simplification algorithmique permet une réduction de complexité de décodage au niveau des noeuds de contrôle. En effet, la fonction non linéaire $\phi(t)$ est remplacée par

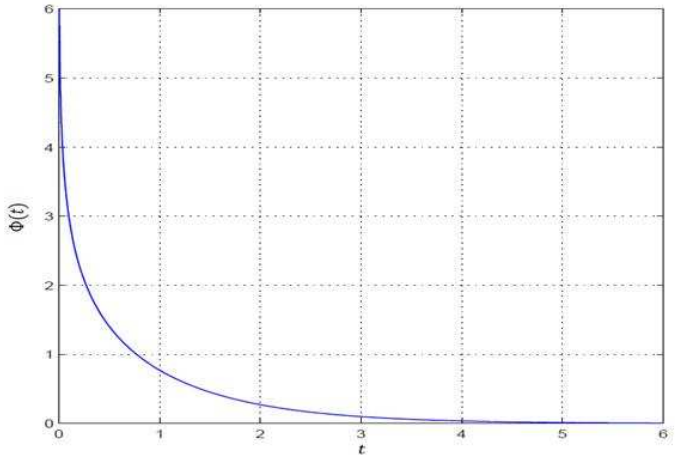


FIGURE 3.10 – L'allure de la fonction $\phi(t)$.

une simple fonction minimum "Min". Il en résulte donc une importante réduction des ressources mémoires nécessaires et la complexité de l'entité de mise à jour. Une seconde propriété très intéressante de cet algorithme réside dans sa robustesse face à une mauvaise estimation des log-rapports de vraisemblance associés aux observations en sortie du canal. En effet, du fait de l'utilisation de l'opérateur minimum, tous les calculs se font à un coefficient multiplicatif près.

Par exemple, dans le cas d'une modulation BPSK (Binary Phase Shift Keying en anglais) et d'un canal à bruit blanc additif et gaussien (AWGN) de variance σ^2, le log-rapport de vraisemblance du bit reçu y_i, à présenter l'entrée du décodeur, est égal à $2y_i/\sigma^2$ (voir la relation (3.27)). Dans le cas d'un décodage par l'algorithme Min-Sum, on peut présenter à l'entrée du décodeur un log-rapport de vraisemblance égal à $\alpha' y_i$. Ceci permet donc de s'affranchir de l'estimation de la variance du bruit [20].

Toutes les étapes de cet algorithme sont identiques à celles de l'algorithme Log-Domain sauf que la relation(3.32) devient comme suit :

$$L(r_{ji}) = \prod_{i' \in R_{j/i}} \alpha_{i'j} \min_{\{i' \in R_{j/i}\}} (\beta_{i'j}) \tag{3.38}$$

3.6 Encodage des codes LDPC

Comme nous l'avons mentionné auparavant, les codes LDPC ont la particularité d'être définis par leur matrice de contrôle de parité H. En effet, la manière triviale de déterminer le mot de code, est d'utiliser la matrice génératrice G calculable à partir de la matrice H, en utilisant la

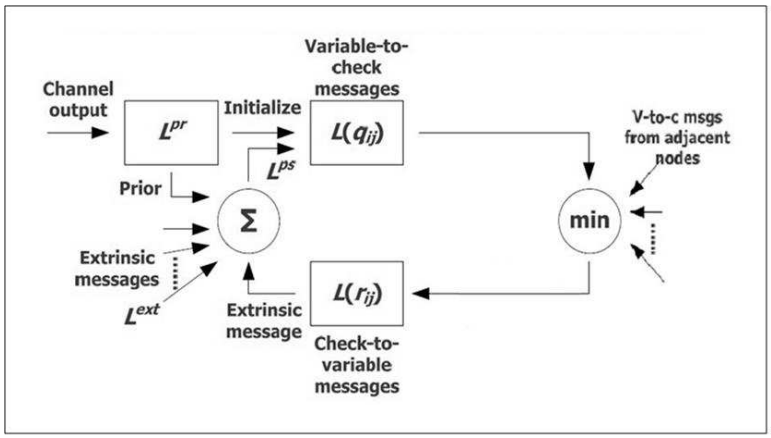


FIGURE 3.11 – Schéma illustratif de l'algorithme Min-Sum.

relation 2.20. Comme la matrice H est de faible densité, malheureusement, la matrice génératrice est généralement dense, ce qui entraîne une grande complexité d'encodage.

Afin de réduire la complexité d'encodage, plusieurs techniques d'encodage ont été introduites. Ces techniques peuvent être divisées en deux grandes familles :

- Les techniques d'encodage structurées.
- Les techniques d'encodage pseudo-aléatoires.

Les techniques d'encodage pseudo-aléatoire permettent de générer la matrice H (M,N) d'une manière pseudo-aléatoire, par contre, dans les techniques structurées, la matrice H est figée et à chaque fois, on la génère à l'aide d'un algorithme, parmi les algorithmes les plus utilisés, on peut citer : *Repeat Accumulate*, *Quasi-Cyclic*, *turbo-structured LDPC* ...etc [58].
D'après les résultats présentés dans la référence [20], les codes LDPC pseudo-aléatoires donnent des performances meilleures que celles des codes LDPC structurés, mais avec une complexité un peu plus grande. On voit bien qu'il y a un compromis à faire entre la performance et la complexité. Généralement, les techniques d'encodage pseudo-aléatoires sont utilisées dans les travaux de recherche (c'est le cas de notre étude), car cette technique permet de générer facilement des codes pour effectuer des simulations sur les performance des codes LDPC dans un système donné. Une fois l'étude faite, la partie d'implémentation et de standardisation nécessite l'introduction d'une technique de codage structurée, car généralement toutes les applications qui utilisent les codes LDPC sont des applications en temps réel, donc elles nécessitent un encodeur avec une complexité faible [2].

Le Tableau (3.6) montre quelques exemples de techniques d'encodage structurées utilisées par quelques standards de télécommunication.

Le standard	La technique utilisée	La taille N	Le rendement R	Références
DVB-S2	Repeat-Accumulate	N=68000	$R = \frac{1}{2}$	[43] [59]
CCSDS	Quasi-Cyclique Codes	N=8176	$R = \frac{7}{8}$	[46] [60]
802.16e WIMAX	Repeat-Accumulate	N=2304	R variable	[45] [61] [62]

TABLE 3.6 – Exemples de techniques de codage utilisées par quelques standards de télécommunication.

Comme on l'a précisé précédemment, notre étude rentre dans les travaux de recherche du laboratoire GSM sur les codes LDPC et les systèmes MIMO (Multiple Input Multiple Output en anglais), pour cela nous allons utiliser une technique d'encodage pseudo-aléatoire basée sur les algorithmes de Radford M. Neal.

3.7 Algorithme d'encodage de Radford Neal

Radford Neal a proposé dans [4] une méthode d'encodage simple, efficace et rapide, cette méthode est composée de deux parties :

1. **Partie 1** : Construction d'une matrice H de faible densité et de lignes linéairement indépendantes, dont les positions des éléments non nuls sont choisies d'une manière aléatoire.
2. **Partie 2** : Calcul des bits de parité à partir de la matrice H et la séquence d'information.

Par convention, on met les M bits de parité en premier et les K bits d'information en dernier pour former un code LDPC systématique.

$$x = [\, p \mid c \,] \tag{3.39}$$

L'algorithme présenté ci-dessous permet de construire une matrice pseudo-aléatoire H (M,N) de faible densité. La première partie de cet algorithme consiste à initialiser la matrice H à zéro et à fixer le nombre d'éléments non nuls par colonne, ensuite, distribuer ces éléments non nuls d'une manière pseudo-aléatoire et de façon à ce que les lignes de la matrice obtenue soient linéairement indépendantes.

La deuxième partie de l'algorithme consiste à vérifier s'il y a au minimum deux éléments non nuls par ligne[5]. Dans le cas où il y a un seul élément non nul par ligne, l'algorithme va ajouter des éléments non nuls de façon à ce que les lignes soient toujours linéairement indépendantes. Enfin, la troisième partie de cet algorithme consiste à éliminer les cycles de longueur 4.

5. Une équation de parité nécessite au minimum deux bits de données.

1.1 **Begin**

1.2 Fix w_c for all the columns of H

1.3 Set H to zero

1.4 **for** i=0 to N-1

1.5 Choose randomly w_c positions from the M positions of the column i

1.6 Set $H_{(w_c i)}$ to 1

Verification of the number of "1" in the rows :

1.8 **for** j=0 to M-1

1.9 **if** $w_r(j) < 2$ **then**

1.10 Add "1" so as to have at least $w_r = 2$

Elimination of all the 4-length cycles :

1.11 **if** no-cycle = 1 **then**

1.12 eliminate all the 4-length cycles

1.13 **end**

Comme on l'a mentionné précédemment, la deuxième partie de la méthode permet de calculer les bits de parités p, pour cela R. Neal a proposé trois algorithmes de calcul :

- L'algorithme d'encodage dense.
- L'algorithme d'encodage mixte.
- L'algorithme d'encodage basé sur la décomposition L-U.

3.7.1 Algorithme d'encodage dense

Cet algorithme consiste à décomposer la matrice H en une matrice carrée A (M,M) et une matrice B $(M,N-M)$, en réarrangeant les colonnes de la matrice H de telle sorte que A soit une matrice *non-singulière*.

$$H = [\, A \mid B \,] \tag{3.40}$$

En remplaçant (3.39) et (3.40) dans (2.19), on obtient :

$$[\, A \mid B \,]\, [\, p \mid c \,]^t = 0 \tag{3.41}$$

Ce qui donne par la suite :

$$A.p \;+\; B.c = 0 \tag{3.42}$$

Comme on est dans un corps de Galois F_2 (le cas binaire), (3.42) peut s'écrire de la manière suivante :

$$A.p \;=\; B.c \tag{3.43}$$

d'où :

$$p = A^{-1}.B.c \tag{3.44}$$

En calculant l'inverse de la matrice A, puis le produit matriciel $A^{-1}B$, on déduit les M bits de parité par la multiplication de la matrice résultante avec c.

L'inconvénient de cette méthode est que le temps de calcul devient très grand (proportionnel à M x (N-M)) pour une matrice H de grande taille.

3.7.2 Algorithme d'encodage mixte

Pour améliorer ou diminuer le temps de calcul de l'algorithme précédent, il est préférable de calculer p en deux étapes :

1. Calculer $z = B.c \Longrightarrow$ Le temps de calcul est proportionnel à M.
2. Calculer $p = A^{-1}z \Longrightarrow$ Le temps de calcul est proportionnel à M^2.

Pour M<N-M c-à-d quand le rendement R est supérieur ou égal à 1/2, cet algorithme est meilleur par rapport au précédent en terme de temps de calcul, car son temps de calcul est proportionnel à M^2 tandis que l'autre est proportionnel à $M(N-M)$,

3.7.3 Algorithme d'encodage basé sur la décomposition L-U

Pour diminuer encore plus le temps de calcul, R. Neal a proposé une autre méthode basée sur la décomposition L-U de la matrice A, cette méthode permet de décomposer la matrice A en une matrice triangulaire inférieure L (Low) et une matrice triangulaire supérieure U (Up).

La relation (3.40) sera écrite de la manière suivante :

$$H \;=\; [\, LU \mid B \,] \tag{3.45}$$

avec

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{M1} & l_{M2} & \cdots & l_{MM} \end{bmatrix} \quad et \quad U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1M} \\ 0 & u_{22} & \cdots & u_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{MM} \end{bmatrix} \tag{3.46}$$

Cette factorisation matricielle permet de résoudre le système d'équations linéaires défini par la relation (3.43).

En posant $z = B.c$ et en remplaçant (3.45) dans (3.43), on obtient :

$$LU\,.p = z \tag{3.47}$$

En posant $U.p = t$, La relation (3.47) peut s'écrire sous la forme matricielle suivante :

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{M1} & l_{M2} & \cdots & l_{MM} \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_M \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_M \end{bmatrix} \tag{3.48}$$

On trouve les composantes de t par des substitutions élémentaires. Cette étape est dite *forward substitution* (en anglais), puisqu'on résout le système en descendant de t_1 à t_M. Il reste à calculer les composantes du vecteur p en résolvant le système triangulaire supérieur :

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1M} \\ 0 & u_{22} & \cdots & u_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{MM} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_M \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_M \end{bmatrix} \tag{3.49}$$

Cette fois-ci, on monte de p_M à p_1. Cette étape est dite *backward substitution* (en anglais).

Il existe une variété d'algorithmes permettant de faire la décomposition L-U. Dans ce manuscrit, nous allons présenter trois algorithmes utilisés par R. Neal :

- L'algorithme First Column.
- L'algorithme Minimal Column.
- L'algorithme Minimal Product.

Algorithme First Column

Le principe de l'algorithme First Column est le suivant :

- Chercher pour chaque ligne i la première colonne qui contient un élément non nul dans la ligne i et la colonne i.
- Réarranger ces colonnes pour former une matrice carrée A (M,M) ne portant que des "1" sur la diagonale.
- Déduire à partir de la matrice A, la matrice triangulaire supérieure U et inférieure L.
- Les $N - M$ colonnes restantes forment la matrice B .

2.1 **Begin**

2.2 Set U and L to all zeros

2.3 Set F to H

Rearrangement of the columns

2.4 **for** j=0 to M-1

2.5 Find a non-zero element of F that is in row i, column i, or in a later row/column.

2.6 Rearrange rows and columns of F and H from i anword to put this element in row i, column i.

2.7 Copy column i of F up to row i to column i of U.

2.8 Copy column i of F from row i to comumn i of L

2.9 Add row i of F to later rows with a 1 in column i.

Deduction of the matrix L and U

2.10 **for** i=0 to M-1

2.11 L(i to M-1,i)=F(i to M-1,i)

2.12 U(0 to i,i)=F(0 to i,i)

Deduction of the matrix B

2.13 Set B to the last N-M columns of the rearranged H.

2.14 **end**

Algorithme Minimal Column

L'algorithme Minimal Column utilise le même principe, sauf que cette fois-ci, on cherche la colonne qui contient à la fois un élément non nul dans la ligne i et la colonne i, et le minimum d'éléments non nuls dans la colonne i.

Algorithme Minimal Product

Dans l'algorithme Minimal Product, on cherche la colonne qui contient à la fois un élément non nul dans la ligne i et la colonne i, et le minimum d'éléments non nuls dans la ligne i et la colonne i.

Les Figures 3.12 et 3.13 montrent une comparaison, en terme de nombre d'opérations effectuées par noeud de contrôle en fonction de M, entre les trois algorithmes décrits précédemment.

On voit bien qu'en augmentant M, le nombre d'opérations effectuées par noeud de contrôle augmente d'une manière linéaire mais avec des pentes différentes pour les trois algorithmes.

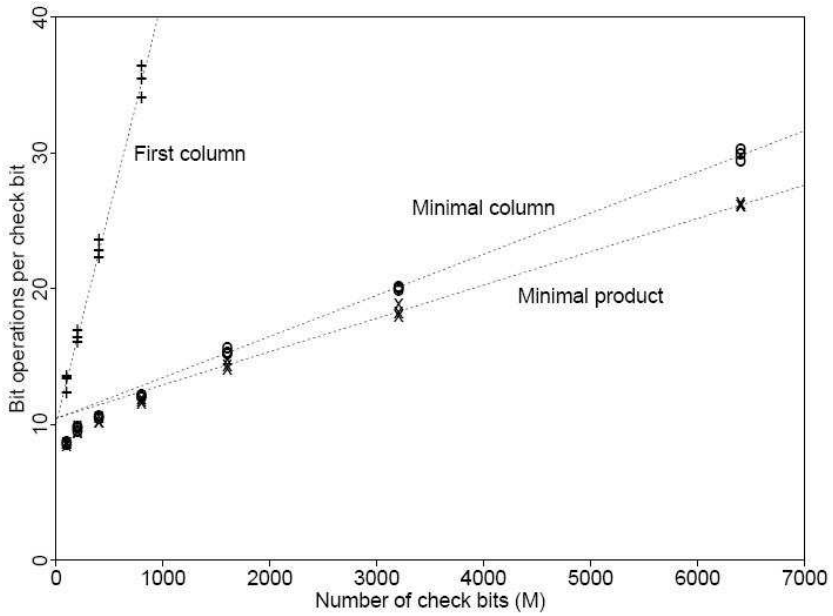


FIGURE 3.12 – Comparaison entre les trois algorithmes de décomposition L-U pour $w_c = 3$ et R=1/2 (Courbes reproduites de la référence [4]).

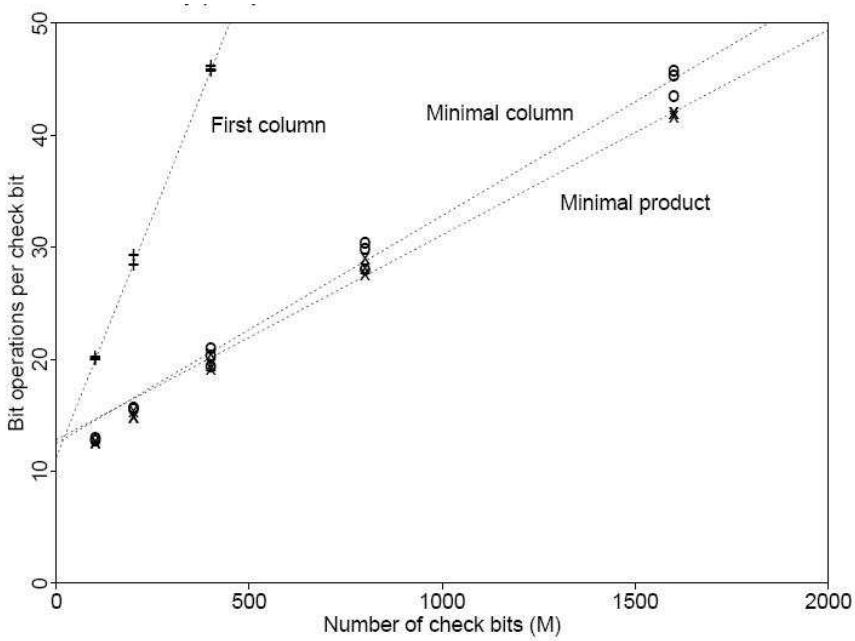


FIGURE 3.13 – Comparaison entre les trois algorithmes de décomposition L-U pour $w_c = 4$ et R=1/2 (Courbes reproduites de la référence [4]).

On remarque aussi que le temps de calcul pris par l'algorithme First Column est beaucoup plus grand par rapport aux autres algorithmes, alors que c'est l'algorithme Minimal Product qui

donne de meilleures performances en terme de temps de calculs.

En augmentant le nombre d'éléments non nuls par colonne, On remarque que la complexité de calcul augmente. Si on prend l'exemple de l'algorithme Minimal Product avec $M = 2000$ et $w_c = 3$, le nombre d'opérations effectuées est d'environ 15 opérations, tandis que pour $w_c = 4$ le nombre d'opérations est de 50 opérations par noeud de contrôle.

Pour la suite de notre étude et pour toutes les simulations que l'on présentera dans la section 3.8, nous allons utiliser l'algorithme Minimal Product.

3.8 Etude des performances des codes LDPC

Dans cette section, une évaluation des performances des codes LDPC est présentée. Pour cela, nous avons utilisé le logiciel MATLAB pour programmer toute la chaîne de simulation et cela à cause de la simplicité de programmation sous MATLAB. Mais au cours des simulations, il s'est avéré nécessaire de basculer vers le langage C afin de faire quelques simulations et cela pour des raisons qu'on définira plus tard.

L'organisation du reste de la section est donnée sous la forme suivante :

- Présentation de la chaîne de simulation.
- Définition des conditions de simulation et les hypothèses.
- Présentation des résultats.
- Interprétations et conclusions.

3.8.1 Présentation de la chaîne de simulation

Le modèle de simulation est schématisé dans la Figure 3.14. Un générateur pseudo aléatoire génère des blocs de données binaires de taille K, ces blocs sont codés par un encodeur LDPC, puis modulés et transmis via un canal AWGN. A la réception, le processus de détection est réalisé pour pouvoir estimer les bits transmis, on compare les bits reçus et les bits transmis afin de déterminer le taux d'erreur binaire BER (Bit Error Rate en anglais), qu'on définit par la relation suivante :

$$BER = \frac{\text{Nombre de bits erronés}}{\text{Nombre de bits transmis}} \tag{3.50}$$

Les principaux paramètres et hypothèses utilisés dans la chaîne de simulation sont résumés dans le Tableau 3.7 :

Paramètre	Valeur/Caractéristique
Type du code LDPC	Code irrégulier
La matrice H	Pseudo aléatoire, sans cycle de longueur 4
Le poids des colonnes	$w_c = 5$
La technique d'encodage utilisée	Algorithme Minimal Product de R. Neal
Type du canal	Canal Gaussien à bruit blanc additif (AWGN)
Estimation de la variance du bruit σ^2	Parfaite
Type de modulation	BPSK (Binary Phase Shift Keying)
Le critère d'évaluation des performances	BER en fonction de E_b/N_0

TABLE 3.7 – Les paramètres de simulation utilisés pour évaluer les performances des codes LDPC.

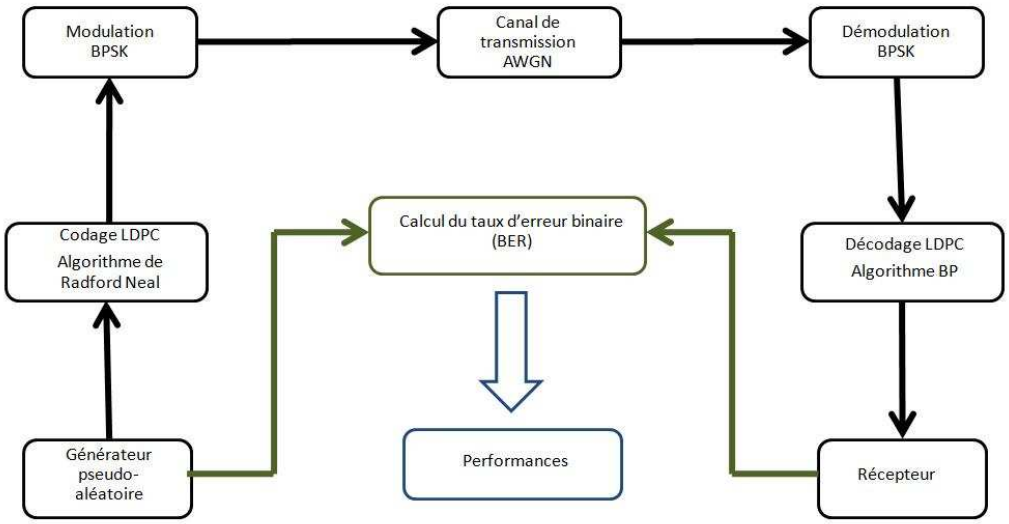


FIGURE 3.14 – Le modèle de simulation utilisé pour l'évaluation des performances des codes LDPC .

