N° attribué par la bibliothèque :

| | | R | E | I | | | | | | |

UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

École Doctorale Sciences Technologies Santé

THÈSE

Pour obtenir le grade de

Docteur de l'Université de Reims Champagne-Ardenne

Spécialité : Génie informatique, Automatique et Traitement du Signal

Par

Hassen BEN MAAD

Optimisation des stratégies de décodage des codes LDPC dans les environnements impulsifs : application aux réseaux de capteurs et ad hoc

Soutenue à l'Université de Reims Champagne Ardenne le 29 Juin 2011 devant le jury composé de :

Président

Martine Lienard	Professeure, Université de Lille 1

Rapporteurs :

Hichem Snoussi	Professeur, l'UT de Troyes
Charly Pouillat	Maître de Conférences, ENSEA Cergy-Pontoise

Examinateurs

Guillaume Gelle	Professeur, Université de Reims
Laurent Clavier	Maître de conférences, TELECOM Lille 1
Alban Goupil	Maître de conférences, Université de Reims

CReSTIC

IRCICA

A mes parents ma fiancée et mes frères. A tous ceux que j'aime..

Penser importe plus que savoir ; combien de savants ne pensent pas ? Savoir du reste est nécessaire, mais savoir quoi ? Ceci est la question.

Victor Hugo, Océan prose.

Remerciements

Je tiens ici à exprimer toute ma reconnaissance aux personnes qui m'ont aidé, encouragé et soutenu pendant ces années de préparation au Doctorat.

Je veux tout d'abord remercié **Mme. Martine Lienard** pour m'avoir fait l'honneur d'accepter de présider ce jury.

Je tiens à adresser ma profonde gratitude et sincère reconnaissance aux **M. Hichem Snoussi** et **M. Charly Poulliat** pour avoir accepté d'être rapporteurs de cette thèse. Je les remercie pour l'intérêt qu'ils ont porté à ce manuscrit ainsi que pour leurs remarques constructives et encourageantes.

Cette thèse a été réalisée en collaboration au sein du laboratoire ***CReSTIC*** sous la direction du Professeur **Janan Zaytoon** ainsi qu'au laboratoire ***IRCICA*** sous la direction du Professeur **Paul Alain Roland**. Je voudrais les remercier de m'avoir accueilli dans leurs équipes, ainsi que pour la confiance qu'ils m'ont témoignée durant cette thèse.

Toute ma reconnaissance pour mes directeurs de thèse, **M. Guillaume Gelle**, **M. Laurent Clavier** et **M. Alban Goupil** pour m'avoir constamment suivi durant ce travail. Votre disponibilité, votre complémentarité, vos qualités scientifiques dont j'ai essayé au maximum de m'imprégner, font que sans votre précieuse aide ce travail n'aurait sans doute pu voir le jour. En y associant en plus vos qualités humaine, sachez quel plaisir ce fut de travailler à vos cotés.

Je remercie aussi ma famille, et en particulier mes parents, pour leur patience et la compréhension dont ils ont fait preuve tout au long de mes études.

Et puis un très grand merci à mon future épouse **Ihsen**, qui m'a soutenu tout au long de ce travail, et qui a su apporter avec talent et enthousiasme les touches finales à ce manuscrit.

Je voudrais aussi remercier tout le personnel du ***CReSTI***C et du ***IRCICA*** ainsi que mes chers collègues au sein des deux laboratoires, avec lesquels j'ai partagé tant de choses et qui ont fait de ma vie à Reims et Lille une expérience inoubliable.

Enfin, je remercie d'avance tous les futurs lecteurs et lectrices qui prendront la peine de parcourir ce manuscrit, car remercier quelqu'un, c'est avant tout s'assurer de son indulgence…

Hassen Ben Mâad (Juin 2011)

Table des matières

Introduction générale **1**

1 Contexte de l'étude **1**
2 Objectifs de l'étude et contributions **2**
3 Plan de l'étude **2**

Canal AISαSN **6**

1 Introduction **7**
2 Modèles des environnements non-gaussien **7**

2.1 Le modèle gaussien 8
2.2 Modèles non gaussiens 9

2.2.1 Modèle mélange de gaussiennes 9

2.3 Modèles à queues lourdes 10

2.3.1 Modèle de Laplace 11
2.3.2 Modèle de gaussienne généralisé 12

2.4 Modèles à queues algébriques 13

2.4.1 Modèle de Pareto 13
2.4.2 Modèle *t* de student 14
2.4.3 Modèle α-stable 14

3 Lois α-stables **16**

3.1 Définitions des lois α-stables 17
3.2 Densité de probabilité 19

4 Les moments des processus α-stables **22**

4.1 Moments fractionnaires d'ordre positif 22
4.2 Moments fractionnaires d'ordre négatif 23
4.3 Limites des moments fractionnaires 24
4.4 Moments d'ordre logarithmique 24

5 Puissance des processus α-stable **25**

5.1 Impossibilité d'une puissance quadratique 25
5.2 Puissance géométrique 26

5.2.1 Définition 26
5.2.2 Propriétés 27
5.2.3 Puissance géométrique d'une distribution α-stable symétrique 28

6 Le rapport signal sur bruit géométrique **28**
7 Capacité du canal AISαSN **28**

7.1 Définition du canal AISαSN 29
7.2 Capacité d'un canal AISαSN à entrée binaire 29
7.3 Comportement de la capacité du canal AISαSN 31

8 Performance de la chaine de transmission dans les environnements AISαSN 32
9 Conclusion 33

Les codes LDPC 36

1 Introduction 37
2 Les codes LDPC 37

2.1 Définition et type de code LDPC 37
2.2 Construction des codes 39
2.3 Encodage 40

3 Représentations graphiques d'un code LDPC 40

3.1 Graphe factoriel 41
3.2 Graphe factoriel normal 41

4 Décodage des codes LDPC 43

4.1 Décodage MAP 43
4.2 Algorithme Sum-Product 44
4.3 Algorithmes de décodage 45

4.3.1 Bit-flipping, Gallager A et B 46
4.3.2 Belief Propagation (BP) 46
4.3.3 Algorithme Min-Sum 49

5 Analyse des performances asymptotiques des codes LDPC 50

5.1 Exemple d'analyse asymptotique : le canal à effacement 51
5.2 Évolution de Densité 52
5.3 Approximation gaussienne et EXIT charts 53
5.4 Utilisation des analyses asymptotiques 53

6 Les paramètres d'entrer du démappeur LDPC : les LLR 54
7 Performances asymptotiques des codes LDPC dans les environnements AISαSN 55
8 Conclusion 57

Performances des codes LDPC dans un environnement AISαSN 60

1 Introduction 61
2 Codes LDPC dans les environnements AISαSN 62

2.1 Cas du démappeur optimal 62
2.2 Cas des démappeurs basés sur des expressions analytiques des LLR 64

3 Démappeurs robustes en environnement α-stable 66

3.1 Le démappeur Linéaire 66
3.2 Le démappeur de Clipping 68

3.2.1 Études basées sur les LLR 69
3.2.2 Études basées sur la puissance de bruit après clipping 77
3.2.3 Étude basée l'Évolution Densité 82

3.2.4 Bilan des différentes études....86
3.2.5 Sensibilité à une mauvaise estimation de α....89
3.3 Le démappeur Hole Puncher....89
3.4 Forme approchée du démappeur....94
4 Décodage Min-Sum....99
4.1 Cas des démappeurs optimal et basés sur la vraisemblance....99
4.2 Cas du démappeur de Clipping....100
5 Conclusion....105
Conclusions et Perspectives....109
Bibliographie....112

Table des figures

Fig.1.1 : Densités de probabilités d'une distribution normale pour différents μ et σ^2 9
Fig.1.2 : Densité de probabilité d'une distribution de mélange de gaussienne 10
Fig.1.3 : Densités de probabilités d'une distribution de Laplace pour différents μ et b 11
Fig.1.4 : Densités de probabilités d'une distribution gaussienne généralisée pour différents p 12
Fig.1.5 : Réalisation de signaux SαS pour différents valeurs d'alpha. 19
Fig.1.6: Densités de probabilité d'une distribution SαS pour différents α 21
Fig.1.7 : Effet de α sur la queue de la densité de probabilité de la distribution SαS 21
Fig.1.8 : Effet de γ sur la densité de probabilité de la distribution SαS 21
Fig.1.9 : Effet du paramètre de symétrie sur la densité de probabilité de la distribution SαS 22
Fig.1.10 : Les moments de lois α-stables pour différentes valeurs de α 23
Fig.1.11 : Comportement du LLR en fonction de la sortie du canal AISαSN pour un γ fixe et différents α 30
Fig.1.12 : Comportement du LLR en fonction de la sortie du canal AISαSN pour un α fixe et différents γ. 31
Fig.1.13 : Variation de la capacité du canal AISαSN en fonction de γ et pour différents α 32
Fig.1.14 : Chaine de transmission sans fil 32
Fig.1.15 : Performance de la chaine de transmission dans les environnements AISαSN 33
Fig.2.1 : Exemple de graphe factoriel 41
Fig.2.2 : Exemple de graphe factoriel normal représentant $Z_G(Y)$. 42
Fig.2.3 : Exemple de sommation dans un graphe factoriel normal. 42
Fig.2.4 : Exemple de code LDPC sous forme de graphe factoriel normal. 43
Fig.2.5 : Exemple de sommation dans un graphe factoriel normal. 44
Fig.2.6 : Décomposition par un bloc. 45
Fig.2.7 : Graphe factoriel normale pour le décodage MAP. 46
Fig.2.8 : Récupération de la valeur d'un bit. 47
Fig.2.9 : Graphe factoriel normale pour le décodage MAP d'un bit. 48
Fig. 2.10 : Illustration de trois régions caractérisant les performances d'un code LDPC 50
Fig. 2.11 : Différents LLR en fonction de sortie du canal 55
Fig.2.12 : Performances asymptotiques des différents types de codes LDPC utilisant différents demappeurs dans les environnements AISαSN (a) α=1, (b) α=1.5, (c) α=2 57

Fig.3.1 : Performance de la chaine de transmission avec et sans codes LDPC dans différents environnements AISαSN(a)α = 2, (b) α = 1.5, (c)α = 1.2 63
Fig.3.2 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=2, (b) α=1.8, (c) α=1.5, (d) α=1.2 65
Fig.3.3 : Variation des performances asymptotiques des codes LDPC en fonction de variation de la pente pour différents α 67
Fig.3.4 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=1.4, (b) α=1.8 68
Fig. 3.5 : Comportement des différents démappeurs en fonction de la variation du sortie de canal y. 69
Fig. 3.6 : comportement des démappeurs de clipping et optimal en fonction de la variation du sortie de canal y. 69
Fig.3.7 : Variation de la pente P en fonction des paramètres α et γ. 70
Fig.3.8 : Variation du seuil H en fonction des paramètres α et γ. 71
Fig.3.9 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=1.2, (b) α=1.4, (c) α=1.8 73
Fig.3.10 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=1.2, (b) α=1.4, (c) α=2 76
Fig.3.11 : P_{arit} = f (H/P) pour différents α et pour un γ = 0.7 78
Fig.3.12 : P_{arit} = f (H/P) pour différents γ et pour un α = 1.8 78
Fig.3.13 : H/P = f (α, γ) où P_{arith} est minimale pour différents pentes 79
Fig.3.14 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=1.8, (b) α=1.4 80
Fig.3.15 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=2, (b) α=1.4, (c) α=1.2 82
Fig.3.16 : Comportement des performances asymptotiques du démappeur de Clipping en fonction des variations des P et H dans un environnement AISαSN (α=1.5) 83
Fig.3.17 : Comportement des performances asymptotiques dans différents environnements AISαSN : (a) α=1.2 (b) α=2 84
Fig.3.18 : comportement du seuil de Clipping H en fonction de variation de α 85
Fig.3.19 : comportement de la pente P en fonction de variation de α 85
Fig.3.20 : comportement du rapport H/ P en fonction de variation de α 85
Fig.3.20 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=1.2, (b) α=1.4, (c) α=1.8, (d) α=2 88
Fig.3.21 : Comparaison des performances du démappeur de Clipping pour un P et H fixes et pour différents α proches α = 1.4 89

Fig.3.22 : Comportement des performances asymptotiques du démappeur Hole Puncher en fonction des variations de P et H dans un environnement AISαSN (α=1.5) 90
Fig.3.23: Comportement des performances asymptotiques dans différents environnements AISαSN : (a) α=1.8 (b) α=1.2 91
Fig.3.24 : comportement du seuil de retour en zéro H en fonction de variation de α 91
Fig.3.25 : comportement de la pente P en fonction de variation de α 92
Fig.3.26 : Comparaison des performances des démappeurs de Clipping et Hole puncher pour des P et H trouvés par évolution densité (a) α =1.2, (b) α =1.8, (c) α =2, ... 93
Fig. 3.27 : Comportement du démappeur approche zéro en fonction de la variation du sortie de la canal y. 94
Fig.3.28 : Comportement des performances asymptotiques du démappeur Approch zero en fonction des variations de P et H dans un environnement AISαSN (α=1.5) 95
Fig.3.29 : Comportement des performances asymptotiques dans différents environnements AISαSN : (a) α=1.2 (b) α=1.8 96
Fig.3.30 : comportement du seuil H en fonction de variation de α 96
Fig.3.31 : comportement de la pente P en fonction de variation de α 97
Fig.3.32 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α = 1.2, (b) α = 1.4, (c) α = 1.8 98
Fig.3.33 : Comparaison des performances du code LDPC (Mac-20000) utilisant BP et MS dans différents environnements AISαSN : (a)α=1.8, (b) α=1.2 100
Fig.3.34 : Variation des performances asymptotiques des codes LDPC en fonction de variation du taux de clipping pour différents α 101
Fig.3.34 : Variation des performances asymptotiques des codes LDPC en fonction de variation du seuil de clipping pour différents α 101
Fig.3.35 : Performances du code LDPC (Mac-20000) utilisant le démappeur de clipping de différents seuils de clipping dans un environnement AISαSN α = 1.2 102
Fig.3.36 : Performances du code LDPC (Mac-20000) utilisant le démappeur de clipping de différents taux de clipping dans un environnement AISαSN α = 1.2 102
Fig.3.36 : Variation des performances du code LDPC (Mac-20000) en fonction du taux de clipping pour γ = 0.3 et α = 1.2 103
Fig.3.36 : Variation des performances du code LDPC (Mac-20000) en fonction du seuil de clipping pour γ = 0.3 et α = 1.2 103
Fig.3.37 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a) α = 1.2, (b) α = 1.4, (c) α = 1.8 105

Introduction générale

1 Contexte de l'étude

L'essor considérable des communications numériques constitue une des principales révolutions technologiques des dernières décennies que ce soit la téléphonie mobile, l'internet, les transmissions par satellites, par courant porteur ou sans fil, etc. Elles permettent d'avoir des systèmes de transmission de plus en plus performants en termes de haut débit, de qualité de service et de mobilité qui témoignent d'un véritable besoin de nos sociétés actuelles. Cette révolution n'aurait pu avoir lieu sans le développement et la conception de circuits complexes et miniaturisés permettant d'implémenter efficacement des algorithmes de décodage de plus en plus performants.

Ce boom technologique a ouvert de nombreuses voies de recherche dans les systèmes de communications numériques et plus précisément dans le domaine du codage. Ainsi, la découverte des Turbo-codes et, plus généralement du principe itératif appliqué au traitement du signal, a révolutionné la manière d'appréhender un système de communications numériques. Cette avancée notable a permis la re-découverte des codes correcteurs d'erreurs inventés par R. Gallager en 1963, appelés codes Low Density Parity Check (LDPC). La découverte de ces deux grandes familles de codes a permis d'avoir des systèmes de transmissions très performants et témoigne de nos véritables besoins. Ils sont inclus dans la définition de nouveaux standards de communications. En 1999, dans le domaine des radiocommunications mobiles de 3ème génération, la norme européenne UMTS (Universal Mobile Telecommunications System) ainsi que son homologue américain CDMA2000 (Code Division Multiple Access) ont retenu les Turbo-codes pour les services de données à haut débit. Les codes LDPC ont fait leur apparition dans le monde industriel en 2003 par leur introduction dans le domaine de la diffusion vidéo numérique par satellite de 2ème génération, DVB-S2 (Digital Video Broadcasting - Satellite - Second Generation) ainsi que dans les standards IEEE 802.16e (Wimax mobile), IEEE802.11n (Wifi), etc. L'existence d'outils permettant une analyse théorique des codes LDPC, leurs algorithmes de décodage itératifs avec une complexité linéaire et le fait qu'ils ne soient pas brevetés contrairement aux turbo-codes nous ont motivés pour mener des travaux dans le domaine des codes LDPC.

Cependant, dans les systèmes de communication tant filaire que sans fil, le phénomène d'interférence est présent et est souvent un sérieux frein au développement de ces dits systèmes. En effet, cette interférence modifie et donc perturbe le signal d'intérêt. Cette interférence apparaît sous de multiples formes. Nous pouvons citer par exemple l'interférence entre symboles liée à la présence d'un canal de propagation de type multi-trajets, l'interférence entre utilisateurs (comme en systèmes UWB, en réseaux *ad hoc* et de capteurs), l'interférence entre porteuses (dans un système OFDM mal synchronisé en fréquence) ou encore l'interférence entre différents canaux (comme le problème de diaphonie en ADSL ou bien encore le problème des canaux adjacents en communication sans fil). Dans ce cadre, il est important de bien modéliser l'interférence afin de limiter son impact négatif. Différents modèles ont été proposés dans ce contexte. L'étude de tels modèles est importante pour deux raisons majeures. Tout d'abord, d'un point de vue fondamental, elle apporte une meilleure compréhension des phénomènes. Ensuite, du point de vue pratique, elle oriente la conception des systèmes de transmission ainsi que le choix des stratégies de communication.

2 Objectifs de l'étude et contributions

L'objectif principal de ce travail de recherche est de concevoir une stratégie simple et robuste de décodage des codes LDPC dans des environnements impulsifs. Cette thèse s'est axée sur deux grands domaines :

- Le premier représente l'étude, l'analyse et la modélisation des phénomènes impulsifs qui apparaissent dans les systèmes de transmission sans fil ;

- Le second concerne les modifications à faire sur les entrées des décodeurs LDPC pour faire face aux environnements impulsifs. Ces modifications cherchent la simplicité de mise en œuvre, la robustesse tout en gardant des performances respectables grâce à une optimisation de leur paramétrisation.

La contribution essentielle est l'amélioration des performances des codes LDPC dans un environnement impulsif dont nous avons déterminé la capacité. Cette amélioration vient principalement de la fonction permettant de calculer le rapport de vraisemblance à l'entrée du décodeur. Nous constatons que la fonction linéaire considérée dans un environnement gaussien n'est pas performante en cas de bruit impulsif. Nous proposons également des fonctions simplifiées pour déterminées cette vraisemblance dans le cadre du décodage par propagation de croyance. Enfin, nous étudions également l'intérêt que pour présenter l'algorithme de décodage Mim-Sum dans notre contexte.

3 Plan de l'étude

Le cœur de ce mémoire est composé de trois chapitres dont le détail de chacun des chapitres est décrit ci-dessous.

Le premier chapitre est consacré à l'étude au choix d'un modèle de distribution permettant une représentation précise et fiable des phénomènes impulsifs qui apparaissent dans les systèmes de transmission.

Dans la première partie de ce chapitre, différentes distributions possibles sont présentées. Il en ressort que les distributions α-stables modélisent bien les phénomènes impulsifs, rares mais de forte amplitude dans ces systèmes. Ce type de distributions est stable par convolution et constitue l'ensemble des lois qui vérifient un théorème central limite généralisé. Elles sont à queues lourdes et, sauf dans le cas gaussien qui fait partie de cette classe, de variance infinie. La conséquence est que le rapport signal sur bruit tel qu'il est habituellement défini, est nul.

Afin de résoudre ces problèmes, nous étudions en deuxième partie de ce chapitre les façons de caractériser ces processus impulsifs à variance infinie. La théorie des statistiques d'ordre zéro, basée sur les moments d'ordre logarithmique, est une solution. Ces moments permettent de définir une nouvelle notion de puissance : la puissance géométrique. Nous en dérivons une mesure de la qualité d'une transmission : le rapport signal à bruit géométrique.

Par la suite, en troisième partie de ce chapitre, nous décrivons le canal de transmission pour un environnement impulsif : le canal à bruit additif symétrique α-stable à échantillons indépendants. En continuité avec le canal nommé Additive White Gaussian Noise (AWGN), nous avons choisi de le

nommer Additive Independent Samples α-Stable Noise (AISαSN). Ce type de canal a une capacité au sens de Shannon qui dépend des paramètres de la distribution α-stable.

A la fin de ce chapitre nous proposons de montrer l'effet de la variation de la nature de ce canal sur le comportement des performances d'une transmission BPSK sans codage.

Le second chapitre constitue une présentation des codes LDPC incluant les notations et les outils algorithmiques indispensables à une bonne compréhension du document. Dans un premier temps nous introduisons les codes LDPC en décrivant leurs propriétés, les algorithmes d'encodage ainsi que leurs différentes représentations graphiques. Le décodage par l'algorithme de propagation de croyance ou par les algorithmes approchés comme le Min-Sum sont présentés dans une deuxième partie du chapitre. L'analyse de la convergence des codes lors de la phase de décodage est indispensable. L'outil permettant cette analyse, nommé évolution densité, est détaillé dans une troisième partie. Il permet d'évaluer les performances asymptotiques des codes LDPC et, en particulier, de comparer les performances espérées à la capacité des environnements AISαSN. Enfin nous proposons à la fin de ce chapitre différentes solutions afin de déterminer le logarithme du rapport de vraisemblance à l'entrée du décodeur LDPC. Ces calculs utilisent les lois α-stables. Comme leur densité probabilité n'a, dans les cas général, pas d'expression analytique, nous testons également quelques méthodes approchées mais permettant un calcul analytique. Les différents résultats obtenus dans cette partie seront pris comme références pour le chapitre suivant.

Le chapitre 3, quant à lui, est essentiellement consacré à étudier la robustesse des codes LDPC dans les environnements impulsifs modélisés par la distribution α-stable. La première partie de ce chapitre présente l'effet du démappeur, qui transforme la sortie du canal en une entrée utilisable par l'algorithme de décodage des codes LDPC, sur les performances dans les environnements AISαSN. Nous proposons plusieurs approches. Pour construire les premiers, nous avons utilisé les densités de probabilités exactes de la distribution α-stable calculées numériquement. Nous en avons aussi dérivés grâce des densités de probabilité qui présentent des expressions analytiques. Parmi ces différents démappeurs, il n'en existe cependant pas qui soient à la fois simples et robustes quelque doit la nature de l'environnement AISαSN.

Cette dernière constatation, nous amène dans la deuxième partie de ce chapitre à proposer d'autres types de démappeurs qui s'adaptent facilement à tous les environnements AISαSN. Dans ce contexte le démappeur linéaire, celui de clipping, le hole puncher et une fonction approchée ont été étudiés. Le démappeur linéaire avec sa forme simple, défini simplement par sa pente, donne aux signaux puissants une trop grande importance. Le démappeur de clipping, défini par deux paramètres : une pente et un seuil, permet de limiter l'impact des grandes valeurs en les écrêtant. Encore plus radical, le démappeur hole puncher, également défini par deux paramètres annule les grandes valeurs des signaux. Enfin le démappeur approché est une version intermédiaire entre le clipping et le hole puncher en laissant un plancher avant l'annulation des valeurs hors limite. Il est inspiré par l'allure du logarithme du rapport de vraisemblance exact dans un bruit α-stable.

N'ayant pas de solution analytiques pour déterminer les meilleurs paramètres de ces différents démappeurs, nous proposons plusieurs solutions afin de les obtenir en fonction des paramètres de l'environnement, c'est-à-dire des paramètres de la distribution α-stable du bruit : en nous basant sur l'allure des logarithmes du rapport de vraisemblance, en utilisant l'évolution de densité ou en minimisant la puissance de bruit en sortie du démappeur.

Finalement nous étudions l'algorithme de décodage Min-Sum qui permet de réduire la complexité de l'algorithme de décodage par propagation de croyance et qui permet également de réduire le nombre de paramètres significatifs des démappeurs, l'échelle (la pente) n'ayant plus d'impact. Il reste cependant moins performant que l'algorithme BP.

Une conclusion générale synthétise les différentes idées et résultats présentés dans ce document et propose des perspectives pour des travaux futurs.

Canal AISαSN

1 Introduction
2 Modèles des environnements non-Gaussien

2.1 Le modèle Gaussien
2.2 Modèles non Gaussiens

2.2.1 Modèle mélange de Gaussiennes

2.3 Modèle à queues lourdes

2.3.1 Modèle de Laplace
2.3.2 Modèle de Gaussienne généralisé

2.4 Modèle à queues algébriques

2.4.1 Modèle de Pareto
2.4.2 Modèle T généralisé
2.4.3 Modèle α-stable

3 Lois α-stables

3.1 Définitions des lois α-stables
3.2 Densité de probabilité

4 Les moments des processus α-stables

4.1 Moments fractionnaires d'ordre positif
4.2 Moments fractionnaires d'ordre négatif
4.3 Limites des moments fractionnaires
4.4 Moments d'ordre logarithmique

5 Puissance des processus α-stable

5.1 Impossibilité d'une puissance quadratique
5.2 Puissance géométrique

5.2.1 Définition
5.2.2 Propriétés
5.2.3 Puissance géométrique d'une distribution α-stable symétrique

6 Le rapport signal sur bruit géométrique GSNR
7 Capacité du canal AISαSN

7.1 Définition du canal AISαSN
7.2 Capacité d'un canal AISαSN à entrée binaire
7.3 Comportement de la capacité du canal AISαSN

8 Performance de la chaine de transmission dans les environnements AISαSN
9 Conclusion

1 Introduction

Une limite importante des réseaux de capteurs denses sera très certainement l'interférence. Si actuellement les couches réseaux essaient de la limiter par des protocoles de *sensing* et en interdisant l'accès à la ressource à des interférents potentiels, plusieurs phénomènes peuvent rendre cette approche hasardeuse [1]. Tout d'abord elle n'est pas très efficace, limitant fortement la réutilisation spatiale quand des communications simultanées seraient rendues possibles par une couche physique robuste. Ensuite l'effet des canaux multi-trajets est rarement pris en compte dans ce genre d'étude et les résultats pratiques peuvent en être affectés. Plutôt que d'essayer d'éliminer l'interférence par une approche réseau, de nombreux travaux comme l'alignement d'interférence [64], les réseaux MIMO [2], et les détecteurs multi-utilisateurs [3] essaient d'éliminer cette interférence ou de ne pas la considérer comme de l'interférence mais comme une information. Ces approches se heurtent cependant aux problèmes de synchronisation et à la nécessité de connaître précisément l'environnement. Ces points sont incompatibles avec des réseaux de capteurs bas coût. Il nous semble alors opportun de mettre un peu de complexité additionnelle dans les capteurs afin qu'ils puissent fonctionner avec un niveau d'interférence et de bruit plus élevés, offrant deux avantages : la possibilité de bien recevoir dans plus de bruit mais aussi, en conséquence, celle de transmettre moins fort et donc de réduire les interférences. Dans ce cadre, il est important de bien modéliser l'interférence. De nombreux travaux ont montré que l'hypothèse gaussienne habituelle n'est pas adaptée. Elle ne représente pas bien le caractère impulsif de l'interférence. D'autres modèles doivent donc être adoptés. L'étude de tels modèles est importante pour deux raisons majeures. Tout d'abord, d'un point de vue fondamental, elle apporte une meilleure compréhension des phénomènes. Ensuite, du point de vue pratique, elle oriente la conception des systèmes de transmission ainsi que le choix des stratégies de communication.

Notre travail dans ce chapitre s'est axé sur l'étude et le choix d'un modèle permettant une représentation précise et fiable des phénomènes impulsifs qui apparaissent dans les systèmes de transmission. Il se devise en trois grandes parties. La première partie présente les différentes lois proposées dans la littérature pour modéliser les phénomènes qui apparaissent dans les systèmes de transmission. La deuxième partie introduit les distributions α-stable en présentant leur définition, leurs propriétés ainsi que leurs intérêts pour modéliser des processus impulsifs. Elle propose également différentes études pour la caractérisation des processus α-stables et, en particulier, discute le problème de variance infinie. La dernière partie décrit le modèle de canal de transmission utilisé dans la suite de la thèse en calculant sa capacité.

2 Modèles des environnements non-gaussien

Les méthodes de traitement du signal utilisées dans les communications ont été historiquement dominées par l'hypothèse que les processus et les phénomènes qui apparaissent dans ces systèmes de transmission sont gaussiens. Cette hypothèse permet généralement d'obtenir des solutions analytiques compactes et rapides mais elle est restrictive car elle ne prend pas en compte la grande variabilité des données. De ce fait, de nombreux travaux se sont intéressés à l'étude de ces phénomènes afin de trouver d'autres modèles mieux adaptés. Différentes distributions ont été proposées dans la littérature et nous en présentons quelques-uns dans la suite.

2.1 Le modèle gaussien

En statistique, une des lois les plus utilisées est la loi gaussienne ou normale [4], [5]. Elle se caractérise par deux paramètres : la moyenne μ et la variance σ^2. Un théorème important, justifiant bien souvent la bonne adéquation de la gaussienne, est *le théorème central limite* [6], [7] : toute somme de variables aléatoires indépendantes et identiquement distribuées (i.i.d) tend vers une variable aléatoire gaussienne. Le théorème 1 est la version exacte du théorème. Il est important de noter qu'un élément important manque dans la version raccourcie que nous venons de faire : la variance des variables additionnées doit être finie.

Théorème 1 (théorème central limite)

Soit X_1, X_2, … une suite de variables aléatoires indépendamment et identiquement distribuées (i.i.d) de moyenne μ et de variance finie σ^2. Si $N \to \infty$ alors la somme normalisée

$$S_N = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{1.1}$$

Converge vers une variable gaussienne de moyenne μ et de variance σ^2/N.

Conceptuellement, le théorème central limite explique la nature gaussienne des processus qui sont générés par la superposition d'un très grand nombre de petites causes aléatoires indépendantes et qui présentent des distributions de probabilités identiques. C'est le cas par exemple du bruit thermique, qui est généré par la superposition d'un grand nombre d'interactions aléatoires indépendantes au niveau moléculaire.

Si X est une variable aléatoire réelle qui suit une loi d'espérance μ et de variance σ^2, elle est dite gaussienne (notée par $\mathcal{N}(\mu,\sigma^2)$) si et seulement si elle possède une densité de probabilité $f(x)$ définie, pour tout nombre réel x, par :

$$\mathcal{N}(x \mid \mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma^2}\right)^2} \tag{1.2}$$

Les variables aléatoires gaussiennes ont été très largement étudiées dans la littérature [8]

Des exemples de densités de probabilités gaussiennes en fonction des paramètres μ et σ^2 sont représentés figure 1.1 :

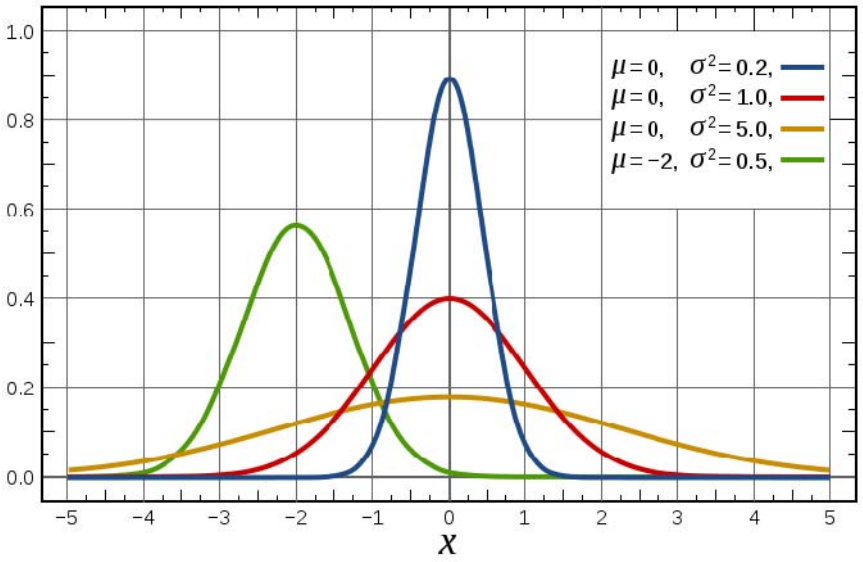


Fig.1.1 : Densités de probabilités d'une distribution normale pour différents μ et σ^2

Cette loi apparait comme une loi universelle car s'adaptant à de nombreuses situations. Parmi ces situations, nous trouvons le domaine de théorie du signal où de très nombreux algorithmes supposent que les signaux étudiés obéissent à une loi gaussienne ou s'ajoutent à un bruit gaussien. À titre d'exemples, le signal du fouillis radar ou le bruit thermique dans les circuits électroniques sont de nature gaussienne. Cette hypothèse permet généralement d'obtenir des solutions analytiques compactes et rapides mais elle est restrictive. Une limitation est en particulier qu'elle a du mal à prendre en compte une grande variabilité dans les données. Or sur le plan pratique dans le domaine des télécommunications, de nombreux phénomènes rencontrés présentent cette importante variabilité. Citons par exemple les bruits rencontrés lors de transmissions sur le réseau électrique [9], les lignes d'abonnés numériques [10], les communications HF, les systèmes de transmission mobile (UMTS) [11], les systèmes UWB [12], les interférences dans les réseaux *ad hoc* [13], les interférences dans des systèmes multi-utilisateurs sans fils [14], [15], etc. L'hypothèse classique sur la nature gaussienne du bruit, justifiée par le biais du théorème centrale limite, n'est plus valable. Ceci nous conduit à rechercher d'autres lois mieux adaptées aux phénomènes impulsifs.

2.2 Modèles non gaussiens

Dans de nombreuses applications du traitement du signal et en particulier les télécommunications, nous rencontrons des phénomènes à faible probabilité d'apparition mais à fortes amplitudes dites de nature impulsive. Cette constatation a motivé depuis un certain temps la recherche dans ce domaine afin de trouver des distributions mieux adaptées. Différents modèles ont été proposés et nous en présentons quelques-uns dans la suite.

2.2.1 Modèle mélange de gaussiennes

Le mélange de gaussiennes [15], [16] est un modèle statistique représenté par une densité de probabilité vu comme une somme de plusieurs gaussiennes pondérées. Nous pouvons l'écrire comme suit :

$$f(x) = \sum_{n=1}^{N} \alpha_n \mathcal{N}(x \mid \mu_n, \sigma_n) \qquad (1.3)$$

Avec $\mathcal{N}(x| \mu_n, \sigma_n)$ est une densité gaussienne multi-variée décrite par un vecteur moyen μ_n, une matrice de covariance σ_n et la pondération est donnée par le coefficient α_n. Cette densité représente l'avantage d'une relative simplicité mathématique car les densités des gaussiennes ont une forme analytique bien connue.