3.8.2 Comparaison entre les différents algorithmes de décodage

La Figures 3.15 représente les différentes implémentations de l'algorithme de décodage BP (Belief Propagation) avec un processus de décodage de 10 itérations et un code LDPC irrégulier de taille $N = 1024$ et de rendement R= 1/2.

Comme attendu, on voit bien que le décodage Hard donne des performances médiocres par rapport au décodage Soft, d'ailleurs on remarque que le décodage Hard présente une région de non convergence d'environ 7.8 dB et un seuil de convergence situé à 6 dB, tandis que dans le décodage Soft et pour le cas de l'algorithme Log-Domain, la région de non convergence est 2 dB

uniquement et un seuil de convergence situé à 1.5 dB. Si on se fixe comme référence à un BER $= 10^{-5}$, on voit bien que le décodage Hard apporte uniquement un gain de 1 dB par rapport au cas sans codage, alors que le cas Soft apporte au moins un gain de 5.5 dB. Delà, on peut dire que tout ces résultats confirment la raison pour laquelle tous les standards introduisant les codes LDPC utilisent le décodage Soft au lieu du décodage Hard.

Pour le cas Soft, Les deux courbes données en vert et en bleu dans la Figure 3.15, permettent d'évaluer l'influence de l'approximation donnée par la relation (3.38). On voit bien que pour un BER $=10^{-4}$, ces deux algorithmes de décodage (Log-Domain et Min-Sum) donnent respectivement des gains de 4.5 dB et 4.25 dB par rapport au cas sans codage. En comparant les performances de ces deux algorithmes, nous avons constaté que l'algorithme sous-optimal Min-Sum est une bonne approximation de l'algorithme Log Domain. Nous avons aussi constaté, au cours des simulations, que l'exécution de l'algorithme Min-Sum met beaucoup moins de temps par rapport à l'exécution de l'algorithme Log Domain.

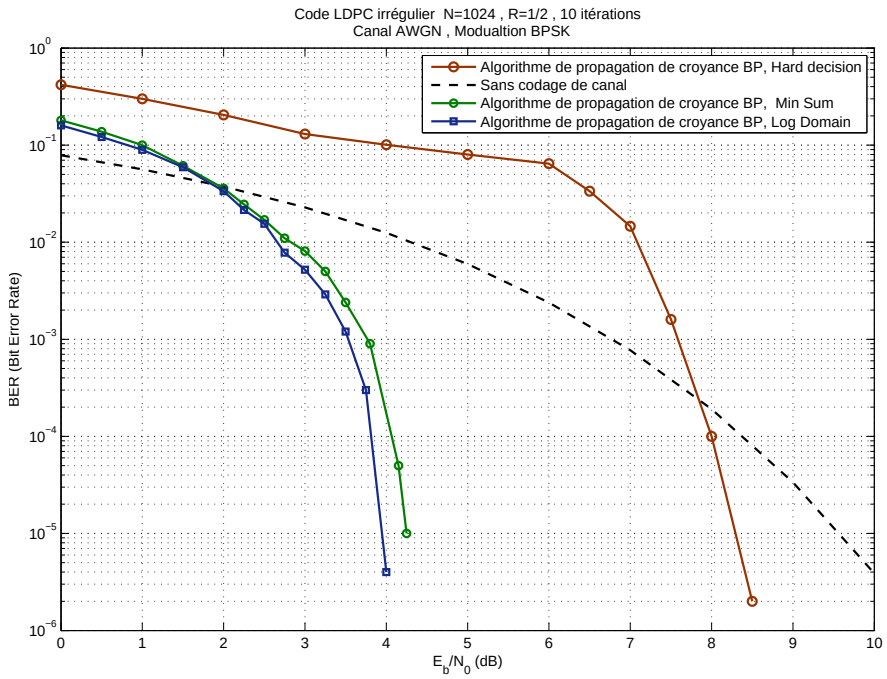


FIGURE 3.15 – Comparaison des performances entre les différents algorithmes de décodage pour N=1024.

3.8.3 Influence de la taille du code sur les performances

Le but de cette simulation est de voir l'influence de la taille N sur les performances des codes LDPC. Pour cela, on fixe R=1/2 et le nombre d'itérations du processus de décodage à 10.

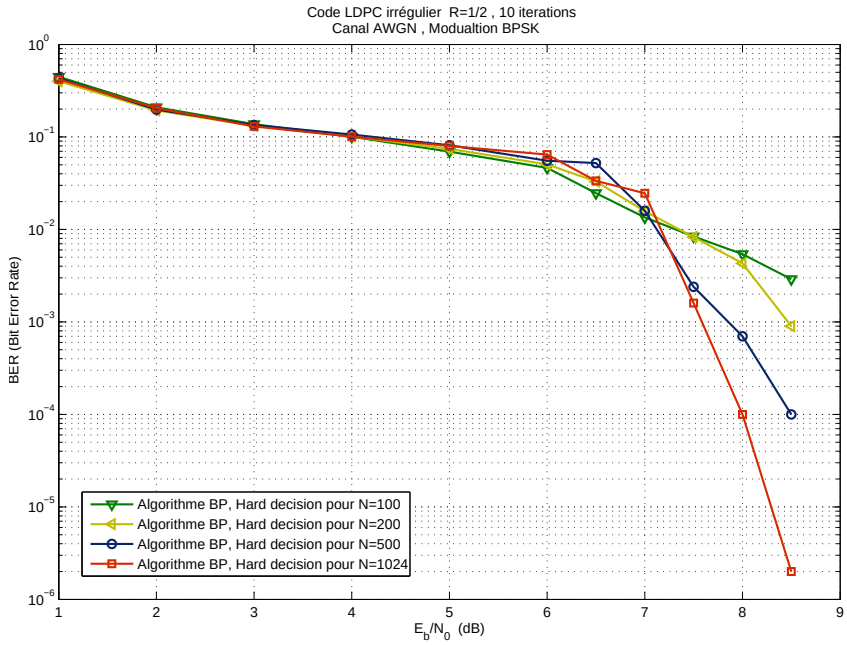


FIGURE 3.16 – Influence de la taille du code sur les performances pour un décodage Hard.

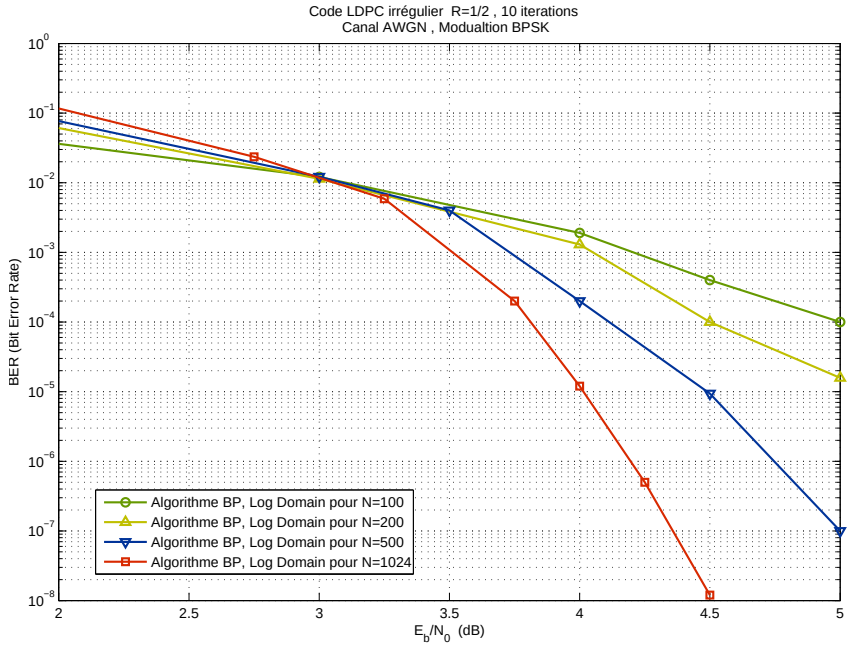


FIGURE 3.17 – Influence de la taille du code sur les performances pour l'algorithme de décodage Log Domain.

D'après les résultats présentés dans la Figure 3.16, on remarque qu'avec l'augmentation de N, il résulte une amélioration de performances, par exemple pour un BER égal à 10^{-3}, le code LDPC irrégulier de taille N=1024 apporte un gain de 0.8 dB par rapport à celui de N=200, et de 0.3 dB par rapport à celui de N=500. On remarque aussi qu'en augmentant N, le seuil de convergence reste presque constant et c'est les pentes des graphes qui augmentent avec l'augmentation de N.

Par contre dans les Figures 3.17 et 3.18 qui correspondent respectivement au décodage Soft Log Domain et Min-Sum, on voit clairement l'amélioration apportée par l'augmentation de N. Par exemple la Figure 3.17 montre que pour un BER égal à 10^{-5}, le code LDPC irrégulier de taille N=1024 apporte un gain de 0.6 dB par rapport à celui de N=500 et un gain de 1.2 dB par rapport à celui de N=200. D'autre part, nous avons constaté que plus N est grand plus le seuil de convergence décale vers les E_b/N_0 faibles et les pentes des graphes deviennent encore plus grandes. Par exemple pour N=1024 et $E_b/N_0 = 4.5$ dB, le taux d'erreur binaire est de l'ordre de 10^{-8}, cela signifie 1 bit erroné parmi cent millions de bits transmis. Ce résultat montre clairement la puissance des codes LDPC.

En augmentant encore plus la taille du code N ($N \geq 1000$), nous avons constaté que le logiciel MATLAB prend beaucoup de temps, car il manipule des matrices de taille très grande [6].

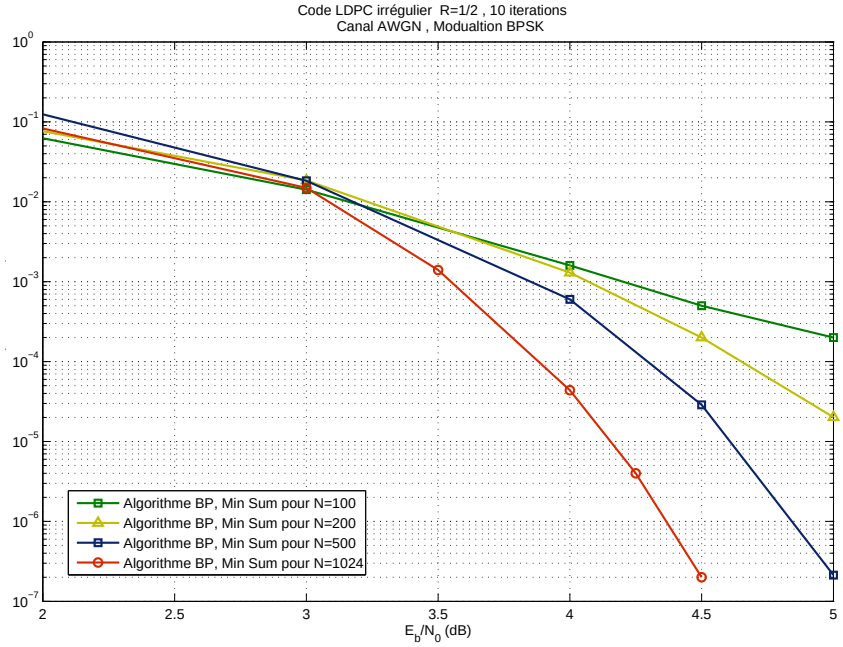


FIGURE 3.18 – Influence de la taille du code sur les performances pour l'algorithme de décodage Min-Sum.

6. A titre d'exemple, dans le cas N=1000 et en utilisant un ordinateur Core 2 Duo 3 Go de RAM, les simulations ont mis tout un week-end pour obtenir un point de $BER = 10^{-6}$.

Pour contourner ce problème, il s'est avéré nécessaire de passer au langage C. Pour cela nous avons utilisé des fonctions prédéfinies dans la bibliothèque de Radford Neal disponibles sur son site [63].

La Figure 3.19 montre le cas des codes LDPC de taille N= 1000, 5000 et 10000 décodés par l'algorithme Log Domain (10 itérations). Dans cette figure, on voit encore plus clairement l'effet de la taille du code N sur les performances. D'ailleurs, on voit bien que pour N=10000, le taux d'erreur binaire BER = 10^{-7} uniquement pour E_b/N_0 =1.8 dB. Ces résultats confirment la puissance des codes LDPC irréguliers surtout pour N très grand.

Les résultats obtenus dans cette partie justifient pourquoi tous les standards de communication utilisant des codes LDPC choisissent des tailles N très grandes, par exemple N=68000 pour le standard DVB-S2 et N=8176 pour le standard CCSDS.

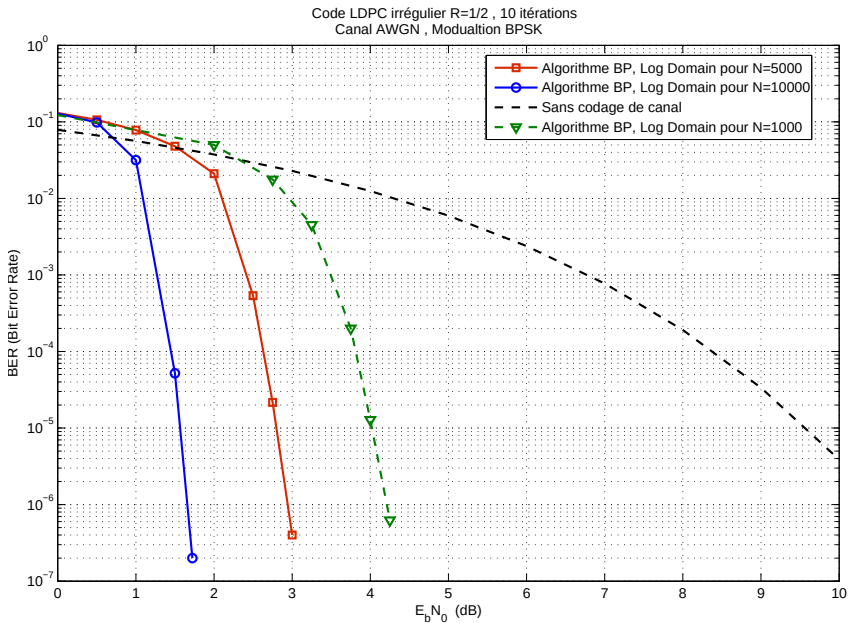


FIGURE 3.19 – Influence de la taille du code sur les performances pour l'algorithme de décodage Log-Domain.

3.8.4 Influence du nombre d'itérations du processus de décodage sur les performances

Dans cette simulation, nous étudions l'effet du nombre d'itérations sur les performances. Pour cela, nous allons considérer l'algorithme Min-Sum et cela à cause de sa rapidité d'exécution par

rapport à l'algorithme Log Domain.

La Figure 3.20 montre l'effet du nombre d'itérations sur les performances pour N=200. On remarque qu'il y a une amélioration à chaque fois qu'on augmente le nombre d'itérations, par exemple entre 2 et 10 itérations, il y a un gain de 0.8 dB pour un BER=10^{-2}. Mais à partir de 15 à 20 itérations, on ne voit pas d'amélioration, c'est-à-dire que les courbes commencent à se saturer.

La Figure 3.21 montre un comportement identique pour N=1024, c'est-à-dire que l'augmentation du nombre d'itérations améliore les performances du décodage jusqu'à un nombre maximum d'itérations où l'algorithme de décodage converge.

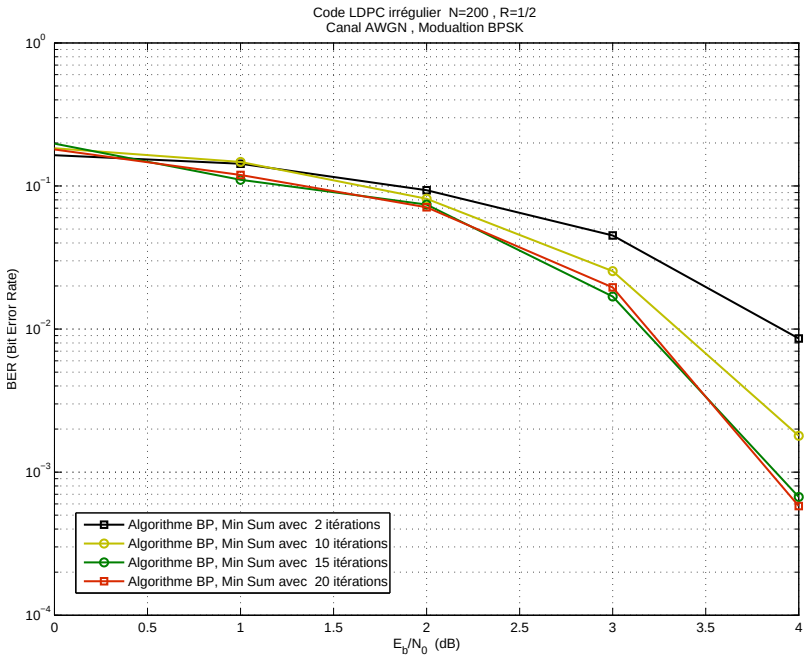


FIGURE 3.20 – Influence du nombre d'itérations sur les performances des codes LDPC pour N=200 et un décodage Min-Sum .

3.8.5 Influence du rendement de codage sur les performances

Dans cette dernière partie, nous étudions l'effet du rendement de codage R sur les performances des codes LDPC irréguliers. Pour cela, on prend N=500 et on considère un décodage Min-Sum avec 20 itérations.

La Figure 3.22 montre une comparaison des performances entre les codes LDPC irréguliers de rendement R égal respectivement à 1/4, 1/2 et 4/5 en terme de BER en fonction du rapport signal sur bruit SNR. D'après les résultats obtenus, on voit bien que le code LDPC irrégulier de rendement R=1/4 est le plus performant, puis vient celui avec R=1/2 et enfin R=4/5.

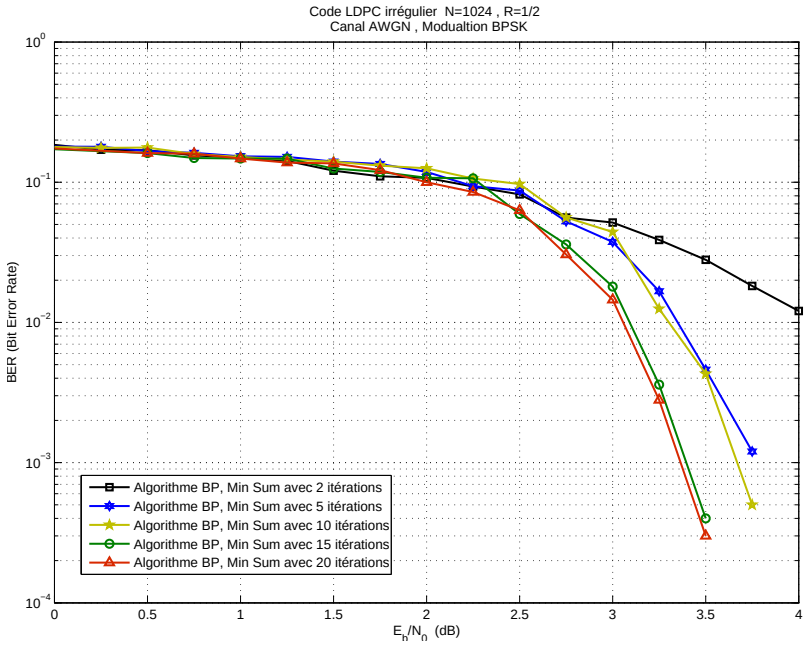


FIGURE 3.21 – Influence du nombre d'itérations sur les performances des codes LDPC pour N=1024 et un décodage Min-Sum .

Pour un BER $=10^{-3}$, le gain apporté en SNR (et non pas en E_b/N_0) par le code de rendement R=1/4 est de 2 dB par rapport à celui de R=1/2 et de 5.9 dB par rapport à celui de R=4/5. On peut interpréter ces résultats par le fait que pour R=1/4, on associe 3 bits de redondance à chaque bit de données, et pour R=1/2, à chaque bit de données est associé un bit de redondance. Alors que, pour R=4/5, pour 4 bits de données, on associe 1 seul bit de redondance. Delà, on constate que plus on augmente le nombre de bits redondants, plus les performances sont meilleures. Mais d'un côté, on perd en débit et en efficacité spectrale du système.

On voit bien que le critère BER en fonction du SNR ne montre pas cet inconvénient. C'est pour cette raison qu'on préfère le critère BER en fonction de E_b/N_0.

On rappelle que :

$$\frac{E_b}{N_0} = \frac{SNR}{R} \tag{3.51}$$

La Figure 3.23 montre une comparaison en terme de BER en fonction de E_b/N_0. Cette fois-ci, c'est le code LDPC irrégulier de rendement R=1/2 qui montre de meilleures performances. D'après ces résultats, il s'avère qu'un code LDPC de rendement R=1/2 représente un bon compromis entre la protection contre les erreurs et l'efficacité spectrale, cela veut dire qu'un seul bit redondant pour chaque bit informatif est suffisant pour assurer une bonne protection contre les erreurs et une bonne efficacité spectrale du système. Ce résultat affirme pourquoi tous les

standards utilisant les codes LDPC n'introduisent jamais les codes de rendement R=1/4 (Voir le Tableau 3.6).

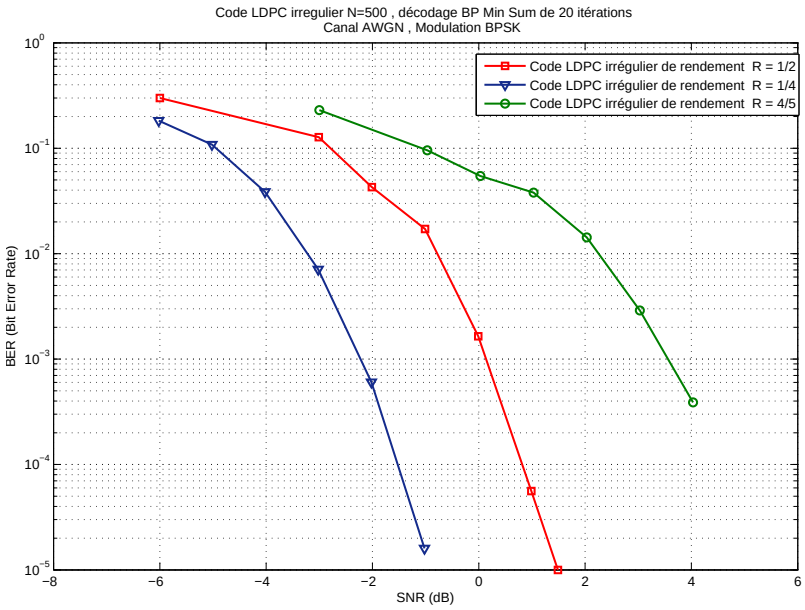


FIGURE 3.22 – Influence du rendement R sur les performances des codes LDPC (BER =f(SNR)).

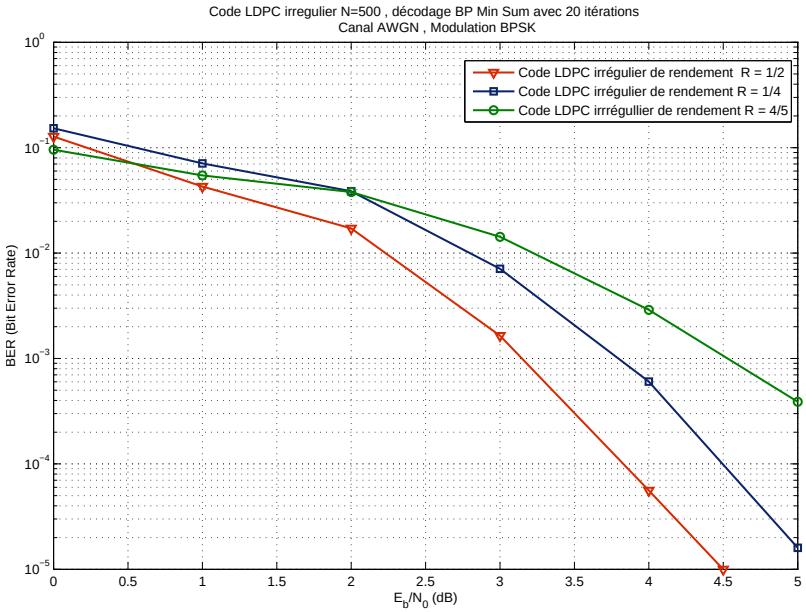


FIGURE 3.23 – Influence du rendement R sur les performances des codes LDPC (BER =f(E_b/N_0)).

3.9 Conclusion

Ce chapitre a fait l'objet d'une étude détaillée des codes LDPC. Nous avons commencé cette étude par l'introduction de l'algorithme de propagation de croyance et ses algorithmes dérivés (sous optimaux), puis l'algorithme d'encodage de Radford Neal. Après avoir présenté tous ces algorithmes, nous avons effectué des simulations afin d'évaluer les performances des codes LDPC dans un canal gaussien. Et à partir des résultats de simulations, nous avons tiré les conclusions suivantes :

- Le décodage Soft est beaucoup plus performant que le décodage Hard.
- L'algorithme sous optimal Min-Sum représente une très bonne approximation de l'algorithme de décodage Log-Domain.
- L'implémentation de l'algorithme Min-Sum dans des applications en temps réel représente une bonne solution, contrairement à l'algorithme Log-Domain.
- L'augmentation de la taille du ode LDPC entraîne une amélioration de performances, cette amélioration est remarquable surtout pour $N \geq 1000$.
- L'augmentation du nombre d'itérations améliore les performances de décodage jusqu'à un nombre d'itérations maximum où l'algorithme de décodage converge.
- Plus R est petit plus la protection contre les erreurs est meilleure, mais à partir d'un certain seuil, le système commence à perdre en efficacité spectrale.

A partir de tous ces résultats, la conclusion générale tirée de toute cette étude est la puissance des codes LDPC irréguliers surtout pour des tailles grandes. Leur puissance ainsi que leur efficacité sont à l'origine de la multitude d'applications de ces codes dans les systèmes de communication émergeants. En se basant sur ces résultats pour le choix des paramètres des codes LDPC, nous allons appliquer, dans le chapitre suivant, ces codes dans un système MIMO.

Chapitre 4

Application des codes LDPC dans un système MIMO

Dans ce chapitre, nous allons implémenter le codeur/décodeur LDPC dans un système MIMO (Multiple Input Multiple Output en anglais) et voir ce qu'il apporte par rapport au codeur/décodeur convolutif classique. Pour cela, nous allons tout d'abord définir le canal radio mobile en précisant ses imperfections, puis nous introduirons brièvement les systèmes MIMO et les gains apportés par rapport au cas mono antenne à l'émission et à la réception SISO (Single Input Single Output en anglais). Ensuite, nous présenterons les différents codes espace-temps retenus pour notre étude. Dans la deuxième partie de ce chapitre, nous montrerons d'abord le contexte de l'application, ensuite, nous étudierons les performances des codes LDPC appliqués dans un système MIMO. Pour cela, nous commencerons par modéliser le canal radio mobile, puis nous décrirons la chaîne de transmission et les conditions de simulations, et enfin, nous présenterons nos résultats ainsi que nos conclusions.

4.1 Introduction

Une des caractéristiques essentielles des futurs systèmes de télécommunications sans-fil est l'évaluation vers les débits élevés tout en envisageant des mobilités de plus en plus importantes des usagers. Face aux imperfections du canal radio mobile, l'usage des réseaux d'antennes et des techniques de traitement d'antennes s'avère très efficaces pour préserver la qualité de transmission des données. En particulier, une technique récente consiste à utiliser des antennes multiples à la fois à l'émission et à la réception, connue sous le nom de systèmes MIMO. Les systèmes MIMO permettent d'atteindre des taux de transmission très élevés, et de ce fait, ils sont considérés comme l'une des voies les plus prometteuses pour les nouvelles générations de communications mobiles (la $4^{ème}$ génération de téléphonie mobile) et les prochaines normes du standard WIMAX.

4.2 Le canal radio mobile

Comme on l'avait défini auparavant, le canal de propagation décrit le support physique de la transmission, qui peut être invariant dans le temps[1] (e.g : le cas d'une transmission sur câble) ou variant dans le temps (e.g : cas d'une transmission dans un canal radio mobile). La variation dans le temps de ce dernier est due essentiellement à la propagation par trajets multiples et à la mobilité du récepteur [13] (voir Figure 4.1). En plus de la variation dans le temps, le canal radio mobile souffre de plusieurs imperfections comme l'évanouissement (Fading en anglais), l'interférence, l'effet Doppler ...etc.

4.2.1 Propagation multi-trajet

La propagation multi-trajet se produit par suite de réflexion, dispersion et de diffraction de l'onde électromagnétique transmise à travers différents obstacles (bâtiments, montagnes, arbres...etc.). Ainsi à la réception, beaucoup de signaux arrivent de différentes directions avec des retards, des atténuations et des phases diverses. La superposition de ces derniers donne des variations d'amplitude et de phase du signal reçu, qui rend difficile la récupération du signal d'information d'origine [13], [64].

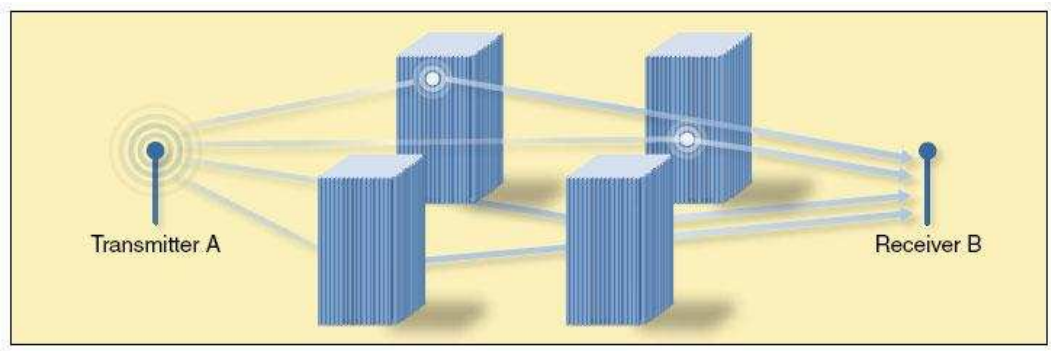


FIGURE 4.1 – La propagation multi-trajet dans un milieu urbain.

Si on suppose que le canal est stationnaire au sens large, c'est-à-dire que les caractéristiques du canal restent constantes pendant de courtes périodes de temps ou distances spatiales, la réponse impultionnelle d'un canal radio mobile est donnée par la relation suivante :

$$h_c(\tau, t) = \sum_{p=0}^{N_p-1} a_p \exp j(2\pi f_{D_p} t + \phi_p)\delta(\tau - \tau_p) \tag{4.1}$$

où :

1. Un système est variant dans le temps si sa réponse impulsionnelle varie avec le temps, cette variation est due au changement continuel des caractéristiques physiques du milieu de transmission.

$$\delta(\tau - \tau_p) = \begin{cases} 1 & \text{si} \quad \tau = \tau_p \\ 0 & \text{sinon} \end{cases} \tag{4.2}$$

et a_p, f_{D_p}, ϕ_p et τ_p [2] sont respectivement l'amplitude, la fréquence de Doppler, la phase et le retard de propagation associés au trajet p ($p = 0, \cdots, N_p - 1$).

La fréquence Doppler représente le décalage de la fréquence centrale f_c du signal transmis, et elle est causée généralement par la mobilité du récepteur [13].

$$f_{D_p} = \frac{v\, f_c\, cos(\alpha_p)}{\nu_c} \tag{4.3}$$

où :

v : la vitesse de déplacement du récepteur.

ν_c : la vitesse de propagation de l'onde électromagnétique, dans l'air $c = 3.10^8$.

α_p : l'angle entre $\overrightarrow{v}$ (vitesse de déplacement) et $\overrightarrow{k}$ (direction de propagation de l'onde électromagnétique).

En appliquant la transformée de Fourier (TF) à la relation (4.1), on obtient la fonction de transfert du canal [3] :

$$H_c(f,t) = \sum_{p=0}^{N_p-1} a_p \exp j(2\pi(f_{D_p} t - f\tau_p) + \phi_p) \tag{4.4}$$

La Figure 4.2 représente un exemple de la réponse impultionnelle et la fonction de transfert d'un canal radio mobile.

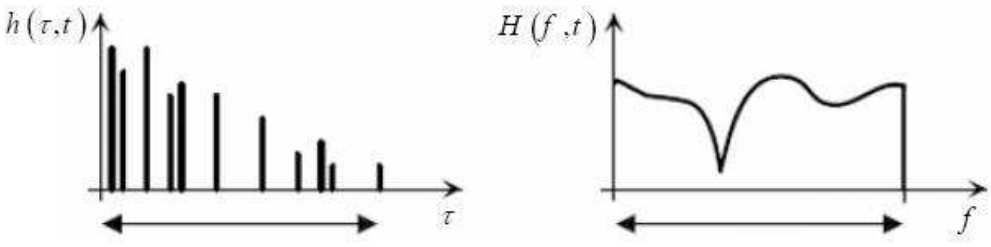


FIGURE 4.2 – Variation de la réponse impulsionnelle et de la fonction de transfert dans le temps.

2. Toutes ces grandeurs sont variables dans le temps.
3. On note par H_c la fonction de transfert du canal pour la distinguer de la matrice de contrôle de parité H.

4.2.2 Evanouissement du canal

L'évanouissement ou le Fading peut être défini comme étant la variation de la puissance du signal en fonction du temps ou de la distance. On peut distinguer deux types de Fading :

Fading à court terme

Ce sont les fluctuations de la puissance du signal reçu sur un intervalle de temps ou un déplacement suffisamment petit (voir Figure 4.3). Il est causé par le multi-trajet, l'effet Doppler et la bande passante du signal émis.

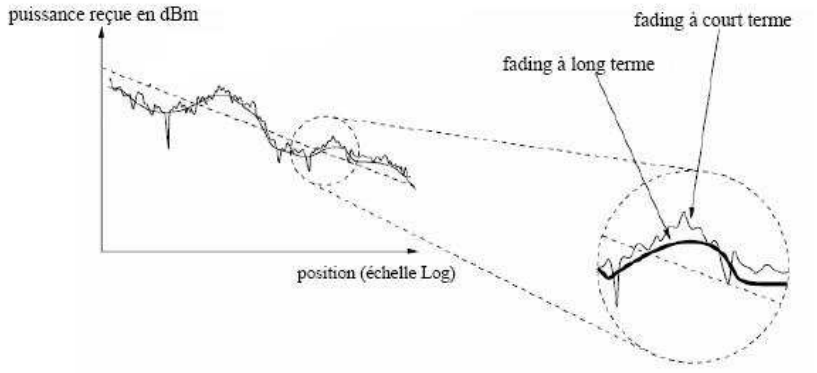


FIGURE 4.3 – Illustration des deux types de Fading.

Fading à long terme

Ce sont les fluctuations de la puissance du signal reçu, mesurée sur un déplacement de plusieurs dizaines de longueurs d'onde ou sur un intervalle de temps suffisamment grand (voir Figure 4.3), il est appelé aussi « Log-Normal Fading » ou bien « Shadowing ». Il est dû principalement à l'effet de masque généré par des obstacles (montagnes, arbres, immeubles, ...etc.).

4.3 Le principe des systèmes MIMO

Avant, l'effet de multi-trajet était vu comme étant un point négatif du canal radio mobile. Mais en 1995 Telatar [65] a montré que cet effet peut être exploité pour accroître le débit de transmission des systèmes de communication sans-fil, et cela, par l'introduction d'un système multi-antennes à l'émission et à la réception (Mutiple Input Multiple Output MIMO en anglais). Le principe de base des systèmes MIMO consiste à combiner les signaux judicieusement tant à l'émission qu'à la réception pour exploiter la diversité spatiale qui réduit les effets

d'évanouissements, ou pour augmenter le débit de transmission.

Les systèmes MIMO présentent deux avantages majeurs par rapport aux systèmes composés d'une seule antenne à l'émission et à la réception (SISO) :

1. Amélioration de la qualité du lien en s'affranchissant des évanouissements du canal, et cela grâce à l'apport de diversité spatiale (voir section 4.5.1).
2. Augmentation du débit d'information sans augmenter la bande passante ou la puissance du signal émis, et cela grâce au multiplexage spatial (voir section 4.5.2).

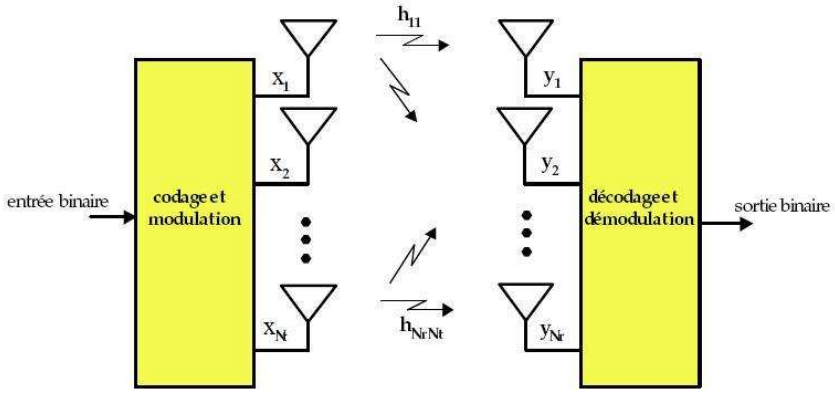


FIGURE 4.4 – Système de transmission sans fil MIMO.

Soit un système MIMO composé de N_t antennes à l'émission et de N_r antennes à la réception comme illustré sur la Figure 4.4. Le signal reçu y_j à chaque instant sur la j-ième antenne de réception est la somme des symboles bruités issus des N_t signaux transmis [5] :

$$y_j = \sum_{i=1}^{N_t} h_{ij} \varkappa_i + n_j \tag{4.5}$$

où h_{ij} est le gain complexe du canal non sélectif en fréquence entre l'antenne d'émission i et l'antenne de réception j, $\varkappa_i$ représente le symbole émis à partir de l'antenne i et n_j est le bruit additif qui est modélisé par des échantillons indépendants et suivant une loi gaussienne centrée de variance $\sigma^2 = N_0/2$ par dimension réelle.

Soit la matrice du canal MIMO H_M [4] de dimension (N_r,N_t) suivante :

4. On utilise cette notation pour la distinguer de la matrice de contrôle de parité H définie précédemment.

$$H_M = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1N_t} \\ h_{21} & h_{22} & \cdots & h_{2N_t} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_r1} & h_{N_r2} & \cdots & h_{N_rN_t} \end{bmatrix} \tag{4.6}$$

La relation (4.5) peut s'écrire sous la forme matricielle suivante :

$$y \;=\; H_M \,\varkappa \;+\; n \tag{4.7}$$

où $\varkappa$, y et n sont respectivement les vecteurs d'émission de dimension (N_t x1), de réception de dimension (N_r x 1) et de bruit de dimension (N_r x 1).

4.4 Capacité d'un canal MIMO

La capacité du canal est un paramètre important pour l'évaluation des performances des canaux MIMO [5]. En effet, pour une liaison donnée, elle permet de connaître la quantité maximum d'information en Shannon/s/Hz (ou bits/s/Hz) qu'il est possible de transmettre sur un canal de propagation.

Pour un canal MIMO de N_t antennes à l'émission et N_r antennes à la réception tel que $N_t \leq N_r$ et les puissances des antennes émettrices sont identiques, la capacité est définie par l'équation suivante :

$$\mathcal{C}(\rho, N_t, N_r) = \log_2(det(I_{N_r} + \frac{\rho}{N_t} H_M H_M^H)) \tag{4.8}$$

où $\rho = E_s/N_0$ est le rapport d'énergie par symbole sur la densité de puissance du bruit et I_{N_r} est une matrice identité (N_r, N_r) [65], [66].

En utilisant la décomposition en valeurs singulières (SVD) (voir Annexe B), un canal MIMO défini par la matrice H_M peut être décomposé en r canaux SISO parallèles, dont la puissance du signal est donnée par leurs valeurs propres. Ainsi, la relation (4.8) peut être écrite de la manière suivante :

$$\mathcal{C}(\rho) = \sum_{i=1}^{r} \log_2(1 + \frac{\rho}{N_t}\lambda_i) \tag{4.9}$$

où λ_i $(i = 1, \cdots, r)$ sont les valeurs propres non nulles de $H_M^H H_M$ [5].

Comme la capacité instantanée donnée en (4.8) et (4.9) est une variable aléatoire. Il est préférable d'utiliser une forme plus pratique pour la décrire, pour cela, on utilise la capacité

5. $(.)^H$ est l'opérateur Hermitien qui signifie la transposée conjuguée de la matrice.

moyenne ou ergodique. La capacité moyenne ou ergodique s'obtient en calculant l'espérance sur toutes les réalisations possibles du canal MIMO.

$$\mathcal{C}(\rho, N_t, N_r) = E\{\log_2 det(I_{N_r} + \frac{\rho}{N_t} H_M H_M^H)\} \tag{4.10}$$

Sur la Figure 4.5 nous présentons la capacité ergodique pour plusieurs configurations (variation du nombre d'antennes à l'émission et à la réception) en fonction du rapport E_s/N_0. Les canaux de transmission utilisés pour évaluer ces capacités sont des canaux non sélectifs en fréquence, indépendants et identiquement distribués (i.i.d) suivant la loi de Rayleigh et varient pour chaque bloc. On voit bien que pour chaque accroissement de 3 dB, la capacité du canal MIMO augmente de $min(N_t, N_r)$ Shannon/s/Hz. Delà, on constate que la capacité du canal MIMO croît linéairement avec $min(N_t, N_r)$ [65], [67].

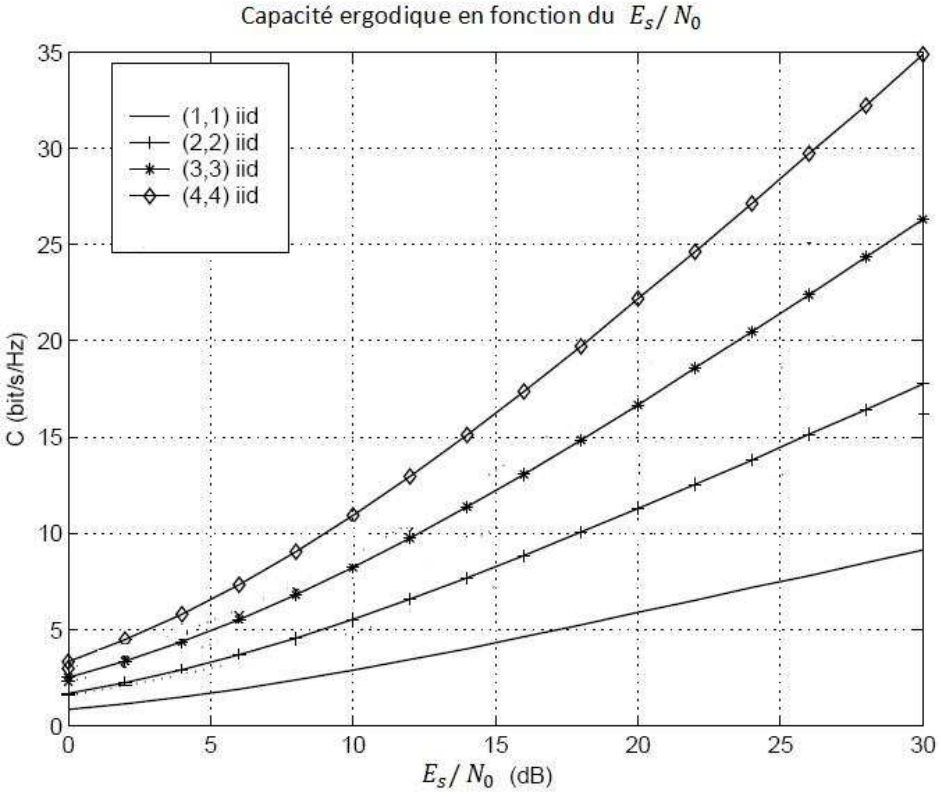


FIGURE 4.5 – Capacité ergodique pour différentes configurations du canal MIMO (Courbes reproduites de la référence [5]).