La figure suivante présente la construction de cette densité de probabilité à partir de quatre différentes densités de probabilité gaussiennes:

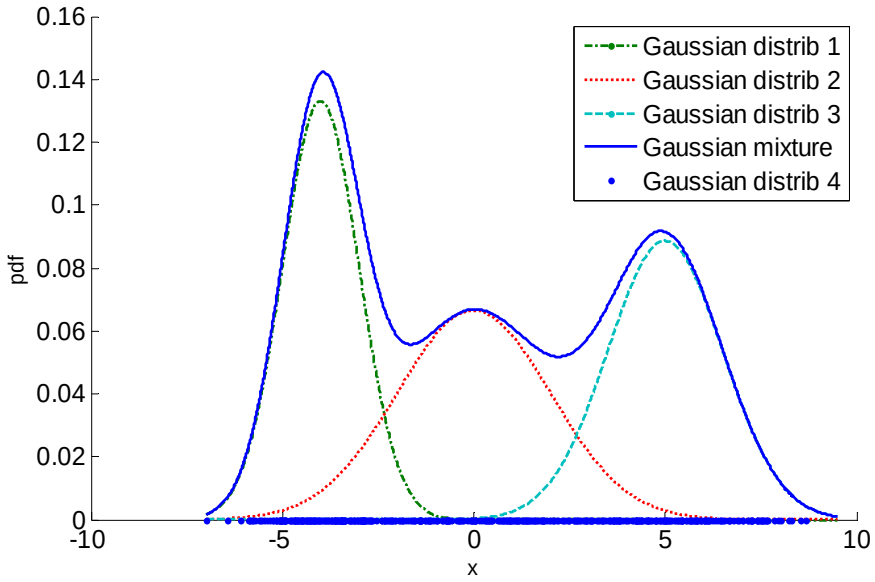


Fig.1.2 : Densité de probabilité d'une distribution de mélange de gaussienne

Cette distribution a été utilisée afin de modéliser les processus impulsifs qui apparaissent dans les systèmes de transmission comme les bruits impulsifs dus aux multi trajets dans les systèmes de transmission par satellite [17], les interférences multi-utilisateurs (MAI) qui apparaissent dans les systèmes UWB [18]. Ce modèle présente cependant certains inconvénients. En particulier il nécessite l'estimation des paramètres μ_n, σ_n, et α_n, par exemple par un processus itératif nommé par Expectation Maximisation [19], qui peut toutefois s'avérer complexe.

Cette approche n'est qu'une approximation de la nature impulsive du bruit. Elle permet de rehausser les queues en plaçant des distributions avec des poids plus faibles assez éloignées de la moyenne mais le résultat n'est pas pour autant une distribution à queue lourde.

2.3 Modèles à queues lourdes

Dans cette partie nous présentons la notion de distribution à queue lourde et les différentes classes correspondantes à ce type de distributions. Ces distributions, liées à la théorie des valeurs extrêmes [20], permet de modéliser plusieurs phénomènes rencontrés dans différentes disciplines : finances, hydrologie, télécommunication, géologie… Plusieurs définitions ont été associées à ce type de distributions en fonction d'un critère de classification. La caractérisation la plus simple et celle basée

sur la comparaison avec la loi normale [21]. Nous disons, alors, que la distribution d'une v.a. X est à queue lourde si :

$$C_K = E\left[\frac{(X-\mu_X)^4}{\sigma_X}\right] > 3 \qquad (1.4)$$

où μ_x et σ_x^2 sont la moyenne et la variance respectivement de X.

Ce qui est équivalent à dire qu'une distribution a une queue lourde si et seulement si son coefficient d'aplatissement, C_k, est supérieur à celui de la loi normale (pour laquelle $C_k = 3$). La différence entre la loi normale et une loi à une queue lourde a été illustrée par Hubert (1996) [22].

La caractérisation, donnée par l'équation (1.4), est très générale et ne peut être appliquée que si les moments d'ordre inférieurs ou égaux à 4 existent. Par conséquent aucune discrimination, pour les distributions ayant un moment d'ordre 4 infini, ne peut être faite si nous ne considérons que ce critère. Or dans le contexte de nos travaux différents modèles vérifiant cette hypothèse de moments infinis seront présentés par la suite.

2.3.1 Modèle de Laplace

La distribution de Laplace [23] est une densité de probabilité continue connue sous le nom de loi *double exponentielle*. Sa densité est en fait l'association de deux lois exponentielles accolées dos à dos. Elle est aussi le résultat de la différence de deux variables exponentielles indépendantes [24]. Sa densité de probabilité est la suivante :

$$f(x \mid \mu, b) = \frac{1}{2b} e^{\left(-\frac{|x-\mu|}{b}\right)} \qquad (1.5)$$

Le réel μ est un paramètre de location et b > 0 un paramètre d'échelle. Quelques exemples de représentations graphiques sont donnés sur la figure suivante :

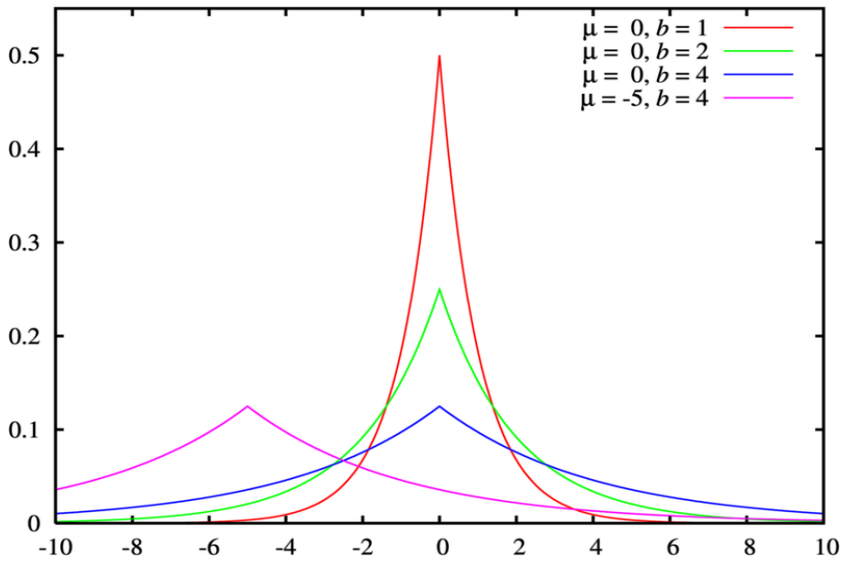


Fig.1.3 : Densités de probabilités d'une distribution de Laplace pour différents μ et b

Comme l'indique cette figure, la nature de cette distribution dépend essentiellement de la valeur du paramètre d'échelle b, plus sa valeur est petite plus la distribution est impulsive. Cette loi est à queues lourdes car elle présente des queues plus épaisses que la loi normale.

Ce type de distribution a été utilisé afin de modéliser les phénomènes impulsifs qui apparaissent dans les systèmes de transmission comme, par exemple, les interférences dans les systèmes UWB [25], ou encore le bruit intervenant dans les systèmes de détection [26]. Ce modèle de distribution peut être vu comme exemple de distributions à queues lourdes. Mais la forme pointue de sa densité de probabilité questionne sur sa validité en pratique. En effet, le *principe de Winsor* indique que, en pratique, la distribution doit être "normale au milieu" [27].

2.3.2 Modèle de gaussienne généralisé

La distribution gaussienne généralisée (GGD) [28] est un modèle typique qui généralise les distributions gaussiennes et de Laplace selon la valeur de son paramètre nommé exposant caractéristique p. Cette distribution est décrite par la densité de probabilité suivante :

$$f(x \mid \mu, \sigma, p) = \frac{p}{2\sigma\Gamma\left(\frac{1}{p}\right)} . \mathrm{e}^{-\left(\left|\frac{(x-\mu)}{\sigma}\right|^{p}\right)} \tag{1.6}$$

Γ (.) est la fonction gamma [29] donnée par :

$$\Gamma(x) = \int_{0}^{\infty} t^{x-1} . \mathrm{e}^{-t}\, dt \qquad x > 0 \tag{1.7}$$

Cette distribution est décrite par trois paramètres : la moyenne μ, le facteur d'échelle σ et le facteur de forme ou l'exposant caractéristique p qui est une mesure du largueur de pic de la distribution. A partir de la forme mathématique simple de sa densité de probabilité, nous pouvons généraliser différents types de distributions : gaussienne (p = 2), Laplace (p = 1) et loi uniforme (p → ∞). Ceci est présenté sur la figure 1.4 :

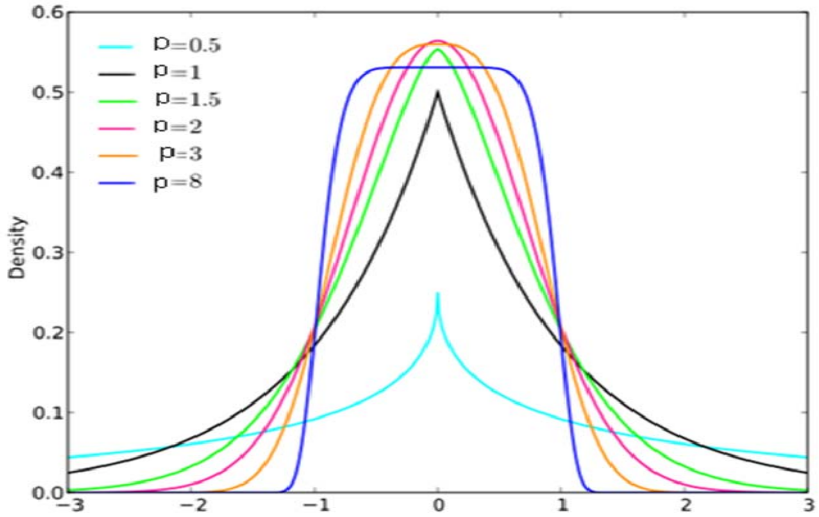


Fig.1.4 : Densités de probabilités d'une distribution gaussienne généralisée pour différents p

Comme le montre cette figure, la nature de cette distribution dépend essentiellement de la valeur de l'exposant caractéristique p. Plus sa valeur est petite plus la queue de la distribution devient importante.

Cette distribution a été utilisée dans plusieurs domaines comme la modélisation de bruit atmosphérique, le codage en sous bande des signaux audio et vidéo [30], [31], la modélisation des interférences dues aux accès multiples (MAI) dans les systèmes UWB[32], IR-UWB [33], dans les réseaux *ad hoc* [34], dans les systèmes multi-utilisateurs [35].

Malgré l'intérêt de cette loi dans plusieurs applications pratiques, elle présente quelques limitations :

- Quand p < 1, cette distribution hérite la forme pointue ou *peaky shape* pour la distribution de Laplace qui se traduit par une incapacité à modéliser ces phénomènes physiques en pratiques.

- Les queues des distributions gaussiennes généralisées se distinguent par une décroissance exponentielle, en contraste avec la décroissance algébrique fréquemment observés dans les processus à queues assez lourdes qui apparaissent dans la pratique [36], [37], [38].

Ceci nous amène à étudier d'autres types de distributions de types queue lourdes, ou à queues algébriques, afin de modéliser plusieurs phénomènes et processus qui apparaissent dans plusieurs disciplines comme la finance, la géologie ou les systèmes de communication sans fil, etc.

2.4 Modèles à queues algébriques

Une variable aléatoire X est dite à queues algébriques [39] s'il vérifie cette relation

$$\Pr(|X| > x) \sim \theta(x^{-\alpha}) \qquad pour \quad x \to \infty \tag{1.8}$$

Le paramètre α, nommé indice de la queue, détermine le niveau d'impulsivité de la distribution. Plus la valeur de α est petite, plus la densité est impulsive et plus sa queue s'alourdit.

Ces processus à queues algébriques possèdent des queues plus lourdes que les processus gaussiens et gaussiens généralisés. En effet les queues de ces processus sont à décroissance exponentielle [39].

Nous présentons maintenant trois grandes familles de distributions à queues algébriques qui jouent un rôle significatif dans la modélisation des processus impulsifs qui apparaissent dans les systèmes de transmission.

2.4.1 Modèle de Pareto

Une variable aléatoire X de paramètres α et x_0, suit une loi de Pareto [40], [41], sa fonction de répartition s'écrit :

$$F(x) = \begin{cases} 0 & si \quad x < x_0 \\ 1 - (\frac{x_0}{x})^{\alpha} & si \quad x > x_0 \end{cases} \tag{1.9}$$

Avec x_0 est la valeur minimale (positive) que peut prendre X et k est un réel positif souvent nommé indice de Pareto.

Dans ce cas la fonction de densité de probabilité de cette *v.a* X est donnée par :

$$f(x) = \frac{\alpha}{x_0}\left(\frac{x_0}{x}\right)^{\alpha+1} \qquad x > x_0 \tag{1.10}$$

Les distributions de Pareto sont des distributions continues [42] à queues algébriques [38].

2.4.2 Modèle *t* de student

La famille de distributions *t* de student a été introduite par Hall en 1966 comme modèle empirique pour le bruit atmosphérique en communication radio [43]. Cette distribution est décrite par la densité de probabilité donnée par l'équation suivante :

$$f(x/\alpha,\sigma^2) = \frac{c}{\left(\alpha\sigma^2 + x^2\right)^{\frac{1+\alpha}{2}}} \tag{1.11}$$

où α est l'exposant caractéristique (α > 0), σ représente la dispersion de la distribution (σ > 0) et la constante de normalisation qui est donnée par l'équation :

$$c = \frac{\Gamma\left(\frac{1+\alpha}{2}\right)}{\sqrt{\pi}\Gamma\left(\frac{\alpha}{2}\right)}\alpha^{\frac{\alpha}{2}}\sigma^{\alpha} \tag{1.12}$$

Cette distribution présente une relative simplicité mathématique de sa densité de probabilité. Elle permet également de généraliser différentes distributions selon la valeur de son exposant caractéristique α :

- Quand α tend vers +∞ cette distribution devient une distribution gaussienne de variance σ² ;
- Plus la valeur de α diminue plus l'impulsivité de la distribution augmente ;
- Quand α tend vers 0 la distribution devient très impulsive ;
- Quand α =1, le modèle correspond à la distribution de Cauchy.

2.4.3 Modèle α-stable

Les distributions α-stables, appelées aussi distributions stables, ont été introduites par Lévy en 1925 [44]. Elles constituent une classe très riche de lois de probabilité capables de représenter différentes asymétries et des queues très lourdes. Ces lois sont utilisées dans de nombreux domaines tels que les télécommunications [46], la finance [45], l'informatique [47], le biomédical… Plusieurs livres sont consacrés à ces lois : Zolotarev [48] a étudié les lois α-stables dans le contexte uni-varié.

Samorodnitsky et Taqqu [39] ont étudié de manière approfondie plusieurs propriétés de ces lois dans le cas uni-varié comme dans le cas multi-varié. Nikias et Shao [37] ont appliqué ces lois dans le domaine du traitement du signal, comme la détection et la classification de signaux, la mise au point de récepteurs optimaux et sous-optimaux en présence de signaux impulsifs et l'estimation des paramètres d'une distribution α-stable.

Nous avons opté dans la suite de ce manuscrit pour les distributions α-stables pour essentiellement deux raisons :

- la première est la propriété de stabilité. Si la distribution gaussienne s'avère être un excellent modèle dans de nombreuses applications, une explication en est le théorème de la limite centrale. Les distributions α-stables constituent l'ensemble des lois qui vérifient ce théorème mais dans une version généralisée, la différence provenant de la variance (finie ou infinie) des variables sommées. L'extension du théorème aux variables à variance infinie et donc au cadre plus général des processus α-stables permet en particulier une meilleure représentation des variations brusques d'un système ou des évènements rares ;

- la seconde est l'existence de justifications théoriques à la bonne adéquation de ce modèle. Les récents travaux de Win *et al.* sont une bonne illustration de ces justifications mais il en existe de nombreuses dans la littérature (Sousa, Illow, Nikias, ElGhannudi, …).

Notre modèle concerne l'interférence d'accès multiples (MAI) dans les réseaux, et plus spécifiquement les réseaux *ad hoc*. Dans un réseau *ad hoc* dense, cette MAI peut devenir la composante principale du bruit. Souvent considérée dans d'autres contextes comme la somme de nombreuses variables aléatoires indépendantes et identiquement distribuées, l'approximation gaussienne est utilisée pour la modéliser, et ceci avec succès. Cependant, la configuration *ad hoc* modifie ses caractéristiques. L'amplitude des impulsions reçues varie de façon importante et l'impact du MAI est fortement conditionné par la présence ou non d'interférents forts. Sousa [49] a montré qu'une variable aléatoire α-stable en est alors un bon modèle. Ce même résultat a été démontré dans le cas d'un *shadowing lognormal* et d'évanouissements rapides de Rayleigh [50]. Pinto [51] quant à lui calcule l'expression exacte du taux d'erreurs pour une communication à bande étroite sujette à des interférents Ultra Large Bande (UWB) et du bruit additif gaussien. Un cadre où les interférents UWB sont spatialement organisés suivant un champ de Poisson infini et communiquent de façon asynchrone est proposé. Ce cadre résulte en un MAI α-stable symétrique à la sortie du récepteur bande étroite et il est largement développé par Win *et al.* dans [52]. El Ghannudi *et al.* [53] appliquent également ce modèle dans le cadre d'un réseau *ad hoc* basé sur des transmissions d'impulsions sub-nanosecondes transposées à 60 GHz.

Pour donner une idée de la preuve mathématique aboutissant au modèle α-stable, considérons, sous une forme assez générique, l'expression du MAI :

$$Z = \sum_{i=1}^{k} a_i \psi_i \tag{1.13}$$

où a_i est l'atténuation de l'utilisateur interférent i (représentant donc l'impact du canal radio) et Ψ_i est lié à la couche physique employé. La variable k est le nombre d'interférents. Les $(\Psi_i)_{i=1,\ldots,k}$ sont des variables aléatoires indépendantes, identiquement distribuées et bornées. Leurs densités de probabilité

sont paires. Les $(a_i)_{i=1,\ldots,k}$ sont des variables aléatoires positives, indépendantes, identiquement distribuées. Leurs propriétés déterminent la nature de Z. Supposons que nous soyons placés dans un trafic dense : k est très grand. Comme les $(\Psi_i)_{i=1,\ldots,k}$ ont une variance finie, il en est de même pour les produits $(a_i\ \Psi_i)_{i=1,\ldots,k}$ et il s'ensuit, d'après le théorème de la limite centrale, que la somme normalisée :

$$Z^{(k)} = \frac{1}{\sqrt{k}} \sum_{i=1}^{k} a_i \psi_i \tag{1.14}$$

converge en loi vers une distribution normale de variance var $(a_i.\Psi_i)$. Cela signifie que $Z^{(k)}$ est asymptotiquement gaussienne et, en conséquence, qu'elle peut être approchée par une distribution normale quand k est suffisamment grand.

Cependant, dans un cadre général, le nombre d'interférents n'est pas toujours très grand et il est plus précis de le modéliser par une variable aléatoire. Comme habituellement, dans des situations similaires, [49], [51], nous modélisons k par une distribution de Poisson. La clef du raisonnement est de ne plus considérer a_i comme une variable de variance finie. Si ceci est un peu contraire à notre intuition première, la puissance de l'interférence est en fait à mettre en relation avec la puissance du lien utile a_0, où l'indice 0 fait référence au lien utile. Dans le cas où ce lien est bien plus long que le lien interférent, la valeur relative observée de a_i/a_0 peut être très importante. Ces évènements, bien que peu fréquents, ont un impact important sur la communication et donne une nature impulsive au bruit. Une modélisation par des moments d'ordre 2 infinis permet alors une nouvelle représentation du phénomène. Cela revient en fait à négliger le champ proche mais cela a peu d'impact sur la validité du modèle. Cette hypothèse s'avère dans la plupart des cas avoir un impact négligeable.

Cette variance infinie rend obsolète le modèle gaussien. En utilisant la théorie des processus de Poisson composés, nous pouvons montrer que la variable aléatoire Z est dans le domaine d'attraction d'une distribution α-stable. Par conséquent elle peut être modélisée par une loi α-stable. Pour le démontrer, il faut étudier la fonction caractéristique du MAI et montrer qu'elle peut se mettre sous la forme :

$$\varphi_Z(\omega) = E\left[e^{j\omega Z}\right] = \exp\left(-\left|\delta\omega\right|^{\alpha}\right) \tag{1.15}$$

La distribution du MAI est alors une distribution α-stable symétrique autour de zéro avec un exposant caractéristique α qui dépend du canal radio et une dispersion qui dépend des paramètres de la couche physique.

3 Lois α-stables

La propriété de stabilité, le théorème de la limite centrale et la caractérisation parfaite par des moments d'ordre un (la moyenne) et d'ordre deux (la variance ou la covariance) sont des propriétés qui font de la loi gaussienne une des lois les plus utilisées en modélisation statistique. Cependant, bien que les calculs d'inférence statistique soient maintenant bien maîtrisés, l'hypothèse gaussienne s'avère parfois trop restrictive, en particulier dans certains domaines pour lesquels il faut prendre en compte une plus grande variabilité des données. Il semble alors assez naturel de s'orienter vers les lois α-stables. Si leur moment d'ordre 2 est infini dès que α est strictement inférieur à 2, elles sont une famille de distributions paramétriques, à laquelle appartient la gaussienne, qui vérifient la propriété de stabilité.

Le seul fait que les lois stables ont une queue de type lourde ou bien asymptotiquement parétienne ne suffit pas pour justifier leur importance. Il existe deux raisons profondes : la première provient d'un théorème que nous détaillerons dans cette partie dit *théorème central limite généralisé* et qui accorde bien le statut de *lois limite* aux lois α-stables. La deuxième raison provient de la propriété de stabilité qui affirme que toute combinaison linéaire de variables aléatoires réelles (v.a.r) α-stables est aussi de loi α-stable.

3.1 Définitions des lois α-stables

Définition ***(Propriété de Stabilité)*** [37], [39], [48]. Une v.a.r. X est dite à distribution stable ou elle suit une loi stable si pour tous nombres réels positifs A et B, il existe un nombre positif C et un nombre réel D tels que :

$$AX_1 + BX_2 \overset{d}{=} CX + D \tag{1.16}$$

où X_1 et X_2 sont des v.a.r. indépendantes et de même loi que X. Lorsque D vaut 0, on parle de la loi strictement stable.

La loi de la somme de deux variables aléatoires indépendantes est le produit de convolution des lois respectives de ces deux variables. Cette définition justifie donc la nomination « stable » car elle fait intervenir la stabilité par produit de convolution.

Théorème *(Théorème Central Limite Généralisé)* [37], [39]***:***

Sans limitation de variance finie, pour toute suite de variables aléatoires i.i.d. $X_1,\ldots,X_n$, une suite a_n de nombre réels positifs et une suite b_n nombres réels, la somme normalisée

$$\frac{(X_1 + \ldots.. + X_n)}{a_n} + b_n \tag{1.17}$$

converge en distribution vers une variable stable.

La démonstration de ce théorème se trouve dans la référence [55] (page 338).

Ces dernières définitions introduisent les variables α-stables mais ne donnent aucune précision sur la distribution de ces variables. La définition suivante, équivalente aux précédentes, donne leur fonction caractéristique.

Définition : *Fonction caractéristique des lois stables (**Levy-Khinchin**) :*

Soit X une v.a modélisée par une distribution α-stable [37], [39] alors sa fonction caractéristique notée par ϕ(t) pour tout t dans R s'écrit:

$$\Phi(t) = \exp(i\mu t - |\gamma t|^{\alpha}\,[1 + j\beta sign(t) w(t,\alpha)]) \tag{1.18}$$

$$w(t,\alpha)=\begin{cases}\tan\dfrac{\alpha\pi}{2} & ,\text{si} \quad \alpha\neq 1\\ 2/\pi\log|t| & ,\text{si} \quad \alpha=1\end{cases} \tag{1.19}$$

$$\operatorname{sgn}(t)=\begin{cases}1 & \text{si} \quad t>0\\ 0 & \text{si} \quad t=0\\ -1 & \text{si} \quad t<0\end{cases} \tag{1.20}$$

Les quatre paramètres α, β, γ, et μ caractérisent d'une manière unique les distributions stables, ce qui signifie que les lois stables sont paramétriques. Pour un praticien de ce domaine, il est important de connaitre la signification statistique de chaque paramètre ainsi que son influence sur l'allure de la courbe de la densité de probabilité ou de la fonction de répartition.

- Le paramètre α est appelé exposant caractéristique ou indice de stabilité, $0 < \alpha \leq 2$. Il caractérise l'épaisseur de la queue de la distribution. Quand α est proche de 2, la probabilité d'observer des valeurs de la variable aléatoire loin de la position centrale est faible. Une valeur proche de 0 signifie au contraire que la masse de la queue est importante et donc les valeurs éloignées de la position centrale plus probables. L'impact de la valeur de α est illustré sur les figures 1.6 et 1.7.

- Le paramètre μ réel est appelé paramètre de position, de location ou de centrage. Il mesure la tendance centrale de la distribution. Pour $\alpha > 1$, μ représente la moyenne et si $0 < \alpha < 1$, alors μ représente la médiane.

- Le paramètre γ est appelé paramètre de dispersion (ou d'échelle), $\gamma > 0$. Il mesure la dispersion de la distribution autour du paramètre de position μ. Ce paramètre correspond à la moitié de la variance dans le cas où $\alpha = 2$.

- Le paramètre β est appelé paramètre de symétrie, $\beta \in [-1, 1]$. Il mesure la dissymétrie de la densité de probabilité. Si $\beta = -1$, la distribution est totalement asymétrique à gauche. Si $\beta = 1$, la distribution est totalement asymétrique à droite. Pour $\beta = 0$, la distribution est symétrique par rapport au paramètre μ (figure 1.9).

Par convention, $S_\alpha(\mu, \beta, \gamma)$ représente une loi α-stable de paramètres α, μ, β et γ.

Une distribution α-stable est dite *symétrique α-stable* (*SαS*) lorsque $\beta = 0$. Dans ce cas, la fonction caractéristique définie par l'équation (1.18) peut être simplifiée et devient :

$$\Phi(t) = \exp(i\mu t - |\gamma t|^{\alpha}) \tag{1.21}$$

Une distribution α-stable est dite symétrique en zéro ($\mu = 0$) quand elle a pour fonction caractéristique :

$$\Phi_{\alpha}(t)=\exp(-|\gamma t|^{\alpha}) \tag{1.22}$$

Une *v.a* X de paramètres α, γ et de paramètre de position μ peut avoir la combinaison $X = \mu + \gamma.S_\alpha$ où S_α est de fonction caractéristique $\exp(-|t|^{\alpha})$.

La principale propriété de la loi SαS est la stabilité. Grace à la forme de la fonction caractéristique $\Phi_\alpha(t)$ (1.22), il est simple de montrer que si des *v.a* sont indépendantes alors :

$$aS_\alpha + bS_\alpha + c = \sqrt[\alpha]{a^\alpha + b^\alpha} S_\alpha + c \tag{1.23}$$

Il est possible aussi d'étendre le résultat à un nombre plus important de *v.a i.i.d* :

$$\sum_i a_i S_\alpha = \left(\sum_i a_i\right)^{1/\alpha} S_\alpha \tag{1.24}$$

Pour illustrer la différence des lois α-stables avec la loi de Gauss, nous présentons sur la figure suivante des réalisations des variables aléatoires i.i.d. SαS d'exposants caractéristiques α = 1.5, 1.9 et une réalisation gaussienne.

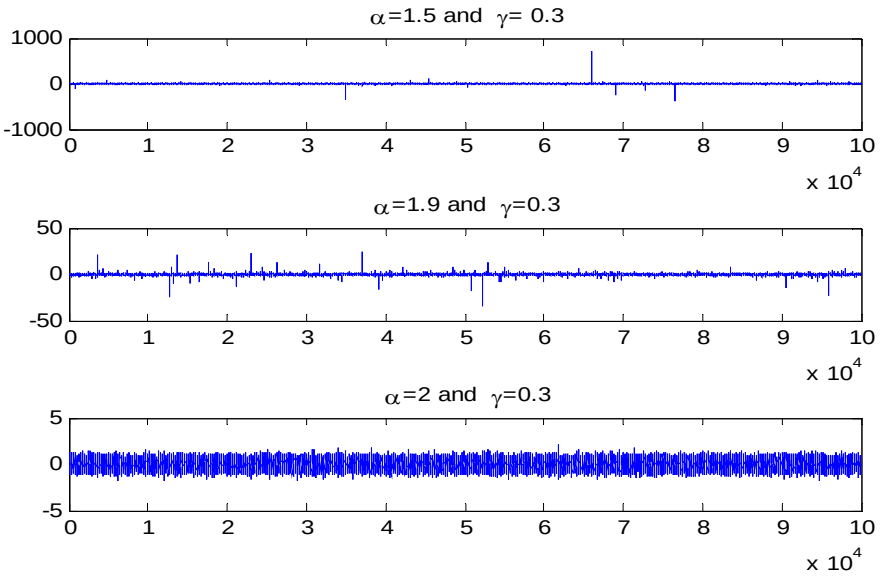


Fig.1.5 : Réalisation de signaux SαS pour différents valeurs d'alpha.

Nous remarquons que plus α est petit, plus la *v.a* SαS est impulsive d'amplitudes importantes avec des faibles probabilités d'apparition. L'environnement dans ce cas est dit de nature impulsive.

3.2 Densité de probabilité

Pour les variables aléatoires α-stables, il n'existe pas d'expression explicite de la densité de probabilité dans le cas général. Cependant nous pouvons obtenir une expression sous la forme de la transformée de Fourier inverse de la fonction caractéristique $\Phi_\alpha(t)$:

$$f(x;\alpha,\beta,\gamma) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \exp(-itx)\Phi_\alpha(t)dt \tag{1.25}$$

Quand la distribution est symétrique (β = 0) autour de zéro (μ = 0) (SαS en zéro), la fonction caractéristique est réelle et paire ce qui permet de simplifier l'expression de la de la densité de probabilité :

$$f(x;\alpha,\gamma)=\frac{1}{\pi}\int_0^{+\infty}\exp(-\left|\gamma t\right|^{\alpha})\cos(xt)dt \qquad (1.26)$$

Cette densité de probabilité a les propriétés suivantes :

- elle dépend que de α et γ ;
- elle vérifie cette l'égalité : *f (x; α, β, γ) = f(-x; α,-β, γ)* ;
- c'est une fonction bornée ;
- elle est de classe infinie : C_∞.

La démonstration de ces propriétés se trouve dans la référence [48].

A ce jour l'évaluation explicite de l'intégrale (1.22) n'est possible que dans les trois cas particuliers suivants :

- La loi *normale* ou de *Gauss* S_2 (0, $\sigma^2/2$, μ) dont la densité de probabilité est donnée par l'équation (1.27):

$$f_2(x;\gamma)=\frac{1}{\sqrt{4\Pi\gamma^2}}\exp(-\frac{(x-\mu)^2}{4\gamma^2})=\frac{1}{\sqrt{2\Pi\sigma^2}}\exp(-\frac{(x-\mu)^2}{2\sigma^2}) \qquad (1.27)$$

- La loi de *Cauchy* (α = 1) S_1 (0, γ, μ) dont la densité de probabilité est donnée par l'équation (1.28):

$$f_1(x;\gamma)=\frac{\gamma}{\pi(\gamma^2+(x-m)^2)} \qquad (1.28)$$

- La loi de *Lévy* $S_{1/2}$ (0, 1, μ) avec les paramètres α = 1/2 et β = (+1) dont la densité de probabilité est donnée par cette équation :

$$f_{1/2}(x;0,\gamma)=\frac{\gamma}{\sqrt{2\Pi}}.\frac{1}{(x-m)^{3/2}}\exp\left\{\frac{\gamma^2}{2(x-m)}\right\} \qquad (1.29)$$

Pour une meilleure illustration des densités α-stables, nous présentons dans les figures 1.6 et 1.7 le comportement des densités de probabilité quand la valeur de α varie. Nous remarquons que plus α est petit, plus la queue de la densité devient lourde, ce qui signifie un comportement plus impulsif de la variable aléatoire.