4.5 Les gains apportés par des systèmes MIMO

Dans cette section, nous présentons les différents gains apportés par la diversité spatiale et le multiplexage spatial.

4.5.1 Gain de diversité spatiale

Dans un canal de transmission sans fil, la puissance reçue varie dans le temps, en fréquence et dans l'espace. La diversité est alors utilisée pour combattre ces évanouissements. L'idée principale consiste à utiliser à la réception, plusieurs répliques du signal (exploiter l'effet de multi-trajet). Plus le nombre de répliques augmente, plus la probabilité que toutes les répliques subissent simultanément un évanouissement diminue. Classiquement dans les systèmes SISO, les diversités temporelle et fréquentielle sont exploitées par le codage de canal. Avec les systèmes MIMO, on dispose d'une nouvelle forme de diversité : *la diversité spatiale*. L'ordre de diversité est fonction du nombre d'antennes à l'émission et à la réception, par exemple : pour N_t=2 et N_r=1, l'ordre de diversité est de 2 et pour N_t=2 et N_r=2, l'ordre de diversité spatiale est égal à 4 [68]. En pratique, pour mettre à profit cette diversité, il est nécessaire d'utiliser un code spatio-temporel (voir section 4.6).

4.5.2 Gain de multiplexage

Comme mentionné dans la section précédente, lorsque les évanouissements des différents canaux sont indépendants, il est alors possible de voir le canal MIMO comme un ensemble de canaux SISO en parallèle. En transmettant des flux d'information dans chacun de ces canaux, il est possible d'augmenter le débit d'information. Il en résulte un gain dit de multiplexage [5].

Il existe deux approches pour exploiter le potentiel des canaux MIMO : le multiplexage spatial et le codage spatio-temporel. Dans le multiplexage spatial, des flux de données indépendants sont transmis sur les différentes antennes d'émission, maximisant ainsi le débit transmis. A l'opposé, les codes spatio-temporel offrent à la fois, de la diversité et du gain de codage tout en améliorant l'efficacité spectrale.

4.6 Le codage spatio-temporel

Pour un système MIMO composé de N_t antennes à l'émission et de N_r antennes à la réception dont les coefficients du canal MIMO restent constantes pendant T intervalles de temps élémentaires (modèle de canaux à évanouissement par bloc). A l'émission, les symboles d'information s_i appartenant à l'alphabet A_s, sont groupés dans un vecteur $s = [s_1, s_2, \cdots, s_Q]^t$ de dimension (Q x 1). En appliquant un codage espace-temps, on associe s à la matrice code suivante de

dimension N_t x T [68] :

$$X = \begin{bmatrix} \varkappa_{11} & \cdots & \varkappa_{1T} \\ \vdots & \vdots & \ddots \quad \vdots \\ \varkappa_{N_t 1} & \cdots & \varkappa_{N_t T} \end{bmatrix} \tag{4.11}$$

où le symbole $\varkappa_{ij}$ appartient à l'alphabet $A_{\varkappa}$.
On définit le rendement du code espace-temps R_{STC} par le rapport entre la taille de l'alphabet A_s et le nombre d'intervalles de temps T (channel use en anglais).

$$R_{STC} = \frac{Q}{T} \tag{4.12}$$

En considérant que le canal est non sélectif en fréquence. A partir de la relation 4.5, on peut écrire la relation matricielle suivante :

$$Y = H_M X + N \tag{4.13}$$

où Y et N[6] sont respectivement les matrices de réception et de bruit de dimension (N_r x T).

D'une façon générale, les codes spatio-temporels peuvent être classés en deux catégories : les codes orthogonaux et les codes non orthogonaux .

Dans notre cas, on s'intéresse au code orthogonal d'Alamouti qui est considéré parmi les codes appelés STBC (Space-Time Block Code) [69], [70], ainsi qu'au schéma non-orthogonal simple de multiplexage spatial.

4.6.1 Le code orthogonal d'Alamouti

Alamouti a proposé dans [71] une méthode simple pour exploiter la diversité espace-temps en utilisant une construction très simple à deux antennes d'émission. Chaque antenne enverra les deux symboles codés qui lui sont fournis l'un après l'autre. Afin d'obtenir les symboles orthogonaux entre les antennes, la première antenne enverra les symboles d'informations de façon systématique et la deuxième antenne enverra ces symboles conjugués, dans l'ordre inverse, et avec l'un des deux inversés (voir Figure 4.6).

On considère le cas $N_t = 2$, $Q = T = 2$ et donc $R_{STC} = 1$. A l'instant 1, les symboles s_1 et s_2 sont transmis respectivement sur les antennes 1 et 2, puis à l'instant 2, les symboles $-s_2^*$ et s_1^* sont transmis sur les antennes 1 et 2. Ainsi sous forme matricielle, on a :

$$X_{Al} = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix} \tag{4.14}$$

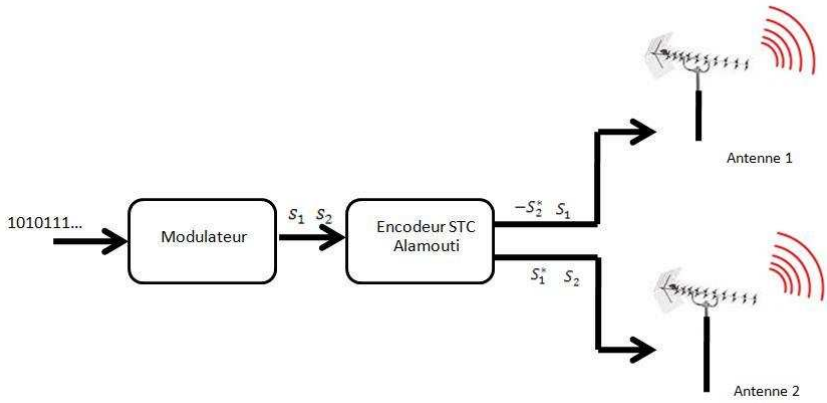


FIGURE 4.6 – Schéma d'un codeur d'Alamouti $N_t = 2$ et $Q = T = 2$.

La propriété d'orthogonalité est donnée par la relation suivante :

$$X_{Al}.X_{Al}^{H} = (|s_1|^2 + |s_2|^2) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.15}$$

Cette propriété permet d'éliminer au niveau des antennes de réception les interférences inter-antennes, contrairement au cas des codes non-orthogonaux.

4.6.2 Le multiplexage spatial

Le multiplexage spatial permet de maximiser le débit d'information sur un canal MIMO et cela par la répartition des symboles sur les antennes d'émission. Ce schéma est celui utilisé dans l'architecture BLAST (Bell Labs Layered Space-Time) [67].

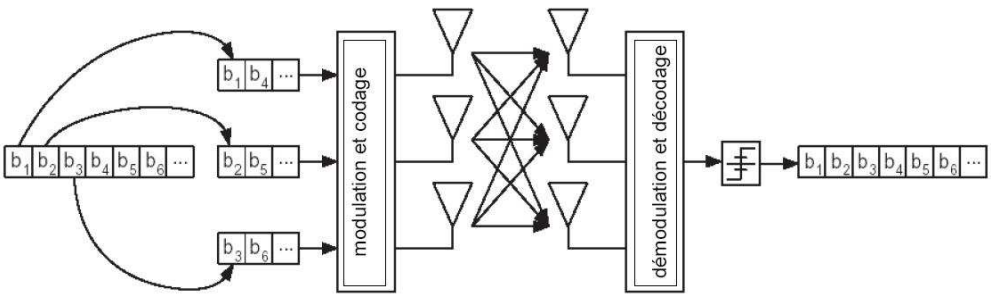


FIGURE 4.7 – Synoptique d'un schéma de multiplexage spatial.

Pour la suite, nous allons considérer un schéma de multiplexage spatial (MUX) où $N_t = 2$,

6. On a mis Y et N en majuscules pour spécifier des matrices.

Q = 2, T = 1 et donc $R_{STC} = 2$, la matrice de code est la suivante :

$$X_{MUX} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \tag{4.16}$$

Quant au décodage espace-temps, on distingue plusieurs techniques de détection , comme le décodage linéaire et le décodage par soustraction successive d'interférences [68] [47] .

Dans le cas de notre étude, nous allons utiliser la technique de détection MMSE (Minimum Mean Square Error) faisant partie du décodage linéaire (voir Annexe C).

4.7 Le contexte de l'application

Comme on l'a spécifié précédemment, notre travail rentre dans les travaux de recherche de l'équipe GSM sur les codes correcteurs d'erreurs appliqués aux systèmes MIMO. Le contexte de ce travail est sur le standard WIMAX, qui envisage d'introduire les codes LDPC et les systèmes MIMO dans la prochaine norme IEEE 802.16 m [61]. Pour cela, une première étude a été faite par mon encadreur le Pr. M. A. Khalighi sur les systèmes MIMO en utilisant un codeur/décodeur convolutif classique.

Notre application consiste donc à enlever le codeur/décodeur convolutif et à le remplacer par un codeur/décodeur LDPC irrégulier. Ensuite, nous allons évaluer ses performances et voir l'apport d'un code correcteur d'erreur puissant sur les performances du système.

4.8 Modélisation du canal radio mobile

Dans cette section, Une modélisation du canal de transmission radio mobile est présentée. pour cela, nous allons adopter les hypothèses suivantes :

• On considère des canaux à entrées et sorties multiples (N_t antennes à l'émission et N_r antennes à la réception) dont les coefficients de Fading (à court terme) de ces canaux sont indépendants et identiquement distribués (i.i.d) suivant la loi de Rayleigh définie comme suit :

$$f(t, \sigma_r) = \frac{t}{\sigma_r^2} \exp(\frac{-t^2}{2\sigma_r^2}) \tag{4.17}$$

où t représente le coefficient de Fading et σ_r l'écart type.

• Le modèle du canal sans fil qu'on va utiliser, est *le canal à évanouissement par blocs* (block Fading channel), qui doit son nom au fait que le canal change N_c fois par trame, ce nombre de changements N_c est appelé *l'ordre de diversité temporelle* qui peut être exploité pour réduire le Fading à la réception [68].

Par exemple : si on a $N_c = 4$, alors le Fading change 4 fois pour chaque trame envoyée à travers le canal d'une manière indépendante et identiquement distribuée (i.i.d).
pour $N_c = 1$, on retrouve le cas d'un canal *quasi-stationnaire*, pour lequel le canal reste inchangé pendant la transmission de toute la trame.
On suppose aussi que le changement du Fading d'un bloc à l'autre se fait d'une manière instantanée et indépendamment de la valeur précédente.

• Sous l'hypothèse d'une modulation à bande étroite (canal non sélectif en fréquence), sans interférences inter-symboles, le modèle équivalent en bande de base est :

$$Y_k = \theta H_{M_k} X_k + \sigma N_k \qquad k = 1, 2, \cdots, N_c \tag{4.18}$$

l'indice k indique le bloc, Y_k est la matrice (N_r x T) du signal reçu, X_k est la matrice (N_t x T) du signal envoyé, H_{M_k} est la matrice (N_r x N_t) qui collecte les coefficients du canal et N_k est la matrice (N_r x T) du bruit additif blanc et gaussien. Les nombres réels positifs θ et σ sont utilisés pour normaliser la puissance du signal utile et la variance du bruit. Les coefficients de H_{M_k} et N_k sont des variables gaussiennes complexes à symétrie circulaire, indépendant et identiquement distribués (i.i.d.) avec une espérance nulle et une variance égale à 1.

• On suppose que le canal est normalisé (c'est-à-dire qu'il n'apporte pas de gain ni d'atténuation) et qu'il est parfaitement connu à la réception.

• On considère le cas d'une transmission mono-porteuse, c'est-à-dire que les symboles sont transmis sur une seule fréquence.

• On considère le cas mono-utilisateur, c'est-à-dire qu'il n'y a pas d'interférences inter-utilisateurs.

4.9 Etude des performances des codes LDPC dans une architecture MIMO

Dans cette section, une évaluation des performances des systèmes MIMO en utilisant des codes LDPC irréguliers est présentée. Pour cela, nous allons implémenter le codeur/décodeur LDPC programmé en C (dont les paramètres du codage et du décodage sont choisis en se basant sur les résultats obtenus dans le chapitre 3) dans la chaîne de transmission, illustrée par les Figures 4.8 et 4.9, programmée également en langage C.

L'organisation du reste de cette section est comme suit :

– Description de la chaîne de transmission.

- Définition des conditions de simulation ainsi que les hypothèses.
- Présentation des résultats de simulations.
- Interprétations et conclusions.

En plus du taux d'erreur binaire (BER) défini dans la section 3.8, Nous allons utiliser un nouveau critère d'évaluation de performance appelé *taux d'erreur par trame* FER (Frame Error Rate en anglais), défini par la relation suivante :

$$\text{FER} = \frac{\text{Nombre de trames erronées}}{\text{Nombre de trames transmises}} \tag{4.19}$$

En utilisant le codage de canal et le codage espace-temps dans un système MIMO, le rendement global du système sera :

$$R_{Glob} = R.R_{STC} \tag{4.20}$$

où :
R : Le rendement du code correcteur d'erreurs.
R_{STC} : Le rendement du code espace-temps.

4.9.1 La chaîne de transmission

Les Figures 4.8 et 4.9 montrent respectivement les schémas bloc de la chaîne de transmission à l'émission et à réception.

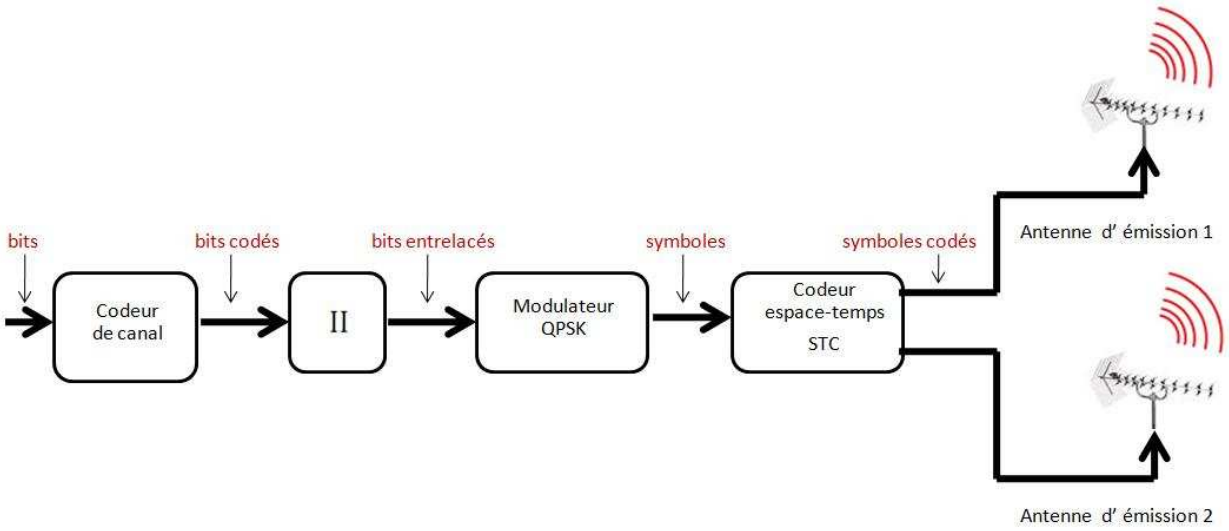


FIGURE 4.8 – Schéma bloc de la chaîne de transmission à l'émission.

A l'émission, on génère une séquence d'information binaire à l'aide d'un générateur pseudo-aléatoire, ensuite cette séquence sera codée par un codeur convolutif de type Non-Récursif Non-Systématique NRNSC ou un codeur LDPC irrégulier basé sur l'algorithme de R. Neal. Le mot de code issu du codeur de canal est ensuite entrelacé à l'aide de l'entrelaceur Π. Une fois que le mot de code est entrelacé, le modulateur QPSK permet d'effectuer une transformation (mapping) bit/symbole. Ces symboles sont ensuite codés par un codeur espace-temps (le code d'Alamouti ou le schéma MUX). Les symboles codés sont par la suite envoyés par les deux antennes émettrices 1 et 2.

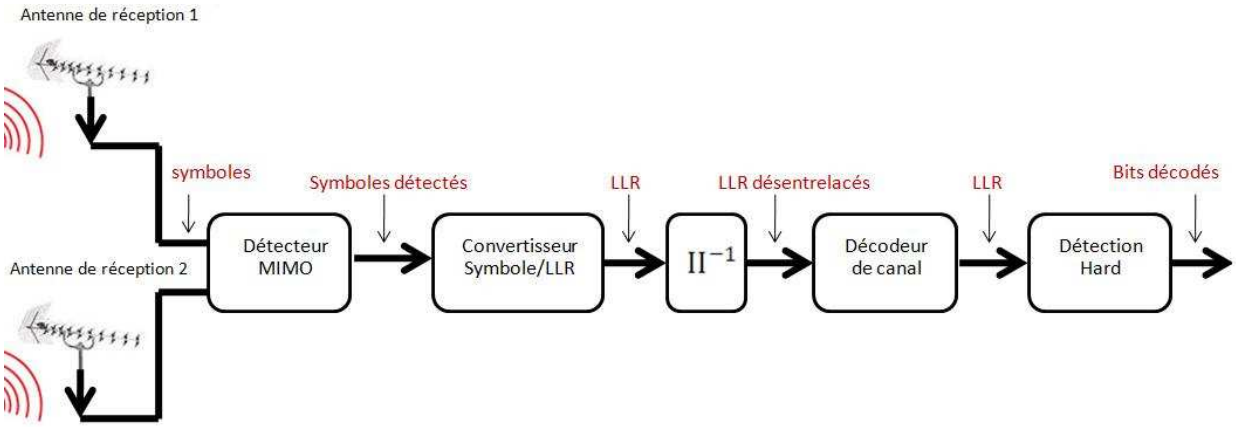


FIGURE 4.9 – Schéma bloc de la chaîne de transmission à la réception.

A la réception, les antennes réceptrices 1 et 2 reçoivent les symboles bruités et interférés. Ensuite, une détection MIMO est faite par la technique MMSE (décodage espace-temps). Puis, les symboles détectés sont convertis en des LLR [7] (Log-rapport de vraisemblance) et désentrelacés par le désentrelaceur Π^{-1}. Ensuite à la sortie du décodeur Max-Log-MAP (le cas des codes convolutifs) ou le décodeur BP Min-Sum (le cas des codes LDPC irréguliers), on obtient des LLR qui vont être transformés en des bits décodés à l'aide de décodage Hard (défini déjà par la relation (3.35) dans la section 3.5.3).

4.9.2 Conditions et hypothèses de simulations

Toutes les conditions et les hypothèses considérées dans les simulations sont résumées dans le Tableau 4.1 :

Remarque : Pour le calcul de E_b/N_0, on tient compte du gain apporté par les antennes de réception, pour cela on rajoute $10\log_{10}(N_r)$.

7. Car le décodeur de canal Max-Log MAP et le décodeur BP Min-Sum travaillent avec les LLR.

Paramètre	Valeur/Caractéristique
Le canal	Canal MIMO (2x2) $N_t = N_r = 2$ Canal à évanouissement par blocs (Block Fading Channel) Canaux i.i.d suivant la loi de Rayleigh Diversité temporelle N_c : 1 et 8 Canal normalisé Estimation parfaite du canal à la réception
Le codage convolutif	Code Non-Récursif Non-Systématique (NRNSC) La taille de la contrainte ν : 3 et 7 Rendement de codage R : 1/2 Algorithme de décodage : Max-Log-MAP
Le codage LDPC	Code LDPC irrégulier La taille du code N : 1024, 2048 et 4096 Rendement de codage R : 1/2 Algorithme de décodage : BP Min-Sum Nombre d'itérations : 20
La modulation	QPSK (Quadrature Phase Shift Keying)
L'entrelacement	La taille de la trame : N Entrelaceur pseudo-aléatoire de taille N
Le codage espace-temps	Le schéma MUX : $Q = 2$, $T = 1$ et $\overset{R}{}_{STC} = 2$ Le code orthogonal d'Alamouti : $Q = 2$, $T = 2$ et $\overset{R}{}_{STC} = 1$ Technique de détection : MMSE
Le critère d'évaluation des performances	BER en fonction de E_b/N_0 FER en fonction de E_b/N_0

TABLE 4.1 – Les paramètres de simulation utilisés pour évaluer les performances des codes LDPC dans un système MIMO.

4.9.3 Résultats et discussions

- **Le choix du code espace-temps**

Les Figures 4.10 et 4.11 présentent respectivement les courbes BER=f(E_b/N_0) et FER=f(E_b/N_0) pour un système MIMO utilisant respectivement un code espace-temps d'Alamouti et le multiplexage spatial (sans codage de canal). On peut voir que le code orthogonal d'Alamouti donne de meilleures performances par rapport au cas du multiplexage spatial. Par exemple, pour un taux d'erreur binaire BER=10^{-4}, le code d'Alamouti apporte un gain de presque 10 dB en terme de E_b/N_0 par rapport au cas de multiplexage spatial.

Ces résultats sont expliqués par le fait que le code d'Alamouti est muni de la propriété d'orthogonalité qui lui permet d'éliminer les interférences inter-antennes, mais au détriment de débit. En effet, le schéma MUX donne un débit deux fois plus grand par rapport à celui du code d'Almouti.

Pour la suite de nos simulations, nous allons considérer le code orthogonal d'Alamouti $M_t = M_r = 2$.