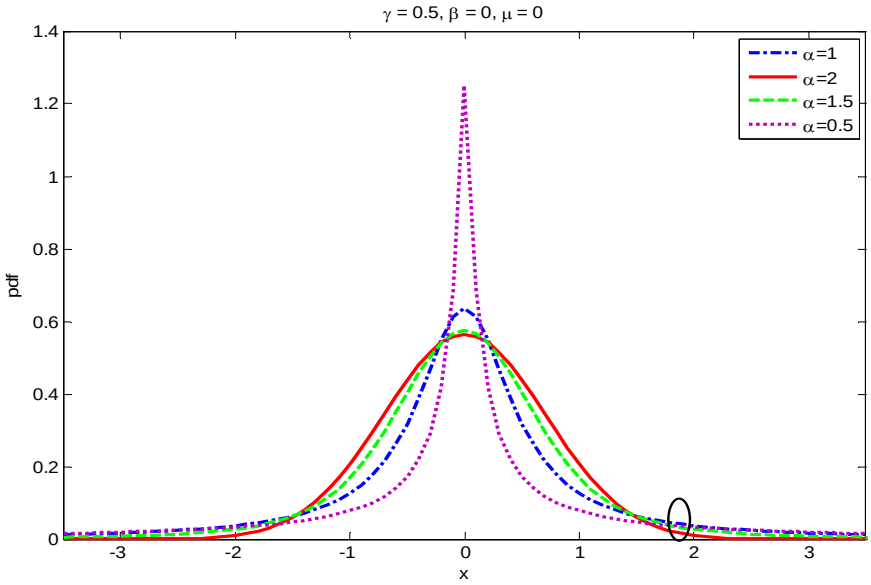


Fig.1.6: Densités de probabilité d'une distribution SαS pour différents α

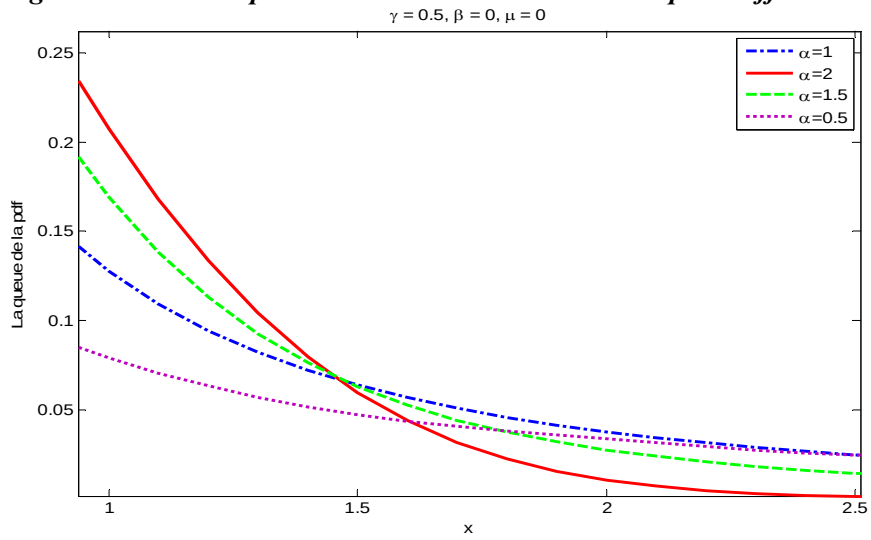


Fig.1.7 : Effet de α sur la queue de la densité de probabilité de la distribution SαS

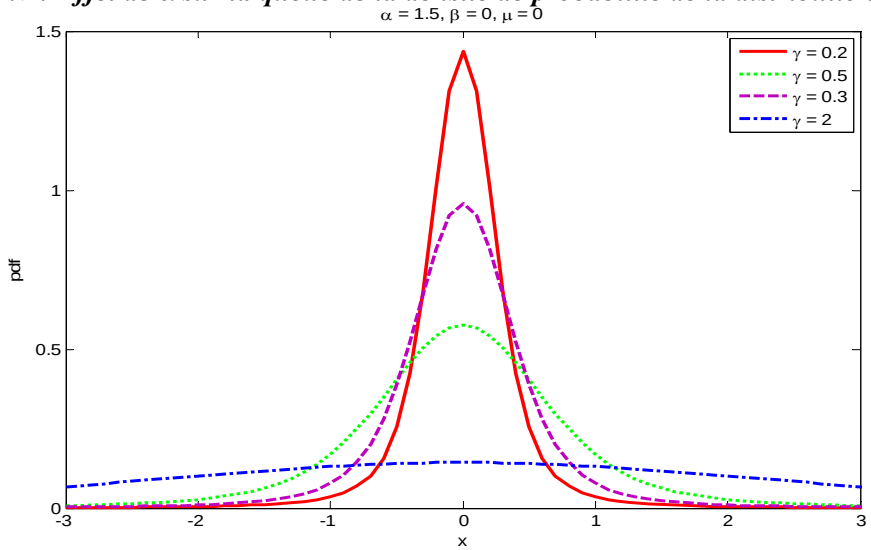


Fig.1.8 : Effet de γ sur la densité de probabilité de la distribution SαS

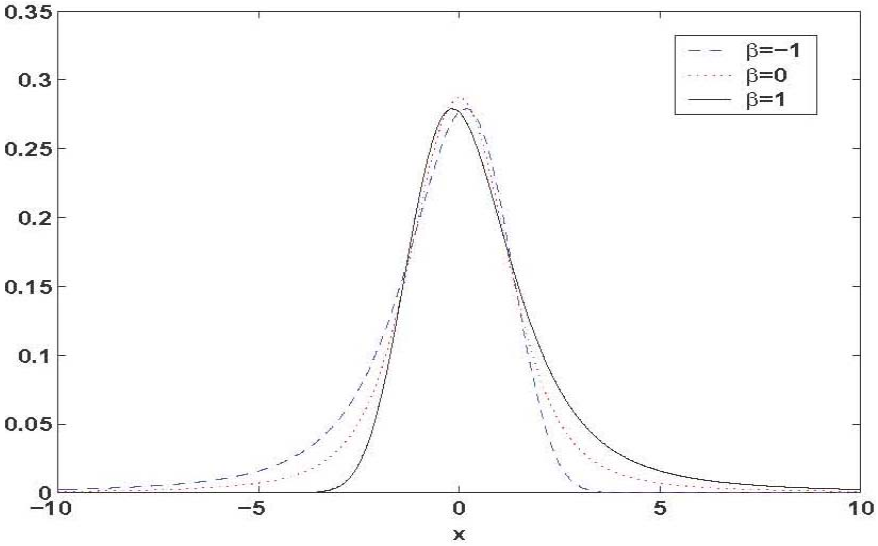


Fig.1.9 : Effet du paramètre de symétrie sur la densité de probabilité de la distribution SαS

Nous observons ensuite sur la figure 1.8 l'impact de la dispersion : l'augmentation de la valeur de γ augmente la variabilité autour de la valeur centrale. Sur la figure 1.9 nous visualisons l'impact du paramètre de symétrie. Une valeur négative de β augmente l'importance de la queue pour les valeurs négatives tandis qu'une valeur positive renforce la partie à droite de la valeur centrale.

Dans les applications en télécommunication, l'une des difficultés apportée par les distributions α-stables est leur variance infinie dès que α est strictement inférieur à 2. La conséquence est que, si nous utilisons ces distributions pour modéliser l'interférence, le rapport signal sur bruit tel qu'il est habituellement défini, sera nul. Différentes solutions ont été proposées afin de pallier ce problème. Ces différentes études sont présentées dans ce qui suit.

4 Les moments des processus α-stables

4.1 Moments fractionnaires d'ordre positif

Même si les moments du second ordre d'une *v.a.* SαS avec $0 < \alpha < 2$ n'existent pas, les moments d'ordre inférieur à α existent et s'appellent les moments fractionnaires d'ordre positif [48], [55]. Nous excluons ici le cas $\alpha = 2$ qui est le cas gaussien bien connu. Dans les autres cas ($0 < \alpha < 2$), pour tout v.a. X SαS de paramètre de position nul et de dispersion γ, deux cas se présentent :

- $E|X|^p = +\infty$ pour $p \geq \alpha$
- $E|X|^p < +\infty$ pour $0 < p < \alpha$, et, plus précisément :

$$E|X|^P = C(\alpha,\beta,p)\gamma^{P/\alpha} \qquad 0 < p < \alpha \tag{1.30}$$

$$C(\alpha,\beta,p) = \frac{2^{p-1}\Gamma(\frac{p-1}{\alpha})}{p\int_0^{+\infty} u^{-p-1}\sin^2 u.du}(1+\beta^2\tan^2\frac{\alpha\pi}{2})^{P/2\alpha}\cos(\frac{p}{\alpha}\arctan(\beta\tan\frac{\alpha\pi}{2})) \tag{1.31}$$

Ce résultat a été démontré par Zolotarev dans la référence [48]. Ces résultats indiquent également que lorsque α est strictement inférieur à 2, la variance d'une loi α-stable est infinie. De plus, si α est strictement inférieur à 1, c'est la moyenne qui devient indéfinie.

4.2 Moments fractionnaires d'ordre négatif

Nikias [37] a démontré que les *v.a.r.* SαS ont aussi des moments finis d'ordre négatif. Ce résultat est présenté par la proposition suivante.

Proposition

Soit X une *v.a.r.* SαS de paramètre de position μ nul et de dispersion γ. Alors la formule unifiée pour ces moments d'ordre positif et d'ordre négatif est de la forme :

$$E\left|X\right|^{P} = C(p,\alpha)\gamma^{P/\alpha} \qquad -1 < p < \alpha \tag{1.32}$$

$$C(p,\alpha) = \frac{2^{p+1}\Gamma(\frac{p+1}{2})\Gamma\left(-p/\alpha\right)}{\alpha\sqrt{\pi}\Gamma\left(-p/2\right)}\gamma^{P/\alpha} \tag{1.33}$$

Ce résultat important a été démontré par Nikias dans la référence [37].

La figure suivante présente les moments d'ordre *p* pour différentes valeurs de α correspondant aux propriétés définies dans (1.32) et (1.33).

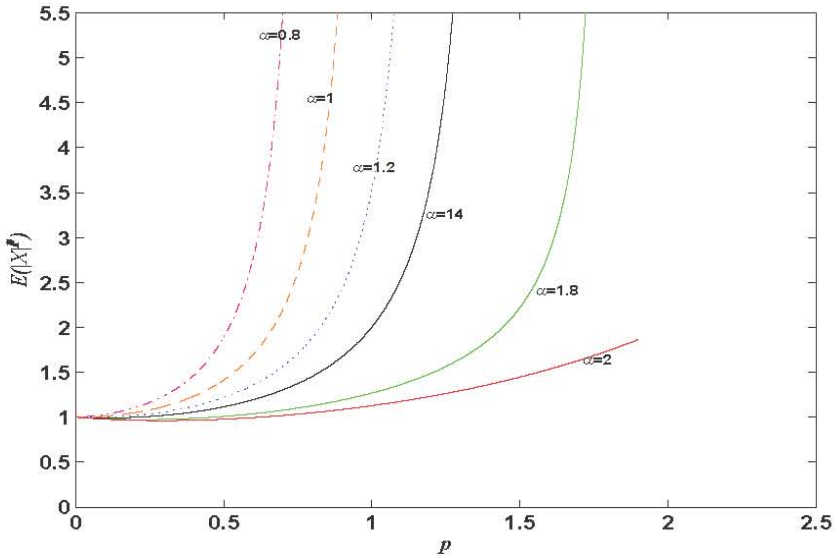


Fig.1.10 : Les moments de lois α-stables pour différentes valeurs de α

4.3 Limites des moments fractionnaires

Cette étude qui est basée sur des moments d'ordre p présente quelques limitations :

- ils n'offrent pas de cadre universel pour la caractérisation des processus à queues algébriques : quand $p \geq \alpha$, les moments deviennent infini et la caractérisation statistique n'est alors plus possible à cet ordre ;

- la restriction du choix de la valeur de p dans un intervalle $(0, \alpha)$ [55] exige une connaissance préalable du α.

Dans le cas où $\alpha < 2$, le $p^{ème}$ moment d'une v.a SαS n'est fini que pour des valeurs de p inférieurs à α. C'est la raison pour laquelle les paramètres γ et μ sont nommés paramètres d'échelle et de localisation et non variance et moyenne.

Afin de résoudre ces limitations et pour bien caractériser les processus impulsif indépendamment de la valeur de α, Gonzalez a proposé dans [56], [57] une autre approche basée sur les statistiques d'ordre zéro (ZOS) qui utilise les moments logarithmiques. Cette théorie permet d'introduire des nouveaux paramètres qui vont permettre de bien caractériser les processus impulsifs à variance infinie. Ces paramètres seront détaillés dans ce qui suit.

4.4 Moments d'ordre logarithmique

Une autre étude a été proposée afin de caractériser ces processus impulsifs à variance infinie. Elle est basée sur des classes de processus dites d'ordre logarithmique [48]. Dans ce cas les processus α-stable sont des processus d'ordre logarithmique comme le montre le théorème suivant :

Théorème

Soit X une v.a à queue algébrique alors

$$E \log|X| < \infty \tag{1.34}$$

Preuve : D'après [57] si X est un processus α-stable, alors il existe p ($0 < p < \alpha$) tel que $E|X|^p < +\infty$. L'inégalité de Jensen [58] garantit que pour la fonction concave f et la v.a Z, $Ef(Z) < f(E.Z)$. Si $f(x) = log|x| / p$ et $Z = |X|^p$ nous obtenons :

$$E \log|X| = E\left(\frac{\log|X|^p}{p}\right) \leq \frac{\log(E|X|^p)}{p} < \infty \tag{1.35}$$

Pour développer un cadre général permettant, quel que soit α, de caractériser les processus α-stables, les moments logarithmiques peuvent être une solution intéressante.

5 Puissance des processus α-stable

5.1 Impossibilité d'une puissance quadratique

Il est important pour les processus α-stables de pouvoir évaluer l' « intensité » du processus, autrement dit sa puissance. C'est un paramètre d'échelle qui permet de comparer l'importance de l'interférence par rapport au signal utile et, éventuellement, aux autres bruits. Classiquement, la puissance arithmétique est utilisée. Elle représente la moyenne des v.a au carré :

$$P(X) = E\left|X\right|^2 \tag{1.36}$$

Il est claire que la valeur définie par (1.36) n'est pas applicable pour un processus α-stable avec $\alpha < 2$. En effet, dès que α est inférieur à 2, le moment d'ordre 2 et donc la puissance définie par (1.36) est infinie. Nous pouvons nous demander alors s'il est possible de définir un opérateur puissance *P* vérifiant quelques propriétés élémentaires de la puissance arithmétique données par (1.37) à (1.40) :

- Une *v.a X* constante de valeur unité présente une puissance aussi égale à l'unité.

$$P\left(1\right) = 1 \tag{1.37}$$

- Si une *v.a X* est multipliée par une constante, sa puissance doit varier d'un facteur a^2 :

$$P\left(a.X\right) = a^2 P(X) \tag{1.38}$$

- Si deux *v.a X* et *Y* sont indépendantes alors leurs puissances s'ajoutent :

$$P\left(X+Y\right) = P(X) + P(Y) \tag{1.39}$$

Nous pouvons facilement montrer qu'il n'est pas possible de définir un tel opérateur dans le cas général des processus α-stables. Soit S_α une v.a. α-stable de paramètres $\mu = 0$ et $\gamma = 1$. Sa fonction caractéristique est $\exp(-|t|^\alpha)$. Grâce à la forme de la fonction caractéristique, il est simple de montrer que pour deux v.a. $S_\alpha^{(1)}$ et $S_\alpha^{(2)}$ i.i.d ($\mu = 0$ et $\gamma = 1$) alors par la propriété de stabilité :

$$aS_\alpha^{(1)} + bS_\alpha^{(2)} + c \overset{d}{=} \sqrt[\alpha]{a^\alpha + b^\alpha}\, S_\alpha + c \tag{1.40}$$

Appliqué à l'opérateur puissance en utilisant les propriétés (1.37) à (1.38), nous obtenons, en partant du membre de gauche :

$$P\left(a.S_\alpha + b.S_\alpha\right) = a^2 P(S_\alpha) + b^2 P(S_\alpha) = (a^2 + b^2) P(S_\alpha) \tag{1.41}$$

Et en partant du membre de droite :

$$P\left(a.S_\alpha + b.S_\alpha\right) = P(\sqrt[\alpha]{a^\alpha + b^\alpha}\, S_\alpha) = (a^\alpha + b^\alpha)^{2/\alpha} P(S_\alpha) \tag{1.42}$$

L'égalité de ces deux dernières équations n'est possible que dans le cas gaussien (α = 2). Il n'est donc pas possible de définir un opérateur de puissance vérifiant les propriétés habituelles de la puissance quadratique (arithmétique) pour les processus α-stables. Toute définition de puissance dans le cas général des α-stable devra donc s'affranchir d'au moins une des propriétés (1.37) à (1.39).

5.2 Puissance géométrique

5.2.1 Définition

Gonzalez dans [56], [57] a proposé une autre forme de puissance, nommée puissance géométrique, qui s'affranchit de la condition de linéarité (1.40).

Définition 4

Soit *X* est *v.a*. La puissance géométrique de cette variable s'écrit :

$$S_0 = S_0(X) = \exp\left[\mathrm{E}\log X^2\right] = \exp\left[2.\mathrm{E}\log|X|\right] \tag{1.43}$$

Gonzalez dans [54] définit l'écart type des processus α-stable plutôt que la puissance géométrique. Nous préférons modifier cet aspect et définir la puissance des *v.a X* par l'équation 1.43.

La puissance géométrique d'une *v.a X* de densité de probabilité *f(x)* s'écrit alors comme suit :

$$S_0 = exp(2.\int_{-\infty}^{+\infty} log(|x|)\, f(x)\, dx) \tag{1.44}$$

A partir de cette formule nous pouvons déterminer les puissances géométriques de quelques distributions usuelles. Elles sont données dans le tableau 1.1.

Tab.1.1 : Puissance géométrique de quelques distributions connues.

Type de distribution	Densité de probabilité	Puissance géométrique
Uniforme	$f(x) = 1/\sigma, \quad \|x\| < \sigma/2$	$S_0(x) = (\sigma/2.e)^2$
Gaussien	$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}$	$S_0(x) = \sigma^2/2.C_g$ avec $\sigma^2 = 2\gamma^2$
Cauchy	$f(x) = \frac{\sigma}{\pi}\frac{1}{\sigma^2 + x^2}$	$S_0(x) = \sigma^2$
Pareto	$f(x) = \alpha.\sigma^2.x^{-\alpha}, \quad x > \sigma$	$S_0(x) = \sigma^2.e^{2/\alpha}$

5.2.2 Propriétés

Cette puissance géométrique présente d'importantes propriétés:

Propriété 1 : (S_0 est un paramètre d'échelle)

Pour tout processus d'ordre logarithmique X et pour toute constante c alors :

(i) $S_0(X) \geq 0$

(ii) $S_0(cX) = c^2 S_0(X)$

Propriété 2 : (S_0 est un indicateur de puissance d'un processus)

Pour tout processus d'ordre logarithmique X et pour toutes constantes c, c_1 et c_2, alors:

(i) $S_0(c) = c^2$

(ii) Si pour toute réalisation *x* de *X*, $0 \leq c_1 < |x| < c_2$ alors $c_1^2 < S_0(X) < c_2^2$

(iii) $S_0(X) = 0$ si et seulement ci $\Pr(X = 0) > 0$, ce qui implique que la nullité de cette puissance quand la masse de probabilité discrète est localisée en zéro.

La démonstration se trouve dans la référence [57] page 34.

Propriété 3: (multiplicativité de S_0)

Pour toute paire de *v.a* X, Y et pour toute constante c :

(i) $S_0(X.Y) = S_0(X).S_0(Y)$

(ii) $S_0(X/Y) = S_0(X)/S_0(Y)$

(iii) $S_0(X^c) = S_0(X)^c$

La démonstration est dans la référence [57] page 36.

Notons que la propriété de multiplicativité est une conséquence directe des propriétés du logarithme. Le résultat n'est pas dépendant la structure de corrélation entre X et Y.

Propriété 4: (L'inégalité de valeur absolue de S_0)

Pour toute paires de *v.a* X, Y alors

$$S_0(|X|+|Y|) \geq S_0(X) + S_0(Y)$$

La démonstration est dans la référence [74] page 37.

5.2.3 Puissance géométrique d'une distribution α-stable symétrique

La puissance géométrique d'une *v.a* α-stable symétrique s'exprime en fonction de α et γ par [48] :

$$S_0 = e^{2.E\log|X|} = \gamma^2.(e^{C_e})^{\frac{2}{\alpha}-2} = \frac{\gamma^2.\left(C_g\right)^{\frac{2}{\alpha}}}{C_g{}^2} \tag{1.45}$$

où $C_g = e^{C_e} \approx 1.78$ est une constante qui représente l'exponentiel de la constante d'Euler.

La puissance géométrique nous permet de caractériser des processus impulsifs à variance infinie qui apparaissent dans les systèmes de transmission, mais elle ne permet pas de caractériser les performances de ces systèmes. Cette puissance permet par la suite de définir un nouveau rapport SNR que nous appellerons SNR géométrique (GSNR). Ce nouveau rapport sera détaillé plus dans ce qui suit.

6 Le rapport signal sur bruit géométrique

Dans le cadre d'un processus impulsif de variance infini, le SNR standard du second ordre est nul et devient donc inutile pour caractériser le système de transmission et en présenter les performances. Dans ce cas, il est possible d'utiliser la puissance géométrique qui nous permet de définir un indicateur universel : le SNR géométrique (GSNR). Ce rapport est défini par l'équation (1.46) :

$$\mathrm{GSNR} = \frac{1}{2C_g}.\frac{A^2}{S_0} \tag{1.46}$$

où A est l'amplitude du signal utile reçu.

La constante de normalisation $2.C_g$ garantit que dans le cas gaussien, le GSNR redevient le SNR standard.

Différents systèmes numériques sont comparés en analysant la probabilité d'erreur bit (BER) ou paquet (FER) en fonction du rapport signal sur bruit SNR. Mais dans notre cas, ces performances sont présentées en fonction du rapport signal sur bruit géométrique GSNR pour les systèmes non codés ou en fonction de $(E_b/N_0)_{\text{géométrique}}$ pour les systèmes codés qui sont différents du SNR standard. Ceci représente un inconvénient pour comparer ces performances utilisant ces paramètres et les résultats qui se trouvent dans la littérature. Afin de résoudre ces problèmes nous avons proposé de présenter les comportements de ces systèmes en fonction de la variation de la dispersion γ (proportionnel à la variance dans le cas AWGN) du bruit impulsif. Ce choix sera utilisé le long de cette thèse.

7 Capacité du canal AISαSN

La majorité des recherches dans le domaine de théorie du signal supposent que le bruit dans les systèmes de transmission suit une loi normale. Le canal de transmission dans ce cas est modélisé par un canal à bruit blanc gaussien additif — AWGN en anglais pour Additive White Gaussian Noise. Quand le bruit est non gaussien, les formules de capacité du canal obtenues dans le cas AWGN ne sont

naturellement plus applicables. Dans la suite de cette section, nous introduisons la notion de canal à bruit additif SαS et nous étudions la capacité de ce canal en fonction du paramètre α.

7.1 Définition du canal AISαSN

Nous appellerons le canal de transmission utilisé dans la suite de cette thèse par le canal à bruit additif symétrique α-stable à échantillons indépendants noté par AISαSN, dont la relation d'entrée/sortie est :

$$Y = X + W \tag{1.47}$$

où X est le signal d'entrée du canal et W est le bruit impulsif modélisé par une v.a. SαS. Comme les échantillons de bruit sont indépendants entre eux, ce canal est dit sans mémoire. Nous supposerons de plus que les entrées sont binaires et donc que les sorties sont symétriques. Ces types de canaux sont qualifiés de BMSC pour Binary Memoryless Symmetric Channel.

7.2 Capacité d'un canal AISαSN à entrée binaire

La capacité C d'un canal est la quantité maximale d'information pouvant y transiter avec une probabilité d'erreur aussi faible que souhaitée [60]. Elle est donnée par la formule suivante :

$$C = \max I(X,Y) \tag{1.48}$$

où la maximisation se fait sur la distribution de la variable X. C'est le maximum d'information mutuelle I entre le signal source X et le signal reçu Y. Dans le cas d'une source binaire, nous pouvons développer l'expression (1.48) :

$$\begin{aligned} C &= \max I(X,Y) = H(Y) - H(Y/X) \\ &= \int \left(-P_Y(y).\log_2 P_Y(y) + \frac{1}{2}\sum_{x=\pm 1} P_{Y/X}(y/x).\log_2 P_{Y/X}(y/x) \right) dy \\ &= \int \frac{1}{2}\sum_{x=\pm 1} P_{Y/X}(y/x).\log_2 \frac{P_{Y/X}(y/x)}{\frac{1}{2}\left(P_{Y/X}(y/1)+P_{Y/X}(y/-1)\right)} dy \\ &= \int P_{Y/X}(y/1).\log_2 \frac{P_{Y/X}(y/1)}{\frac{1}{2}\left(P_{Y/X}(y/1)+P_{Y/X}(y/-1)\right)} dy \end{aligned} \tag{1.49}$$

Le passage de la première à la seconde ligne s'obtient grâce à l'hypothèse d'une entrée X binaire (+1 ou -1) équiprobable. Le dernier passage, quant à lui, suppose la symétrie du canal qui est justifié par le caractère additif du bruit centré.

Dans ce cas la capacité de ce canal [60], [61], [62] va être calculée à partir la formule (1.49).

Le canal AISαSN est symétrique alors $P(y / x=1) = P(-y / x=-1)$. Soit $f(y)$ la densité de probabilité à la sortie du canal quand le symbole 1 est émis. Alors $P(y / 1) = f(y)$ et $P(y / -1) = f(-y)$. Le logarithme du rapport de vraisemblance (LLR) prend alors l'expression particulière suivante :

$$m(y) = \log \frac{P_{Y/X}(y/1)}{P_{Y/X}(y/-1)} = \log \frac{f(y)}{f(-y)} \qquad (1.50)$$

Par conséquent, la capacité peut s'écrire en remplaçant le rapport $f(-y) / f(y)$ par $e^{-m(Y)}$,

$$C = \int f(y).\log \frac{f(y)}{\frac{1}{2} f(y)(1+e^{-m(y)})} dy = \int f(y)(1-\log(1+e^{-m(y)})).dy \qquad (1.51)$$

La capacité s'exprime par conséquent comme l'espérance mathématique $E[1-\log(1+e^{-m(Y)})]$.

L'expression $m(Y)$ est une variable aléatoire notée M qui permet de calculer la capacité directement sous la forme d'une moyenne :

$$C = 1 - E[\log(1+e^{-M})] \qquad (1.52)$$

Les LLR $m(y)$ sont calculés numériquement à partir de la fonction caractéristique de la distribution SαS (1.22) du bruit. Elle dépend de l'exposant caractéristique α et de la dispersion γ. Les figures suivantes présentent le comportement de ces LLR $m(y)$ en fonction de variations des paramètres α et γ de cet environnement AISαSN.

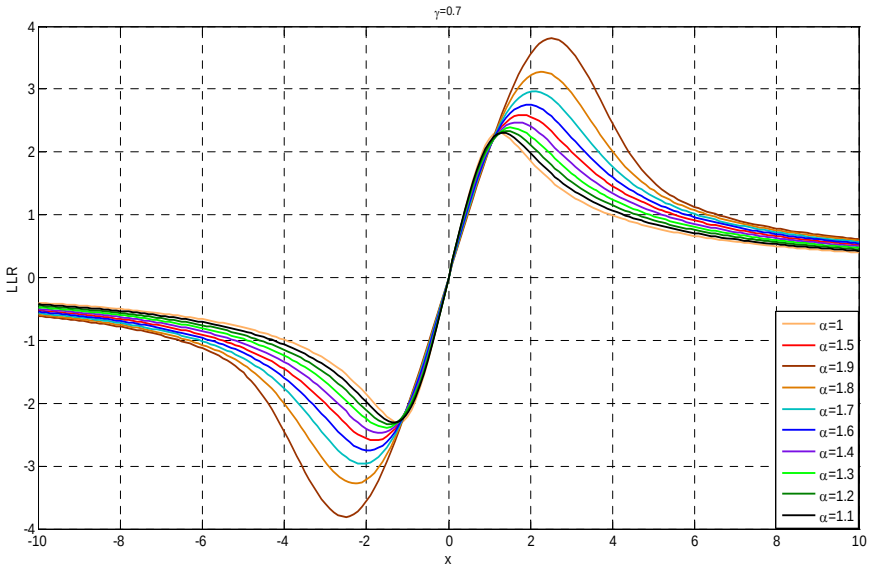


Fig.1.11 : Comportement du LLR en fonction de la sortie du canal AISαSN pour un γ fixe et différents α.

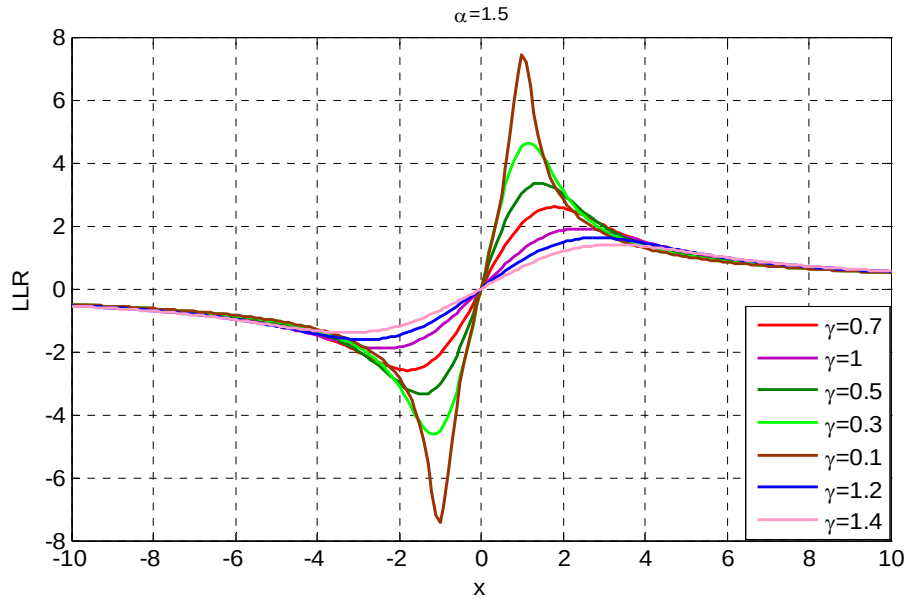


Fig.1.12 : Comportement du LLR en fonction de la sortie du canal AISαSN pour un α fixe et différents γ.

Nous remarquons à partir des figures 1.11 et 1.12 que les LLR dépendent fortement des paramètres α et γ de ce type du canal. L'amplitude de ces LLR augmente en fonction de la diminution de dispersion ou quand α tend vers 2. Ces LLR vont être utilisés afin de montrer le comportement de la capacité du canal AISαSN**.**

Afin d'estimer cette capacité, une méthode de Monte-Carlo est appliquée. La capacité estimée sur N-échantillons de la variable aléatoire M est donnée par (1.53).

$$C \approx 1 - \frac{1}{N}\sum_{i=1}^{N} \log(1 + e^{-m_i}) \qquad (1.53)$$

Cette capacité estimée dépend de la dispersion γ et de l'exposant caractéristique α du bruit. Dans ce qui suit nous présentons l'effet de ces paramètres sur le comportement de la capacité du canal AISαSN.

7.3 Comportement de la capacité du canal AISαSN

La figure 1.13 montre l'évolution de la capacité en fonction de la variation de γ pour différentes valeurs de α.

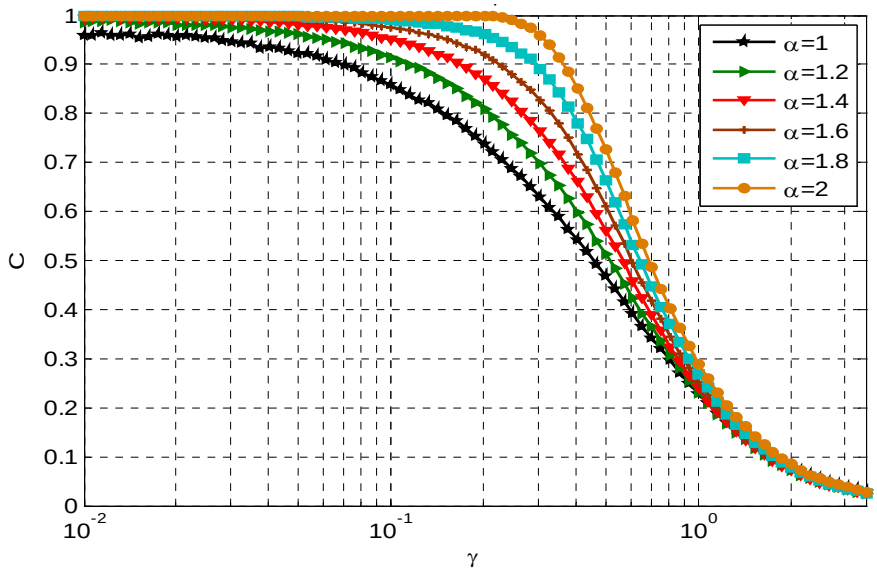


Fig.1.13 : Variation de la capacité du canal AISαSN en fonction de γ et pour différents α

Nous remarquons que la capacité se dégrade quand la valeur de la dispersion γ augmente ou quand la valeur de l'exposant caractéristique α augmente. En particulier il est beaucoup plus difficile d'atteindre une capacité très proche de 1 quand l'impulsivité du bruit augmente.

Dans la plage de bruit qui se présente en pratique, la capacité du canal dépend fortement des paramètres caractéristiques du canal. Ce canal de transmission AISαSN sera intégré par la suite dans la chaine de transmission qui sera utilisée le long de cette thèse.

8 Performance de la chaine de transmission dans les environnements AISαSN

Afin de rapprocher notre travail de thèse aux systèmes de transmission sans fil dans le cas réel, nous avons utilisé la chaine de transmission suivante.

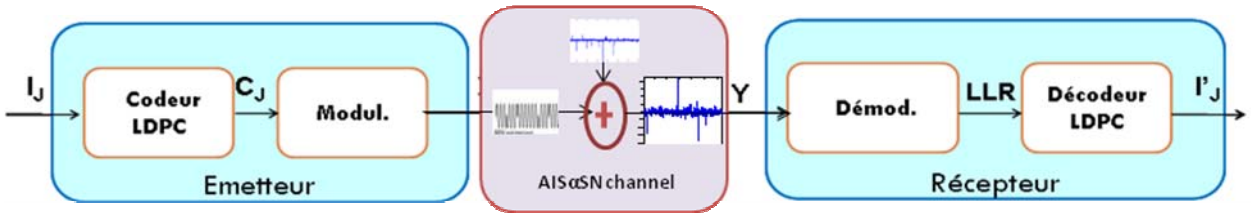


Fig.1.14 : Chaine de transmission sans fil

La chaîne est composée de trois parties : l'émetteur, le canal de transmission AISαSN et le récepteur. Les bits d'information I_j sont générés par la source puis codés par le codeur LDPC pour donner les bits des mots de codes C_j, qui portent et l'information et la redondance. Ces derniers vont être modulés par le modulateur *BPSK* qui au bit 0 associe X = +1 et au bit 1 associe X = -1. La puissance unité est assigné de cette manière au signal modulé X. Ce signal est ensuite émis et va être propagé à travers le canal de transmission AISαSN.

A la réception ce signal Y va être démodulé puis décodé par le décodeur LDPC pour aboutir au final les bits d'information I_j' estimés.

La figure 1.15 présente le comportement d'une transmission BPSK sans codage dans différents environnements AISαSN (valeur α). Nous retrouvons en ordonné la dispersion γ pour laquelle nous montrons le comportement des performances de cette chaine sans codage pour différentes valeurs de *α*. À noter que les différentes courbes ne sont pas vraiment comparables car les puissances de bruit ne le sont pas pour des exposants caractéristiques différents du bruit.

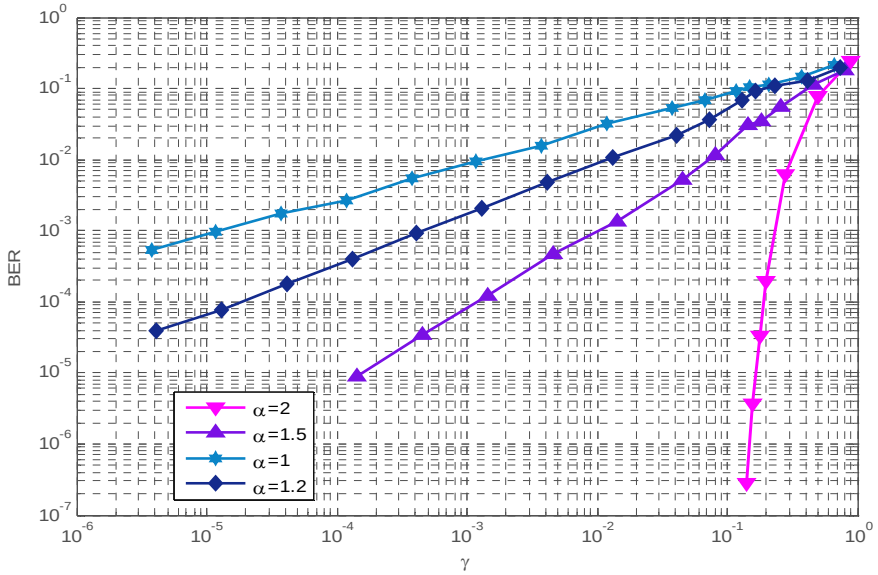


Fig.1.15 : Performance de la chaine de transmission dans les environnements AISαSN

Nous constatons que plus le bruit devient impulsif (α petit), plus il est difficile d'atteindre de faibles taux d'erreurs. En effet, des impulsions fortes viennent régulièrement perturber la communication alors que dans le cas gaussien, quand la variance est faible, il n'y a quasiment plus d'erreurs qui se produisent.

Plus précisément, la probabilité d'erreur décroit exponentiellement avec γ dans le cas gaussien, *i.e.* lorsque α = 2, mais décroit seulement polynomialement lorsque α < 2.

9 Conclusion

Au cours de ce chapitre nous avons présenté certaines distributions permettant de modéliser des processus impulsifs qui apparaissent dans les environnements de transmission. Notre choix s'est porté sur la distribution symétrique α-stable qui par sa propriété de stabilité et par sa justification théorique nous semble appropriée pour représenter l'interférence dans les réseaux.

Nous avons alors introduit le canal de transmission à bruit additif symétrique α-stable à échantillons indépendants noté par AISαSN. Nous avons déterminé la capacité de ce canal.

Mais, cette distribution à queues lourdes est à variance infinie quand α est strictement inférieur à 2. Ce fait entraine la nullité du rapport SNR standard. Une des solutions proposée est d'utiliser la théorie des statistiques d'ordre zéro proposée par Gonzalez afin de caractériser ces types processus. Cette théorie, basée sur les moments d'ordre logarithmique, permet de caractériser ces processus par la puissance géométrique et la qualité d'une transmission par le SNR géométrique.

Il nous a semblé cependant plus judicieux de se référer directement à la dispersion du processus α-stable considéré pour caractériser l'intensité du bruit. Cela nous interdit de comparer directement les résultats obtenus pour différents α mais ce n'est pas notre objectif dans ce travail et le recours direct à la dispersion nous paraît plus clair.

Les codes LDPC

1 Introduction
2 Les codes LDPC
2.1 Définition et type de code LDPC
2.2 Construction des codes
2.3 Encodage
3 Représentations graphiques d'un code LDPC
3.1 Graphe factoriel
3.2 Graphe factoriel normal
4 Décodage des codes LDPC
4.1 Décodage MAP
4.2 Algorithme Sum-Product
4.3 Algorithmes de décodage
4.3.1 Bit-flipping, Gallager A et B
4.3.2 Belief Propagation (BP)
4.3.3 Algorithme Min-Sum
5 Analyse des performances asymptotiques des codes LDPC
5.1 Exemple d'analyse asymptotique : le canal à effacement
5.2 Évolution de Densité (ED)
5.3 Approximation gaussienne et EXIT charts
5.4 Utilisation des analyses asymptotiques
6 Les paramètres d'entrer du démappeur LDPC : les LLR
7 Performances asymptotiques des codes LDPC dans les environnements AISαSN
8 Conclusion

1 Introduction

Les codes LDPC (Low Density Parity Check) font leur première apparition en 1962 dans une monographie de Gallager [66]. Hormis quelques exceptions comme les travaux de Zyablov *et al.* [100] en 1975, de Tanner [84] en 1981 et de Wiberg [70], ces codes n'ont pas suscité suffisamment d'intérêt au sein de la communauté de la théorie du codage. La raison principale de ce délaissement en est la complexité du décodage par rapport aux capacités calculatoires de l'époque.

Alors que la communauté des codes correcteurs se focalisait principalement sur des codes basés sur des structures algébriques fortes, en 1993, Berrou et Glavieux présentent dans [65] les turbo-codes qui sont basés sur deux concepts : la concaténation de deux sous-codes et le décodage utilisant une rétroaction.

Les turbo-codes [71] ont alors ouvert la voie à des codes correcteurs permettant de s'approcher de la limite de Shannon avec des codes de longueur gérable et un décodage pouvant être mis en œuvre de façon effective.

MacKay redécouvre dans [72] la monographie de Gallager et remet au goût du jour les codes LDPC qui ne sont pas soumis à un brevet contrairement aux turbo-codes. Cet article de MacKay présente des constructions de codes LDPC et montre leur bonnes performances par de nombreuses simulations.

Notre travail lors de ce chapitre s'est alors axé sur l'étude des codes LDPC en présentant ces différentes propriétés, en décrivant ces différents algorithmes de décodages ainsi que les outils permettant d'analyser leurs performances asymptotiques.

En effet ce chapitre se devise en quatre grandes parties. La première partie est consacrée à la présentation des codes LDPC. La deuxième partie est réservée à introduire les différents algorithmes de décodages de ces codes. Les outils d'analyse des performances asymptotiques des codes LDPC sont introduits dans la troisième partie. La dernière partie présente le comportement des performances asymptotiques de quelques démappeurs LDPC dans les environnements AISαSN.

2 Les codes LDPC

2.1 Définition et type de code LDPC

Les codes LDPC se sont des classes des codes blocs linéaires, définis par une matrice binaire de parité particulièrement creuse, c'est-à-dire qu'une grande majorité des coefficients sont nuls et seulement quelques-uns valent 1.

S'il faut quantifier l'aspect creux d'une matrice, il s'agirait de la proportion de coefficients non nuls de la matrice de parité relativement à sa taille. Dans le cadre des codes binaires, si ce rapport varie linéairement avec la taille du code alors il s'agit d'un code LDPC alors que la variation est quadratique dans le cadre d'un code binaire classique.

Les mots d'un code linéaire en bloc sont les vecteurs d'un sous-espace de dimension k immergé dans un espace vectoriel de dimension n. Le paramètre n s'appelle la longueur du code alors que le rapport

$R = k / n$ définit le rendement qui mesure la quantité d'information contenue dans un bit d'un mot de code.

Grâce à sa structure, un code linéaire en bloc se construit en se donnant une base du sous-espace vectoriel — ou de façon équivalente grâce à une matrice génératrice G de taille $k \times n$ et de rang plein dont les lignes sont constituées des vecteurs de la base. Ainsi un mot de code x est le produit d'un vecteur $\boldsymbol{c}$ contenant k bits d'information qui constitue le message et de la matrice génératrice G.

Toutefois, il est possible de prendre la forme duale afin de caractériser les codes linéaires. En effet, il existe une matrice H de rang plein, dite de contrôle ou encore de parité, de taille $(n - k) \times n$ telle que $H\, x^T = 0$ pour tous les mots de code x. Si x n'est pas un mot de code, alors le vecteur $\boldsymbol{s}$, appelé syndrome, résultat du produit matriciel précédant n'est pas nul. De par sa définition, ce syndrome ne dépend que des erreurs de transmission et non du mot de code émis.

Les codes LDPC sont une famille de code uniquement définis par la matrice de parité H pseudo aléatoires de faible densité. Cette matrice doit de plus avoir la propriété d'être creuse, c'est-à-dire que la densité d'éléments non nuls de la matrice doit varier au plus linéairement avec la longueur du code.

Initialement, Gallager avait défini les codes LDPC par des contraintes plus fortes. En effet, il considère dans [66] que la matrice de parité doit contenir exactement ρ valeurs non nulles par ligne et λ valeurs non nulles par colonne. Par conséquent, chaque élément du mot de code participe à λ équations de contrôle de parité et chacune de ces équations est composée de ρ bits. Cet ensemble de codes LDPC est par conséquent paramétré par λ, ρ et dont le rendement vérifie $R = k / n > 1 - (\lambda / \rho)$ où la dernière égalité est presque surement une égalité.

Lorsque les contraintes précédentes sont assouplies, l'ensemble des codes LDPC sera qualifié de codes irréguliers. Dans ce cas, un ensemble sera paramétré par deux séquences λ_i et ρ_j. Les valeurs λ_i comptabilisent le nombre de valeurs non nulles associées à une colonne de poids i. Les valeurs ρ_j sont définies de la même façon pour les lignes.

Ces paramètres semblent abscons mais ils sont directement reliés aux performances asymptotiques des codes LDPC. De plus il est possible de les relier à la proportion de colonnes de poids i et à la proportion de lignes de poids j. Cette dernière paramétrisation facilite les algorithmes de construction des matrices de parité.

Naturellement les poids maximaux des lignes et des colonnes doivent être limités pour que l'ensemble des matrices de parité ainsi paramétrées reste des matrices creuses.

Hormis la généralisation de la structure de la matrice de parité, d'autres généralisations des codes LDPC sont envisageables. Seuls les codes binaires ont été considérés jusqu'à maintenant. Or il est possible de définir des codes LDPC dont les symboles des mots de codes ne sont plus binaires mais appartiennent à un corps finis de taille supérieur à 2 [73]. Les extensions corps considérés sont classiquement d'ordre 2^p et donc de caractéristique 2. Les valeurs non nulles de la matrice de parité sont alors des éléments inversibles du corps finis.

Cette généralisation n'est pas vide de sens puisqu'elle permet d'avoir des codes performants tout en ayant des algorithmes de décodage adaptés utilisables [74].

Une seconde généralisation incluant la précédente a été développée dans [75]. Les symboles des mots de code appartiennent maintenant à un groupe abélien prédéfini. Les éléments non nuls de la matrice

ne sont plus des éléments d'un corps mais des homomorphismes du groupe. Bien que très générale, ce type de code permet la construction d'un algorithme de décodage rapide. La grande force de cette approche est qu'elle permet d'avoir des équations de parité particulièrement générales grâce à la non nécessité d'inversibilité des éléments de la matrice de parité.

Il existe encore d'autre type de généralisation de code LDPC où les symboles n'appartiennent pas au même ensemble, etc.

2.2 Construction des codes

Les codes LDPC ne sont pas des codes prédéfinis mais forment un ensemble de codes tel que ceux utilisés dans les démonstrations de la théorie de l'information. Pour des raisons pragmatiques, il est nécessaire de construire la matrice de parité H d'un des représentants de l'ensemble visé.

La première méthode consiste à construire aléatoirement un élément de l'ensemble. Toutefois, les performances des codes LDPC sont analysées en moyenne et il se peut que le représentant trouvé ne soit pas un bon candidat.

Il devient donc nécessaire de pouvoir comparer les candidats sans toutefois le faire par des simulations. C'est pourquoi quelques critères ont été développés pour classer les candidats. Un premier critère est donné par la circonférence du graphe de Tanner associé au code (cf. ci-dessous). En effet ce paramètre du graphe intervient directement dans les performances de l'algorithme de propagation de croyance utilisé pour le décodage.

En faisant une recherche gloutonne de graphe en utilisant ce critère de circonférence, ou ses dérivés, Hu *et al.* ont, dans [76], construit l'algorithme PEG (Progressive Edge growth) qui garantit une circonférence de graphe minimum. Cet algorithme est le premier du type et à générer un certain nombre d'améliorations comme par exemple celle de [77].

L'approche aléatoire de la construction est parfois problématique car elle demande un stockage de la matrice de parité. Cette matrice ne comportant pas de symétrie, ce stockage peut être gourmand. De plus les algorithmes de décodage ne pourront pas utiliser avec profit de ces symétries. C'est pourquoi plusieurs approches structurées ont été investies.

Une telle approche est présentée dans [78] utilise la relation d'incidence des droites et des points des géométries finies, euclidiennes ou projectives. Les symétries de cette structure sont alors très fortes : il est impossible de différencier les variables les unes des autres. D'autres méthodes de ce type sont l'objet de [79].

La théorie des groupes fut aussi utilisée pour la construction de matrice de parité. Par exemple les codes LDPC de Margulis utilisent deux copies du groupe $SL_2(p)$ pour construire des matrices de codes de paramètres (3, 6).

Malheureusement, même si la construction est esthétiquement intéressante, ces codes présentent des particularités qui en font des candidats peu fiables comme le montre l'article de MacKay et Postol [80].

D'autres constructions semi-structurées ne font pas appel à l'algèbre. Nous pouvons citer par exemple les proto-graphes ou les codes LDPC multi-edges qui sont aussi intéressants du point de vue de l'optimisation de leurs paramètres [81].

2.3 Encodage

Les codes LDPC sont définis de façon duale par leur matrice de parité. L'utilisation effective de tels codes nécessite cependant de pouvoir associer à un vecteur d'information c un mot de code x. Ce passage se faisant principalement grâce à la matrice génératrice du code.

Malheureusement, la matrice génératrice d'un code LDPC n'est pas nécessairement creuse. Comme les codes sont en pratique suffisamment longs, cet aspect pose un certain nombre de problèmes. Le premier concerne le stockage d'une matrice de taille conséquente mais non creuse. Le second problème se situe au niveau de la multiplication entre cette matrice génératrice et le vecteur d'information. Cette multiplication aura une complexité de l'ordre de $O(n^3)$, ce qui peut être prohibitif.

Une solution au problème de l'encodage a été proposée dans [82] où les auteurs utilisent une décomposition par bloc de la matrice de parité qui garde alors une forme creuse. L'encodage devient alors rapide de par cette faible densité de coefficients non nuls. En effet, la complexité de la plupart du processus d'encodage est ramenée à l'ordre $O(n^2)$. Il reste cependant une multiplication par une petite matrice pour finir l'encodage. Les auteurs, grâce à une étude basée sur le décodage des codes LDPC sur le canal à effacement ont pu estimer la taille de cette matrice. Cette taille étant faible en pratique, cette méthode d'encodage est relativement envisageable.

Une approche utilisant une décomposition LU est aussi envisageable [83]. Elle nécessite un calcul préliminaire lourd, mais une fois la décomposition trouvée, l'encodage est rapide. L'empreinte mémoire nécessaire au stockage des matrices issues de la décomposition reste faible d'où une complexité quasiment linéaire.

La dernière approche consiste à construire directement une matrice de parité sous forme systématique. Cela permet d'encoder et de récupérer le mot d'information simplement par lecture. L'autre forme simple est semi-systématique consiste à utiliser plusieurs « diagonales » non nulles. La difficulté se porte alors sur la construction des codes LDPC supportant cette contrainte de forme.

3 Représentations graphiques d'un code LDPC

Un code LDPC peut également être représenté, en plus de sa matrice de contrôle de parité H, sous une forme graphique. Cette représentation, qualifiée de graphe de Tanner [84], permet de décrire et de visualiser la fonction de répartition à calculer pour, dans notre cas, faire le décodage. Ainsi, il sera possible de visualiser l'algorithme de décodage.

Les codes LDPC se prêtent bien à ce type de représentation car la fonction à évaluer est une somme de produits de fonctions. Ces fonctions sont de deux types : les fonctions indicatrices de la satisfaction des parités et les fonctions donnant les valeurs des a priori des symboles issus du canal $\Pr([y_i|x_i])$.

L'article de Loeliger [85] donne un panorama général des représentations graphiques. Ci-dessous, deux types de graphes sont passés en revue : les graphes factoriels et les graphes factoriels normaux.

3.1 Graphe factoriel

Un graphe factoriel représente une fonction à évaluer qui est factorisée sous la forme d'une somme de produits. Ce type de graphe est présenté dans [86]. Des idées similaires sont généralisées dans [87].

Le graphe factoriel est un graphe biparti dont les nœuds sont de deux types : les nœuds de variable et les nœuds de parité. Les nœuds de variable représentent les symboles du mot de code et les nœuds de parité correspondent aux équations de parité. Un nœud de variable ***v*** est relié à un nœud de parité ***c*** par une branche, si et seulement si, l'élément correspondant à la $c^{ème}$ ligne et la $v^{éme}$ colonne de la matrice de contrôle de parité H est non nul. Par convention, les nœuds de variable seront représentés par des cercles et les nœuds de parité par des carrés.

A titre d'exemple, soit un code LDPC régulier de type (2,4) dont la matrice de parité vaut.

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \qquad (2.1)$$

Le graphe factoriel correspond à cette matrice de parités H, est donné sur la figure 2.1.

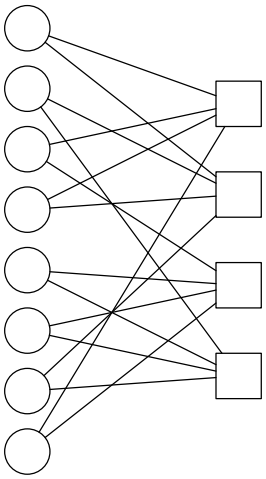


Fig.2.1 : Exemple de graphe factoriel

La représentation par un graphe d'un code LDPC nous permet aussi d'introduire la notion de cycle. Un cycle existe dans un graphe dès lors qu'il y a un chemin pour quitter u nœud et y revenir sans passer par les mêmes branches. Le nombre de branches traversées détermine la longueur du cycle. Un graphe sans cycle est appelé un arbre.

3.2 Graphe factoriel normal

Afin d'expliquer l'algorithme de décodage nous utiliserons de préférence les graphes factoriels normaux qui normalisent les graphes factoriels précédents. Ils ont été initiés par Forney dans [88] où ils permettent d'obtenir des résultats théoriques de dualité intéressants. Cette représentation récupère le nom de « Forney Factor Graph » dans [85]. Ils sont de nouveaux mis au premier plan par leur grande

énéralité [89]. L'explication ci-dessous de l'algorithme de propagation de croyance sera basée sur cet article.

Les graphes factoriels normaux permettent de représenter graphiquement une fonction qui s'écrit sous la forme

$$Z(x) = \sum_{y} \prod_{k} f_k(x_k, y_k) \tag{2.2}$$

où $\boldsymbol{x_k}$ est un sous ensemble des variables x_i et y_k un sous ensemble des variables muettes y.

Un graphe factoriel normal représente les fonctions f_k par des nœuds, souvent dessinés sous forme de boite. Les variables muettes y_i sont les arcs du graphe. Enfin les variables non muettes x_i sont des demi-arcs qui sont graphiquement représentés par des dongles qui illustrent les entrées/sorties du graphe. Grâce à ces dongles, il est possible de représenter un sous-graphe par une seule boîte et inversement.

Par exemple, le graphe présenté la figure 2.2 représente une fonction de répartition dépendant de Y. La boîte G soit un graphe imbriqué soit une fonction ayant un paramètre Y.

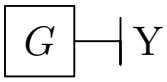


Fig.2.2 : Exemple de graphe factoriel normal représentant $Z_G(Y)$.

Si deux boîtes G et H sont connectées par une arrête étiquetée Y, comme le montre la figure 2.3, alors la fonction représentée est une sommation :

$$Z_{GH} = \sum_{y} Z_G(Y) Z_H(Y) \tag{2.3}$$

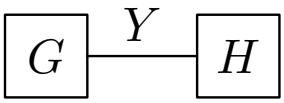


Fig.2.3 : Exemple de sommation dans un graphe factoriel normal.

Cet exemple montre que le nom des arêtes n'est pas nécessaire et qu'elle permet seulement d'écrire mathématiquement la somme sur un indice qui est une variable muette.

Le graphe factoriel normal du code LDPC d'exemple précédent est dessiné sur la figure 2.4. Dans ce cas la différence avec l'image du graphe factoriel n'est pas clairement visible mais pourtant cette différence est fondamentale : dans un cas les variables sont des sommets dans l'autre des arcs. À cause de cette contrainte de normalisation, il est nécessaire d'introduire des nœuds d'égalité, qui sont alors des fonctions et non des variables.

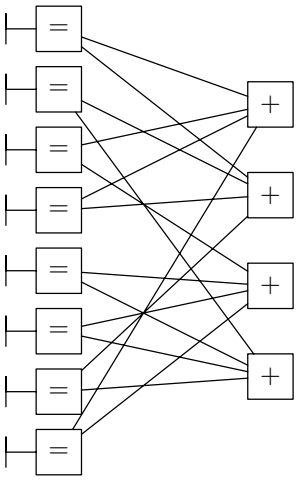


Fig.2.4 : Exemple de code LDPC sous forme de graphe factoriel normal.

4 Décodage des codes LDPC

L'avantage des codes LDPC par rapport aux autres codes c'est qu'ils présentent un décodage pertinent dont la complexité varie seulement linéairement avec la longueur du code. Cet aspect important est principalement dû à la faible densité de la matrice de parité.

En effet, en écrivant le critère MAP (Maximum-A-Posteriori) pour le décodage d'un bit, l'aspect creux apparaît clairement au travers de la condition d'appartenir au code.

4.1 Décodage MAP

La relation entre le mot de code ***x*** émis et le mot ***y*** reçu se résume par la probabilité conditionnelle Pr([y|x]). Par hypothèse, le canal est supposé sans mémoire, par conséquent, $\Pr([y|x]) = \Pi_i \Pr([y_i|x_i])$.

Le décodage MAP du mot de code se calcul une fois la fonction de Pr([x|y]) évaluable. La règle de Bayes permet de l'obtenir à partir du modèle de canal et du code

$$\Pr[x \mid y] \alpha [x \in C] \Pr[y \mid x] = [x \in C] \prod_i \Pr[y_i \mid x_i] \qquad (2.4)$$

où la notation [P] représente la fonction indicatrice de la proportion P qui vaut 1 si P est vraie et 0 sinon.

Le décodage MAP du $i^{ème}$ bit du mot de code est immédiat à partir de la connaissance de $\Pr([x_i|y])$. Obtenir cette valeur s'obtient par marginalisation de l'a posteriori précédent,

$$\Pr[x \mid y] \alpha \sum_{x \setminus i} [x \in C] \prod_i \Pr[y_i \mid x_i] \qquad (2.5)$$

où $\sum_{x\setminus i}$ indique une sommation sur tous les bits x_j sauf le $i^{ème}$.

L'appartenance d'un vecteur x au code C correspond à la conjonction des validités de chaque parité. Cette condition s'écrit sous forme d'un produit de fonctions indicatrices de validité de parité.

À titre d'exemple, si nous prenons la matrice de parité du dernier code LDPC de type (2,4), la fonction indicatrice du code se développe sous la forme

$$\begin{aligned}\left[x \in C\right] = \; & \left[x_1 + x_3 + x_4 + x_8 = 0\right] \\ & \times\left[x_1 + x_2 + x_4 + x_7 = 0\right] \\ & \times\left[x_1 + x_3 + x_4 + x_8 = 0\right] \\ & \times\left[x_3 + x_5 + x_6 + x_8 = 0\right] \\ & \times\left[x_2 + x_5 + x_6 + x_7 = 0\right]\end{aligned} \qquad (2.6)$$

où les additions sont binaires, c'est-à-dire des " ou exclusif " entre les bits.

En intégrant la fonction indicatrice, le calcul de l'a posteriori fait intervenir une somme de produits de fonctions. L'aspect creux de la matrice de parités induit le fait que les fonctions intervenant dans cette somme de produits ne comportent qu'un nombre limité d'arguments. Les algorithmes de décodage utilisent cet aspect pour limiter leur complexité.

4.2 Algorithme Sum-Product

La description du calcul de la fonction de répartition reprend les étapes de l'article de Forney et Vontobel [89].

Les graphes factoriels normaux représentent une fonction de répartition de la forme d'une somme de produits. Il est possible d'utiliser cette représentation pour en déduire un algorithme de calcul basé sur la factorisation de la fonction de répartition.

Cet algorithme de passage par message se décrit récursivement en utilisant des blocs qui représentent des sous-graphes. Si la fonction de répartition peut s'écrire sous la forme

$$Z_{GH} = \sum_{y} Z_G(Y) Z_H(Y) \qquad (2.7)$$

telle qu'elle est représentée sur la figure 2.3, alors il est possible de décomposer en deux partie cette fonction. Cette décomposition est donnée par la figure 2.5.

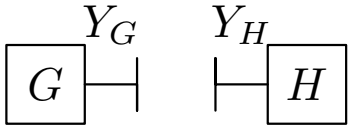


Fig.2.5 : Exemple de sommation dans un graphe factoriel normal.

Ainsi, le calcul de la fonction Z_{GH} se fait grâce à deux messages : le bloc G envoie au bloc H le message $\mu_{G\to H}(Y) = Z_G(Y)$ et H envoie au bloc G le message $\mu_{H\to G} = Z_H(Y)$. Comme $Z_G(Y)$ ne dépend pas de $Z_H(Y)$, le message $\mu_{G\to H}(Y)$ peut se calculer récursivement. Il en est de même du message $\mu_{H\to G}(Y)$.

Il reste toutefois à calculer les messages du type $\mu_{G\to H}(Y)$ provenant des graphes de la forme dessinée sur la figure.2. Soit la boîte G est primaire et dans ce cas elle représente une fonction prédéfinie qui peut être évaluée. Soit la boîte G est un sous-graphe qui peut être développé sous la forme de la figure

2.6. Naturellement le nombre de boîtes connectées à la fonction *f* est fixé ici à deux à titre d'exemple mais peut être plus ou moins élevé.

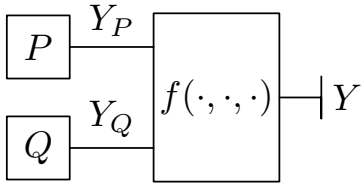


Fig.2.6 : Décomposition par un bloc.

Dans ce cas, la boîte *P* envoie à *f* le message $\mu_{P\to f}(Y_P) = Z_P(Y_P)$, la boîte *Q* envoie à *f* le message $\mu_{Q\to f}(Y_Q) = Z_Q(Y_Q)$. À partir de ces messages la boîte *f* peut calculer la fonction de répartition suivante qui pourra être alors un message sortant :

$$Z(Y) = \sum_{Y_P, Z_Q} \mu_{Q\to f}(Y_Q)\mu_{P\to f}(Y_P) f(Y_P, Y_Q, Y) \qquad (2.8)$$

L'équation du calcul des messages traversant une arête ainsi que celle permettant le calcul du message sortant d'une partie d'un graphe factoriel normal permettent le calcul récursif et effectif de la fonction de répartition.

Il faut toutefois noté que ce calcul est réellement possible uniquement si le graphe factoriel normal est un arbre. Et dans ce cas il est exact. En effet, les sommets d'un arbre peuvent être mis dans un ordre, qualifié de topologique, tel que les décompositions par les arêtes soit bien du type de la figure.3. Autrement, il y a une dépendance qui se créée entre les messages et aucun ordre de calcul des messages n'est possibles.

Toutefois, une valeur approchée de la fonction de répartition est calculable en itérant les calculs précédents plusieurs fois et en initialisant les messages transitant sur les arêtes par des valeurs par défaut, leurs mises à jour se feront progressivement au cours des itérations.

4.3 Algorithmes de décodage

La majorité des codes LDPC ne peuvent pas être décodés algébriquement de par leur longueur importante et le peu de structure de leur matrice de parité. Ainsi les algorithmes de décodage reposent, à l'instar des turbo-codes, sur un processus itératif.

Ces algorithmes ne sont pas optimaux quant au décodage MAP mais permettent toutefois d'approcher la limite de Shannon sur des canaux simples. Le non optimalité s'explique par le fait que le graphe représentant les codes LDPC comporte des cycles. Ces cycles sont inévitables car, dans [90], les auteurs montrent que la distance minimum des codes LDPC qui ont une représentation sans cycles est faible quel que soit le rendement.

Les algorithmes BP et Min-Sum constituent les algorithmes de décodage les plus importants. Il est toutefois possible de les modifier en vue d'améliorer les performances ou encore de les simplifier en vue d'une mise en œuvre matérielle effective.

4.3.1 Bit-flipping, Gallager A et B

Gallager propose dès sa monographie [66] un algorithme de décodage itératif nommé par *bit-flipping*. Le principe général de cet algorithme est le suivant : dans un graphe où des valeurs sont associées aux nœuds, l'algorithme fait passer l'information le long des arêtes du graphe et modifie la valeur des nœuds en fonction de l'information qu'il a reçu via les arêtes. L'algorithme converge vers un état stable après un certain nombre d'itérations.

Les algorithmes A et B de Gallager ne sont utilisables que sur le canal binaire symétrique. Il s'agit de faire un passage de message entre les nœuds de variable et ceux de parité.

Pour les deux algorithmes, une parité envoie un message à la variable qui est la valeur que doit prendre le bit de la variable pour satisfaire la parité en supposant que toutes les autres variables sont correctes.

Un nœud de variable reçoit des parités des messages lui indiquant quel doit être sa valeur pour les satisfaire. Il enverra un message à une parité uniquement en fonction des messages des autres parités et du bit reçu.

Dans le cas de l'algorithme A, le message envoyé par une variable est par défaut la valeur reçue par le canal. Toutefois si tous les nœuds de parité, hormis le destinataire du message, s'accorde sur une autre valeur, cette dernière est envoyée.

L'algorithme B diffère de la version A en utilisant le vote à la majorité pour le choix du message — hors destinataire mais avec le bit issu du canal.

4.3.2 Belief Propagation (BP)

Le décodage des codes LDPC par l'algorithme de propagation de croyance (BP pour Belief Propagation) est l'instanciation de l'algorithme Sum-Product présenté précédemment.

Pour ce faire, la représentation graphique de la figure.4 n'est pas suffisante car elle ne fait pas intervenir les valeurs issues du canal. C'est pourquoi la figure.7 fait intervenir les fonctions f_i qui dépendent des valeurs reçues.

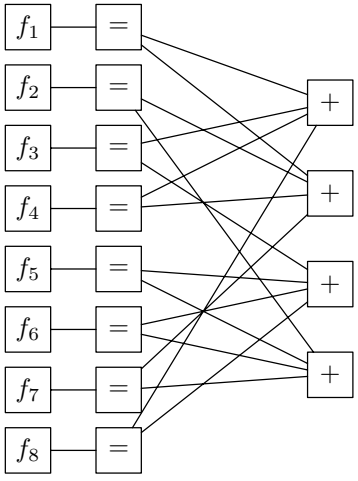


Fig.2.7 : Graphe factoriel normale pour le décodage MAP.

Le graphe de la figure 2.7 représente la fonction de répartition

$$Z_{MAP} = \sum_{x} \prod_{p} [p_satisfaite] \prod_{i} \Pr[y_i \mid x_i] \qquad (2.9)$$

où x balaye l'ensemble des mots binaires de taille n, p balaie celui des parités du code et $\Pr[y_i|x_i]$ est la vraisemblance du $i^{ème}$ bit de x. La fonction [*p satisfaite*] est la fonction indicatrice qui vaut 1 si la parité p est satisfaite par le mot binaire x.

Le lien entre Z_{MAP} et la figure 2.7 est immédiat. Les boîtes f_i sont les fonctions de vraisemblance du ***i*** bit reçu par le canal. Les boîtes + sont les fonctions indicatrices d'une parité. Enfin les boîtes = sont nécessaires pour satisfaire la contrainte des graphes factoriels normaux qui imposent que les variables muettes soient représentées par une seule arête. Ces boîtes permettent alors de faire des clones des "indices" de sommation.

Le calcul de la constante Z_{MAP} par l'algorithme Sum-Product est suffisant pour calculer la probabilité a posteriori de chaque bit. En effet, une fois les messages calculés après un certain nombre d'itération de l'algorithme, il est possible de récupérer $\Pr[x_i|y]$. Il suffit, pour cela, de faire la construction de la figure 2.8 pour s'en convaincre. Cette construction devra se faire au niveau de l'arête entre une boîte = et la boîte f_i correspondante.

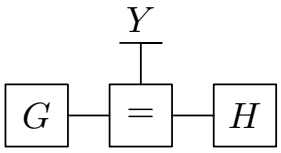


Fig.2.8 : Récupération de la valeur d'un bit.

En effet, la figure 2.8 représente la fonction de répartition

$$Z_{GH}(Y) = \sum_{Y_G, Y_H} Z_G(Y_G) Z_H(Y_H) \left[Y_G = Y_H = Y \right] \qquad (2.10)$$

La fonction indicatrice implique que seuls les indices respectant $Y_G = Y_H = Y$ interviennent dans la sommation, c'est-à-dire que $Z_{GH}(Y)$ vaut simplement le produit

$$Z_{GH}(Y) = Z_G(Y) Z_H(Y) \qquad (2.11)$$

Ce produit est calculable une fois les messages $\mu_{fi \to =}$ et $\mu_{= \to fi}$ calculés lors du calcul de Z_{MAP}.

Pour le code LDPC précédent, cette astuce revient à calculer la fonction de répartition du graphe de la figure 2.9.

En effet la fonction de répartition de ce graphe vaut

$$\Pr[x \mid y] \alpha \sum_{x \setminus i} [x \in C] \prod_{i} \Pr[y_i \mid x_i] \qquad (2.12)$$

Pour résumer, et pour fixer l'algorithme tel qu'il est utilisé dans les simulations de nos travaux, l'algorithme de décodage fait plusieurs itérations.

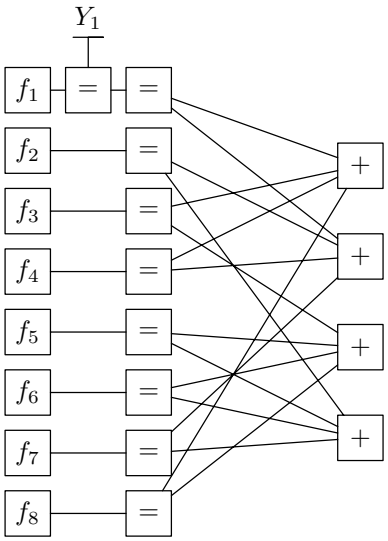


Fig.2.9 : Graphe factoriel normale pour le décodage MAP d'un bit.

Les données utilisées par cet algorithme sont : les vraisemblances de chaque bit reçu, les messages $\mu_{c\to v}$ et $\mu_{v\to c}$ transitant sur les arêtes connectant les variables et les parités.

Chaque itération est composée d'une passe sur les variables Data-Pass et d'une passe sur les parités Check-Pass. La première passe modifie les messages $\mu_{v\to c}$ alors que la seconde modifie $\mu_{c\to v}$. Les messages $\mu_{ch\to v}$ sont issus du canal et sont envoyés à la variable v. Les notations c, c' dénotent des parités et v, v' des variables. Le Data-Pass se résume par le calcul suivant appliqué à chaque variable v et pour chaque parité c qui lui est connectée

$$\mu_{v\to c}(x) = \mu_{ch\to v}(x) + \sum_{c'\neq c} \mu_{c'\to v}(x) \tag{2.13}$$

Tandis que le Check-Pass applique la formule suivante

$$\mu_{c\to v}(x) = \mu_{ch\to v}(x) + \sum_{x_{v'}} \prod_{v'\neq v} \mu_{v'\to c}(x_{v'}) \tag{2.14}$$

où la somme sur $x_{v'}$ est binaire.

L'objectif, qui correspond à une approximation de la probabilité a posteriori de chaque bit est donné par la somme de tous les messages vers la variable correspondante :

$$app_v(x) = \mu_{ch\to v}(x) + \sum_{c} \mu_{c\to v}(x) \tag{2.15}$$

Le décodage binaire revient alors à prendre le x qui maximise $app_v(x)$ pour toutes les variables v.