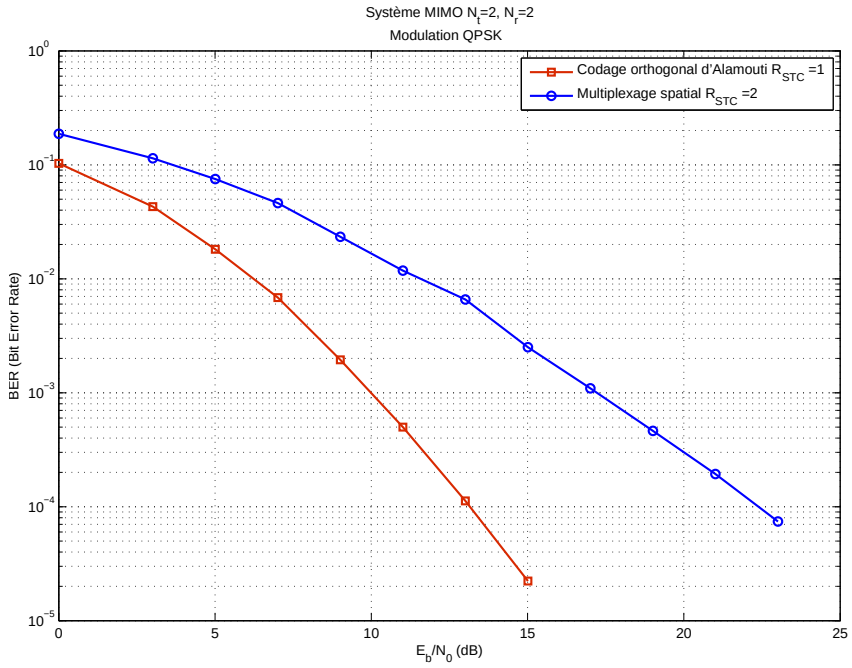


FIGURE 4.10 – Comparaison des performances en terme de BER entre le code orthogonal d'Alamouti et le schéma de multiplexage spatial MUX.

- **Evaluation des performances du codeur/décodeur LDPC**

Dans cette partie, nous présenterons les résultats de simulations pour deux valeurs de diversité temporelles : $N_c = 1$ et $N_c = 8$.

En absence de diversité temporelle $N_c = 1$ (canal quasi-cyclique), les Figures 4.12 et 4.13 montrent respectivement une comparaison des performances en terme de BER et de FER entre les codes convolutifs et les codes LDPC implémentés dans un système MIMO.

En analysant la première figure, c'est-à-dire en terme de BER, on remarque que les codes LDPC apportent une amélioration par rapport au cas des codes convolutifs et cela à partir d'un seuil de convergence égal à 5 dB. Pour un BER=10^{-4}, le code LDPC N=1024 apporte un gain de 1.5 dB en terme de E_b/N_0 par rapport au code convolutif de longueur de contrainte $\nu = 7$

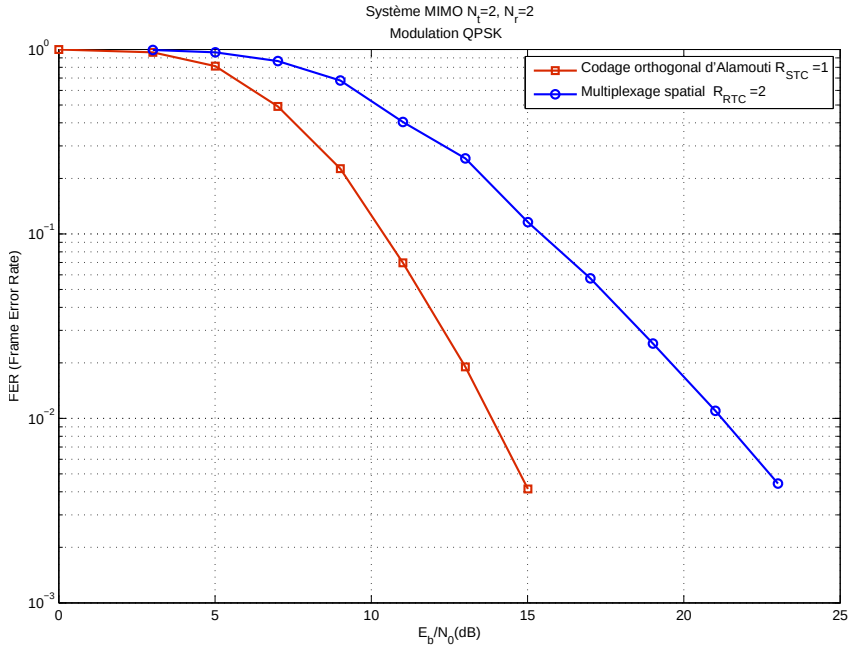


FIGURE 4.11 – Comparaison des performances en terme de FER entre le code orthogonal d'Alamouti et le schéma de multiplexage spatial MUX.

et un gain de 3 dB par rapport au cas sans codage de canal. D'autre part, on constate que l'augmentation de la longueur de contrainte ν des codes convolutifs et la taille N des codes LDPC n'améliorent pas les performances du système dans le cas d'un canal quasi-stationnaire. D'ailleurs, on voit bien que les codes convolutifs de longueur de contrainte ν=3 et ν=7 donnent pratiquement des résultats identiques, et la même tendance pour les codes LDPC de taille N= 1024, 2048 et 4096.

Si on fait la comparaison en terme de FER, on remarque que l'amélioration est un peu plus remarquable par rapport à celle obtenue en terme de BER. Par exemple, pour un FER=10^{-3} le code LDPC de taille N=1024 apporte respectivement un gain de 2 dB, 3 dB et environ 7 dB en terme de E_b/N_0 par rapport aux codes convolutifs de longueur de contrainte $\nu = 7$, $\nu = 3$ et par rapport au cas sans codage de canal. Cette fois-ci on voit bien que l'augmentation de la taille de la contrainte ν entraîne une amélioration de performances (entre $\nu = 7$ et $\nu = 3$, le gain est de 2 dB pour un FER=10^{-3}). Par contre, l'augmentation de la taille N des codes LDPC n'apporte pas vraiment une grande amélioration.

D'après les résultats qu'on a obtenu pour le cas d'un canal quasi-stationnaire, on constate que l'implémentation du codeur/décodeur LDPC a amélioré les performances du système MIMO (2x2) par rapport au cas d'un codeur/décodeur convolutif simple. Mais, par rapport à l'étude faite sur les codes LDPC dans le chapitre précédent et les résultats impressionnants qu'on a

obtenu dans le cas d'un canal gaussien, on voit bien que cette amélioration n'est pas aussi importante, ceci est dû essentiellement à l'évanouissement fort du canal (Fading). Car, dans un canal quasi-stationnaire ($N_c = 1$), les caractéristiques du canal restent constantes le long de la transmission d'une trame, et lorsque les trames transmises subissent un Fading fort, ni le codage de canal, ni l'entrelacement, ni non plus le codage espace-temps ne peuvent corriger totalement les erreurs introduites dans les trames transmises via le canal radio mobile. Ces trames perdues entraînent par la suite une augmentation significative du taux d'erreur binaire BER, c'est la raison pour laquelle on ne voit pas bien l'effet des codes LDPC dans la Figure 4.12. C'est à cause de cela aussi, que généralement il est préférable d'utiliser le critère FER en fonction de E_b/N_0 pour évaluer les performances d'un code correcteur d'erreur lorsque le Fading est important.

Les Figures 4.14 et 4.15 montrent respectivement une comparaison en terme de BER et de FER pour un canal de diversité temporelle $N_c = 8$ (c'est-à-dire que les caractéristiques du canal changent 8 fois lors de la transmission d'une trame). Cette fois-ci, on voit clairement l'amélioration des performances. Si on compare ces résultats avec les résultats précédents, on voit bien que pour un $E_b/N_0 = 6$ dB, le BER est de l'ordre de 10^{-6} et le seuil de convergence se situe à environ 4 dB, alors que pour $N_c = 1$, le BER est de l'ordre de 10^{-2} et le seuil de convergence se situe à environ 5 dB.

D'autre part, on remarque que la pente des courbes BER= f(E_b/N_0) et BER= f(E_b/N_0) est plus grande pour le cas des codes LDPC, ce qui entraîne un gain plus important en E_b/N_0. Par exemple pour un BER = 10^{-6}, le code LDPC de taille N=4096 apporte respectivement un gain de 3 dB, 5 dB en E_b/N_0 par rapport aux codes convolutifs de longueur de contrainte $\nu = 7$ et $\nu = 3$. Et pour un BER = 10^{-5}, le code LDPC de taille N=4096 apporte un gain de 10 dB en E_b/N_0 par rapport au cas sans codage.

A partir des résultats présentés en Figures 4.14 et 4.15, on remarque aussi que les performances obtenues en terme de BER et de FER sont semblables (même allure), cela peut être expliqué par le fait que la diversité temporelle est grande, et grâce au codage de canal et l'entrelacement, on arrive à moyenner sur le Fading. Par exemple : pour une diversité temporelle égale à 8 (c'est-à-dire que le coefficient de Fading change 8 fois lors de la transmission d'une trame sur le canal), on suppose que 1/8 de la trame est perdu à cause d'un Fading fort, même si on n'arrive pas à récupérer cette partie, ces erreurs n'influent pas considérablement sur le taux d'erreur binaire BER, contrairement au cas d'un canal quasi-stationnaire.

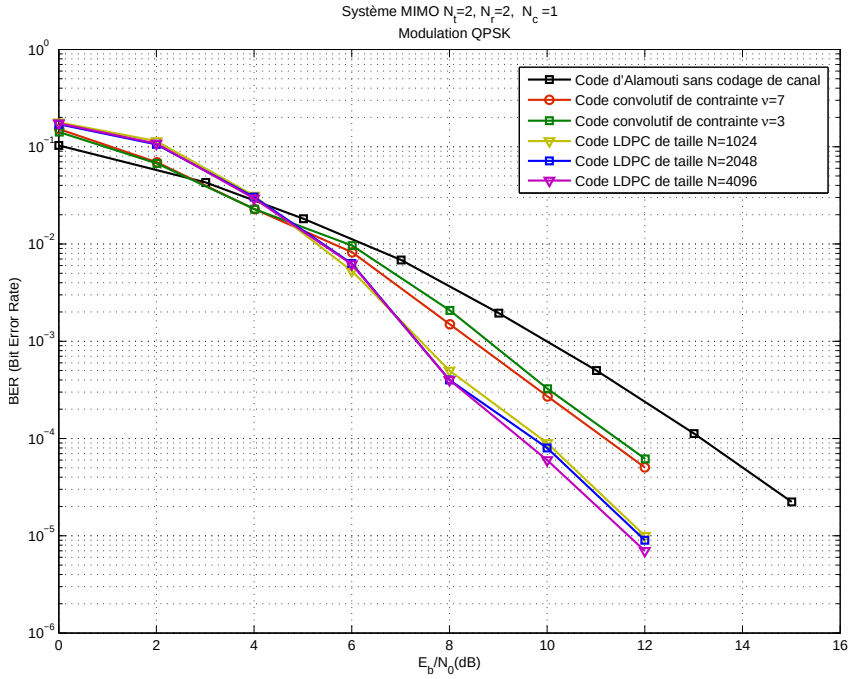


FIGURE 4.12 – Comparaison des performances en terme de BER entre les codes convolutifs et les codes LDPC pour une diversité temporelle $N_c = 1$.

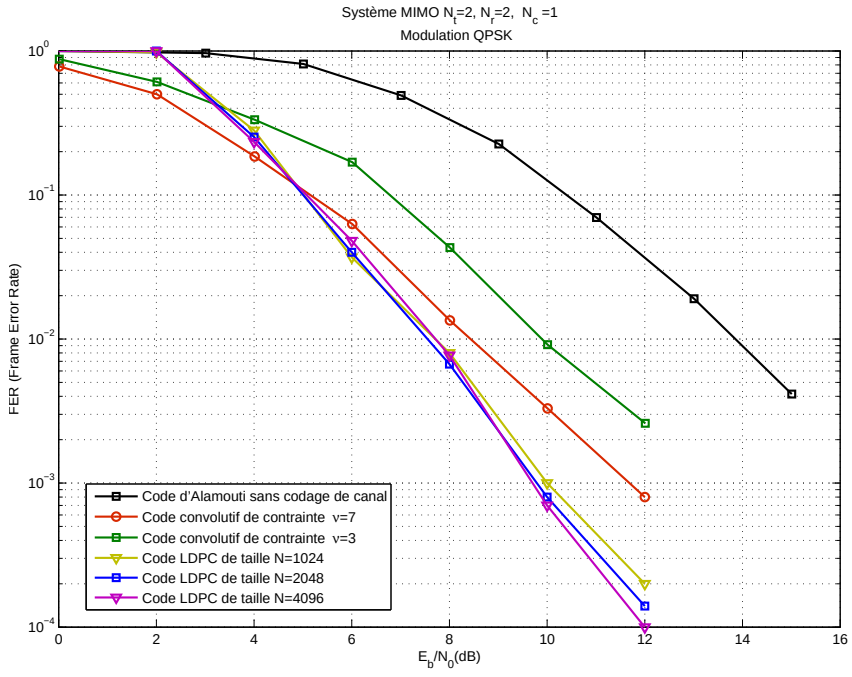


FIGURE 4.13 – Comparaison des performances en terme de FER entre les codes convolutifs et les codes LDPC pour une diversité temporelle $N_c = 1$.

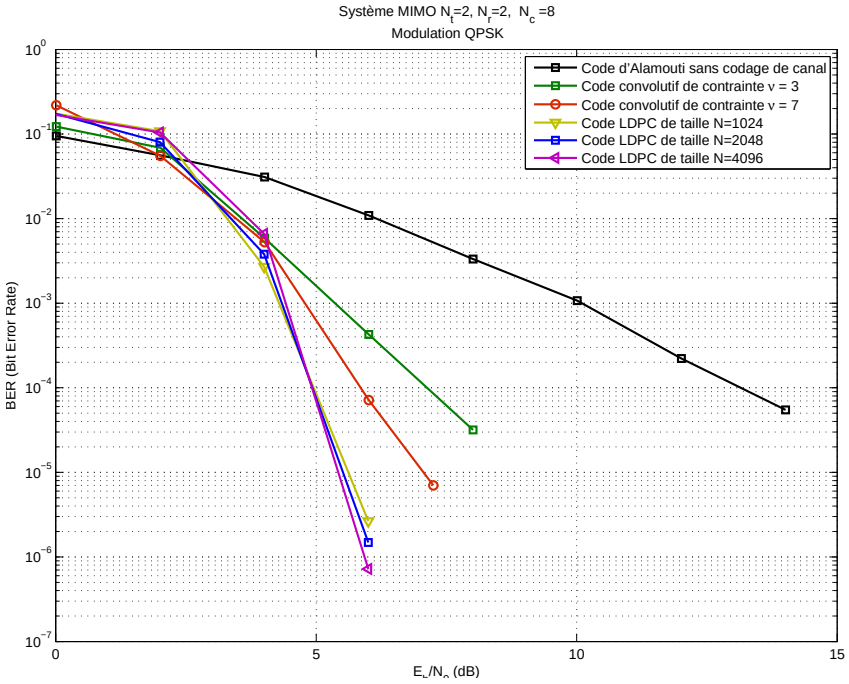


FIGURE 4.14 – Comparaison des performances en terme de BER entre les codes convolutifs et les codes LDPC pour une diversité temporelle $N_c = 8$.

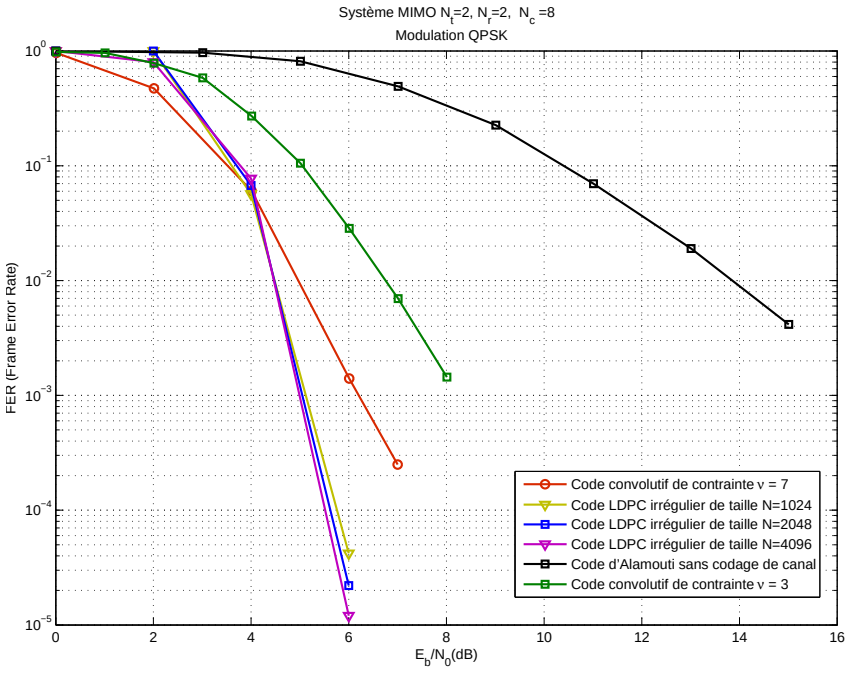


FIGURE 4.15 – Comparaison des performances en terme de FER entre les codes convolutifs et les codes LDPC pour une diversité temporelle $N_c = 8$.

4.10 Conclusion

Dans ce chapitre, nous avons effectué la deuxième partie du stage qui correspond à l'étude de performances des codes LDPC irréguliers dans un système MIMO. Pour cela, nous avons introduit brièvement le principe des systèmes MIMO et les gains apportés par rapport aux systèmes SISO. Ensuite, nous avons présenté le code espace-temps d'Alamouti et le schéma de multiplexage spatial.
Une fois terminé avec les principes fondamentaux des systèmes MIMO, nous avons montré le contexte de l'application qui rentre dans les travaux de recherche de l'équipe GSM sur la prochaine norme IEEE 802.16 m. Par la suite, une modélisation du canal radio mobile et une présentation des conditions de simulation ont été présentées, puis, nous avons abordé l'étude des performances des codes LDPC dans un système MIMO. A partir des résultats de simulations, nous avons tiré les conclusions suivantes :

- Le code espace-temps d'Alamouti donne de meilleures performances par rapport au cas d'un schéma de multiplexage spatial, mais au détriment du débit.
- Les codes LDPC donnent de meilleures performances par rapport aux codes convolutifs classiques.
- L'effet des codes LDPC est important dans le cas d'une diversité temporelle grande.
- En absence de diversité temporelle, même la puissance des codes LDPC ne peut pas éliminer totalement l'effet d'un Fading fort.

Chapitre 5

Présentation du logiciel LDMO

L'objectif de ce dernier chapitre est de présenter le logiciel LDMO. Pour cela, nous donnerons d'abord une présentation générale du logiciel, puis nous montrerons dans un ordre hiérarchique l'intérêt et le mode d'emploi de chaque interface de ce logiciel de simulation.

5.1 Introduction

Le logiciel LDMO (**LD** pour LDPC et **MO** pour MIMO) a été conçu dans le but de regrouper tous les programmes et fonctions développés sous MATLAB et sous le langage C, sous forme d'interfaces graphiques. Ce logiciel permet ainsi de compléter le travail théorique en fournissant un outil de simulation simple à utiliser et une base pour ceux désirant continuer le travail sur les codes LDPC et les systèmes MIMO, afin d'optimiser encore plus mes programmes. Ces interfaces ont été réalisées avec l'outil " GUIDE " de MATLAB et l'éditeur/compilateur C CodeBlocks [72].

5.2 Présentation générale

Le logiciel LDMO s'est voulu un environnement convivial pour l'utilisateur avec des applications simples à exécuter. L'interface graphique principale est organisée de manière hiérarchique avec un menu principal comportant 3 items, permettant ainsi un regroupement selon l'orientation ou l'application. Comme l'illustre la Figure 5.1, les 3 items sont :

(1) → Evaluation de performances des codes LDPC : Cette partie regroupe toutes les simulations effectuées dans le chapitre 3 pour l'évaluation des performances des codes LDPC.

(2)→ Application des codes LDPC dans un système MIMO : Cette partie regroupe toutes les simulations effectuées dans le chapitre précédent pour l'évaluation des performances des codes LDPC dans un système MIMO.

(3)→ Bibliographie : Dans cette partie, nous avons mis toute la documentation qu'on a

utilisée pour l'étude des codes LDPC et les systèmes MIMO, ainsi que tous les programmes écrits sous MATLAB et sous le langage C en format PDF.

Quant à la barre d'outils, elle comporte les menus suivants :

(4)→ **File** : Permettant aux utilisateurs du logiciel d'interagir avec les données de l'ordinateur.

(5)→ **Help** : Fournissant les renseignements nécessaires pour le bon fonctionnement du logiciel.

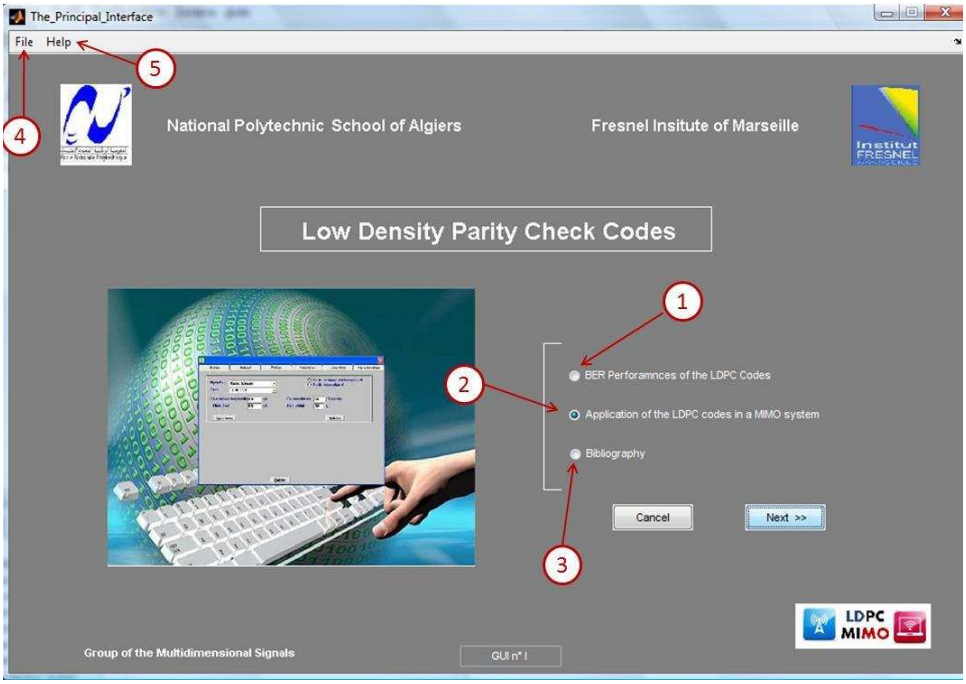


FIGURE 5.1 – GUI n°I : L'interface graphique principale du logiciel LDMO.