Les messages $\mu_{v\to c}(x)$, $\mu_{c\to v}(x)$ et $\mu_{ch\to v}(x)$ sont des fonctions de paramètre x qui représentent la croyance en la valeur de la variable x ou de la parité suivant le type de message. Comme nos travaux se restreignent aux codes LDPC binaires, l'argument sera lui aussi binaire.

Toutefois, comme il s'agit de probabilité, il est possible de résumer les messages μ par une seule valeur. En effet la normalisation $\mu(0) + \mu(1) = 1$ permet de ne garder qu'une valeur. Il y a plusieurs

possibilités de résumés : le rapport de vraisemblance ou son logarithme, la différence de vraisemblance ou autre —cf. [78].

Le résumé le plus utilisé reste cependant le logarithme du rapport de vraisemblance :

$$\mu = \log \frac{\mu(0)}{\mu(1)} \tag{2.16}$$

Cette forme est particulièrement simple car alors le Data-Pass reste une sommation :

$$\mu_{v\to c} = \mu_{ch\to v} + \sum_{c'\neq c} \mu_{c'\to v} \tag{2.17}$$

De même l'opération somme et produit du Check-Pass devient

$$\mu_{c\to v} = 2\tanh^{-1}\left(\prod_{v'\neq v} \tanh\frac{\mu_{v'\to c}}{2}\right) \tag{2.18}$$

Dans cette formulation, une forme de transformée de Fourier rapide est présente. Elle est également à l'oeuvre dans la description de l'algorithme de propagation de croyance pour des codes non-binaires par exemple sur des extensions du corps binaire dans [74] ou encore sur des groupes abéliens [75].

Il est à noter que l'ordre d'exécution des deux équations précédentes n'est pas fixé par l'algorithme ; il fait encore l'objet de recherches actuelles. Nos simulations utilisent fait, pour chaque itération, un passage de Data-Pass pour chaque variable puis un passage de Check-Pass pour chaque parité. L'algorithme s'arrête dès que le syndrome est nul ou lorsque le nombre d'itérations dépasse un certain seuil de l'ordre de la centaine.

4.3.3 Algorithme Min-Sum

La propagation de croyance est un algorithme complexe à mettre en œuvre car le Check-Pass nécessite le calcul d'un produit et l'évaluation des fonctions trigonométriques hyperboliques (tangente et arc tangente hyperboliques). Même si ces fonctions sont tabulées, le nombre d'évaluations reste important.

À fort rapport signal à bruit, peu de messages sont erronées, dans ce cas les magnitudes des messages $\mu_{c\to v}$ sont grandes et leur tangente hyperbolique sera proche de 1 ou de −1. Par conséquent, lors du Check-Pass seul le message de plus faible magnitude sera pris en compte dans le produit. Cette remarque permet de simplifier énormément la complexité du décodage.

Les calculs du Data-Pass restent inchangés :

$$\mu_{v\to c} = \mu_{ch\to v} + \sum_{c'\neq c} \mu_{c'\to v} \tag{2.19}$$

Alors que, le Check-Pass devient une minimisation de la magnitude et un calcul simple du signe :

$$|\mu_{c\to v}| = \min|\mu_{v'\to c}| \tag{2.20}$$

$$\mathrm{sgn}(\mu_{c\to v}) = \prod_{v'\neq v} \mathrm{sgn}(\mu_{v'\to c}) \qquad (2.21)$$

où le produit sur les signes revient à faire un “ou exclusif”.

Hormis sa simplicité, l’algorithme du Min-Sum est intéressant car il est complètement invariant par un facteur d’échelle. En effet, multiplier toutes les entrées par un nombre positif revient à multiplier les sorties du décodeur par ce même facteur. Cet aspect sera utilisé avec profit dans le prochain chapitre.

D’autres simplifications sont possibles, par exemple mettre un biais sur le calcul du min-sum qui peut alors être adaptatif. Mais alors l’homogénéité du décodage est perdue.

5 Analyse des performances asymptotiques des codes LDPC

Dans le cas de l'utilisation de techniques de codage avancées, l'évolution de la performance du code en fonction du rapport signal à bruit peut se diviser en trois régions comme illustrée sur la figure 2.10. La première région correspond à un comportement où le décodage ne converge pas, le décodage a alors souvent tendance à dégrader les performances par rapport à un système non codé : on parle de région de non convergence. A partir d'un certain rapport signal à bruit, appelé seuil de convergence, le décodage rentre dans une phase où la probabilité d'erreur diminue très rapidement avec le rapport signal à bruit : on parle de la région du waterfall. Enfin, il existe une région, le plancher d’erreur ou error floor où la probabilité d'erreur diminue de manière moins rapide que la région du waterfall. Ce comportement est fortement lié à la présence de mots de code de poids faible, et par linéarité du code à la distance minimale du code ainsi que de sa multiplicité. Nous appelons un gain de codage l'écart entre deux systèmes de l'énergie par bit utile nécessaire atteindre un taux d'erreur donné.

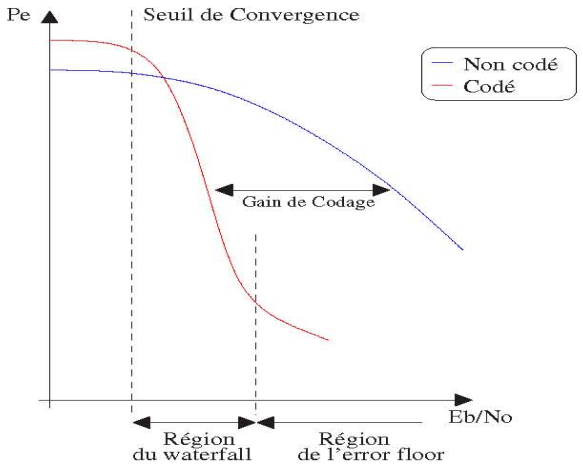


Fig. 2.10 : Illustration de trois régions caractérisant les performances d’un code LDPC

L’étude des performances dans le cas des codes LDPC est difficile. En effet, les performances d’un code particulier peuvent différer des performances moyennes de l’ensemble des LDPC auquel il appartient. Ensuite, l’algorithme BP n’est pas exact sur un graphe contenant des cycles. Or les cycles sont inévitables pour obtenir de bons codes.

Cependant, il est possible d’en faire une analyse asymptotique : en augmentant leur longueur, le comportement des codes LDPC vont se concentrer vers un comportement typique.

Le passage entre les performances du comportement asymptotique vers celle des performances à taille finie n'est pas trivial. Pragmatiquement, pour les tailles et pour le nombre d'itérations des algorithmes de décodage utilisés dans nos travaux, l'analyse asymptotique reste suffisante.

Il existe certaines études des codes LDPC à taille finie [91], [92], [93] mais elles ne sont pas aussi génériques ou générales que les études asymptotiques.

5.1 Exemple d'analyse asymptotique : le canal à effacement

Afin d'expliquer l'analyse des performances asymptotiques d'algorithme BP des codes LDPC, nous avons proposé un exemple simple où le code LDPC sert à corriger les erreurs d'un canal à effacement.

En effet, ce canal transmet correctement les bits présents à son entrée sauf une proportion qui est effacée. Par conséquent, si un bit n'est pas effacé il est sûr. Un tel canal est hypothétique mais il permet une compréhension qualitative des algorithmes de décodage.

Pour ce type de canal, les logarithmes du rapport de vraisemblance ne peuvent prendre que trois valeurs : $+\infty$, $-\infty$ ou 0.

Il est donc possible de dériver les algorithmes du BP pour ces *a priori*. Les messages prendront, eux aussi, seulement ces trois valeurs possibles. Il est par contre impossible que des indéterminations du type $\infty - \infty$ surviennent car les bits ne peuvent pas être corrompus par le canal : les bits non effacés sont une partie d'un mot de code. Par conséquent, dans un souci de simplification, nous pourrons supposer que le mot de code émis est effectivement le mot de code nul ; les messages ne pourront porter que deux valeurs : 0 ou ∞.

Le Data-Pass devient dans ce cas : la variable v envoie un message nul à une parité c seulement si les messages reçus des autres parités et du canal valent tous 0. Sinon le message vaut ∞.

Le Check-Pass devient quant à lui : la parité c transmet à la variable v le message 0 si au moins un des messages reçus des autres variables vaut 0. Sinon le message vaut ∞.

Si à 0 est associé Faux et à ∞, Vrai, alors le Data-Pass correspond à une disjonction, tandis que le Check-Pass correspond à la conjonction. C'est pourquoi il est possible de faire l'analyse du décodage par l'analyse d'un arbre And-Or [96].

Notons p_e la probabilité d'effacement du canal, $p_v^{(l)}$ la proportion de messages valant 0 (effacés) issus des nœuds de variable à l'itération l et $p_c^{(l)}$ celle des messages issus des parités.

Supposons que le code LDPC soit régulier de type (λ,ρ). Nous faisons dorénavant l'hypothèse d'indépendance des messages arrivant sur une variable ou une parité. Alors le Data-Pass nous donne la relation

$$p_v^{(l)} = p_e (p_c^{(l)})^{\lambda-1} \tag{2.22}$$

alors que le Check-Pass implique la relation

$$1 - p_c^{(l)} = (1 - p_v^{(l)})^{\rho-1} \tag{2.23}$$

En regroupant ces deux relations, nous obtenons

$$p_v^{(l)} = p_e(1-(1-p_v^{(l)})^{\rho-1})^{\lambda-1} \tag{2.24}$$

Ainsi, suivre l'évolution du décodage revient à suivre la dynamique de l'équation scalaire précédente. Lorsque le nombre d'itérations tend vers l'infini, cette dynamique se résume alors à l'étude du point fixe de l'équation

$$x^* = p_e(1-(1-x^*)^{\rho-1})^{\lambda-1} \tag{2.25}$$

Ce point fixe dépend de la probabilité d'effacement du canal p_e. En faisant cette étude, il est possible de montrer qu'il y a un effet de seuil. Pour une probabilité d'effacement inférieure à un seuil $p_e < p_e^*$, le point fixe vaut 0 et par conséquent le code LDPC corrige tous les effacements. Par contre si $p_e > p_e^*$ alors le point fixe n'est pas nul et correspond à la proportion minimum de messages non corrigibles.

L'hypothèse d'indépendance des messages utilisée précédemment est très forte. Cependant, il est possible de montrer [62] qu'elle se révèle exacte si la longueur du graphe LDPC tend vers l'infini. En effet, dans ce cas le graphe tend vers un graphe sans cycle, la proportion $p_v^{(l)}$ tend vers x^* et l'effet de seuil existe.

Remarquons que Gallager avait déjà introduit une telle analyse pour le canal binaire symétrique dans [66].

5.2 Évolution de Densité

L'exemple précédent contient la majorité des idées de l'évolution de densité dont l'article de référence est [62] :

- la densité de la loi de probabilité des messages constitue la variable suivie par l'analyse ;
- la longueur du code LDPC justifie la concentration des comportements vers le comportement typique ;
- la présence d'un comportement à seuil sous réserve de symétrie des densités.

Ces résultats nécessitent toutefois que le canal soit sans mémoire, binaire et symétrique. De plus l'algorithme de décodage n'est pas restreint à la propagation de croyance. Il faut par contre qu'il respecte certaines symétries pour que les densités soient analysables.

Enfin une condition de stabilité sur les paramètres du code et ceux du canal permettent de justifier de la convergence de l'algorithme de décodage.

L'évolution de densité n'est pas simple à mettre en œuvre numériquement car elle nécessite de manipuler des fonctions (les densités des messages) et non des nombres. Il faut donc représenter ces fonctions, pouvoir les intégrer, etc. selon l'algorithme de décodage.

Le livre [60] donne une méthode approchée pour cette mise en œuvre. Nous avons préféré appliquer une méthode de Monte-Carlo pour ce faire. Suivre l'évolution de la densité des messages revient ici à

suivre son histogramme. La matrice de parité du code LDPC est simulée de telle sorte qu'elle paraisse de taille infinie : sa construction est dynamique et aléatoire.

5.3 Approximation gaussienne et EXIT charts

L'évolution de densité est un outil performant mais il demande une mise en œuvre difficile et reste par conséquent théorique. Il est utilisable si le nombre d'étude est restreint mais devient impraticable pour faire du *design* de code (cf. ci-dessous) de par sa complexité.

Il devient par conséquent nécessaire d'approcher l'évolution de densité. Il suffit de la simplifier en faisant l'analyse non pas de la densité complète mais d'un paramètre qui la résume. Par exemple, si les densités de probabilité des messages sont approchées par une loi normale, alors il suffit de suivre sa moyenne et sa variance. Cette approche est l'objet de l'article [94] sur l'approximation gaussienne. Or, de par la nécessaire symétrie des algorithmes, il suffit de suivre soit la moyenne, soit la variance car les deux sont dépendantes l'une de l'autre. Le théorème central limite justifie l'utilisation de la loi normale même si elle n'approche pas correctement les densités des premiers messages échangés pendant l'algorithme de décodage.

Ten Brink a développé dans [95] un type d'analyse similaire dans l'esprit de l'approximation gaussienne pour les turbo-codes. Cette analyse est transcriptible aux codes LDPC. Elle consiste à suivre l'information extrinsèque, c'est-à-dire l'information mutuelle entre les messages et leur bit. Cette information mutuelle doit converger vers 1 pour indiquer un décodage parfait. Il est possible de relier ensuite cette information mutuelle à la probabilité d'erreur.

Pour être plus précis, les variables et les parités sont considérées comme deux décodeurs qui s'échangent de l'information qui est analysée par un scalaire : l'information mutuelle. Cet échange d'information justifie le nom d'EXIT chart : EXtrinsic Information Transfert chart.

5.4 Utilisation des analyses asymptotiques

L'analyse asymptotique est le plus souvent utilisée pour optimiser les paramètres des codes LDPC. En effet, en jouant sur les paramètres λ_i et ρ_j, l'optimisation cherche à maximiser le seuil. Il s'agit donc de trouver les paramètres qui permettent un décodage sans erreur sur le pire des canaux envisagés pour un code de longueur infini. Cette méthode est souvent appelée Design de code.

Cette approche a été utilisée avec succès dans [61] pour trouver des codes LDPC qui approche à moins de 0,06dB la capacité de Shannon pour un canal additif blanc et gaussien.

Naturellement, d'autres paramètres sont modifiables selon le type de construction de code étudié. L'article [81] fait la revue de cette utilisation des analyses asymptotiques.

Dans les travaux présentés ici, l'analyse asymptotique ne sera pas utilisée pour optimiser de code mais pour comparer les performances du calcul des *a priori* issus du canal. En effet, celui-ci influence la densité des messages provenant du canal et par conséquent modifie le seuil des codes LDPC.

6 Les paramètres d'entrer du démappeur LDPC : les LLR

Les algorithmes de décodage ainsi que les outils d'analyse des performances asymptotiques des codes LDPC ont besoin de connaître les valeurs des logarithmes de rapport de vraisemblance (LLR) des bits reçus. Ces LLR sont calculés numériquement à partir des densités de probabilités des signaux à l'entrée du démappeur LDPC ou à la sortie du canal de transmission (1.50).

Le canal considéré tout au long de ce travail de thèse est un canal AISαSN (sec.7.1 chap.1) dont le bruit suit une distribution SαS. Cette distribution (1.22), ne permet malheureusement pas d'obtenir une expression explicite du LLR sauf dans les cas particuliers de Gauss (1.27) et de Cauchy (1.28).

Nous avons utilisé ces densités particulières afin de définir des démappeurs LDPC (Gauss et Cauchy) décrites par les LLR suivants :

$$LLR_{\text{Gauss}} = \frac{2Y}{\sigma^2} \tag{2.34}$$

$$LLR_{Cauchy} = Ln\left(\frac{\gamma^2 + (Y_j + 1)^2}{\gamma^2 + (Y_j - 1)^2}\right) \tag{2.35}$$

Chua dans [97], [98], [99] a proposé pour les turbo-codes une approximation de la densité de probabilité qu'il a nommée logistic. Cette densité, basée sur la métrique d'Huber, est donnée par l'équation suivante :

$$f(x, y) = \exp\left(\frac{E_b}{\gamma^2}\rho_{Logistic}.y\right) \tag{2.36}$$

où $\rho_{Logistic} = h^2 \ln(\cosh(\frac{y}{h}))$ est la fonction pénalité et h est un paramètre de non arrêt qui est généralement mis à 1 ou 2. Dans la suite de nos travaux, nous avons choisi de fixer ce paramètre à 2.

Le LLR dans ce cas est calculable et est de la forme :

$$LLR_{Logistic} = \frac{h^2}{\gamma^2}\left[Ln\left(\cosh\left(\frac{Y+1}{h}\right)\right) - Ln\left(\cosh\left(\frac{Y-1}{h}\right)\right)\right] \tag{2.37}$$

Les graphes des fonctions LLR spécifiques précédemment présentées sont tracés sur la figure 2.11.

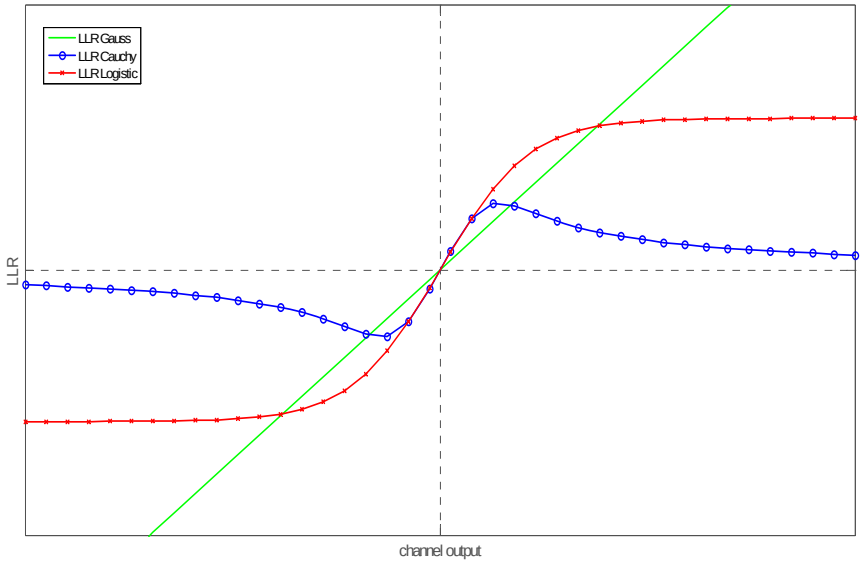


Fig. 2.11 : Différents LLR en fonction de sortie du canal

A partir de ces résultats nous pouvons déduire que:

- le démappeur de Gauss qui est une fonction linéaire, donne aux grandes valeurs de ***y*** une grande importance en le multipliant par sa pente ;
- le démappeur de Cauchy limite l'impacte des grandes valeurs de ***y*** en les affaiblissant pour finir asymptotiquement par les annuler. Ce démappeur de comporte comme le démappeur linéaire pour des faibles valeurs de ***y*** ;
- Le démappeur Logistic se comporte de la même manière que le démappeur de Cauchy pour les faibles valeurs de ***y*** et limite les grandes valeurs de ***y*** en les seuillant.

Ces différents démappeurs qui présentent des LLR prédéfinis seront utilisés le long de ce travail de thèse afin de montrer le comportement des codes LDPC dans les environnements AISαSN.

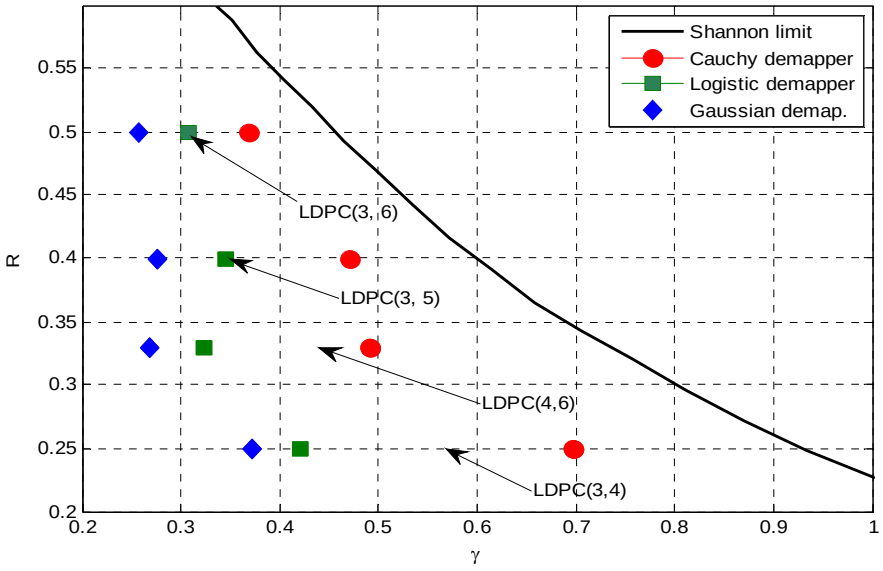
7 Performances asymptotiques des codes LDPC dans les environnements AISαSN

Afin de montrer le comportement des codes LDPC dans les environnements AISαSN nous avons proposé d'analyser ces performances asymptotiques en utilisant l'évolution de densité. En effet, cet outil permet de montrer le comportement des performances asymptotiques des codes LDPC en les comparants aux capacités des environnements AISαSN.

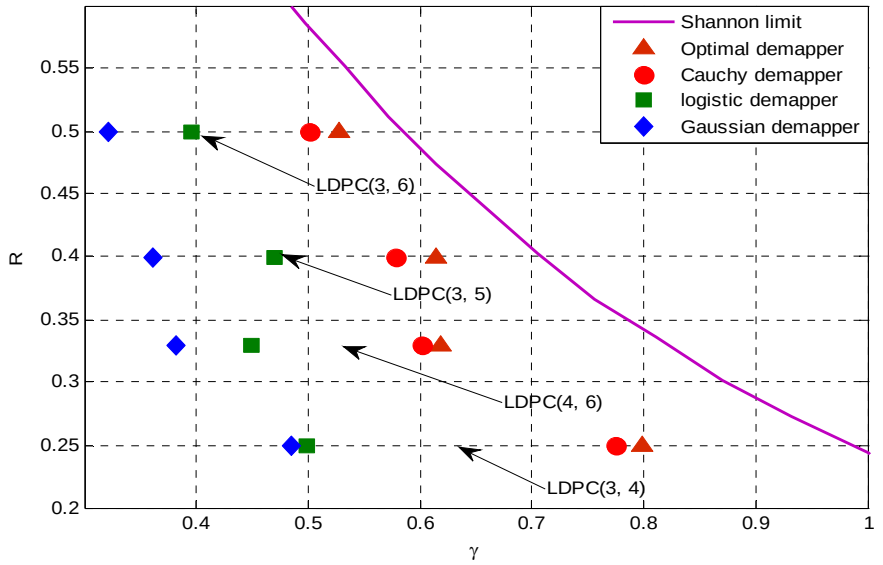
Dans ce contexte, nous avons utilisé les démappeurs présentés à la section précédente ainsi que des codes LDPC réguliers de différents rendements. Le calcul des seuils de convergence par évolution de densité permet de comparer les performances asymptotiques et la capacité des canaux des différents environnements AISαSN.

L'objectif est de montrer les effets du choix du démappeur ainsi celui du type de codes LDPC sur le comportement des performances asymptotiques dans des environnements AISαSN. La figure 2.12 présente les résultats trouvés par évolution de densité pour différents paramètres α du canal AISαSN (α = 1, 1.5 et 2).

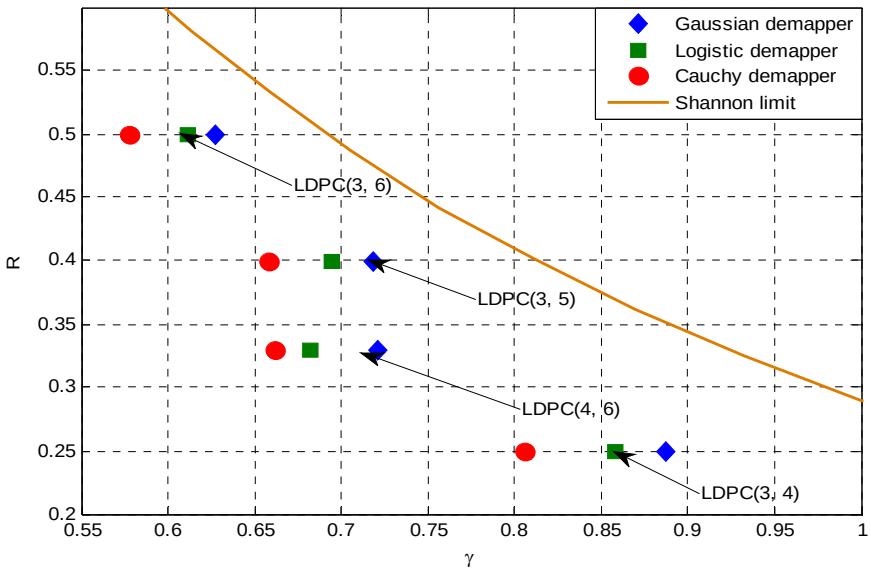
Nous retrouvons en abscisse le seuil, *i.e.* la dispersion maximale γ pour laquelle une transmission sans erreur est possible asymptotiquement, pour différents rendements de codes LDPC et pour différents démappeurs.



(a)



(b)



(c)

Fig.2.12 : Performances asymptotiques des différents types de codes LDPC utilisant différents demappeurs dans les environnements AISαSN (a) α=1, (b) α=1.5, (c) α=2

A partir de ces résultats nous remarquons que

- le démappeur de Gauss avec la forme simple de son LLR est optimal dans le cas d'un environnement AWGN en présentant des performances asymptotiques les plus proches de la capacité de cet environnement. Ses performances se dégradent rapidement et fortement quand l'environnement devient impulsif ($\alpha < 2$) ;

- le démappeur de Cauchy s'adapte bien aux environnements impulsifs en présentant des performances asymptotiques proches de la capacité de ces environnements. Il devient cependant mauvais dans l'environnement AWGN ;

- le démappeur Logistic présente des performances asymptotiques intermédiaires entre celles de Gauss et de Cauchy dans tous les environnements AISαSN ;

- les codes LDPC régulier de type (3, 6) de rendement ½ présentent le moins d'écart avec la capacité. Les performances asymptotiques des différents démappeurs se dégradent quand le rendement des codes diminue et lorsque cet environnement devient impulsif.

En conclusion, les performances asymptotiques des codes LDPC sur un canal AISαSN sont très sensibles à la nature du démappeur, donc de la représentation de l'entrée du décodeur. De plus cette sensibilité dépend du type et du rendement du code LDPC.

8 Conclusion

Au cours de ce chapitre nous avons présenté les codes LDPC en décrivant différents algorithmes de décodages, les outils d'analyse de leurs performances asymptotiques ainsi que différents démappeurs. Nous avons alors étudié le comportement des performances asymptotiques de ces codes dans des environnements AISαSN et les effets de la nature du démappeur.

Dans ce cas nous avons comparé les performances asymptotiques aux capacités des canaux AISαSN. Nous en concluons que les codes LDPC sont très sensibles à la nature du démappeur.

Aucun démappeurs, parmi les différents proposés, n'est universel pour les canaux AISαSN, c'est-à-dire qui s'adaptent facilement à tous les environnements considérés.

Ceci nous amène à étudier dans le prochain chapitre d'autres types de démappeurs très simples et suffisamment robustes vis-à-vis des canaux impulsifs.

Performances des codes LDPC dans un environnement AISαSN

1 Introduction
2 Codes LDPC dans les environnements AISαSN

2.1 Cas du démappeur optimal
2.2 Cas des démappeurs basés sur des expressions analytiques des LLR

3 Démappeurs robustes en environnement α-stable

3.1 Le démappeur Linéaire
3.2 Le démappeur de Clipping

3.2.1 Études basées sur les LLR
3.2.2 Études basées sur la puissance de bruit après clipping
3.2.3 Étude basée l'Évolution Densité (ED)
3.2.4 Bilan des différentes études.
3.2.5 Sensibilité à une mauvaise estimation de α.

3.3 Hole Puncher
3.4 Forme approchée du démappeur

4 Décodage Min-Sum

4.1 Cas des démappeurs optimal et basés sur des exprésions analytiques des LLR
4.2 Cas du démappeur de Clipping

5 Conclusion

1 Introduction

Combinant les modèles α-stables présentés dans le chapitre 1 et les codes LDPC présentés dans le chapitre 2, nous allons étudier maintenant la robustesse des codes LDPC en présence d'un bruit impulsif. En particulier nous avons vu dans le chapitre 2 qu'un gain important pouvait être obtenu en adaptant le calcul du logarithme de vraisemblance — nous appellerons l'opération permettant de passer de la valeur reçue à sa vraisemblance le démappeur — à l'entrée de l'algorithme de propagation de croyance. C'est pourquoi nous étudierons essentiellement cet aspect. Nous évaluerons les performances en utilisant différents démappeurs et étudierons plus en détail un dépmappeur de clipping en essayant de trouver des opérations simples pour définir le seuil et la pente nécessaires à sa définition.

Dans la première partie de ce chapitre, nous montrons que les codes LDPC offrent toujours d'excellentes performances dans un environnement impulsif, du moins quand le démappeur utilisé est le démappeur optimal, c'est à dire le démappeur utilisant la densité de probabilité exacte du bruit. Le fait de ne pas avoir d'expression analytique de cette densité complexifie le calcul mais ne nous interdit pas de le faire à partir des fonctions caractéristiques. Nous montrons ensuite que les performances se dégradent très significativement si le démappeur linéaire (faisant l'hypothèse d'un bruit gaussien) est utilisé. Si nous voulons utiliser une expression analytique, il semble alors plus opportun d'utiliser la distribution de Cauchy qui présente une robustesse très proche de l'optimale pour un large éventail de valeurs de α, bien qu'assez logiquement il perde cette robustesse quand α s'approche de 2. Nous comparons également ces approches au démappeur dit logistique qui a été proposé dans [98].

Parmi les différents démappeurs proposés, il n'en existe cependant pas qui soient à la fois simples et robustes quelque soit la valeur de α. Ceci nous ramènera dans la deuxième partie de ce chapitre à reprendre l'étude du démappeur linéaire puis à introduire une opération non linéaire de clipping pour essayer de nous approcher des performances du démappeur utilisant la vraie densité de probabilité. Ce dernier démappeur, que nous appellerons démappeur de clipping, permet de limiter l'impact des grandes valeurs dues aux phénomènes impulsifs. La difficulté se résume alors à choisir deux paramètres : la pente de la partie linéaire et le seuil à partir duquel le signal est écrêté. Les performances des codes LDPC s'avèrent très sensibles à ces paramètres. N'ayant pas de solution analytique pour déterminer ces paramètres, nous proposons plusieurs solutions afin de les obtenir en fonction des paramètres de l'environnement, plus précisément les paramètres de l'interférence α-stable, c'est à dire l'exposant caractéristique α et la dispersion γ. Trois approches sont proposées. L'une utilise le LLR obtenu avec la vrai densité de probabilité et utilise la pente en 0 et le maximum de cette fonction. Il s'avère que le maximum obtenu est trop important et ne permet pas d'obtenir de bonnes performances. L'évolution de densité permet d'améliorer le choix du seuil. Une seconde approche est basée sur la minimisation de la puissance de bruit après l'écrêtage. Enfin, la troisième approche se base exclusivement sur l'évolution de densité et permet, en particulier, d'obtenir le seuil et la pente en fonction du seul paramètre α.

Dans la suite de ce chapitre nous étudions d'autres démappeurs simples à mettre en œuvre : le hole puncher qui annule les valeurs reçues au delà d'un seuil et un démappeur approché qui mélange le clipping et le hole puncher. Cette dernière approche s'avère apporter un bon compromis performance / complexité.

Enfin, nous étudions également une autre solution pour réduire le nombre de paramètre à définir : l'algorithme Min-Sum au lieu de la propagation de croyance. En effet, cet algorithme n'est plus sensible à la pente et seul le seuil est alors à déterminer

2 Codes LDPC dans les environnements AISαSN

Dans cette partie, les codes LDPC réguliers de types (3, 6) sont déployés dans la chaine de transmission (sec.8 chap.1). L'objectif est d'étudier les performances de ces codes dans les environnements AISαSN. Dans un premier temps nous étudions le comportement des LDPC utilisant le démappeur optimal au sens du maximum de vraisemblance. Calculé numériquement à partir de la fonction caractéristique du bruit, ce démappeur présente les performances asymptotiques les plus proches de la capacité des environnements AISαSN (sec.7 chap.2) et servira de référence pour la suite des travaux. Cependant, l'absence d'une expression analytique de la densité de probabilité donne à ce démappeur une importante complexité. Dans une deuxième étape, nous testons d'autres démappeurs ayant une expression analytique afin de réduire la complexité tout en conservant des performances proches de l'optimal. Une première idée est de se baser sur des distributions α-stables dont la densité de probabilité possède une expression analytique. C'est en particulier le cas pour $\alpha = 1$ (Cauchy) et $\alpha = 2$ (Gauss). Le démappeur de Gauss (linéaire) avec la forme simple de son LLR, offre peu de robustesse dans le cas des environnements impulsifs. Le démappeur de Cauchy, sauf pour $\alpha = 2$, s'avère bien plus robuste. Ceci a déjà été suggéré dans de nombreux travaux en détection [101] ou en transmissions impulsionnelles [53]. Enfin, nous testons également le démappeur logistic proposé par Chua pour des turbo-codes [98]. Il s'avère être un compromis entre les démappeurs de Gauss et de Cauchy.