5.3 GUI n°I.1 : Performances des codes LDPC

En sélectionnant le choix (1) puis en cliquant sur "suivant", l'interface graphique GUI [1] n°I.1 illustrée par la Figure 5.2 sera affichée. Cette interface contient également un menu composé des items suivants :

(1)→ **Codage et décodage LDPC** : Cette interface regroupe l'algorithme d'encodage (Algorithme de R. Neal décrit dans la section 3.7) ainsi que les algorithmes de décodage décrits dans la section 3.5.

(2)→ **La chaîne de transmission** : Dans cette interface, nous avons présenté le modèle de

1. GUI : Interface graphique (Graphic User Interface en anglais).

simulation illustré dans la Figure 3.14.

(3)→ Simulations : Cet item est composé de 3 autres items :

- Influence de la taille du mot de code LDPC sur les performances.
- Influence du nombre d'itérations du processus de décodage sur les performances.
- Influence du rendement de codage sur les performances.

(4)→ Quelques résultats de simulation : Cette interface permet d'afficher quelques résultats de simulation sous forme d'un diaporama.

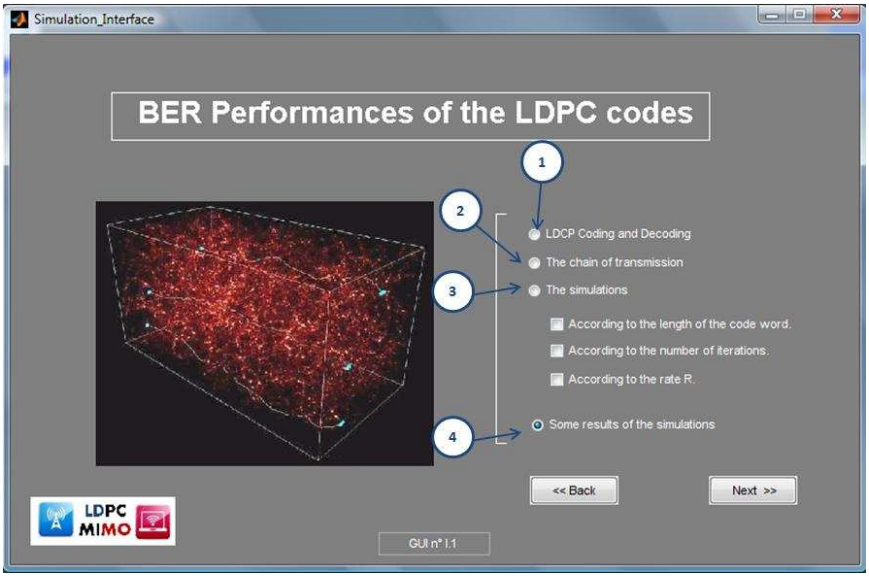


FIGURE 5.2 – GUI n°I.1 : Performances des codes LDPC.

5.3.1 GUI n°I.1.1 : Codage et décodage LDPC

Dans cette interface (voir Figure 5.3) nous avons illustré la partie encodage et décodage LDPC. L'utilisateur de cette interface graphique doit suivre les étapes suivantes :

(1)→ Introduction des paramètres du codeur et du décodeur LDPC :

- N : la taille du code LDPC.
- M : le nombre d'équations de parité.
- w_c : le poids des colonnes.
- Itérations : le nombre d'itérations de l'algorithme de décodage.
- BERMax : le nombre maximal de bits erronés et détectés pour chaque valeur de E_b/N_0.

(2)→ Génération de la matrice de contrôle de parité H [2] par l'une des méthodes de construction suivantes :

2. Les éléments non nuls sont représentés par des points bleus.

– Evenboth : Essayer de placer les éléments non nuls de manière à avoir un poids constant de toutes les colonnes et dans toutes les lignes.
– Evencol : Placer les éléments non nuls de manière à avoir un poids constant de toutes les colonnes.

(3)→ Introduction des valeurs minimale et maximale de E_b/N_0 ainsi que le pas d'incrémentation.

(4)→ Pour la réinitialisation de l'interface, cliquer sur le bouton "Reset" et pour la fermeture de

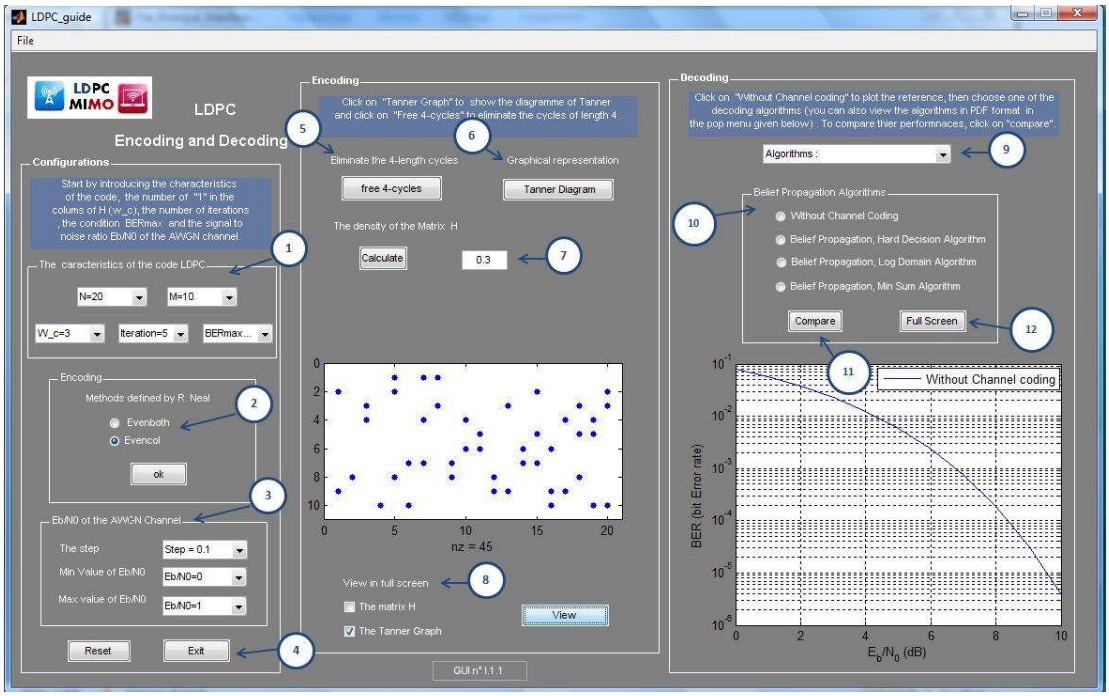


FIGURE 5.3 – GUI n°I.1.1 : Codage et décodage LDPC.

l'interface, elle s'exécute soit en cliquant directement sur "Exit" ou bien sur "File" puis "Exit", ou tout simplement en tapant les touches de raccourci suivantes " Ctrl + E ".

(5)→ Elimination des cycles de longueur 4 en cliquant sur le bouton "Free 4-cycles".

(6)→ La représentation graphique (graphe de Tanner) du code LDPC en cliquant sur le bouton "Tanner Diagram" (voir Figure 5.4).

(7)→ Calcul de la densité de la matrice H en cliquant sur le bouton "Calculate".

(8)→ Visualisation en plein écran de la matrice H ou le diagramme factoriel (Diagramme de Tanner).

(9)→ Voir les algorithmes de décodage en format PDF.

(10)→ Après la sélection des paramètres de simulation dans **(1)**, **(2)** et **(3)**, lancer automatiquement la simulation en cliquant sur l'un des algorithmes de décodage BP. Au cours de la

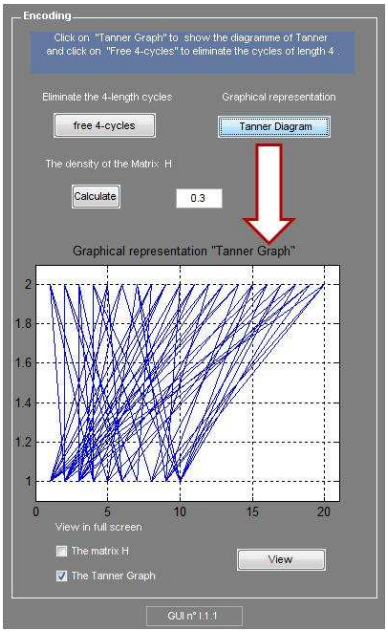
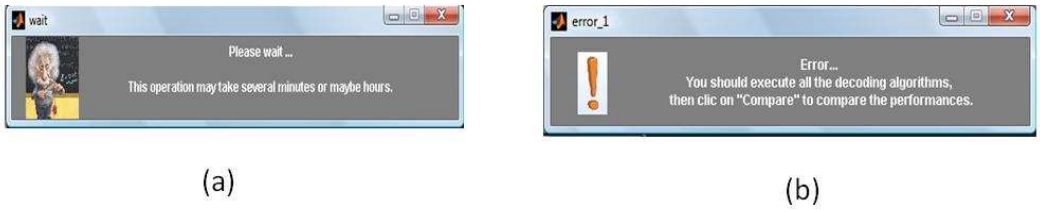


FIGURE 5.4 – Le diagramme de Tanner sous le logiciel LDMO.



(a) (b)

FIGURE 5.5 – (a) Le message d'attente affiché durant les simulations. (b) Le message d'erreur.

simulation[3] un message d'attente sera affiché (voir Figure 5.5 (a)).

(11)→ Comparaison de performances entre les algorithmes de décodage BP, en cliquant sur le bouton "Compare". S'il manque les résultats d'un des trois algorithmes, un message d'erreur sera affiché (voir Figure 5.5 (b)).

(12)→ Visualisation en plein écran des résultats de comparaison.

5.3.2 GUI n°I.1.3 : Simulations

Comme mentionné précédemment, cette partie comporte 3 interfaces graphiques (voir Figure 5.6) regroupant les simulations suivantes :

1. Influence de la taille N sur les performances $\Longrightarrow$ Figure 5.6 (a).
2. Influence du nombre d'itérations sur les performances $\Longrightarrow$ Figure 5.6 (b).

3. Le temps d'exécution des simulations augmente d'une manière exponentielle avec la taille de la matrice H.

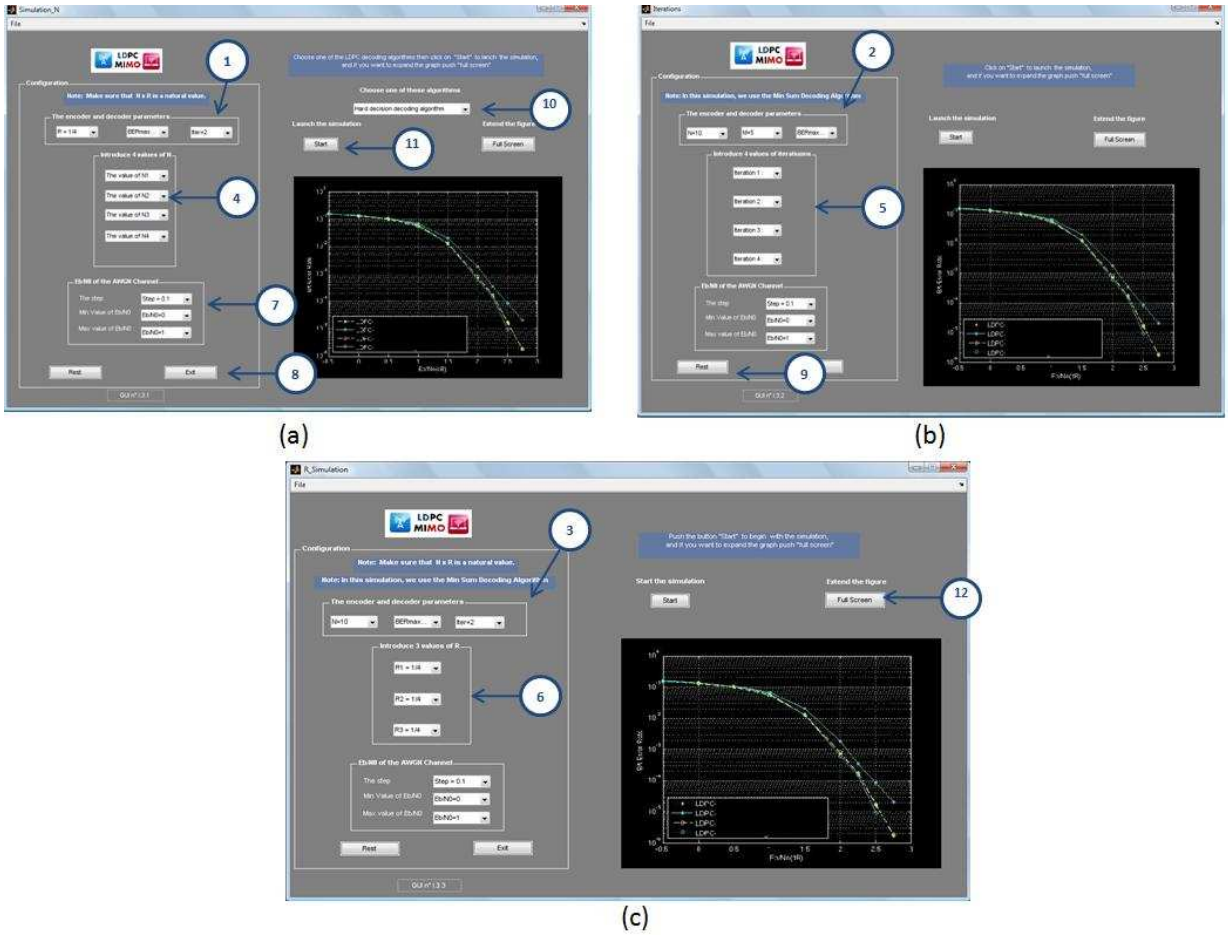


FIGURE 5.6 – GUI n°I.3 : Simulations : (a) GUI n°I.3.1 : selon la taille N. (b) GUI n°I.3.2 : selon le nombre d'itérations. (c) GUI n°I.3.3 : selon le rendement de codage.

3. Influence du rendement de codage sur les performances $\Longrightarrow$ Figure 5.6 (c).

Pour ce qui suit, nous allons donner le rôle de chaque item ou bouton des 3 interfaces illustrées dans la Figure 5.6 :

(1)→ Introduction des paramètres concernant la simulation 1 (le rendement de codage R, le nombre d'itérations de l'algorithme de décodage et le BERMax).

(2)→ Introduction des paramètres concernant la simulation 2 (la taille du code N, le nombre d'équations de parité M et le BERMax).

(3)→ Introduction des paramètres concernant la simulation 3 (la taille du code N, le nombre d'itérations de l'algorithme de décodage et Le BERMax).

(4)→ Sélection de 4 valeurs de la taille du code LDPC (N).

(5)→ Sélection de 4 valeurs du nombre d'itérations de l'algorithme de décodage.

(6)→ Sélection de 3 valeurs du rendement de décodage R.

(7)→ Introduction des valeurs minimale et maximale de E_b/N_0 ainsi que le pas d'incrémentation.
(8) et (9) → Pour la réinitialisation de l'une des 3 interfaces, cliquer sur le bouton "Reset" et pour la fermeture de l'une des 3 interfaces, elle s'exécute soit en cliquant directement sur "Exit" ou bien sur "File" puis "Exit", ou tout simplement en tapant les touches de raccourci suivantes " Ctrl + E ".
(10)→ Sélection de l'algorithme de décodage BP.
(11)→ Début de la simulation, en cliquant sur le bouton "Start".
(12)→ Visualisation en plein écran des résultats de simulation.

5.4 GUI n°I.2 : Application des codes LDPC dans un système MIMO

Pour accéder à cette interface, il suffit juste de choisir (2) dans le menu principal, puis, cliquer sur "suivant".

Comme mentionné dans le chapitre précédent, cette partie a été programmée en langage C, pour cela, l'interface illustrée par la Figure 5.8, comporte deux parties :

- Lancement des simulations sous le langage C.
- Récupération des résultats pour les afficher sous MATLAB.

5.4.1 Lancement des simulations sous le langage C

Pour faciliter l'accès à l'éditeur/compilateur C "CodeBlocks", nous avons créé un lien (1) "Browse" dont le chemin d'accès est illustré dans (2), Après l'ouverture de l'éditeur/compilateur "CodeBlocks" on choisit l'un des projets suivants :

- **Workspace-sanscodage** ⇒ pour le cas d'un codage espace-temps sans codage de canal (Code d'Alamouti ou schéma de multiplexage spatial).
- **Workspace-conv** ⇒ pour le cas d'un code espace-temps d'Alamouti avec un code convolutif.
- **Workspace-ldpc** ⇒ pour le cas d'un code espace-temps d'Alamouti avec un code LDPC irrégulier.

Après la sélection du projet, l'utilisateur du logiciel doit introduire les paramètres de simulations définies au début du fichier main.c du projet sélectionné, puis compiler et exécuter les programmes (voir Figure 5.7). A la fin de l'exécution des programmes, on récupère les résultats à partir des fichiers .dat.

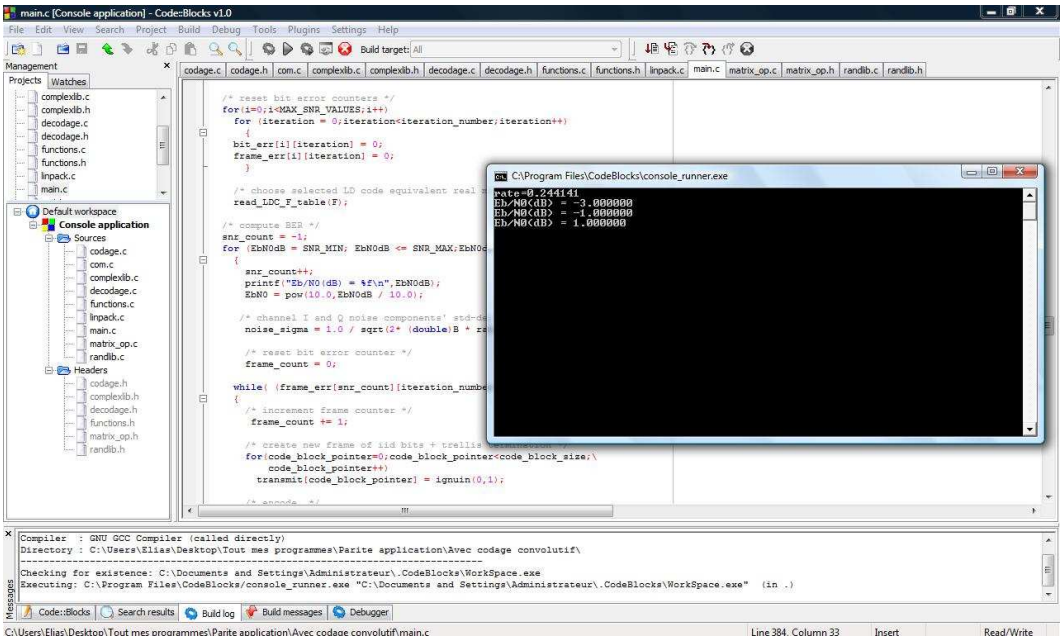


FIGURE 5.7 – Exécution d'une des simulation sous le langage C.

5.4.2 Affichage des résultats

Pour ce qui suit, nous donnerons le rôle des items et des boutons restants dans l'interface graphique présentée par la Figure 5.8 :

(4)→ Introduction des valeurs minimale et maximale de E_b/N_0 ainsi que le pas d'incrémentation.

(5)→ Le nombre d'antennes à la réception.

(6)→ La taille de l'entrelaceur Π.

(7)→ La diversité temporelle N_c du canal.

(8)→ Pour le projet Workspace-sanscodage, sélectionner le type du code espace-temps (Alamouti ou MUX).

(9)→ Pour le projet Workspace-conv, sélectionner la taille de la contrainte ν du code convolutif.

(10)→ Pour le projet Workspace-ldpc, sélectionner la taille du code LDPC N.

(11)→ Télécharger les résultats à partir des fichiers .dat.

(12)→ Sélection du critère d'évaluation de performances BER ou FER.

(13)→ Affichage des résultats sous forme de courbes BER=f(E_b/N_0) ou FER=f(E_b/N_0).

Remarque : Pour visualiser la chaîne de transmission, il suffit de sélectionner soit à l'émission ou à la réception, puis, cliquer sur "Ok" (3).

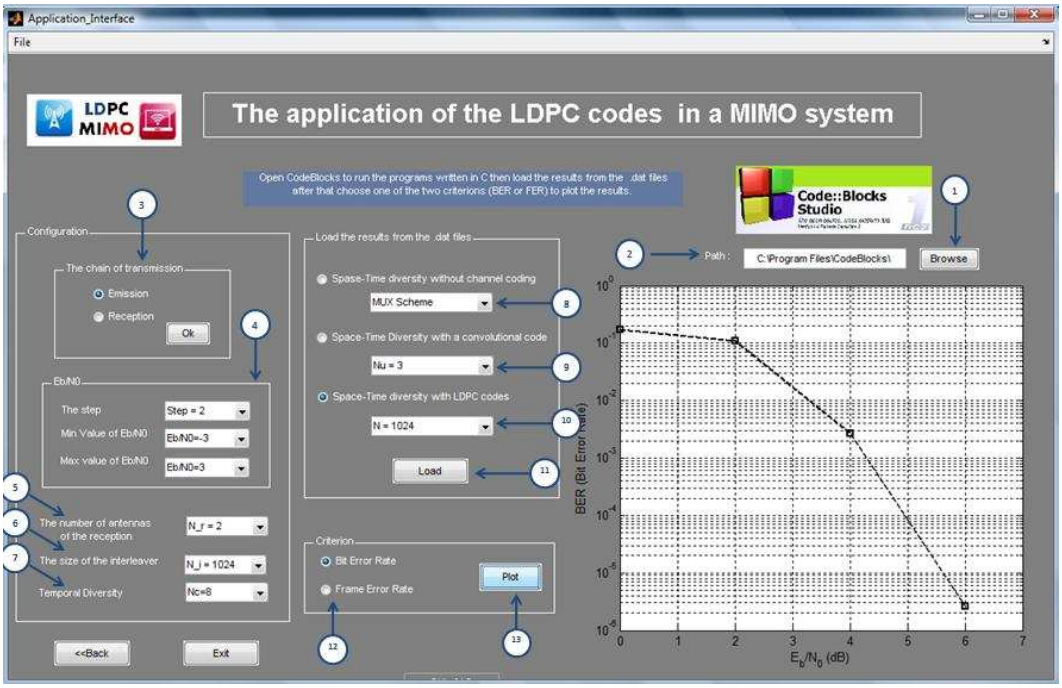


FIGURE 5.8 – GUI n°I.2 :Application des codes LDPC dans un système MIMO.

5.5 GUI n°I.3 : Bibliographie

Le menu (1) illustré dans la Figure 5.9 (a) regroupe la documentation concernant :

- Les codes LDPC (voir Figure 5.9 (b)).
- les systèmes MIMO (voir Figure 5.9 (c)).

Comme l'illustre la Figure 5.9 (b) est divisée en deux parties :

(2)→ Liste contenant tous les documents utilisés pour l'étude des codes LDPC, triés selon le type (Articles, livres, thèses ...etc).
(3)→ A partir de cette liste, l'utilisateur du logiciel peut consulter tous les programmes qu'on a écrit sous MATLAB en format PDF.

L'interface illustrée par la Figure 5.9 (c) est aussi divisée en deux parties :

(4)→ Liste contenant tous les documents utilisés pour l'étude des systèmes MIMO, triés selon le type (Articles, livres, thèses ...etc)..
(3)→ A partir de cette liste, l'utilisateur du logiciel peut consulter tout les programmes qu'on a écrit sous le langage C en format TXT.

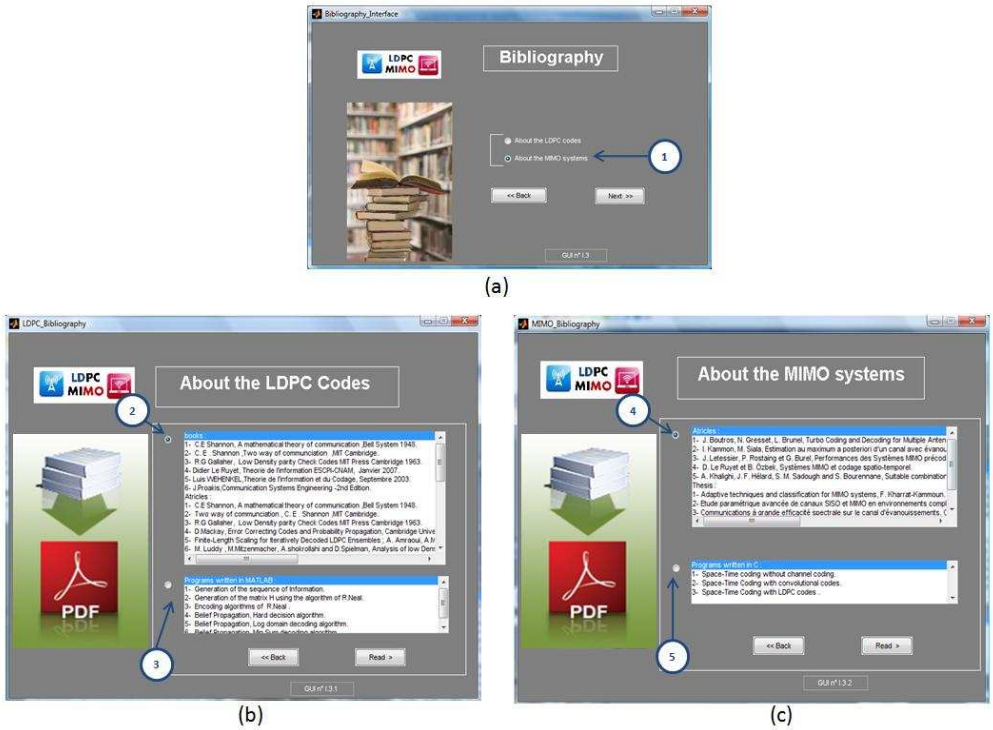


FIGURE 5.9 – (a) GUI n°I.3 : Bibliographie. (b) GUI n°I.3.1 : Sur les codes LDPC. (c) GUI n°I.3.2 : Sur les systèmes MIMO.

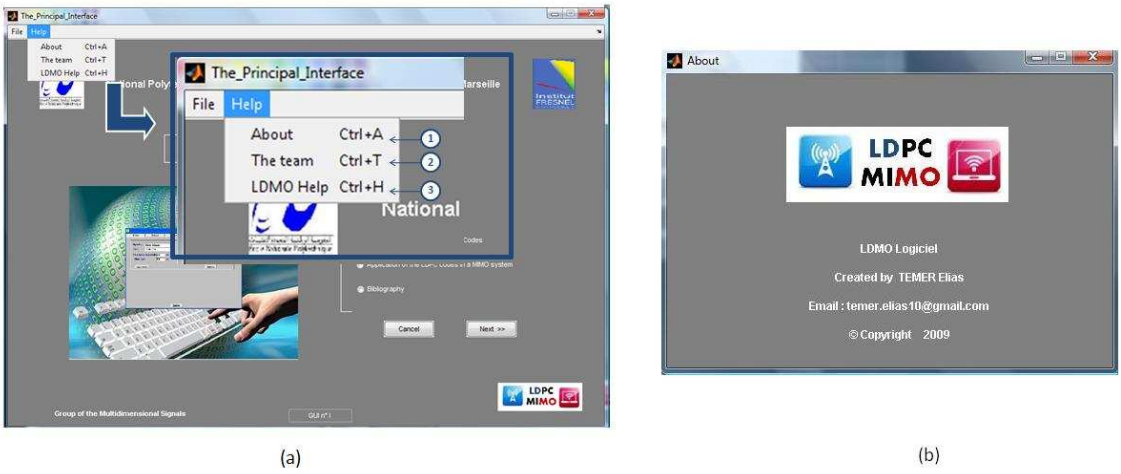


FIGURE 5.10 – (a) Le menu Help. (b) About.

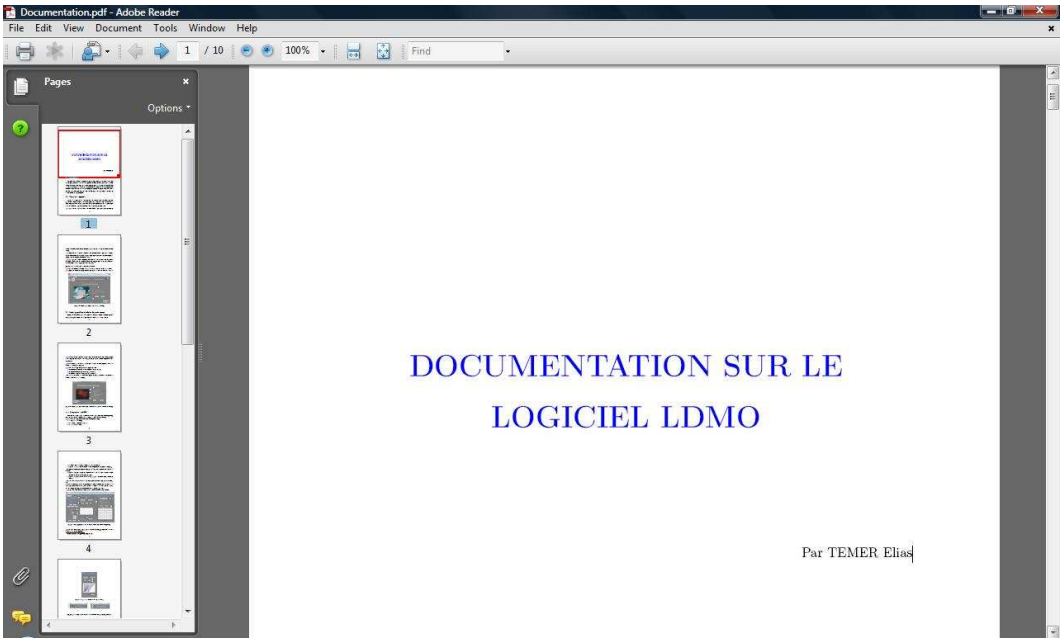


FIGURE 5.11 – Document d'aide donné en PDF.

5.6 Help

En cliquant sur "Help" (5) dans la barre d'outil de l'interface principale, un menu composé de 3 sous-menus sera affiché (voir Figure 5.10 (a)) tel que :

(1)→ About : Fournit des informations sur les réalisateurs du logiciel (voir Figure 5.10 (b)).

(2)→ The Team : Fournit des informations sur l'équipe travaillant sur ce sujet.

(3)→ LDMO Help : Fournit un document en PDF. Ce document explique le mode de fonctionnement du logiciel ainsi que les différentes étapes à suivre pour lancer et afficher les résultats de l'application voulue. Des exemples y sont fournis afin de faciliter l'utilisation du logiciel et de mieux illustrer les commandes. Ainsi, même un amateur pourra facilement s'y adapter.

5.7 Conclusion

Ce petit logiciel représente une synthèse de tout le travail effectué lors des six mois de stage au sein de l'institut Fresnel, en fournissant des outils et des simulations faciles à utiliser. Les interfaces graphiques de ce logiciel sont présentées de manière simple et attractive, afin d'éveiller l'intérêt de l'utilisateur sur ce domaine d'étude, qui est en plein expansion dans le monde des télécommunications. Finalement, nous n'omettrons pas de signaler que la conception reste ouverte à des améliorations, afin d'augmenter et d'optimiser les performances de ce petit logiciel.

Conclusions et perspectives

Les travaux présentés dans ce mémoire sont le fruit de six mois de stage au sein de l'institut Fresnel, plus précisément l'équipe GSM. Le sujet étudié, portant sur l'étude des codes LDPC et leurs applications dans un système MIMO, rentre dans la cadre des travaux de recherche du laboratoire sur la prochaine norme IEEE 802.16 m du standard WIMAX, qui envisage d'introduire les codes LDPC avec les systèmes MIMO.

Arrivés au terme de ce projet et avant d'en évoquer les perspectives, nous nous proposons de faire un bilan rapide des résultats obtenus, à travers le résumé du contenu des cinq chapitres qui ont été développés, tout en soulignant les différents liens les unissant.

En faisant des comparaisons entre les codes correcteurs d'erreurs les plus connus, nous avons constaté que seules les techniques avancées de codage, les turbo-codes et les codes LDPC, peuvent atteindre la limite de Shannon.

Après avoir introduit dans un premier temps, les codes LDPC à travers un bref historique. Nous avons présenté les deux classes des codes LDPC : réguliers et irréguliers ainsi que leurs présentations matricielle et graphique. En comparant leurs performances dans un canal gaussien AWGN, nous avons constaté que les codes irréguliers sont les plus performants. A partir de ce résultat, nous avons considéré le cas des codes irréguliers pour le reste de notre étude. Nous avons, ensuite, abordé la partie décodage et encodage. Concernant la partie décodage, nous avons illustré l'algorithme de propagation de croyances BP dans le cas Hard et Soft ainsi que l'algorithme sous optimal Min-Sum. Et concernant la partie encodage, nous avons présenté l'algorithme d'encodage présenté par R. Neal. Quant à la partie expérimentale, nous avons tout d'abord commencé par introduire la chaîne de transmission, les conditions et hypothèses de simulations, ensuite, nous avons présenté les résultats de simulations, à partir desquelles nous avons tiré les conclusions suivantes :

- Le décodage Soft est beaucoup plus performant que le décodage Hard.
- L'algorithme sous optimal Min-Sum représente une très bonne approximation de l'algorithme

de décodage Log-Domain.

• L'implémentation de l'algorithme Min-Sum dans des applications en temps réel représente une bonne solution, contrairement à l'algorithme Log-Domain.

• L'augmentation de la taille du code LDPC entraîne une amélioration des performances, cette amélioration est remarquable surtout pour $N \geq 1000$.

• L'augmentation du nombre d'itérations améliore les performances de décodage jusqu'à un nombre d'itérations maximum où l'algorithme de décodage converge.

• Plus R est petit plus la protection contre les erreurs est meilleure, mais à partir d'un certain seuil le système commence à perdre en efficacité spectrale.

Cette étude originale a permis de mettre en évidence l'importance du choix de certains paramètres sur les performances du code. En se basant sur ces résultats pour le choix des paramètres du codeur/décodeur LDCP, nous avons fait une évaluation des performances du codeur/décodeur LDPC implémenté dans un système MIMO. Pour cela, nous avons commencé par la présentation du canal radio mobile ainsi que ses imperfections, ensuite, nous avons introduit brièvement le principe des systèmes MIMO ainsi que les gains apportés par rapport aux systèmes SISO. Une fois terminé avec l'étude théorique des systèmes MIMO, nous avons modélisé le canal radio mobile, puis présenté la chaîne de transmission, les conditions et les hypothèses de simulations. A partir des résultats des simulations, nous avons tiré les conclusions suivantes :

• Les codes LDPC donnent de meilleures performances par rapport aux codes convolutifs classiques.

• L'effet des codes LDPC est remarquable dans le cas d'une diversité temporelle grande.

• En absence de diversité temporelle, même la puissance des codes LDPC ne peut pas éliminer totalement l'effet d'un Fading fort.

En dernier lieu, nous avons présenté le logiciel LDMO, qui a été conçu dans le but de regrouper tous les programmes et fonctions développés sous MATLAB et le langage C, sous forme d'interfaces graphiques.

Ce logiciel s'est voulu pratique, interactif et surtout très facile à utiliser. Nous signalons que ce logiciel est ouvert à des améliorations et des perfections.

Perspectives

Le travail réalisé dans ce mémoire consiste en un point de départ à toute une série de travaux possibles. Plusieurs perspectives sont envisageables :

- Augmenter encore plus la diversité temporelle N_c et voir s'il y a encore possibilité d'amélioration des performances du codeur/décodeur LDPC.
- Voir pour quelle valeur de N les performances du codeur/décodeur LDPC sont identiques à celles d'un codeur/décodeur convolutif simple.
- Refaire l'étude avec une estimation de canal, c'est-à-dire qu'à la réception, on ne connaît pas les caractéristiques du canal.
- Enfin, faire une étude comparative détaillée entre le codeur/décodeur LDPC et le codeur/décodeur convolutif en terme de complexité d'implémentation.

Annexe A

Algorithms for LDPC decoder

Steven W. McLaughlin, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA.

The message passing algorithm is an APP algorithm only if the code graph has no cycles. LDPC codes have cycles, so strictly speaking the message passing algorithm does not compute the APPs. However the algorithm performs remarkably well. The cycle-free requirement implies that all code bits $x_0, \ldots, x_{N-1}$ are independent, which clearly they are not. At the end of this section algorithms are given for implementing the message passing algorithm in both probability and log domains. We now derive the message passing algorithm for LDPCs from first principles. Prior to decoding the decoder has the following : a parity check matrix H, its bipartite graph, and N channel outputs r. Let $M(i)$ be the set of parity nodes connected to the code bit x_j, and let N(m) be the set of bit nodes connected to the m^{th} parity check.

Message passing algorithm

The derivation given here most closely follows the description given by Gallager. Using the assumption of code bit independence and Baye's rule the APP $p(x_j = b|S_j, r)$ can be written as

$$p(x_j = b|S_j, r) = Kp(r_j|x_j = b)p(S_j|x_j = b, r) \tag{A.1}$$

where K is a constant for both b=0,1 and can be ignored. The first term is easy to compute, which for Gaussian noise

$$p(r_j|x_j = b) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp \frac{(r_j + (-1)^b)^2}{2\sigma^2} \tag{A.2}$$

and for a BSC (with crossover probability p)

$$p(r_j|x_j = 0) = p^{r_j}(1-p)^{1-r_j} \tag{A.3}$$

$$p(r_j|x_j = 1) = p^{1-r_j}(1-p)^{r_j} \tag{A.4}$$

The second term in A.1 is the probability that all parity checks connected to x_j are satisfied given r and x_j = b. Note that $S_j = \{S_{0j}, \ldots, S_{kj}\}$ is a collection of events, where S_{mj} is the event that the m^{th} parity node connected to x_j is satisfied. Again by independence of the code bits $(c_0, \ldots, c_{N-1})$ this can be written as

$$p(S_j|x_j = b, r) = p(S_{0j}, S_{1j}, \ldots S_{kj}|x_j = b, r) = \prod_{m \in M(j)} p(S_{mj}|x_j = b, r) \tag{A.5}$$

where $p(S_{mj}|x_j = b, r)$ is the probability that the m^{th} parity check connected to the bit x_j is satisfied given $x_j = b$ and r. If $b = 0$ this is the probability that the code bits other than x_j connected to the m^{th} parity check have an even number of 1's. If b = 1, the other code bits must have odd parity. Using this fact we next show that $p(S_{mj}|x_j = b, r)$ has a relatively simple form.

As a preliminary calculation, suppose two bits satisfy a parity check constraint $x_1 \oplus x_2 = 0$, and it is known that $p_1 = P(x_1 = 1)$ and $p_2 = P(x_2 = 1)$.
Let $q_1 = 1 - p_1$ and $q_2 = 1 - p_2$. Then the probability that the check is satisfied is

$$p(x_1 \oplus x_2 = 0) = (1-p_1)(1-p_2) + p_1 p_2 = 2p_1 p_2 - p_1 - p_2 + 1 \tag{A.6}$$

which can be rewritten as

$$2p(x_1 \oplus x_2 = 0) - 1 = (1-2p_1)(1-2p_2) = (q_1 - p_1)(q_2 - p_2) \tag{A.7}$$

Now suppose that L+1 bits satisfy an even parity-check constraint $x_0 \oplus x_1 \oplus x_2 \oplus \ldots \oplus x_L = 0$. Then for known probabilities $\{p_1, p_2, \ldots, p_L\}$ corresponding to the bits $\{x_1, x_2, \ldots, x_L\}$, it is possible to generalize A.7 to find the probability distribution on the binary sum $z_L = x_1 \oplus x_2 \oplus \ldots \oplus x_L$, where $z_L = z_{L-1} \oplus x_L$

$$\begin{aligned} 2p(z_L = 0) - 1 &= (1 - 2P(z_L - 1 = 1))(1 - 2p_L) \\ &= (2P(z_L - 1 = 0) - 1)(1 - 2p_L) \end{aligned} \tag{A.8}$$

where $p_L = p(x_L = 1)$. Applying this recursively yields

$$2p(z_L = 0) - 1 = \prod_{i=1}^{L}(1 - 2p_i) \tag{A.9}$$

or

$$p(z_L = 0) = \frac{1}{2}(1 + \prod_{i=1}^{L}(1 - 2p_i)) = \frac{1}{2}(1 + \prod_{i=1}^{L}(q_i - p_i)) \tag{A.10}$$

Similary it is possible to show

$$p(z_L = 1) = \frac{1}{2}(1 - \prod_{i=1}^{L}(1 - 2p_i)) = \frac{1}{2}(1 - \prod_{i=1}^{L}(q_i - p_i)) \tag{A.11}$$

Returning to our calaculation of $p(S_{mj}|x_j = b; r)$, if $x_j = 0$ then we use $p(z_L = 0)$ and if $x_j = 1$ we use $p(z_L = 1)$.

$$p(S_{mj}|x_j = 0, r) = \frac{1}{2}(1 + \prod_{n' \in N(m)\backslash j}(q^0_{mn'} - q^1_{mn'})) \tag{A.12}$$

$$p(S_{mj}|x_j = 1, r) = \frac{1}{2}(1 - \prod_{n' \in N(m)\backslash j}(q^0_{mn'} - q^1_{mn'})) \tag{A.13}$$

where $q^0_{mn'}$ is the probability that code bit $x_{n'}$ is zero, given r and excluding any information about $x_{n'}$ from parity check m. This exclusion is needed because we desire extrinsic knowledge about $x_{n'}$ from its parity checks to get extrinsic knowledge about x_j. Note that the rightmost product is over all code bits connected to the m'^{th} parity node except for x_j , since we are interested in the even or odd parity of the bits other than x_j.