2.1 Cas du démappeur optimal

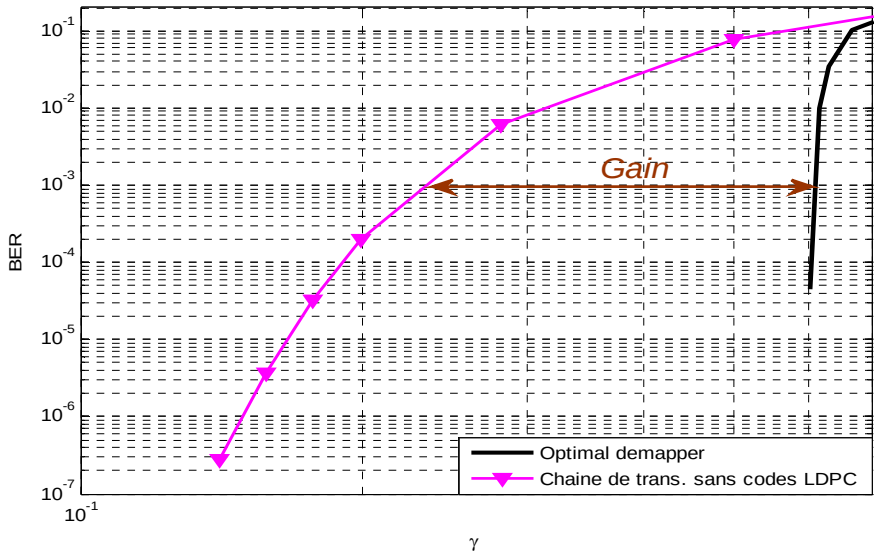
Nous avons dans un premier temps déployé le démappeur optimal au sens du maximum de vraisemblance dont l'expression a été donnée dans le chapitre 1. Nous la rappelons ici pour la clarté de l'exposé :

$$m = LLR = \log \frac{\mathrm{P}_{Y/X}(y/1)}{\mathrm{P}_{Y/X}(y/-1)} = \log \frac{f(y)}{f(-y)}$$

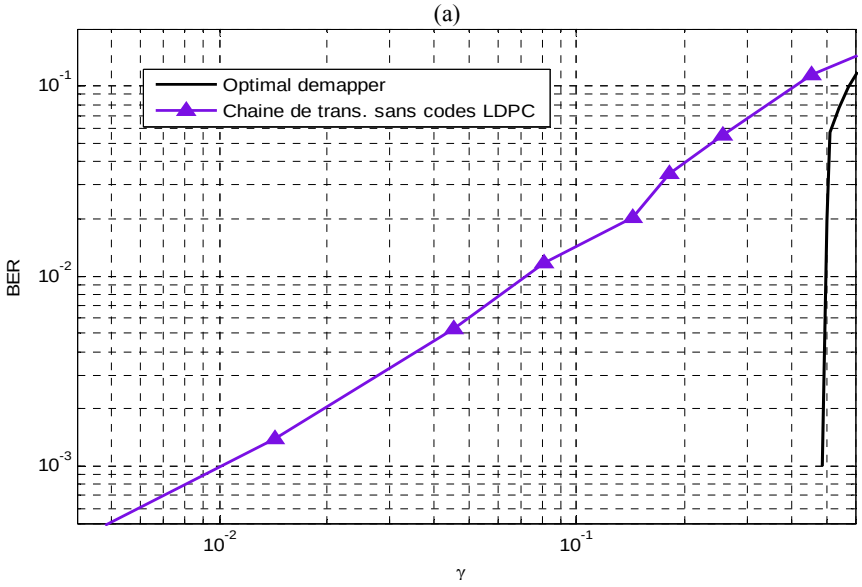
Dans le cas général, ce démappeur ne peut pas être écrit explicitement. Il est alors nécessaire d'avoir recours à un calcul numérique de la transformée de Fourier inverse de la fonction caractéristique $\phi(t)$ de la variable aléatoire SαS qui modélise le processus impulsif.

Cette première étude nous permet de constater l'efficacité des codes LDPC dans un environnement α-stable. Ils apportent un gain considérable aux performances de la chaine de transmission sans codage dans tous les environnements AISαSN (figure.3.1).

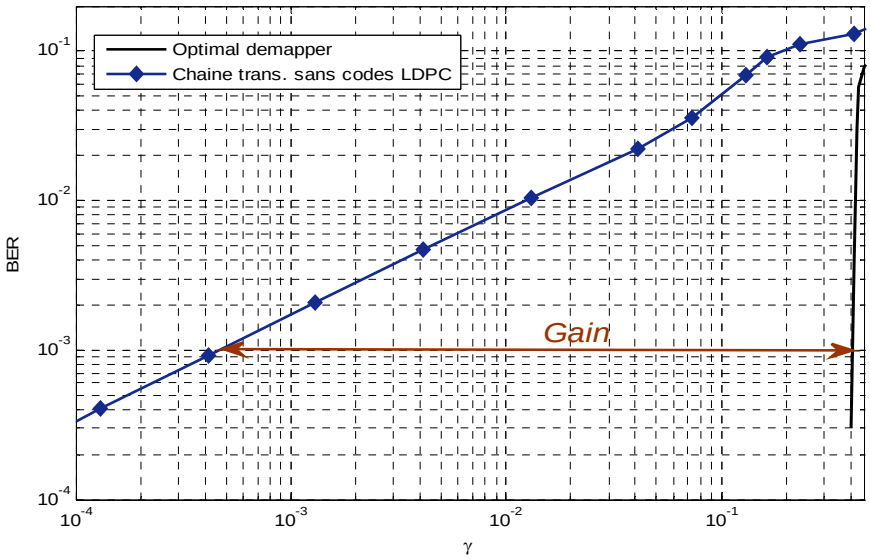
Il faut toutefois prendre des précautions quant aux gains affichés sur la figure 3.1. En effet, l'abscisse donne le facteur d'échelle du bruit du canal de communication et non un rapport signal sur bruit. Par conséquent, ce n'est pas équivalent à une représentation en fonction de E_b/N_0. De plus il n'est pas possible de faire une comparaison de ces gains pour les différents α car les facteurs d'échelle ne sont pas comparables.



(a)



(b)



(c)

Fig.3.1 : Performance de la chaine de transmission avec et sans codes LDPC dans différents environnements AISαSN(a)α = 2, (b) α = 1.5, (c)α = 1.2

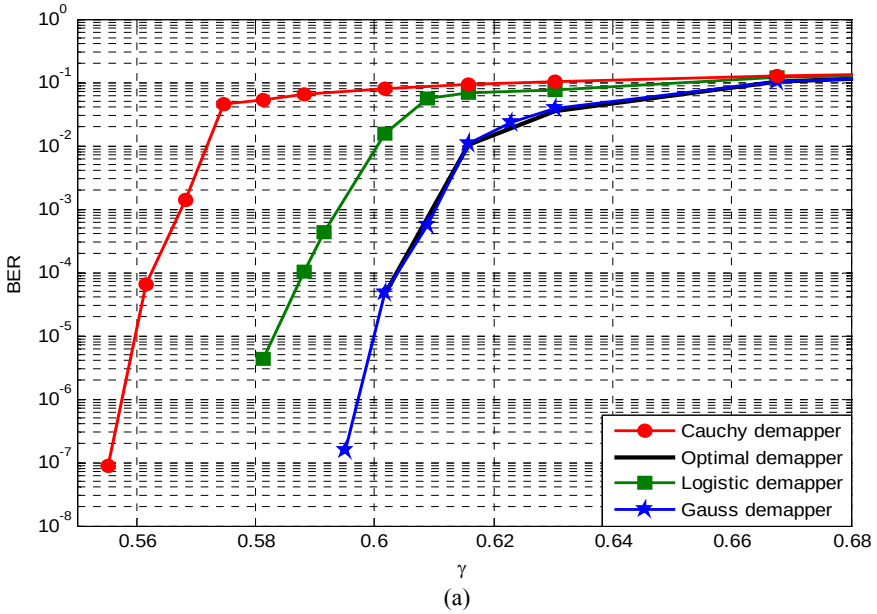
2.2 Cas des démappeurs basés sur des expressions analytiques des LLR

Nous nous posons la question de savoir s'il ne serait pas possible d'utiliser des expressions analytiques qui donneraient des performances proches de l'optimal. Pour cela nous travaillons avec les densités connues pour deux valeurs de α : $\alpha = 2$, la distribution gaussienne et $\alpha = 1$, la distribution de Cauchy (voir les équations (2.34) et (2.35)). Nous testons également le démappeur dit Logistic (équation 2.37). Nous évitions ainsi le calcul numérique nécessaire dans le cas du démappeur optimal. Chacun de ces démappeurs présente des avantages et des inconvénients et leurs performances sont représentées sur la figure 3.2.

- Le démappeur de Gauss présente une forme simple du LLR (fonction linéaire) qui ne dépend que de la dispersion γ. Il est bien évidemment, optimal dans le cas d'un environnement AWGN mais ses performances se dégradent rapidement en comparaison du démappeur optimal dés que l'environnement devient impulsif ($\alpha < 2$) ;

- Le démappeur de Cauchy s'adapte bien aux environnements impulsifs AISαSN ($\alpha < 2$) en présentant des performances proches de l'optimal quelle que soit la valeur de α. Il se dégrade cependant légèrement quand α se rapproche de 2. Il reste également relativement complexe à implémenter ;

- Le démappeur logistic présente l'avantage d'être plus robuste que le démappeur de Cauchy dans le cas gaussien ($\alpha = 2$) mais reste moins bon dans les autres cas d'environnements AISαSN en présentant des performances intermédiaires entres celles de Gauss et de Cauchy. Il reste également relativement complexe à implémenter.



(a)

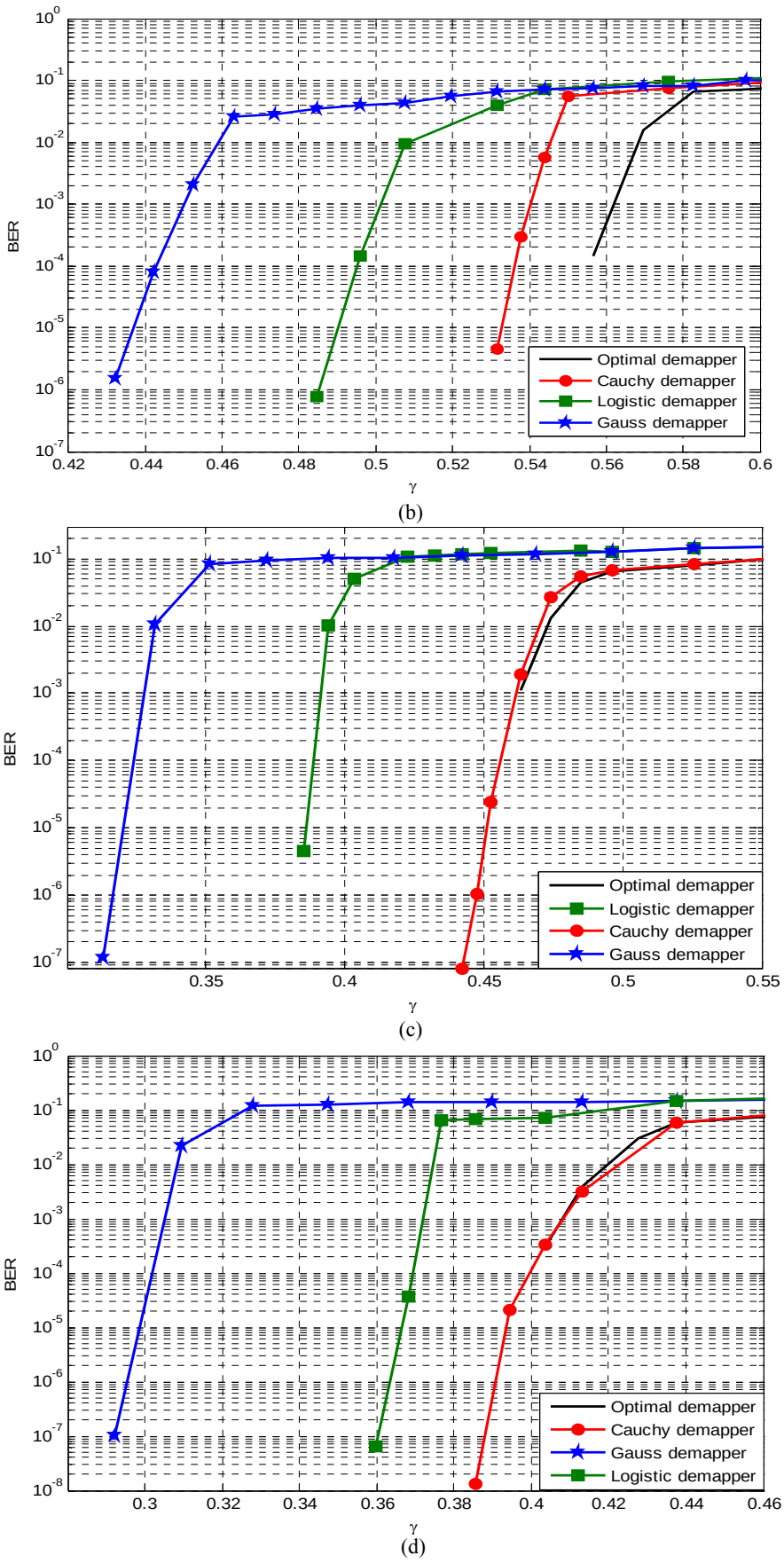


Fig.3.2 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=2, (b) α=1.8, (c) α=1.5, (d) α=1.2

A partir de cette étude, nous concluons que le démappeur de Gauss est inefficace dans un environnement impulsif. L'approche la plus pertinente est d'utiliser l'expression de Cauchy qui offre des performances toujours assez proches de l'optimal avec malgré tout une légère dégradation (0.6 dB pour un TEB = 10^{-6}) quand l'environnement devient AWGN.

Les différentes études proposées précédemment ne permettent cependant pas d'obtenir un démappeur simple et robuste qui s'adapte facilement aux environnements impulsifs. Ceci nous ramène à étudier des solutions plus simples à implémenter mais qui permettent de conserver de bonnes performances tout en s'adaptant au contexte.

3 Démappeurs robustes en environnement α-stable

Nous cherchons des démappeurs robustes, qui s'adaptent à tous les environnements et dont la mise en œuvre est facile. L'environnement est ici caractérisé par les paramètres α et γ. Dans un premier temps nous avons regardé s'il était possible d'avoir un démappeur *linéaire* performant en adaptant sa pente. Nous montrons que ce n'est pas possible. Il semble alors important de réduire l'impact des grandes valeurs et donc de seuiller les valeurs importantes. Nous obtenons alors le démappeur de *clipping* et deux paramètres sont à définir : la pente de la partie linéaire et le seuil. Nous n'avons pas trouvé de méthode analytique pour calculer les meilleurs paramètres possibles. Nous avons donc testé plusieurs approches en utilisant les expressions exactes des LLR, la puissance du bruit après clipping ou l'évolution de densité. Cette dernière approche est robuste et offre l'avantage de ne faire dépendre les paramètres recherchés que de la valeur α.

Nous introduisons ensuite le démappeur *Hole Puncher* qui ne se contente pas de seuiller les grandes valeurs mais les annule pour en éliminer l'impact dans la prise de décision. Cette approche ne s'avère pas performante. Nous proposons alors un démappeur approchant un peu mieux la courbe optimale du LLR. Il peut être vu comme un compromis entre le *clipping* et le *Hole Puncher*.

3.1 Le démappeur Linéaire

Dans un premier temps nous utilisons simplement un démappeur linéaire. Il est représenté sur la figure 3.5 avec d'autres démappeurs que nous présentons par la suite. Son LLR est décrit par l'expression suivante :

$$LLR_{Linear} = P.y \quad \forall \quad y \in R \tag{3.1}$$

où P est la pente et *y* est la sortie du canal (l'entrée du démappeur).

Ce démappeur a l'avantage d'une grande simplicité de mise en œuvre. Nous proposons de déterminer la pente *P* en fonction de α par évolution densité. Les valeurs telles que les codes LDPC présentent les meilleures performances asymptotiques seront tabulées. Le P résultant ne dépend que de α, ce qui n'est pas le cas du démappeur gaussien où P dépend de la dispersion γ (la variance). Nous pouvons donc craindre une certaine dégradation en utilisant cette approche.

La figure 3.3 présente les résultats de l'évolution de densité pour différents α (différents environnements AISαSN). Nous retrouvons en ordonnée la dispersion maximale pour laquelle une transmission sans erreur est possible asymptotiquement. Nous considérons alors que la valeur de *P* où la courbe atteint son maximum donne la pente que nous utiliserons dans nos simulations.

Nous avons également comparé la courbe des performances dans le cas où α = 2 avec la limite des performances asymptotiques des codes LDPC de type (3, 6) utilisant l'algorithme de décodage BP donnée par Richardson et Urbanke dans [62] pour un canal AWGN, performances que nous retrouvons dans nos figures.

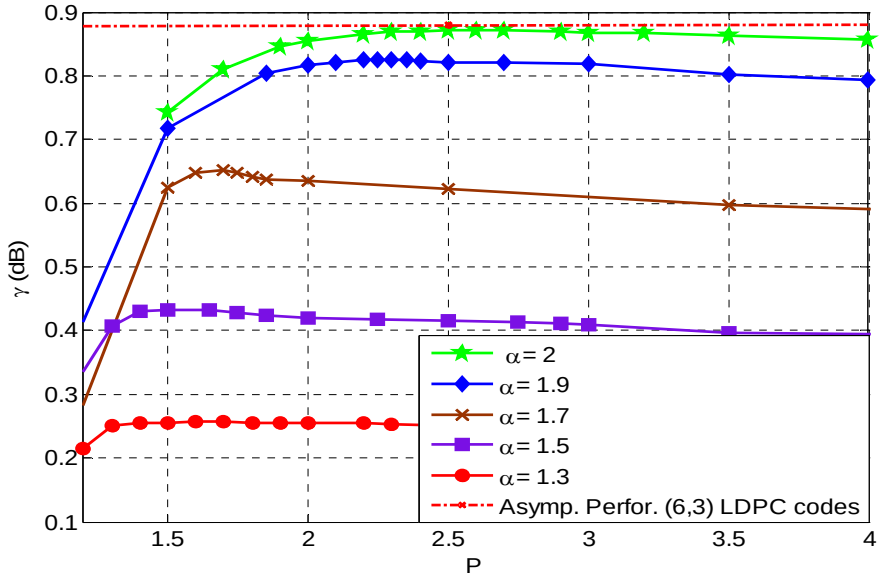
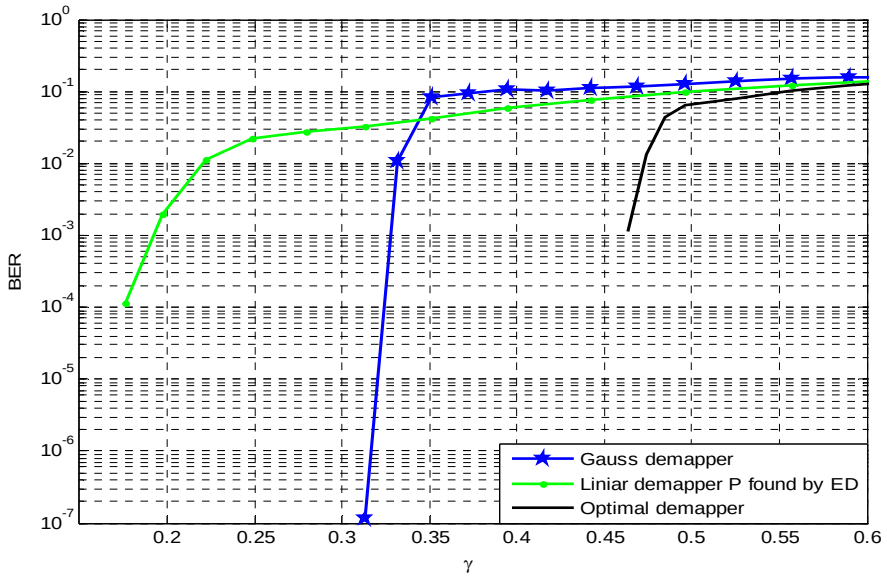
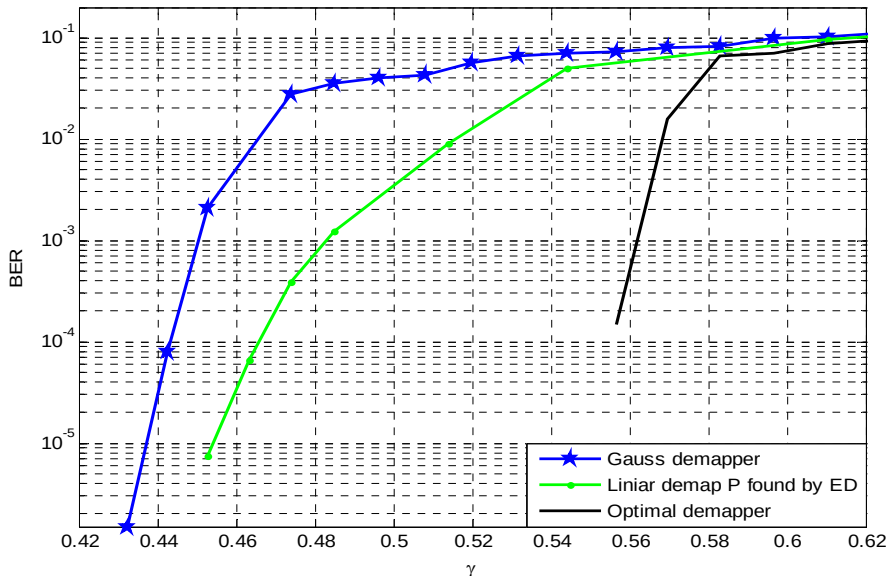


Fig.3.3 : Variation des performances asymptotiques des codes LDPC en fonction de variation de la pente pour différents α

Le *P* ainsi tabulé sera utilisé par le démappeur afin d'étudier ses performances dans les environnements AISαSN. Ces études sont représentées sur la figure 3.4 pour deux valeurs différentes de α (1.4 et 1.8).



(a)



(b)

Fig.3.4 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=1.4, (b) α=1.8

A partir de ces résultats nous concluons que les codes LDPC utilisant un démappeur linéaire avec les valeurs tabulées de P obtenues par évolution de densité donnent de mauvaises performances. En fait, une amélioration serait peut-être possible en modifiant la pente (en la choisissant par exemple en fonction de la dispersion ce que fait le démappeur gaussien qui donne de meileures performances pour $\alpha = 1.8$) mais il est surtout important de réduire l'impact des grandes valeurs de bruit qui masquent complètement l'information. Or un démappeur linéaire donne à ces valeurs une grande importance et, surtout, laisse croire qu'elles sont très fiables alors, qu'en réalité, elles le sont peu. Une première solution pour limiter ce problème est de seuiller les valeurs à partir d'un certain point. Ceci représente le but de notre prochaine étude.

3.2 Le démappeur de Clipping

Contrairement au démappeur linéaire, ce démappeur apporte une solution au problème de l'impulsivité du bruit en limitant l'impact des grandes valeurs. Pour cela elle ramène à la valeur d'un seuil toutes les valeurs qui le dépassent. Ce démappeur est nommé clipping (représenté sur la figure 3.5) et est décrit par l'expression :

$$LLR_{Clipping} = \begin{cases} +H & quand \quad y > H/P \\ P.y & quand \quad -H/P < y < H/P \\ -H & ailleur \end{cases} \tag{3.2}$$

Ce démappeur dépend essentiellement de 2 paramètres : la pente P et seuil de cliping H.

Bien que ce type de démappeur soit largement répandu, peu de travaux sur le choix de ses paramètres ont été présentés. Ils s'avèrent pourtant être assez critiques. N'ayant pas de solution analytique optimale pour les déterminer, nous étudions plusieurs approches afin de les obtenir en fonction des paramètres de l'environnement. Trois études sont proposées. Dans un premier temps nous nous basons

sur les allures des LLR (optimal ou de Cauchy). Dans un second temps, nous étudions la puissance de bruit après seuillage. Enfin, la troisième approche se base sur l'évolution de densité.

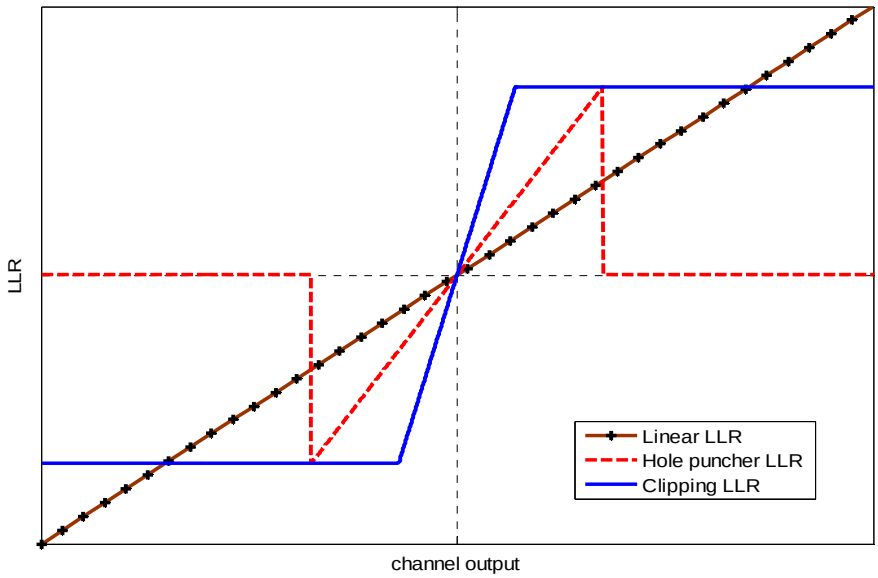


Fig. 3.5 : Comportement des différents démappeurs en fonction de la variation du sortie de canal y.

3.2.1 Études basées sur les LLR

Cette approche détermine les paramètres P et H de telle sorte que la fonction de *demapping* se rapproche des allures des LLR exacts ou de Cauchy. Si la valeur de pente ainsi obtenue semble judicieuse, ce n'est pas le cas du seuil. Nous essayons donc d'améliorer le choix de ce seuil.

a) ***Pente en zéro et maximum du démappeur optimal***

Dans un premier temps, nous choisissons P et H à partir du LLR optimal, comme cela est décrit sur la figure 3.6.

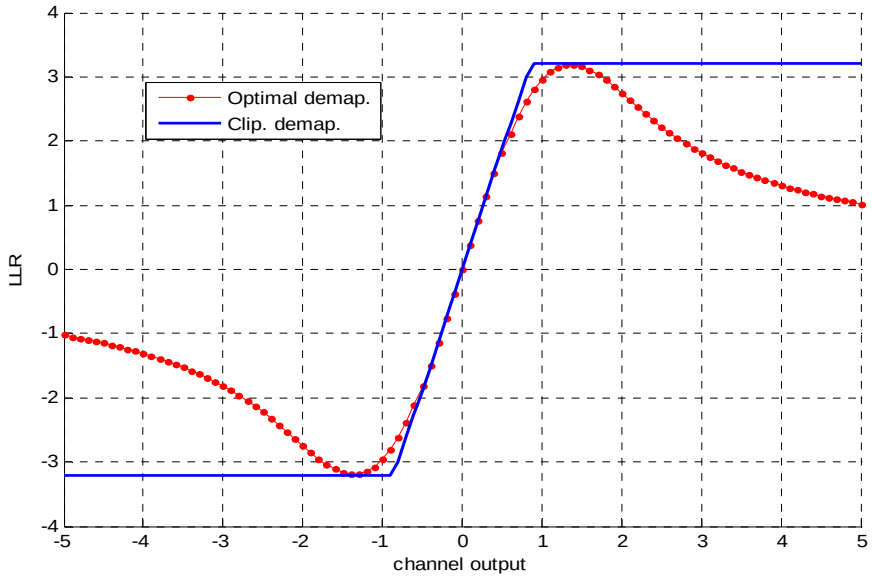


Fig. 3.6 : comportement des démappeurs de clipping et optimal en fonction de la variation du sortie de canal y.

La pente P représente la tangente en zéro du LLR optimal (1.50). Elle est donc la dérivée en zéro du LLR et a pour expression :

$$P = \frac{d(LLR)}{dy}\bigg|_{y=0} = \frac{d}{dy}\ln\left[\frac{f(y-x)}{f(y+x)}\right]_{y=0}$$
$$= \frac{d}{dy}\ln(f(y-x))_{y=0} - \frac{d}{dy}\ln(f(y+x))_{y=0} \qquad (3.3)$$
$$= \frac{f'(-x)}{f(x)} - \frac{f'(x)}{f(x)}$$

Où ***x*** est le symbole *BPSK* transmis (+1 ou -1).

Un calcul numérique permet d'obtenir P. La densité de probabilité de la distribution SαS est la transformée de Fourier inverse de la fonction caractéristique donnée par (1.22). Sa dérivée peut alors s'écrire :

$$f'(x) = \frac{1}{\pi}\int_0^{+\infty} \exp(-|\gamma t|^{\alpha})\sin(xt)dt \qquad (3.4)$$

Puisque la distribution est symétrique alors *f'(x) = -f'(-x)* et l'expression devient :

$$P = \frac{d(LLR)}{dy}\bigg|_{y=0} = 2.\frac{f'(-x)}{f(x)} \qquad (3.5)$$

La pente P dépend de α et γ de la distribution et est présentée par la figure 3.7 pour différentes valeurs de α.

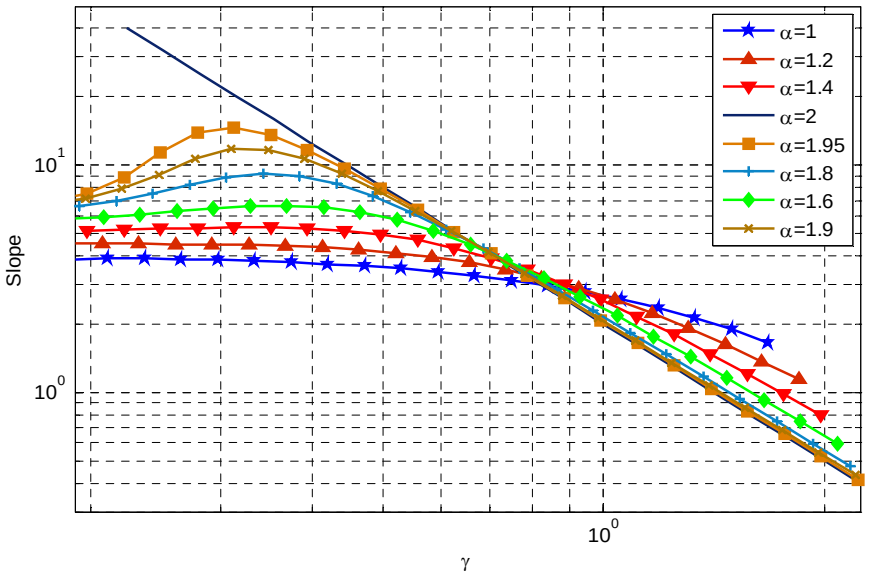


Fig.3.7 : Variation de la pente P en fonction des paramètres α et γ.

Nous remarquons sur la figure 3.7 que la pente croît quand la dispersion γ diminue mais se stabilise puis décroît même dès que α est strictement inférieur à 2. Nous notons également la spécificité du cas gaussien dans l'ensemble des distributions α-stables. Enfin, il apparaît que plus α s'approche de 1, plus

les variations sont faibles. Nous pouvons s'attendre à plus de sensibilité à ce paramètre pour des α proches de 2.

Le seuil de clipping H est aussi déterminé numériquement. Il représente la valeur où le dérivé du LLR optimal s'annule. Nous obtenons cette valeur numériquement par dichotomie pour chaque valeur de α et γ. Les valeurs estimées de H sont représentées sur la figure 3.8.

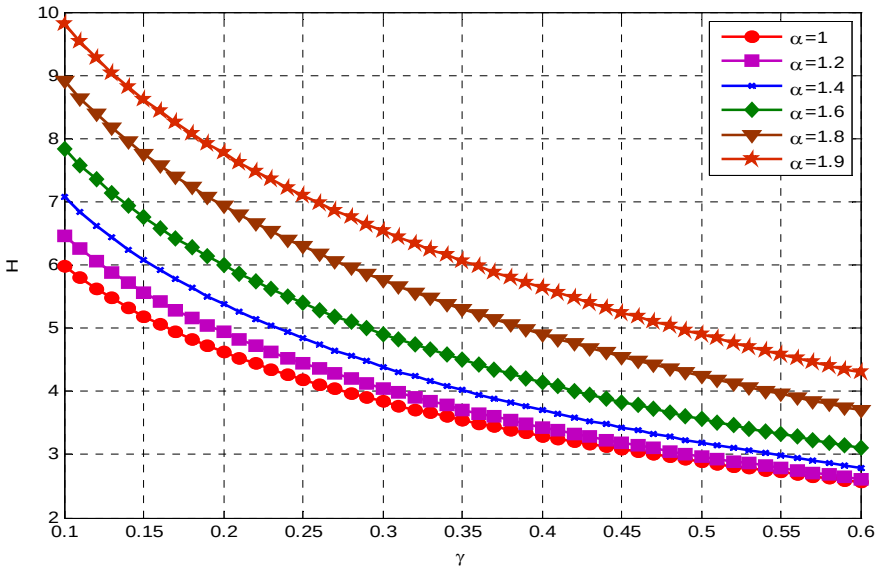


Fig.3.8 : Variation du seuil H en fonction des paramètres α et γ.

Le seuil H augmente quand la dispersion γ diminue ou quand α s'approche de 2. Il devient même infini dans le cas gaussien, ce qui semble logique car le récepteur linéaire (sans seuillage) est alors optimal. Nous représentons sur la figure 3.9 les performances du démappeur ainsi obtenu. Les codes LDPC se comportent relativement bien dans les environnements peu impulsifs ($\alpha \geq 1.8$) mais leurs performances se dégradent quand l'impulsivité augmente ($\alpha < 1.8$). Ceci peut vraisemblablement s'expliquer par les grandes valeurs du seuil de clipping H. En effet, elles ne permettent pas de suffisamment pondérer l'impact des fortes valeurs reçues. Un autre problème est que cette méthode ne permet pas de déterminer de façon analytique les valeurs de H et de P. Plusieurs études ont montré que dans un environnement α-stable, l'utilisation de la loi de Cauchy au lieu de l'expression exacte permettait d'obtenir des résultats analytiques tout en conservant de bonnes performances (comme par exemple en détection). Ceci nous ramène alors à tester d'autres approches. En particulier, déterminer H et P à partir du LLR de Cauchy puis optimiser le seuil par évolution de densité.

b) ***Pente de Cauchy et maximum du démappeur de Cauchy***

Nous avons vu que le clipping est moins performant dans les environnements impulsifs et la méthode utilisée jusqu'à présent se base sur les densités de probabilités du bruit qui n'ont pas d'expression analytique explicite. Bien que ces traitements puissent se faire *off-line* et que nous ne sommes donc pas vraiment concernés par la complexité, il nous semble qu'il pourrait être intéressant de baser notre recherche de H et P sur des expressions analytiques. La solution intuitive qui s'impose est d'utiliser le LLR du démappeur de Cauchy qui a souvent été utilisé avec succès, par exemple pour problèmes de détection dans des bruits avec un α différent de 1. L'utilisation de la loi de Cauchy s'est avérée souvent proche de la solution optimale pour une large gamme de α. Qui plus est, comme nous l'avons vu précédemment, les grandes valeurs de α donnent de grandes valeurs de H et ces grandes valeurs de

H ne sont pas très performantes. Les réduire pourrait être bénéfique. Or c'est pour α = 1 que les valeurs de H obtenues sont le plus faibles (cf. figure 3.8).