Combining these results with A.1 and A.5, we get the final expressions for the APPs.

$$p(x_j = 0|S_j, r) = Kp(r_j|x_j = 0) \prod_{m \in M(j)} \frac{1}{2}(1 + \prod_{n' \in N(m)\backslash j}(q^0_{mn'} - q^1_{mn'})) \tag{A.14}$$

$$p(x_j = 1|S_j, r) = Kp(r_j|x_j = 1) \prod_{m \in M(j)} \frac{1}{2}(1 - \prod_{n' \in N(m)\backslash j}(q^0_{mn'} - q^1_{mn'})) \tag{A.15}$$

Furthermore, some can be viewed as "*parity node*" computations and others as "*bit node*" computations, for example, for $b = 1$, $p(x_j = 1|S_j, r)$ is

$$\overbrace{p(y|x_j = 1)} \prod_{m \in M(j)} \overbrace{\frac{1}{2}(1 - \prod_{n' \in N(m)\backslash j}(q^0_{mn'} - q^1_{mn'}))} \tag{A.16}$$

The notation can be simplified by letting $\delta q_{mj} = q^0_{mj} - q^1_{mj}$ and dedining parity check equations as

$$r^0_{mj} = \frac{1}{2}(1 + \prod_{n' \in N(m)\backslash j} \delta q_{mn'}) \tag{A.17}$$

$$r^1_{mj} = \frac{1}{2}(1 - \prod_{n' \in N(m)\backslash j} \delta q_{mn'}) \tag{A.18}$$

Annexe B

Décomposition SVD d'un canal MIMO

Considérons un système MIMO avec N_t antennes d'émission et N_r antennes de réception tel que $N_t \leq N_r$. On suppose que le récepteur connaît parfaitement le canal alors que l'émetteur ne dispose pas de la connaissance du canal.
Le canal MIMO défini par la matrice H_M peut être décomposé en plusieurs canaux (SISO) parallèles en utilisant la décomposition en valeurs propres (SVD) [5] comme suit :

$$H_M = U\Sigma V^H \tag{B.1}$$

où U et V sont des matrices unitaires et Σ est la matrice diagonale $\Sigma = diag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \cdots, \sqrt{\lambda_r}, 0, \cdots, 0)$ où λ_i $(i = 1, \cdots, r)$ sont les valeurs propres non nulles de $H_M^H H_M$. Le nombre de valeurs propres r est le rang de la matrice de canal H_M et est égal à $min(N_t, N_r)$. En appliquant un pré-traitement aux symboles transmis $(V.\varkappa)$ du coté de l'émetteur

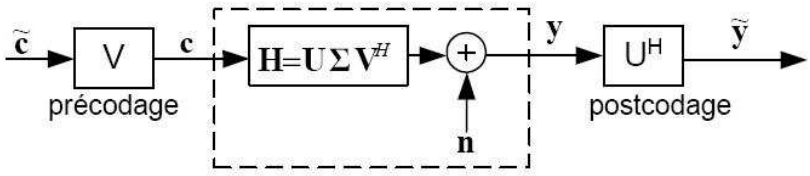


FIGURE B.1 – Décomposition SVD du canal MIMO.

et un post-traitement à la réception $(U^H y)$ on obtient la relation suivante :

$$U^H y = H^H (U\Sigma V^H) V \varkappa + U^H n \tag{B.2}$$

$$\widetilde{y} = \Sigma V^H \widetilde{\varkappa} + \widetilde{n} \tag{B.3}$$

où $\widetilde{n}$ est encore gaussien avec la même variance que n.

L'équation B.3 représente un système équivalent avec r canaux SISO parallèle dont la puissance du signal est donnée par leurs valeurs propres. Ainsi, la capacité instantanée peut être écrite de cette manière :

$$C(\rho) = \sum_{i=1}^{r} \log_2(1 + \frac{\rho}{N_t}\lambda_i) \tag{B.4}$$

où $\rho = E_s/N_0$ est le rapport d'énergie par symbole sur la densité de puissance du bruit [65], [66].

Annexe C

MIMO detection : ST decoding

This part is extracted from the article " **Suitable combination of channel coding and space time schemes for moderate to high spectral efficiency MIMO systems** " written by **M. A. Khalighi, J. F. Hélard, S. M. Sadough and S. Bourennane** [68].

Remark : In this annex, we note by H the matrix of the MIMO channel.

For a MIMO channel described by a matrix H of dimension (N_r x N_t). A transmitted frame of N bits corresponds hence to N_c blocks with independent fades, Now, corresponding to an ST (Space Time) encoded (N_t x T) matrix X, we receive the (N_r x T) matrix Y :

$$Y = HX + N \tag{C.1}$$

where N represents the receiver noise ; its entries are zero-mean complex Gaussian random variables.

General formulation of LD codes

In order to provide a general formulation for the MIMO detector irrespective of the underlying ST scheme, we adopt the formulation of Linear Dispersion codes (LD), proposed in [73]. This formulation applies also to OSTBC and MUX schemes.
Let α_q and β_q be the real and imaginary parts of a symbol s_q, i.e., $s_q = \alpha_q + j\beta_q$. The main parameters of the code are its (N_t x T) dispersion matrices A_q and B_q , q =1,... , Q. The ST generator matrix is then

$$X = \sum_{q=1}^{Q} (\alpha_q A_q + j\beta_q B_q) \tag{C.2}$$

A_q and B_q are in general complex matrices, but like in [73], we consider them of real entries. We further separate the real and imaginary parts of the entries of S and X and stack them row-wise in vectors. Remember that S is the (Q x 1) vector of data symbols prior to ST encoding. We obtain hence the vectors $\mathcal{S}$ of dimension ($2Q$ x 1) and $\mathcal{X}$ of dimension ($2N_t$ x 1) :

$$\mathcal{S} = [\mathcal{R}\{s_1\}\mathcal{I}\{s_1\}, \ldots, \mathcal{R}\{s_1\}\mathcal{I}\{s_1\}]^t \tag{C.3}$$

$$\mathcal{X} = [\mathcal{R}\{X_{(1,1)}\}\mathcal{I}\{X_{(1,1)}\}, \ldots, \mathcal{R}\{X_{(N_t,T)}\}\mathcal{I}\{X_{(N_t,T)}\}]^t \tag{C.4}$$

where $\mathcal{R}$ and $\mathcal{I}$ denote the real and imaginary part operators, respectively. The LD ST encoder can now be considered as a ($2N_tT$ x $2Q$) matrix $\mathcal{F}$ such that

$$\mathcal{X} = \mathcal{F}\mathcal{S} \tag{C.5}$$

$$\mathcal{F} = \begin{bmatrix} \mathcal{F}_1(1,1) & \cdots & \mathcal{F}_Q(1,1) \\ \vdots & \ddots & \vdots \\ \mathcal{F}_1(1,T) & \cdots & \mathcal{F}_Q(1,T) \\ \vdots & \ddots & \vdots \\ \mathcal{F}_1(N_t,T) & \cdots & \mathcal{F}_Q(N_t,T) \end{bmatrix} \tag{C.6}$$

where

$$\mathcal{F}_q(m,t) = \begin{bmatrix} A_q^{\mathcal{R}}(m,t) & -B_q^{\mathcal{I}}(m,t) \\ A_q^{\mathcal{I}}(m,t) & B_q^{\mathcal{R}}(m,t) \end{bmatrix} \tag{C.7}$$

and, for instance, $A_q^{\mathcal{R}}(m,t)$ denotes the $(m,t)^{th}$ entry of $A_q^{\mathcal{R}}$. Receiving the matrix Y, corresponding to a transmitted matrix X, we construct the vector $\mathcal{Y}$ from Y as we did to obtain $\mathcal{X}$. We can write $\mathcal{Y} = \mathcal{H}\mathcal{X} + \mathcal{N}$, where $\mathcal{N}$ is the vector of real AWGN of zero mean and variance σ^2, and the matrix $\mathcal{H}$ of dimension ($2N_rT$ x $2N_tT$) is constructed from the $\mathcal{R}$ and $\mathcal{I}$ parts of the entries H_{ij} of the initial matrix H. It is composed of segments $\mathcal{H}_{\text{ij}}$, i= 1,..., N_r, j= 1,..., N_t, each one of dimension (2T x 2T) :

$$\mathcal{H}_{\text{ij}} = \begin{bmatrix} H_{ij} & 0 & \cdots & 0 \\ 0 & H_{ij} & \cdots & 0 \\ \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & H_{ij} \end{bmatrix} \tag{C.8}$$

where H_{ij} is obtained from each entry H_{ij} of the initial matrix H :

$$H_{ij} = \begin{bmatrix} \mathcal{R}\{H_{ij}\} & -\mathcal{I}\{H_{ij}\} \\ \mathcal{I}\{H_{ij}\} & \mathcal{R}\{H_{ij}\} \end{bmatrix} \tag{C.9}$$

Now, we can consider an equivalent channel matrix $\mathcal{H}_{\text{eq}}$ of dimension ($2N_rT$ x $2Q$) such that

$$\mathcal{Y} = \mathcal{H}\mathcal{F}\mathcal{S} + \mathcal{N} = \mathcal{H}_{\text{eq}}\mathcal{S} + \mathcal{N} \tag{C.10}$$

MIMO soft detection

The detection problem is to find the transmitted data vector S, given the vector Y. For this purpose, we perform iterative Soft-PIC detection, as shown in Figure C.1, that we describe in the following in more detail.

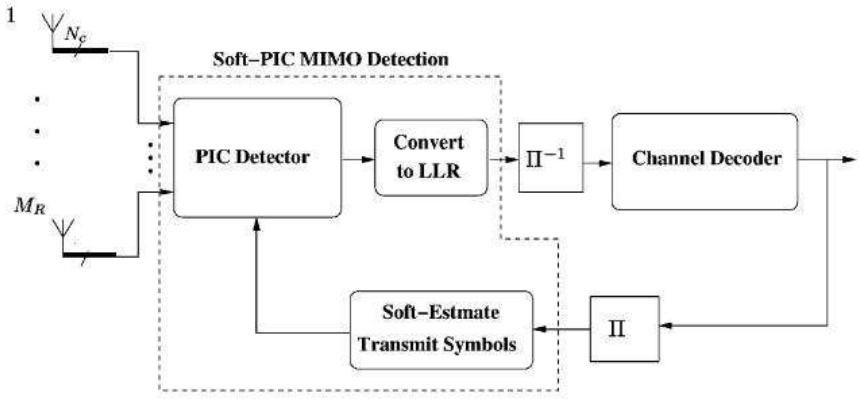


FIGURE C.1 – Block diagram of the receiver.

Let us denote by γ_p the p^{th} entry of $\mathcal{S}$, p = 1,..., 2Q. At the first iteration, where we have no information on the transmitted symbols, to obtain the estimate of γ_p, $\hat{\gamma_p}$, we apply the MMSE (Minimum Mean Square Error) filter w_p to $\mathcal{Y}$:

$$\hat{\gamma_p} = w_p^t \mathcal{Y} \tag{C.11}$$

where

$$w_p^t = \frac{h_p}{(\mathcal{H}_{\text{eq}}{\mathcal{H}_{\text{eq}}}^t + \sigma^2 I)} \tag{C.12}$$

Here, h_p of dimension ($2N_rT$ x 1) is the p^{th} column of $\mathcal{H}_{\text{eq}}$ From the second iteration, we calculate soft estimates of the transmit symbols $\hat{\mathcal{S}}$ using the SISO decoder outputs and use them in the PIC detector to perform interference cancellation followed by simplified MMSE detection :

$$\begin{cases} \text{Interference cancellation : } \hat{\mathcal{Y}}_{\text{p}} = \mathcal{Y} - \mathcal{H}_{\text{p}}\hat{\mathcal{S}}_{\text{p}} \\ \text{MMSE filtering : } \hat{\gamma_{\text{p}}} = w_p^t \hat{\mathcal{Y}}_{\text{p}};\ \ w_p^t = \frac{h_p}{h_p^t h_p + \sigma^2} \end{cases} \tag{C.13}$$

where $\hat{\mathcal{S}}_{\mathrm{p}}$ of dimension ((2 Q -1) x 1) is $\hat{\mathcal{S}}$ with its p^{th} entry removed, and $\mathcal{H}_{\mathrm{p}}$ of dimension ($2N_rT$ x $(2Q-1)$) is $\mathcal{H}_{\mathrm{eq}}$ with its p^{th} column removed. In fact, C.13 is a suboptimal solution to the detection problem, and has a considerable reduced computational complexity, compared to the exact solution. We have verified by simulations that, when used in an iterativ escheme, this suboptimal detector mostly converges to the optimal solution at high enough SNR and when there is enough diversity available, e.g., when the number of receive antennas is large enough. For the case of orthogonal ST schemes, we perform maximum likelihood (ML) detection once (without iteration).

Remark : In our case, we considered the orthogonal scheme of Alamouti and the MUX scheme. So, we consider the PIC detector with one iteration, which will be simplified to the case of the linear detection MMSE.

Bibliographie

[1] D. Le Ruyet, *Théorie de l'information, Codage Source-Canal*, Ecole Supérieure de Conception et de Production industrielles, France, Jan. 2007.

[2] B. Kurkoski, "Introduction to low density parity check codes," 2007.

[3] M. Luby, M. Mitzemmacher, A. Shokrollahi, and D. Spielman, "Improved Low Density Parity Check codes using irregular graphs and belief propagation," *IEEE Transactions on Information Theory*, Apr. 1998.

[4] R. Neal, "Faster encoding for low density parity check codes using sparse matrix methods," *IMA Program on codes, systems and graphical Models, University of Toronto Canada*, 1999.

[5] D. L. Ruyet and B. Ozbek, "Systèmes MIMO et codage spatio-temporel," *CNAM Paris*.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding," *IEEE International Conference on Communications*, vol. 2, May 1993.

[7] N. Aitamer, "Apport de la diversité spatiale pour les transmissions optiques sans fil," Rapport de master, Institut Fresnel, Juillet 2008.

[8] Le site officiel de l'Institut Fresnel, "http ://www.fresnel.fr/," .

[9] J. L. Lacoume, "Modèles et traitements de signaux multidimentionnels," Mai 1987.

[10] H. Nyquist, "Certain factors affecting telegraph speed," *Bell System Technical Journal*, vol. 3, pp. 324–346, 1924.

[11] R. Hartley, "Transmission of information," *Bell System Technical Journal*, July 1928.

[12] C. E. Shannon, "Mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, Oct. 1948.

[13] K.-L. Du and M. N. S. Swamy, *Wireless Communication Systems*, Cambridge University Press, New York, Feb. 2009.

[14] G. Battail, *Théorie de l'information, Application aux techniques de communication*, Collection Pédagogique de télécommunication. Masson, Paris, 1997.

[15] A. Gersho and R. M. Gray, *Vector Quantization and Compression*, Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[16] M. A. Khalighi, "Iterative decoding and detection (turbo methods), principles and related algorithms," Report, INPG University, Gronoble, France, Sept. 2002.

[17] J. G. Proakis, *Digital Communication*, McGraw Hill, 5 edition, 2008.

[18] A. Galvieux, *Codage de canal des bases théoriques aux turbocodes*, Lavoisier, 2005.

[19] J. Oswald et G. Battail, *Théorie de l'information , Analyse diacritique des systèmes*, Edition Masson, 2006.

[20] J. B. Doré, *Optimisation de codes LDPC et leur architectures de décodage et mise en oeuvre sur FPGA*, Ph.D. thesis, INSA de Rennes, Octobre 2007.

[21] A. Le Glaunec, "Corps de Galois : Aide mémoire," .

[22] G. D. Forney, *Concatenated Codes*, Ph.D. thesis, MIT Press, Combridge, 1966.

[23] A. Spataru, *Fondements de la théorie de la transmission de l'information*, 1991.

[24] W. Peterson and E. J. Welden, "Error-correcting codes," *MIT Press, Cambridge*, 1961.

[25] C. Berrou, *Codes et Turbo-Codes ; chapitre 3 limites théoriques*, Springer, 2007.

[26] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the SIAM*, pp. 300–304, June 1960.

[27] ESTI Standard, *Digital Video Broadcasting (DVB) :Framing structure, channel coding and modulation for digital terrestrial television*, Digital Video Broadcasting, 2004.

[28] V. D. Goppa, "A new class of linear error correcting codes," *Problemy Peredachi Informatsi*, vol. 6, pp. 207–212, Sept. 1970.

[29] B. Sakkour, *Etude du décodage des codes Reed-Muller et application à la cryptographie*, Ph.D. thesis, UMA-ENSTA, 2007.

[30] M. Kanemasu, "Golay codes," *MIT Undergraduate Journal of Mathematics*, 2000.

[31] P. Elias, "Error-free coding," *IEEE Transactions on Information Theory*, vol. 4, pp. 29–37, Sept. 1954.

[32] A. Viterbi, "Error bounds for convolutional codes and asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, Apr. 1967.

[33] A. Viterbi and O. K. Omura, *Principales of Digital Communications and Coding*, McGraw Hill, 1978.

[34] J. Hagenouer, E. Offer, and L. Papke, "Iterative decoding of binary blocs and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, pp. 429–445, Mar. 1996.

[35] M. J. Bernhard, "LDPC Codes : a brief Tutorial," Apr. 2005.

[36] R. Gallager, *Low-Density-Parity-Check Codes*, Ph.D. thesis, MIT Cambridge, 1963.

[37] R. Gallager, "Low density parity check codes," *IEEE Transactions on Information Theory*, vol. 8, pp. 21–28, 1962.

[38] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, Sept. 1981.

[39] D. Mackay and R. M. Neal, "Near schannon limit performance of low-density-parity-check codes," *Electronic Letter*, Aug. 1996.

[40] J. Pearl, *Probabilistic reasonning in intellegent systems : Networks of plausible inference*, Morgan kaufman Publeshers, 1988.

[41] S. Y. Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the design of Low Density Parity Check Codes within 0.045 dB of the Shannon Limit," *IEEE Communication Letters*, vol. 2, pp. 58–60, 2001.

[42] F. Hamon, N. Fau, C. Wolinski, and F. Chariot, "A new powerful scalable generic multi-standard LDPC decoder architecture," *Proceedings of the* 16^{th} *IEEE Symposium on Field Programmable Custom Computing Machines, Palo Alto California USA*, 2008.

[43] ISSCC Paper 24.3, *A 135 Mbps DVB-S2 Compliant Codes based on 64800-bit LDPC and BCH Codes*, 2004.

[44] ESTI ; European Standard Telecomminications series, *Digital Vidéo Broadcasting DVB : Second generation framing structure, channel coding and modulation systems, interactive services, new generation and other broadcasting applications*, June 2004.

[45] IEEE Standard for Local and Metropolitain Area Networks, *Part 16 : Air Interface for fixed and mobile broadband wireless access systems*, IEEE Project 802.16e, 2005.

[46] CCSDS Standard, *Low Density Parity Check for use in the Near-Earth and Deep Space Applications*, Orange Book, Sept. 2007.

[47] S. M. Bilfagih, *Transmission multiple porteuses utilisant un codage détecteur/correcteur d'erreur de type LDPC sur canaux MIMO*, Ph.D. thesis, Université de LIMOGES, Mars 2005.

[48] J. Sun, "An introduction to low density parity check codes," *Wireless Communication Research Laboratory Lane Department of Computer Science and Electrical Engineering, West Virginia University*, June 2003.

[49] M. Luby, M. Mitzemmacher, and A. Shokrollahi, "Analysis of low density parity check codes and improved design using irregular graphs," *STOC*, 1998.

[50] N. Wiberg, *Coding and decoding in general graphs*, Ph.D. thesis, Department of Electrical Engineering, Linkoping, Sweden, 1996.

[51] T. J. Richardson and R. Urbanke, "The capacity of LDPC codes under message passing decoding," *IEEE Transactions on Information Theory*, vol. 47, pp. 599–618, Feb. 2001.

[52] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graph and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, Feb. 2001.

[53] T. Bayes, "Studies in the history of probability and statistics : Tomas bayes's essay towords solving a problem in the doctrine of chance," *Biometrika*, vol. 45, pp. 296–315, 1958.

[54] F. Lehmann, *Les systèmes de décodage itératif et leurs applications aux modems filaires et non filaires*, Ph.D. thesis, INP de Gronoble, Dec. 2002.

[55] M. Fossorier, M. Mihajevic, and H. Imai, "Reduced complexity iterative decoding performances," *International Symposium on Information Theory*, Sept. 2005.

[56] J. Chen and M. Fossorier, "Density evolution of two improved BP-Based algorithms for LDPC decoding," *IEEE Transactions on Communications*, vol. 47, May 1999.

[57] F. Gulloud, *Generic Architecture for LDPC Codes Decoding*, Ph.D. thesis, Télécom Paris, July 2004.

[58] J. M. Moure, J. Lu, and H. Zhang, "Structured low density parity check codes," *IEEE Signal Processing Magazine*, vol. 47, pp. 498–519, Jan. 2004.

[59] S. Muller, M. Schreger, M. Kabutz, M. Alles, F. Kienle, and N. Wehn, "A novel LDPC decoder for DVB-S2 IP," *THOMSON Architecture Group and the University of Kaiserslautern*, 2009.

[60] F. Demongel, N. Fau, and N. Drabik, "A Generic Architecture of CCDS Low Density Parity Check Decoder for Near-Earth Applications," *DATE*, 2009.

[61] S. Ahmadi, "An overview of the Next-Generation Mobile WIMAX technology," *IEEE Communications Magazine*, vol. 47, pp. 84–98, June 2009.

[62] 802.16 Standard, *802.16 LDPC Encoder v 1.0 : Product specification XILINX*, July 2006.

[63] Softwares for LDPC Codes, "http ://www.cs-toronto.edu/ radford/ldpc.software," .

[64] E. W. Ghebache, "Evaluation des systèmes radio mobile à MC-CDMA ," Tech. Rep., Ecole Nationale Superieure Polytechnique d'Alger, June 2007.

[65] E. Telatar, "Capacity of multiple antenna gaussian channel," *European Transactions on Telecommunications*, vol. 10, pp. 585–595, Nov. 1999.

[66] G. I. Foschini and M. J. Gans, "On the limits of wireless communications in fading environment when using multiple antennas," *Wireless Personal Communications*, vol. 6, pp. 311–335, 1998.

[67] G. I. Foschini, G. D. Golden, R. A. Valenzuelo, and Wolniansky, "Simplified processing for high spectral efficienty wireless communication employing multi-elements arrays," *IEEE Journal on Selected Areas on Communications*, vol. 17, pp. 1841–1852, 1999.

[68] M. A. Khalighi, J. F. Hélard, S. M. Sadough, and S. Bourennane, "Suitable combination of channel coding and space-time schemes for moderate-to high spectral efficiency MIMO systems," *International journal AEU of Electronics and Communications*, Mar. 2009.

[69] V. Tarokh, N. Seshadri, A. Naguib, and A. R. Calderbank, "Space-time codes dor high data rate wireless communications : Performance criteria in presence of channel estimation error mobility and multipaths," *IEEE Transactions on Communications*, vol. 16, pp. 1451–1458, Oct. 1998.

[70] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "STBC from Orthogonal Designs," *IEEE Transactions on Information Theory*, vol. 45, pp. 1456–1467, July 1999.

[71] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas on Communications*, vol. 16, pp. 1451–1458, 1998.

[72] Documentation sur l'éditeur/compilateur C Code Blocks, "http ://www.codeblocks.org/," .

[73] B. Hochwald B. Hassibi, "High-rate codes that are linear in space and time.," *IEEE Transactions on Information Theory*, 2002.