Nous avons donc étudié les performances du démappeur de clipping dont la pente et le seuil sont déterminés à partir de la distribution de Cauchy. La pente P représente la tangente en zéro du LLR de Cauchy (2.35). Elle est donc la dérivée en zéro du LLR et a pour expression :

$$P_{Cauchy} = \frac{d(LLR_{Cauchy})}{dy}\Big|_{y=0} = \frac{4}{1+\gamma^2} \qquad (3.6)$$

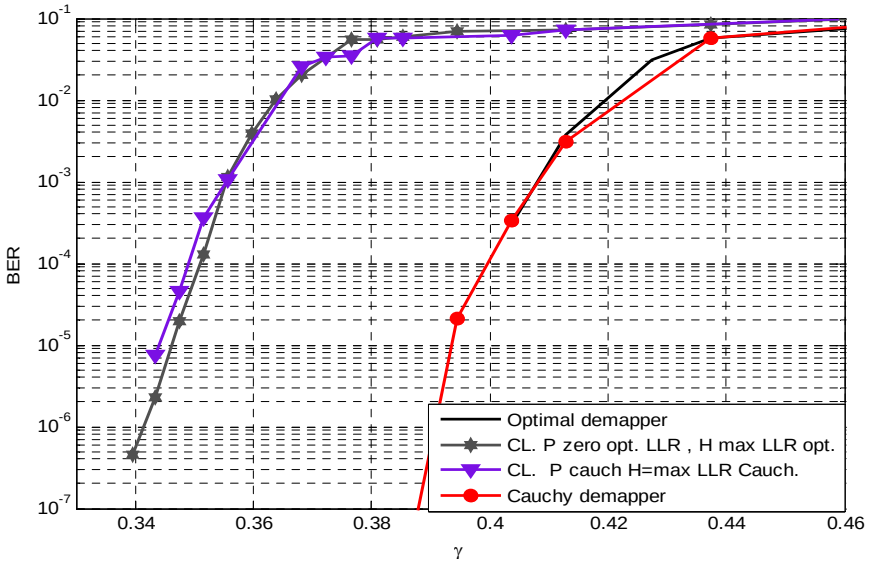
Cette pente dépend de la dispersion γ. Elle est représentée sur la figure (3.7) (LLR exact pour α = 1).

Le seuil de clipping H est également déterminé analytiquement. Il représente la valeur où la dérivée du LLR de Cauchy s'annule et a pour expression :

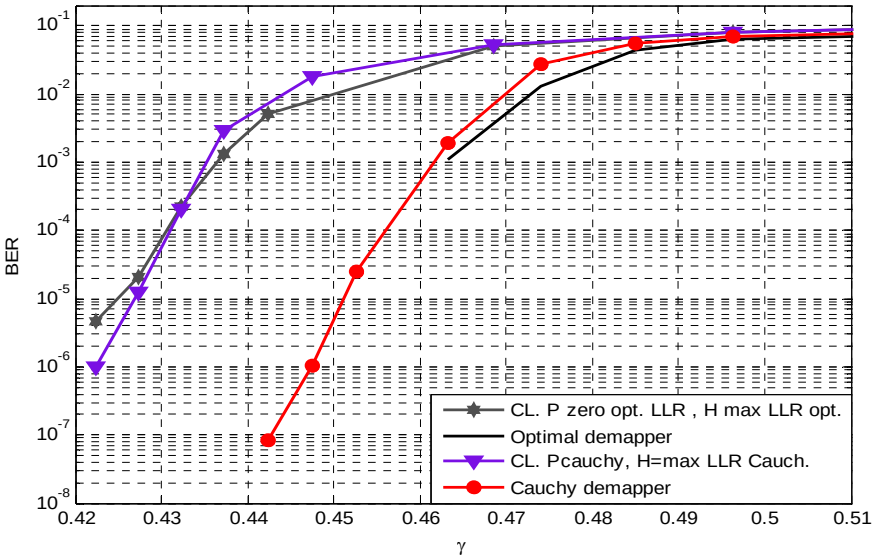
$$H_c = \log\left(\frac{\sqrt{1+\gamma^2}+1}{\sqrt{1+\gamma^2}-1}\right) \qquad (3.7)$$

Ce seuil dépend de la dispersion et nous retrouvons ses valeurs sur la figure 3.8 pour α = 1.

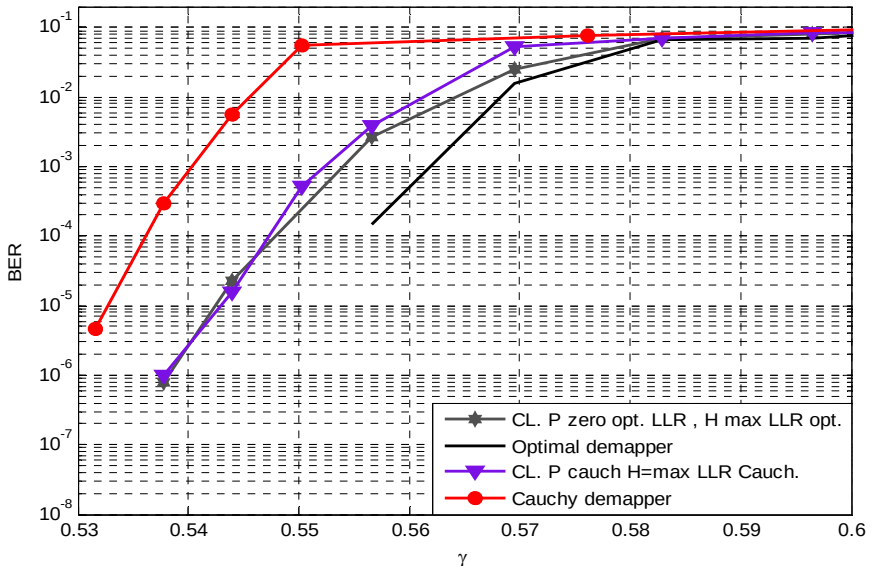
Les performances du démappeur de clipping avec ces paramètres sont présentées par la figure 3.9.



(a)



(b)



(a)

Fig.3.9 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=1.2, (b) α=1.4, (c) α=1.8

Le démappeur de clipping basé sur la distribution de Cauchy se comporte de la même manière que le démappeur de clipping utilisant la pente en zéro du LLR exacte. Ceci peut être expliqué par les valeurs de H et P obtenues dans les deux approches. Le tableau 3.1 donne les valeurs du seuil H et de la pente P pour différentes valeurs de α et de dispersion γ. Les deux dernières colonnes présentent les résultats des sections suivantes mais pour des raisons de commodité de lecture nous préférons faire apparaître le tableau dès maintenant. Les paramètres sont choisis pour que les performances obtenues avec le démappeur de clipping utilisant la pente et le seuil déterminés sur le LLR optimal ou de Cauchy soient inférieures ou égales à 10^{-5}.

Tab.3.1 : Comportement des paramètres P et H dans les environnements AISαSN

α	γ (10^{-5})	$P_{zéro}$	P_{Cauchy}	*H du LLR exacte*	*H du LLR Cauchy*	*H (ED)* P_{zero}	*H (ED)* P_{Cauchy}
1	0.2	4	4	4.7	4.7	2.6	2.5
1.1	0.27	4.1	3.9	4.15	4.05	2.6	2.7
1.2	0.36	4.4	3.8	3.8	3.7	2.6	2.7
1.3	0.41	4.8	3.7	3.5	3.25	2.7	2.7
1.4	0.43	5.2	3.7	3.5	3.15	2.7	2.8
1.5	0.46	5.7	3.6	3.5	3.05	3	2.8
1.6	0.49	6	3.6	3.6	2.9	3	2.9
1.7	0.52	6.2	3.5	3.8	2.9	3	3
1.8	0.54	6.3	3.5	4	2.75	3.6	3.6
1.9	0.55	6.4	3.5	4.6	2.72	3.8	3.7
2	0.56	6.4	3.5	INF	2.7	4	4

Les résultats montrent que la pente et le seuil obtenus par les deux méthodes sont très proches quand α tend vers 1 mais sont assez différents quand α s'approche de 2. Le rapport même H/P semble assez fluctuant. Comme nous avons constaté que les paramètres H et P peuvent grandement influencer les performances du décodeur, le fait d'obtenir des performances proches avec des paramètres significativement différents nous laisse perplexe. Nous n'allons cependant pas creuser plus avant cet aspect car la remarque principale que nous faisons à la lecture de la figure 3.9 est la faible qualité de ces démappeurs. Nous allons donc essayer avant tout de les améliorer. Si cependant nous devions faire un choix maintenant, l'approche du démappeur de clipping basé sur la distribution de Cauchy apporte plusieurs avantages :

- les expressions analytiques (3.6) et (3.7) donnent directement les valeurs de la pente et du seuil ;

- ces valeurs ne dépendent que de la dispersion γ du bruit. Il n'est plus nécessaire d'estimer α. Nous n'avons pas travaillé sur l'estimation des paramètres et nous ne savons pas s'il existe des méthodes permettant d'estimer la dispersion sans estimer α. Quoiqu'il en soit, les figures 3.7 et 3.8 ainsi que le tableau 3.1 ont également montré que les variations de P et H sont les plus lentes pour $\alpha = 1$ par rapport aux autres valeurs de $\alpha > 1$. Ceci peut nous laisser espérer une certaine robustesse par rapport aux erreurs d'estimation.

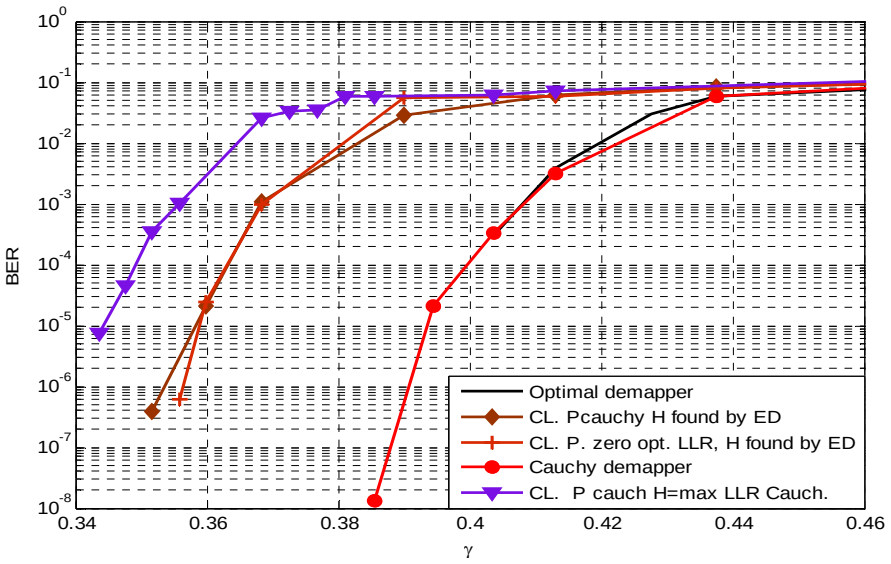
Mais les approches de démappeur de clipping qui se basent sur les LLRs exacts et de Cauchy ne résolvent pas le problème d'impulsivité d'environnement AISαSN**.** Ces performances se dégradent

rapidement quand l'impulsivité augmente. Ceci peut vraisemblablement s'expliquer par les grandes valeurs du seuil de clipping H qui ne permet pas de réduire l'impacte des grandes valeurs de bruit qui masque complètement information. Ceci nous ramène alors à tester d'autres approches en espérant qu'elles s'adaptent mieux à l'impulsivité. Dans ce contexte l'évolution de densité sera adoptée afin d'optimiser ce seuil H toute en gardant les mêmes pentes.

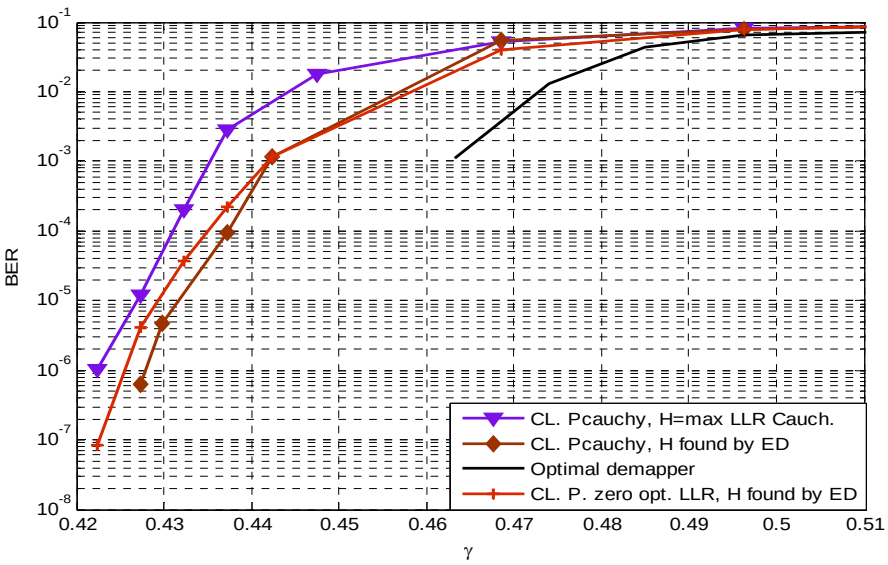
c) ***Pente en zéro et calcul du seuil par évolution de densité***

Nous gardons la même approche que dans les points a) et b) pour déterminer la pente, soit à partir du LLR optimal soit du LLR de Cauchy. Le seuil H est alors déterminé par évolution de densité. Il ne dépendra que de α. Le tableau 3.1 présente le comportement de H trouvé par cette méthode en fonction des variations de α. Ce seuil H est la valeur qui permet (asymptotiquement) une transmission sans erreur pour la plus grande dispersion possible. Nous remarquons que la valeur de H obtenue est significativement inférieure à celle donnée par la première approche.

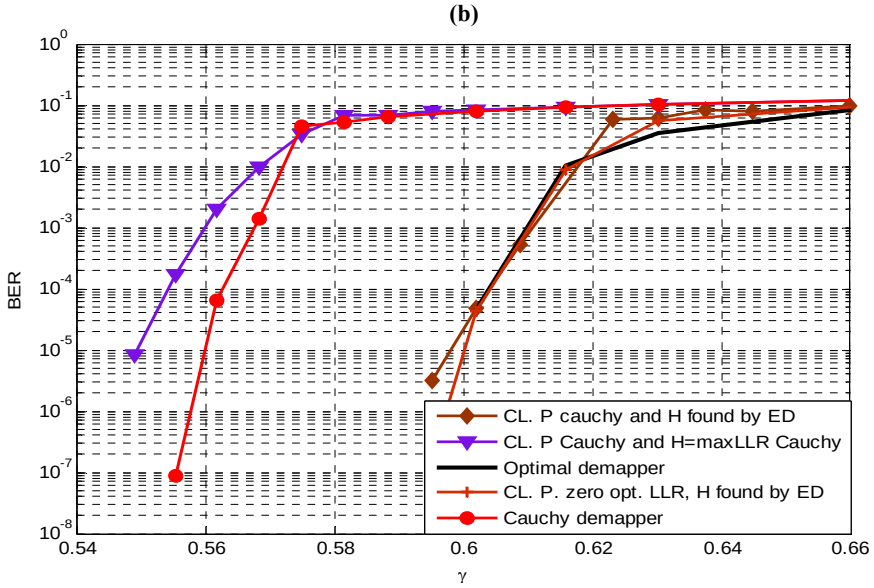
Nous constatons alors sur la figure 3.10 que les codes LDPC utilisant ce type de démappeur se comportent relativement bien dans les environnements AISαSN. Les performances ont cependant tendance à se dégrader quand l'impulsivité augmente (α = 1.2 et 1.4). Dans ce cas, l'utilisation de l'évolution densité permet tout de même d'améliorer les performances. Dans les environnements moins impulsifs (α ≥ 1.8), le clipping s'avère performant, offrant même un meilleur comportement que le démappeur de Cauchy.



(a)



(b)



(c)

Fig.3.10 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=1.2, (b) α=1.4, (c) α=2

La seconde remarque que nous pouvons faire à partir de la figure 3.10 est que les deux approches pour le calcul de la pente donnent des résultats similaires. Nous concluons que les deux méthodes conviennent pour déterminer la pente, donc pourquoi ne pas utiliser la méthode analytique uniquement fonction de la dispersion. Cependant, un ajustement des seuils permet nettement d'améliorer les performances. Nous perdons alors l'intérêt d'une méthode purement analytique puisque nous devons tabuler H en fonction de α et de γ.

Pour conclure, cette première approche permettant d'obtenir la pente et le seuil du démappeur de clipping, nous faisons trois commentaires :

1. un tel démappeur offre une grande simplicité de mise en œuvre comparé aux démappeurs optimaux, de Cauchy ou logistique présentés dans la section 2 de ce chapitre. Il reste cependant très performant bien que le choix des paramètres P et H soit un facteur important pour garantir de bonnes performances ;

2. nous pouvons à partir des courbes du LLR obtenir P et H de façon numérique. Il s'avère cependant que le seuil obtenu est trop important, ne permettant pas de suffisamment réduire l'impact des grandes valeurs. Un meilleur seuil peut être obtenu par évolution de densité et les performances du démappeur ainsi obtenues s'approchent nettement des performances optimales. Ces approches nécessitent des calculs préalables et la tabulation des valeurs P et H ;

3. nous avons également obtenu des formes analytiques pour déterminer *P* et *H* directement à partir de la dispersion du bruit. Cette approche supprime la dépendance par rapport à α mais présente les mêmes performances par rapport aux valeurs obtenues à partir du LLR exact. Le système reste cependant encore assez peu performant car la valeur de H obtenue est trop grande. L'évolution de densité permet d'améliorer la valeur de H mais nécessite à nouveau la tabulation.

3.2.2 Études basées sur la puissance de bruit après clipping

Nous avons dans un second temps pris une approche tout à fait différente pour déterminer la pente et le seuil. Nous étudions la puissance du bruit après l'opération de *clipping*. Les valeurs retenues pour P et H sont celles qui minimisent cette puissance. Cependant, cette opération ne nous permet pas d'obtenir les deux paramètres mais simplement le rapport entre les deux (H/P par exemple). En effet, modifier la pente d'un facteur donné et modifier le seuil de façon similaire ne modifie pas l'impact du clipping sur l'information qu'il nous reste en fin d'opération.

Le bruit après clipping n'est pas un bruit SαS tel que nous le trouvons à l'entrée du démappeur. Nous pouvons alors utiliser la puissance arithmétique pour estimer la puissance du bruit résiduel :

$$S_{arith} = \frac{1}{N}\sum_{i=1}^{N} x_i^2 \tag{3.8}$$

Nous recherchons donc les valeurs de P et H qui minimisent la puissance du bruit après clipping. Nous procédons comme suit :

1. nous générons une séquence binaire (+1, -1), notée *x(n)* ;

2. nous lui ajoutons un bruit SαS, noté $n_\alpha(n)$;

3. nous appliquons au signal reçu $r(n)=x(n)+n_\alpha(n)$ le démappeur de clipping (3.2). Nous notons *y(n)* le signal après ce démappeur ;

4. nous retranchons au signal *y(n)* la séquence source multipliée par la pente et obtenons ainsi le bruit additif après clipping : $z(n)=y(n)-P.x(n)$.

En utilisant (3.8) nous pouvons alors calculer la puissance du bruit résiduel et nous cherchons le rapport H/P qui minimise cette puissance.

Dans un premier temps nous étudions le comportement de la puissance en fonction de la variation du rapport H/P pour différentes valeurs de α et γ. Le résultat est présenté sur les figures (3.11) pour une dispersion γ fixée et différents α et (3.12) pour un α fixé et différentes valeurs de γ.

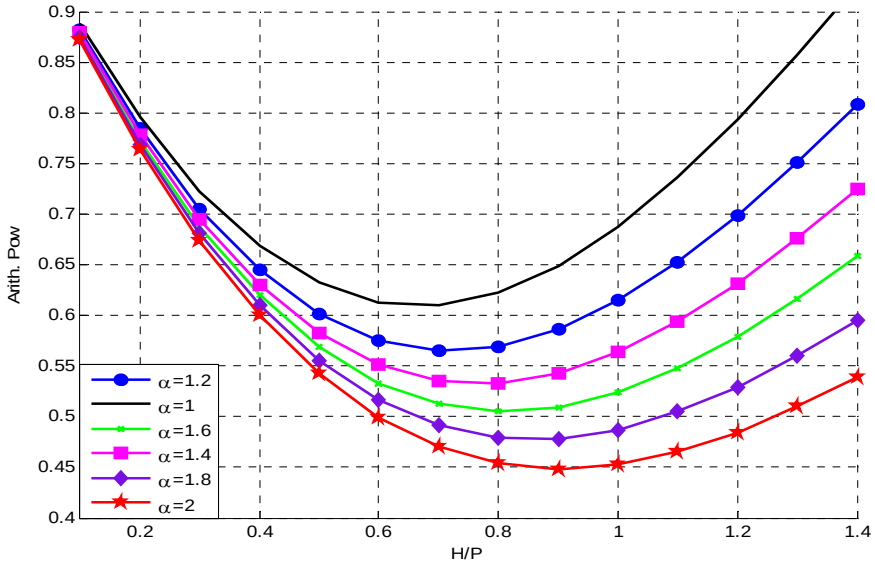


Fig.3.11 : P_{arit} = f (H/P) pour différents α et pour un γ = 0.7

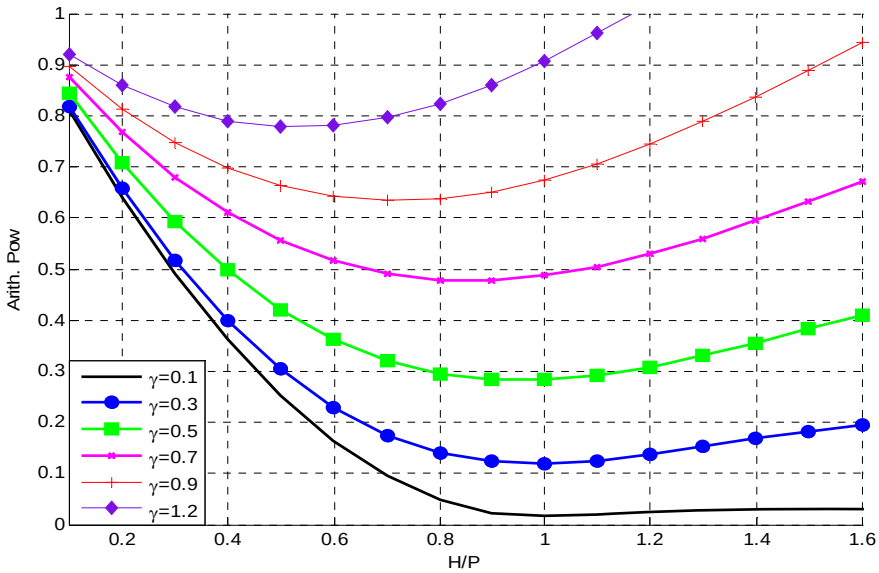


Fig.3.12 : P_{arit} = f (H/P) pour différents γ et pour un α = 1.8

Nous constatons sur les figures 3.11 et 3.12 qu'il existe un H/P minimisant la puissance du bruit après clipping. Ce minimum diminue quand α diminue ou quand γ augmente.

Nous représentons sur la figure 3.13 le rapport H/P qui minimise la puissance de bruit après clipping pour α dans [1,2] et γ dans [0.2, 1]

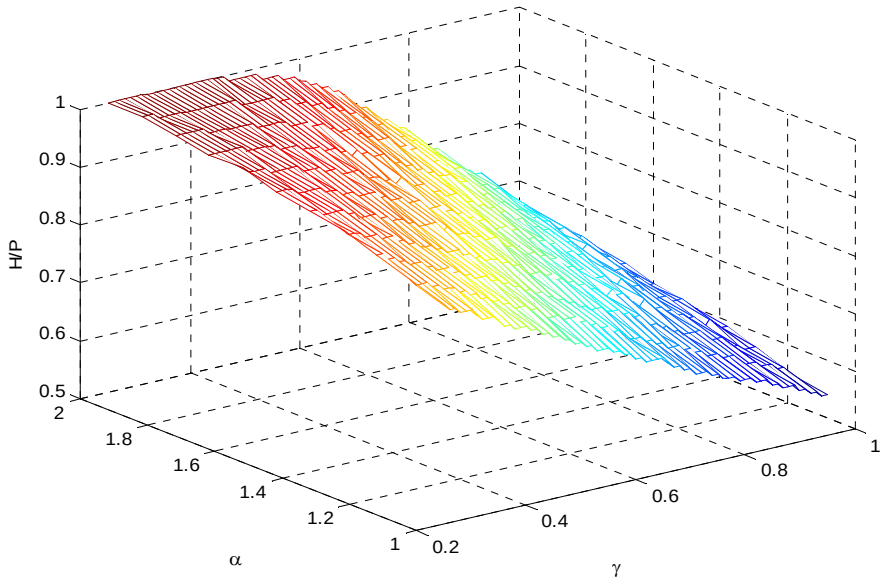


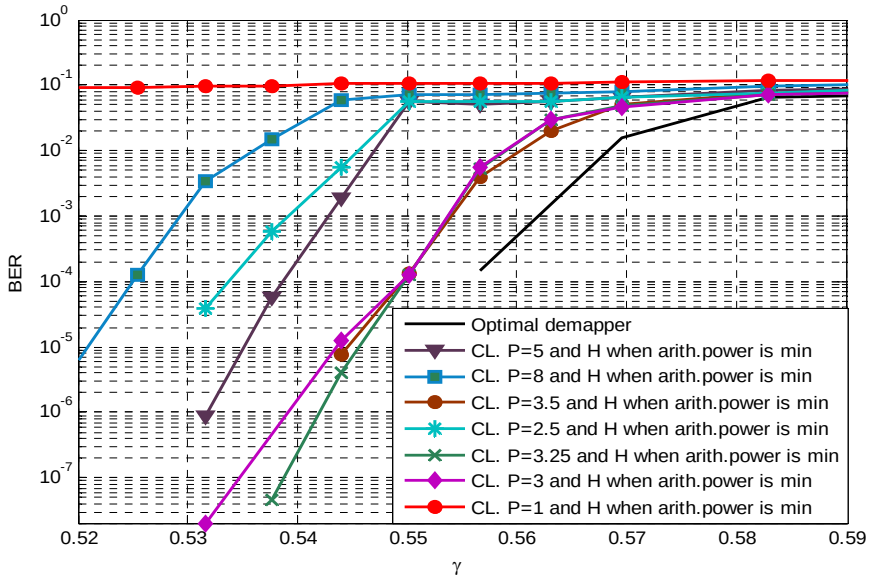
Fig.3.13 : H/P = f (α, γ) où P_{arith} est minimale pour différents pentes

La figure 3.13 montre que le rapport H/P augmente quand α augmente ou quand la dispersion γ diminue. Il reste toujours inférieur ou égal à 1 quelque soit les valeurs de α et γ de notre étude. Plus le bruit devient impulsif, plus il semble approprié de prendre un seuil faible. Nous seuillons en fait en dessous du niveau du signal et ceci même dans le cas gaussien quand la variance du bruit devient importante. Ceci représente peut-être une limite de cette étude car dans le cas gaussien, l'optimal devrait être un seuil infini. Il pourrait alors être intéressant d'étudier la distribution du bruit écrêté et de modifier à nouveau la métrique en conséquence mais ceci sort du cadre de ce travail.

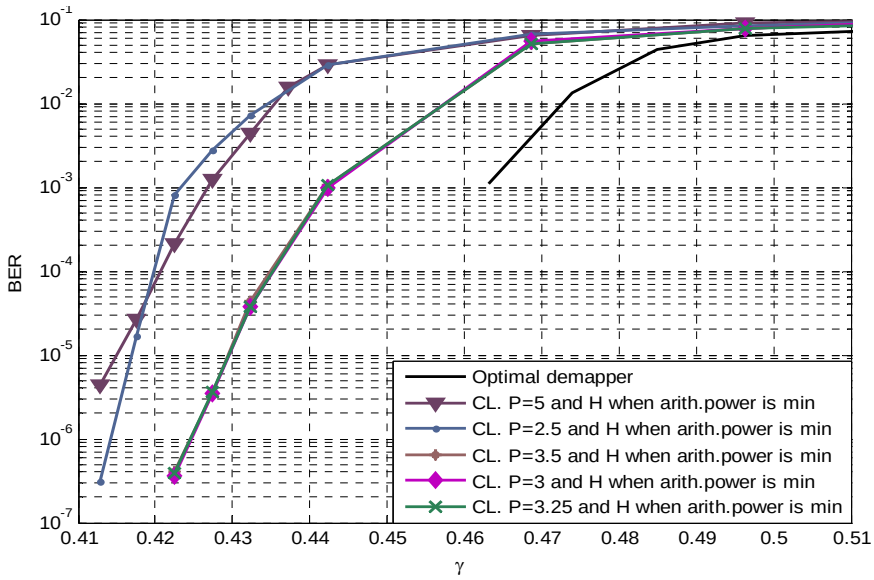
Cette méthode d'estimation de la puissance de bruit après clipping nous permet d'ajuster les paramètres du démappeur. Nous tabulons le rapport H/P obtenu en fonction des paramètres α et γ, indépendamment de la pente P.

Cependant, pour complètement définir le démappeur, il nous manque la définition soit de P, soit de H. D'après l'étude de la partie 3.2.1, la pente nous semble être un paramètre plus simple à déterminer et moins sensible à des erreurs d'estimation de α et γ. Nous préférons donc déterminer P puis le H à partir du rapport tabulé. Deux approches nous semblent alors naturelles. La première consiste à prendre un P fixe (figure 3.14). Intuitivement, d'après la figure 3.7, une valeur aux alentours de 3, où les courbes donnant P en fonction de γ se croisent pourrait être un compromis judicieux. Nous testons cependant plusieurs valeurs pour évaluer la sensibilité au choix de cette pente. La seconde approche qui nous paraît pertinente est de choisir la valeur de P donné par la pente de Cauchy (α = 1).

La figure 3.14 présente le comportement des codes LDPC avec le démappeur de clipping utilisant une pente fixe pour α = 1.8 et 1.4. Nous avons pris une petite valeur de pente (P = 1), de grandes valeurs (P = 8 et 5) et des valeurs proches de 3 : P = 2.5, P = 3, P = 3.25 et P = 3.5.



(a)

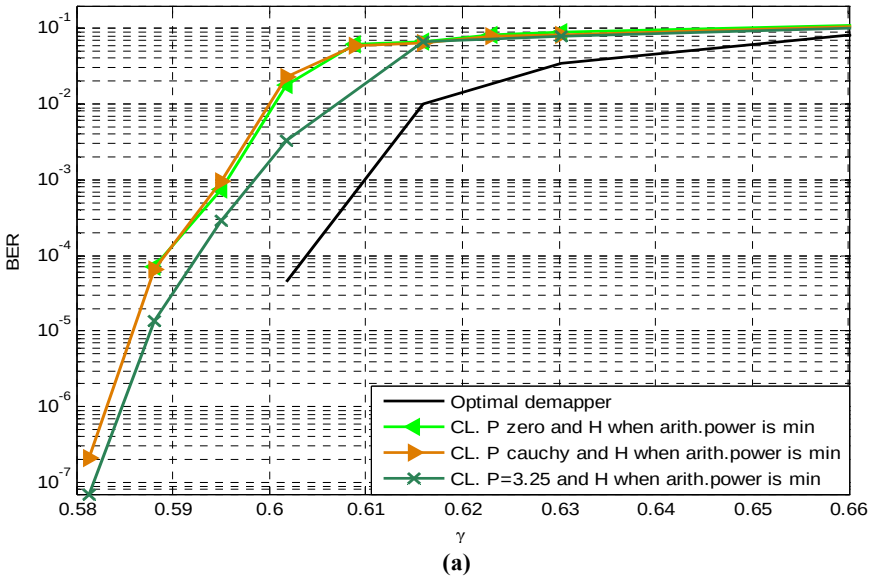


(b)

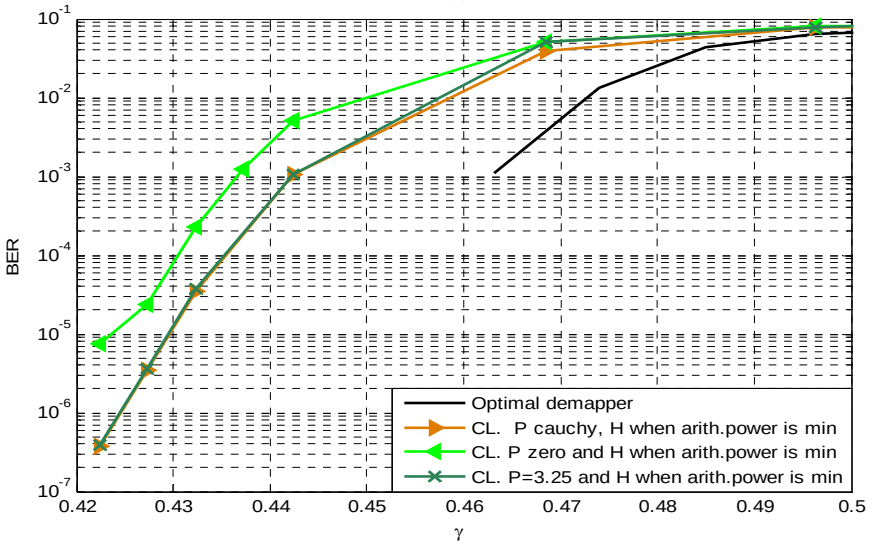
Fig.3.14 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=1.8, (b) α=1.4

Comme nous nous y attendions, le démappeur présente de mauvaises performances quand la pente P prend de petites valeurs (P = 1 ou 2.5) ou de grandes valeurs (P = 5 ou 8). Ceci montre la sensibilité du décodeur à la pente choisie et nous étudions donc le comportement de la chaîne de transmission quand les valeurs sont plus proches de 3 qui, intuitivement, nous semblent un choix plus judicieux. Les performances obtenues avec des valeurs de P entre 3 et 3.5 sont très proches. Si la sensibilité à la pente existe, une large plage de valeurs est acceptable et le choix de P ne nécessite pas une précision extrême. La pente P = 3.25 se comporte bien dans tous les environnements testés et nous la conserverons pour la suite.

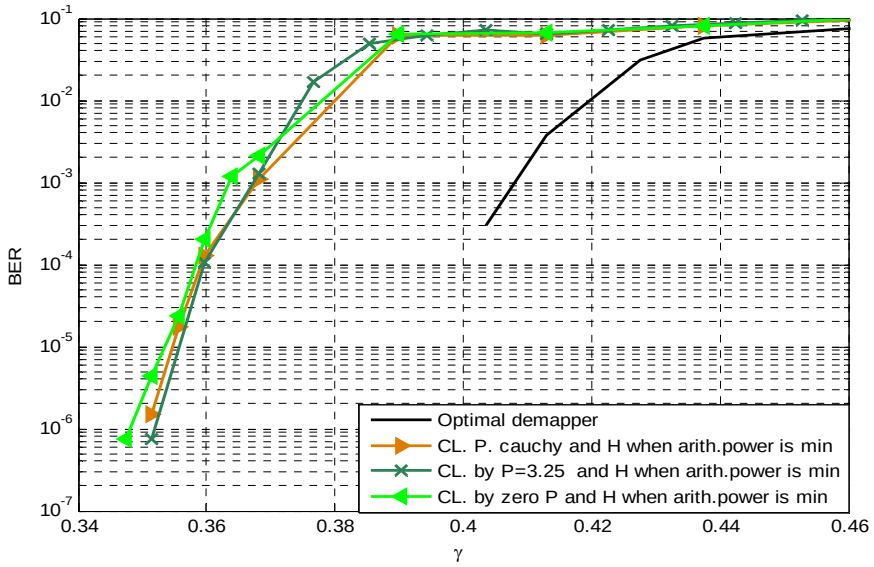
Pour évaluer plus en détail l'impact du choix d'une pente constante sur les performances du système, nous comparons le comportement du décodeur quand le démappeur utilise la pente de Cauchy où la pente en zéro du LLR exact (figure 3.15, pour α = 2, 1.4 et 1.2).



(a)



(b)



(c)

Fig.3.15 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=2, (b) α=1.4, (c) α=1.2

Le choix d'une pente fixe et d'un rapport H/P qui minimise la puissance de bruit après clipping semble tout à fait judicieux, plus performants même que de prendre des pentes variables. Qui plus est la valeur proposée garde de bonnes performances quel que soit la valeur de α choisie.

Cette deuxième proposition pour déterminer les paramètres du démappeur de clipping se base sur la recherche du rapport H/P qui minimise la puissance de bruit à la sortie du démappeur. Ce rapport est fonction de α et γ. Il reste cependant l'un des paramètres à déterminer pour caractériser complètement le démappeur. Les résultats montrent que le choix de la pente peut avoir un impact important sur les performances. Nous proposons de choisir la pente puis de calculer le seuil à partir du rapport H/P qui minimise la puissance de bruit. Nous aurons donc un seuil H en fonction de la pente et des paramètres α et γ. Finalement, nous avons constaté que le démappeur de clipping avec une pente fixe P = 3.25 s'adapte bien aux différents environnements AISαSN.

3.2.3 Étude basée l'Évolution Densité

Nous abordons maintenant une dernière approche pour déterminer les paramètres H et P du démappeur de clipping. L'idée est de se baser sur l'évolution de densité. Pour une valeur donnée de α, nous recherchons les valeurs de H et P qui permettent une transmission asymptotiquement sans erreur pour la plus grande valeur possible de la dispersion γ. Ces valeurs seront tabulées et utilisées dans le décodeur. Le grand intérêt de cette approche, si les résultats obtenus s'avèrent bons, sera de ne faire dépendre le démappeur que de la valeur de α.

Dans un premier temps, nous présentons la valeur de γ qui permet, asymptotiquement, une transmission sans erreur en fonction des valeurs de H et de P. Sur la figure 3.16 nous illustrons ce résultat dans le cas où α = 1.5. Nous représentons également la capacité du canal pour cette valeur de α.

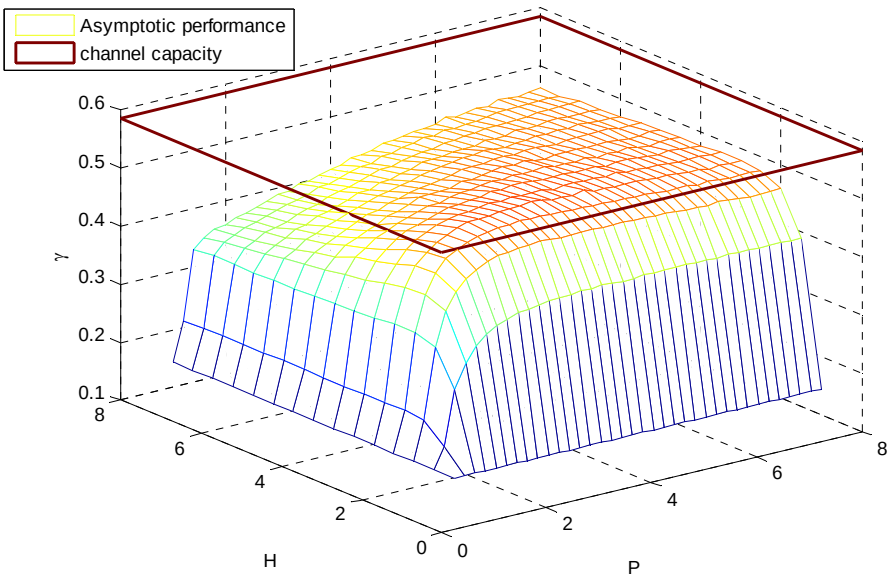
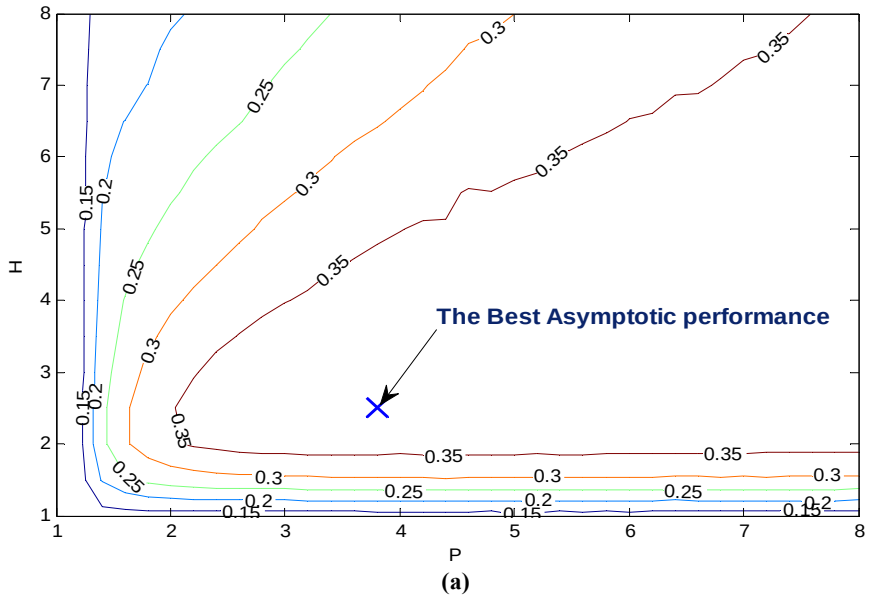


Fig.3.16 : Comportement des performances asymptotiques du démappeur de Clipping en fonction des variations des P et H dans un environnement AISαSN (α=1.5)

Nous remarquons la présence d'un maximum sur cette figure, et donc l'existence d'un couple (H, P) qui permet d'obtenir les meilleures performances asymptotiques. Nous remarquons également la présence d'un plateau relativement important. Ceci peut laisser penser à une certaine robustesse au choix de H et P. La zone optimale semble cependant s'élargir quand P augmente ce qui se rapproche de l'idée que le paramètre H/P est assez important et qu'augmenter P devrait permettre de choisir H sur une plage plus large. Nous avons vu précédemment qu'une trop grande valeur de P réduit cependant significativement les performances du démappeur et qu'il faut analyser avec prudence la présence de ce plateau.

Afin de mieux visualiser la position du H et du P maximisant la valeur de la dispersion, nous représentons sur la figure 3.17 les contours de la courbe de niveau représentée en 3D figure 3.16 pour une petite valeur de α (α = 1.2) et pour le cas gaussien (α = 2). Nous plaçons également sur ces courbes les P et H où la dispersion maximale est atteinte.



(a)

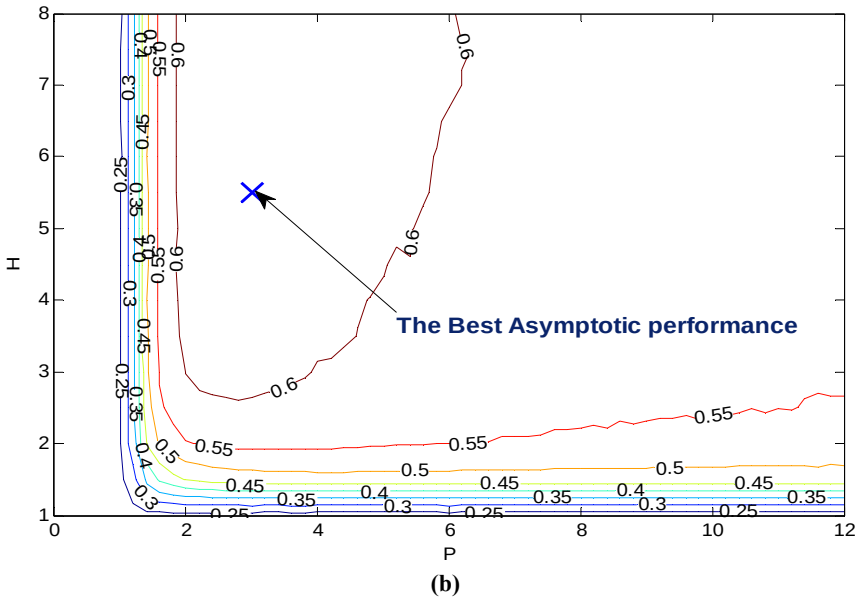


Fig.3.17 : Comportement des performances asymptotiques dans différents environnements AISαSN : (a) α=1.2 (b) α=2

La première chose que nous remarquons sur la figure 3.17 (b) pour le cas gaussien est que le seuil donnant le plus grand γ est élevé quelque soit la pente et que les performances ne se dégradent pas quand le seuil augmente. Ceci est logique car un démappeur linéaire, sans seuil, est optimal dans le cas gaussien. Les résultats issus de l'évolution de densité dépendent d'un algorithme de type *Monte-carlo*. Les conséquences de ce choix sont multiples : les valeurs obtenues sont plus ou moins approchées et elles dépendent des nombres pseudo-aléatoires générés. Ainsi dans le cas gaussiens, la probabilité qu'une valeur dépasse un seuil élevé est très faible et cet événement n'apparaitra sûrement pas dans les simulations. Ainsi la valeur de H reste finie dans le cas gaussien de part ces remarques.

Par contre, dans le cas d'un bruit impulsif (figure 3.17(a)), le nombre de valeurs écrêtées devient significatif et quelle que soit la valeur de P, un seuil trop grand ne limite plus suffisamment l'impact des grandes valeurs et tend à dégrader les performances asymptotiques.

Afin de généraliser cette étude pour différentes valeurs de α, nous représentons sur les figures 3.18 et 3.19 le comportement des paramètres P et H trouvés par l'évolution densité en fonction de la valeur de α.

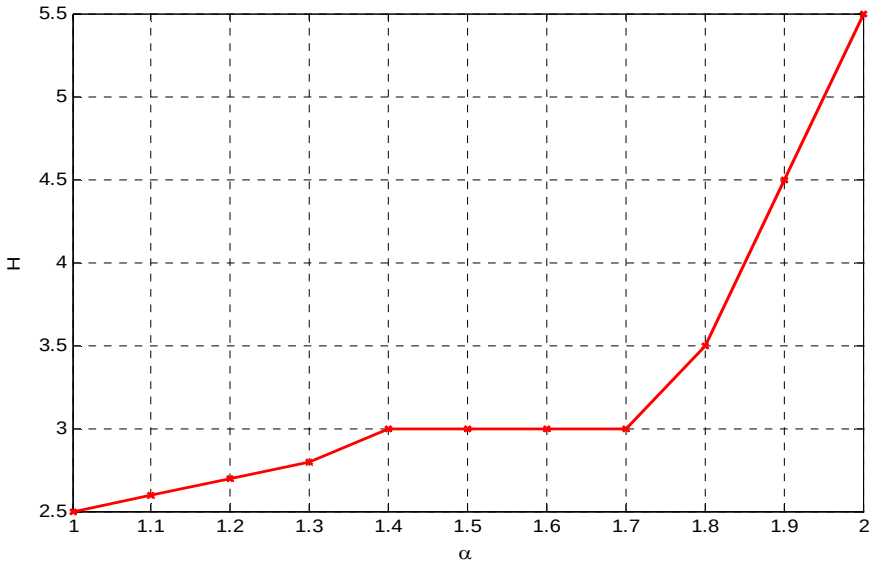


Fig.3.18 : comportement du seuil de Clipping H en fonction de variation de α

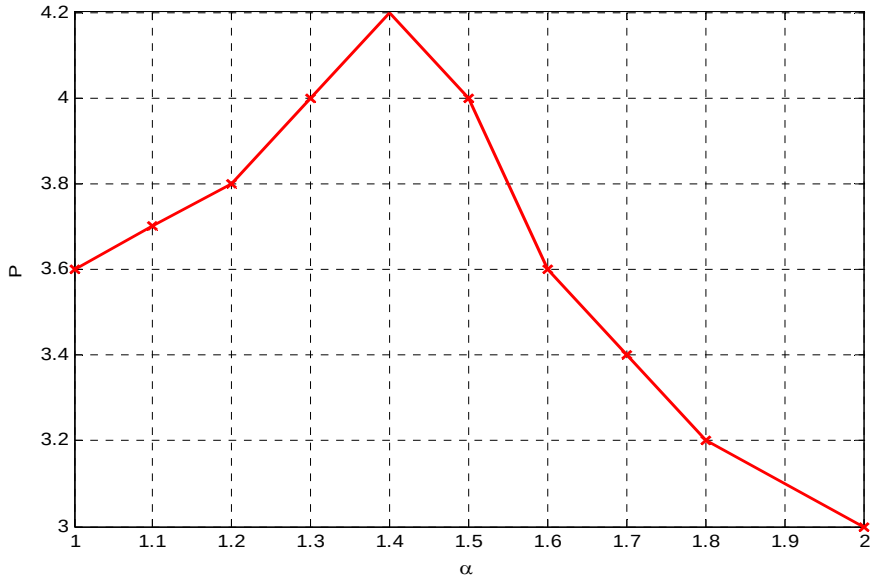


Fig.3.19 : comportement de la pente P en fonction de variation de α

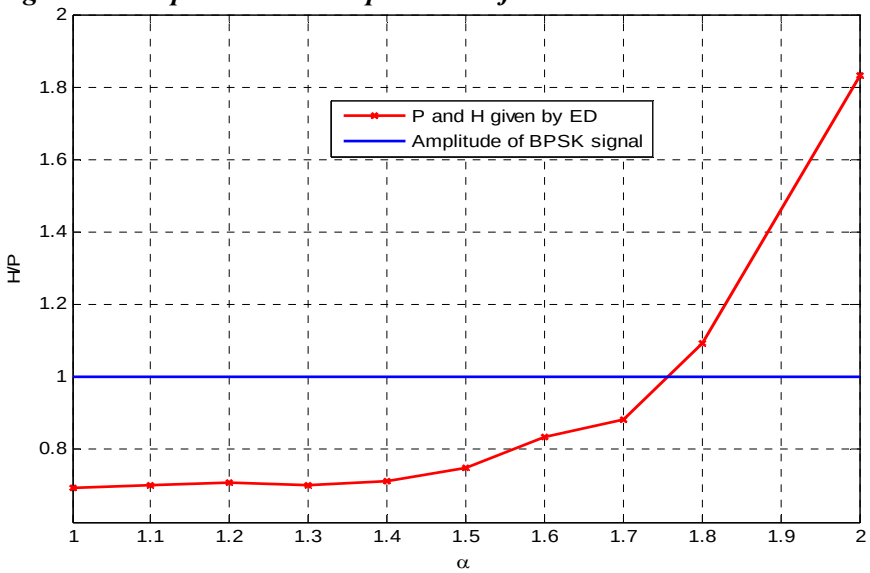


Fig.3.20 : comportement du rapport H/ P en fonction de variation de α

Nous notons tout d'abord que les variations de la pente (figure 3.19) sont moyennes, entre 3 et 4.2, ce qui tend à confirmer les choix de la partie 3.2.2 où nous avions préconisé de choisir une valeur de pente fixée entre 3 et 4 et d'adapter le rapport H/P à l'environnement. Par contre les valeurs du rapport H/P (figure 3.20) doivent être significativement réduites quand l'impulsivité du bruit augmente. Ce rapport doit même être inférieur à l'amplitude du signal dès que α est inférieur à 1.8. Quand α se rapproche de deux par contre, l'impact des valeurs impulsives devient de moins en moins significatif et il vaut mieux tenir compte au maximum des valeurs pondérées, le seuil optimal devenant infini quand $\alpha = 2$.

Un commentaire intuitif de ce résultat est que dans le cas d'un bruit fortement impulsif, l'information « soft » habituellement utilisée et si efficace dans de nombreux décodage, perd complètement de son efficacité et devient même un facteur qui nuit aux performances. Un décodage avec entrée « hard » du décodeur peut s'avérer plus performant.

Les paramètres H et P vont être tabulés en fonction de α et utilisés pour le démappeur de clipping à l'entrée du décodeur. Nous présentons les résultats dans la partie suivante où nous faisons également le comparatif et le bilan de l'ensemble des approches proposées.

3.2.4 Bilan des différentes études.

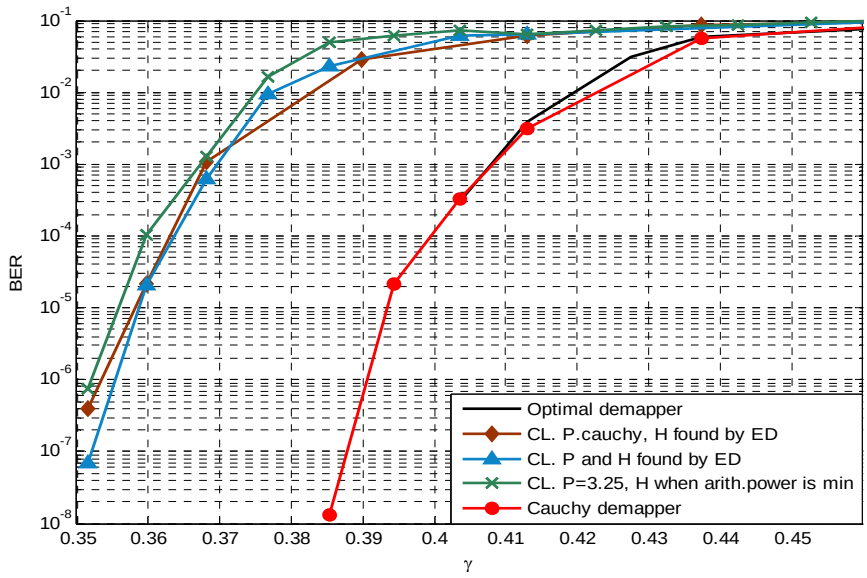
Dans cette section nous présentons les résultats obtenus lors de nos différentes études visant à paramétrer le démappeur de clipping. Les différentes approches retenues sont les suivantes :

1. la pente déterminée par la tangente en zéro du LLR de Cauchy et le seuil par évolution densité. Cette approche nécessite l'estimation de α et γ puis la tabulation des différents paramètres correspondants ; la pente calculée dépend de γ et un seuil H tabulé par évolution densité en fonction de α. Moins complexe à mettre en œuvre ;

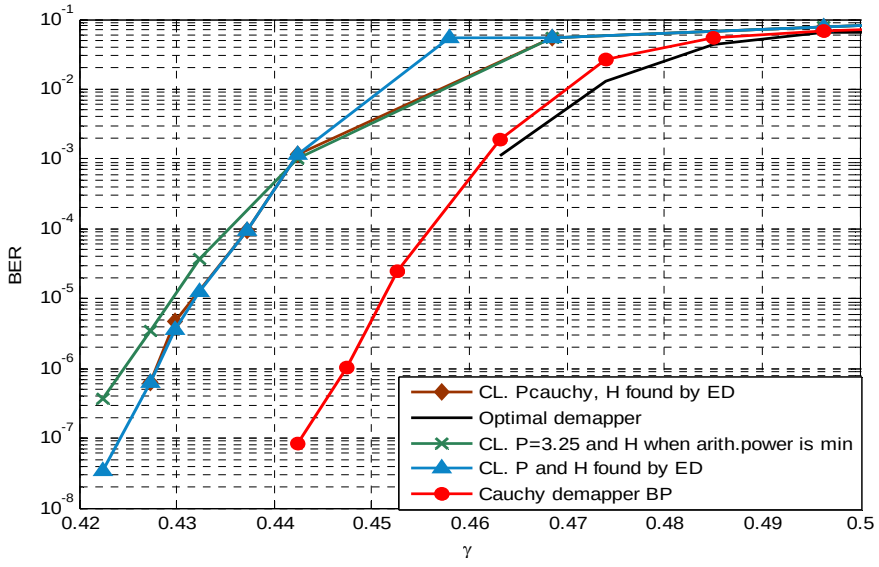
2. la pente fixée à 3.25 et le seuil obtenu en minimisant la puissance arithmétique du bruit après l'opération de clipping. Cette approche nécessite l'estimation de α et γ puis la tabulation du rapport H/P. Bien que déterminer ce rapport soit relativement simple, c'est une opération qui s'avère relativement longue ;

3. la pente et le seuil obtenus par évolution de densité. Cette approche ne nécessite que l'estimation de α puis la tabulation des différents paramètres correspondants.

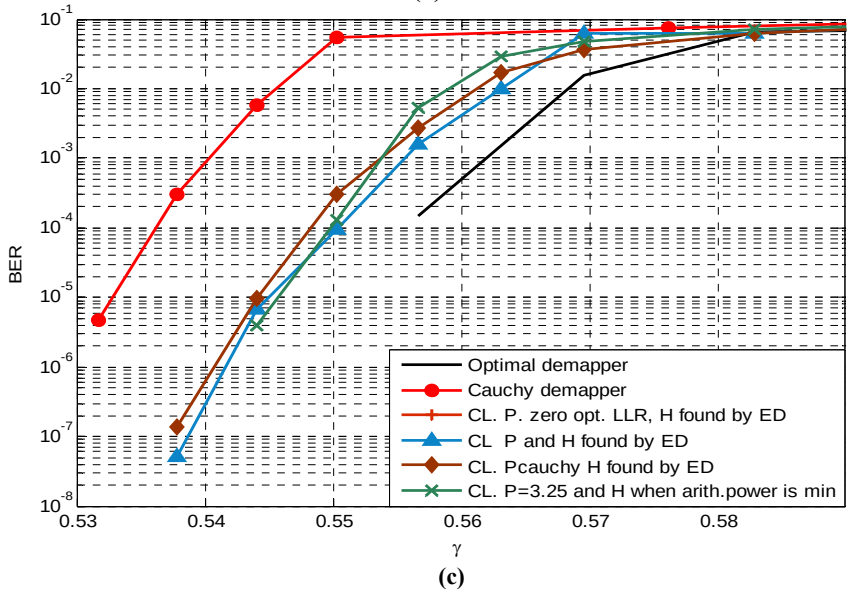
Sur la figure 3.20, nous représentons les performances de ces trois stratégies pour $\alpha = 1.2, 1.4, 1.8$ et 2. Nous représentons également les performances du démappeur optimal obtenu en utilisant l'expression exacte de la densité de probabilité α-stable, le calcul des LLRs demandant alors l'intégration numérique de la fonction caractéristique. Enfin, nous représentons le démappeur de Cauchy dans les deux cas $\alpha = 1.2, 1.4$ et 1.8.


BER
γ
Optimal demapper
CL. P.cauchy, H found by ED
CL. P and H found by ED
CL. P=3.25, H when arith.power is min
Cauchy demapper

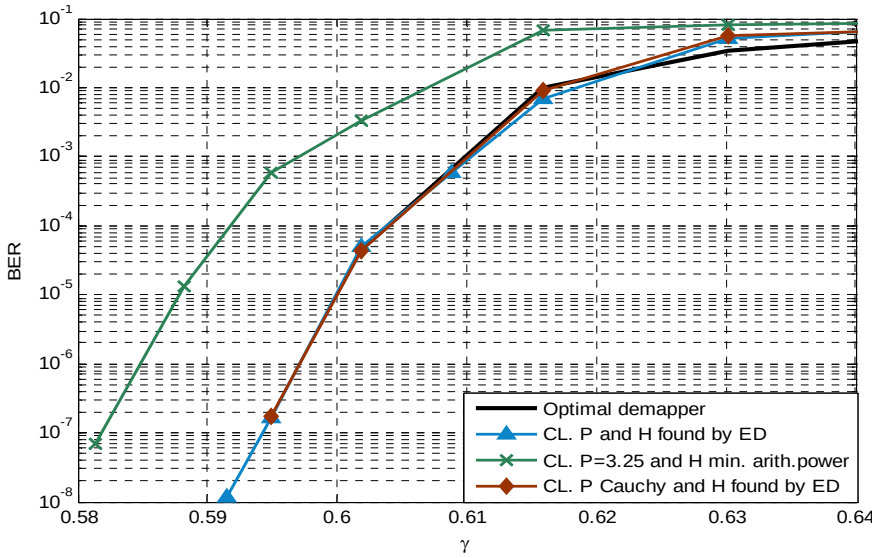
(a)


BER
γ
CL. Pcauchy, H found by ED
Optimal demapper
CL. P=3.25 and H when arith.power is min
CL. P and H found by ED
Cauchy demapper BP

(b)


BER
γ
Optimal demapper
Cauchy demapper
CL. P. zero opt. LLR, H found by ED
CL P and H found by ED
CL. Pcauchy H found by ED
CL. P=3.25 and H when arith.power is min

(c)



(d)

Fig.3.20 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α=1.2, (b) α=1.4, (c) α=1.8, (d) α=2

La remarque la plus globale que nous puissions faire est que les décodeurs 1 (déterminé à partir du LLR) et 4 (par évolution de densité) se comportent de façon équivalente. La dépendance par rapport à la dispersion ne semble donc pas nécessaire à prendre en compte ce qui permet une simplification significative de la réception.

Ensuite, même si la pente nous semblait avoir peut d'influence, la fixer dégrade les performances quand l'impulsivité du bruit disparaît. Un test d'identification interférence impulsive ou non serait alors nécessaire si on veut garder un système robuste même dans un cadre non impulsif. Comme l'estimation de α reste nécessaire pour déterminer le seuil, cette approche reste envisageable. Dans les cas impulsifs, ce choix de pente ne dégrade pas les performances.

Enfin, le démappeur basé sur des expressions analytiques obtenues à partir de la densité de probabilité de Cauchy entraîne une dégradation des performances. Ceci est essentiellement du à la valeur du seuil et l'adaptation du seuil en fonction de α et de γ permet d'obtenir des performances équivalentes aux autres approches. Mais dans ce cas il est nécessaire de tabuler ces valeurs et autant alors tabuler H et P.

Nous pouvons également remarquer que le démappeur de clipping offre un bon compromis entre qualité et complexité. En particulier quand l'impulsivité est limitée, ce démappeur se comporte très bien et reste relativement proche de l'optimal. Cependant quand l'impulsivité augmente, les performances se dégradent quelque peu et le démappeur de Cauchy s'avère plus performant. La question de ce que nous faisons des grandes valeurs se pose alors. Une grande valeur montre que l'information est noyée dans un bruit très fort. Se contenter d'un seuil semble lui laisser toujours une importance trop grande dans le processus de décodage. Le LLR optimal tend à donner à ces grandes valeurs une vraisemblance proche de 0, donc pas d'impact réel dans le processus de décodage. Le Hole Puncher semble alors être une approche logique. Nous l'introduisons dans la partie 3.3 et essayons de l'améliorer dans la partie 3.4 mais cette étude reste brève et de nombreuses extensions sont envisageables. Avant cela nous étudions la robustesse des démappeurs proposés face à une mauvaise estimation du paramètre α.

3.2.5 Sensibilité à une mauvaise estimation de α.

Nous nous contentons dans cette section du démappeur basé sur l'évolution densité. En effet, malgré un travail préalable conséquent, il est sans doute le plus simple dans sa mise en œuvre et ne nécessite de tabuler des valeurs qu'en fonction de α. Pour cela nous considérons le cas d'un environnement AISαSN caractérisé par le paramètre α = 1.4. Nous étudions sur la figure 3.21 le comportement du décodeur quand le démappeur utilise une pente et un seuil obtenus pour des valeurs de α légèrement différentes de 1.4 (α = 1.34, 1.5 et 1.6). Nous les comparons avec le résultat obtenu avec une estimation exacte de α.

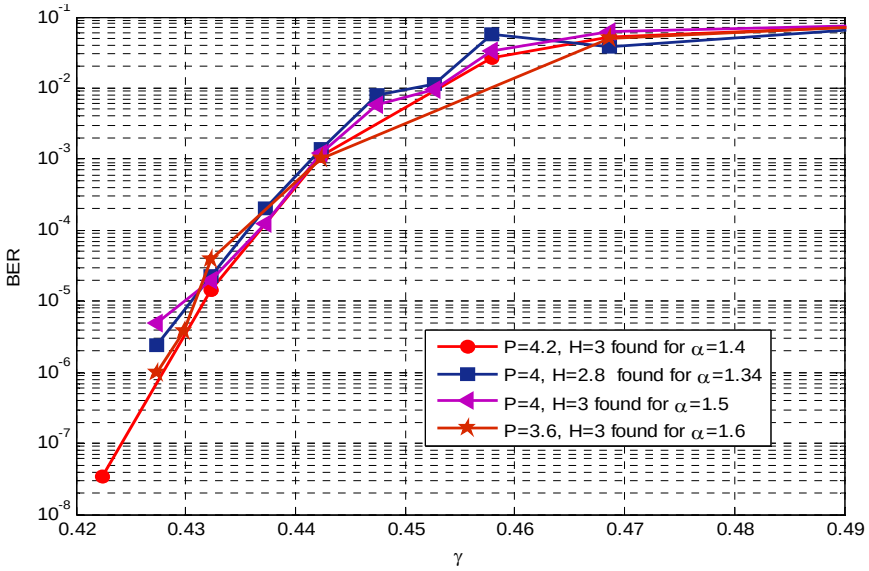


Fig.3.21 : Comparaison des performances du démappeur de Clipping pour un P et H fixes et pour différents α proches α = 1.4

Nous constatons que les performances restent équivalentes malgré les erreurs sur le α. Ceci tend à renforcer l'idée qu'un démappeur de clipping est une bonne solution pour limiter l'impact des bruits impulsifs et que si un choix judicieux de la pente et du seuil est essentiel, la précision du choix n'est pas un facteur critique.

3.3 Le démappeur Hole Puncher

Quand l'impulsivité du bruit augmente, nous avons vu que le démappeur de clipping perd un peu de son efficacité. Il semble alors logique de se dire qu'il faut encore réduire l'impact des valeurs fortes. C'est ce que propose de faire le Hole Puncher (HP) en annulant les valeurs au delà d'un seuil donné. L'expression mathématique de ce démappeur est donnée par (3.9) et il a été représenté sur la figure 3.4.

$$LLR_{HolePuncher} = \begin{cases} P.y & |y| \le H/P \\ 0 & |y| > H/P \end{cases} \qquad (3.9)$$

Ce démappeur est défini par 2 paramètres : la pente P et le seuil de remise à zéro H.

Pour déterminer ces paramètres, nous avons repris l'étude qui nous a semblé la meilleure dans le cas du démappeur de clipping : l'évolution de densité. Sur la figure 3.22, nous présentons les performances asymptotiques de ce démappeur en fonction de la variation de P et de H pour α = 1.5. Nous retrouvons sur l'axe vertical la dispersion maximale pour laquelle une transmission sans erreur est possible asymptotiquement. Nous considérons le maximum de cette courbe pour déterminer les paramètres P et H que nous utiliserons dans nos simulations.

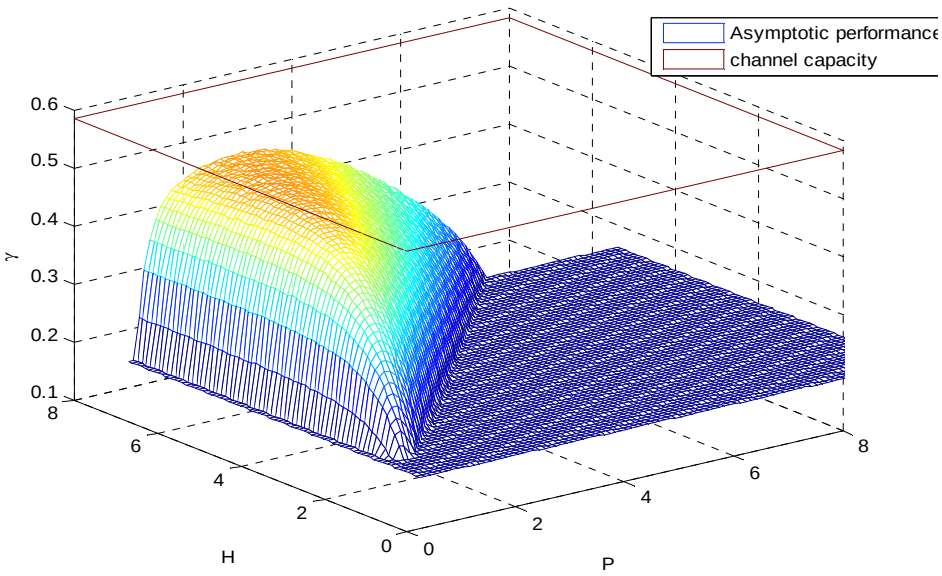
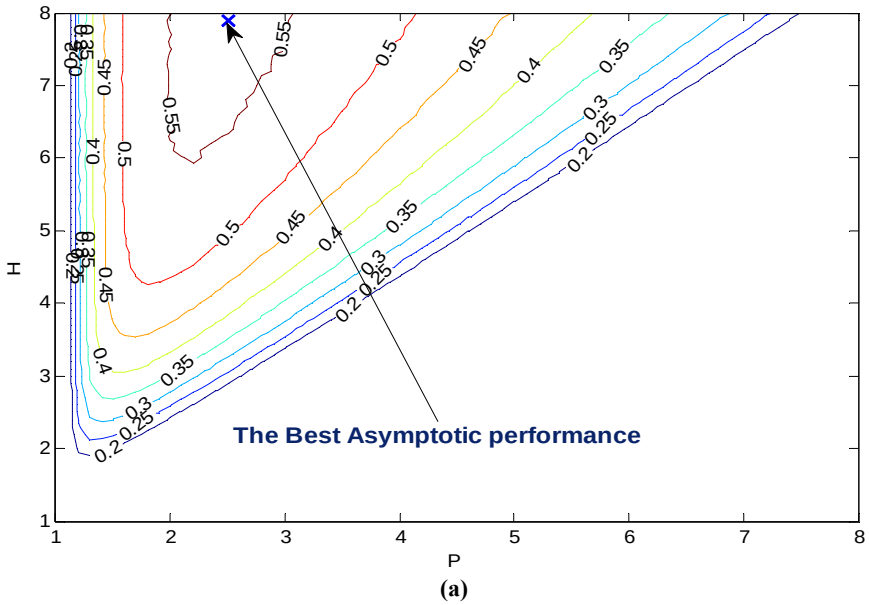


Fig.3.22 : Comportement des performances asymptotiques du démappeur Hole Puncher en fonction des variations de P et H dans un environnement AISαSN (α=1.5)

Comme dans le cas du clipping, il apparaît qu'il existe un coupe (H, P) donnant des performances optimales au démappeur. Sur la figure 3.23, nous représentons les courbes de niveau correspondant au même graphique que le 3.22 pour deux valeurs différentes de α (1.2 et 1.8).



(a)

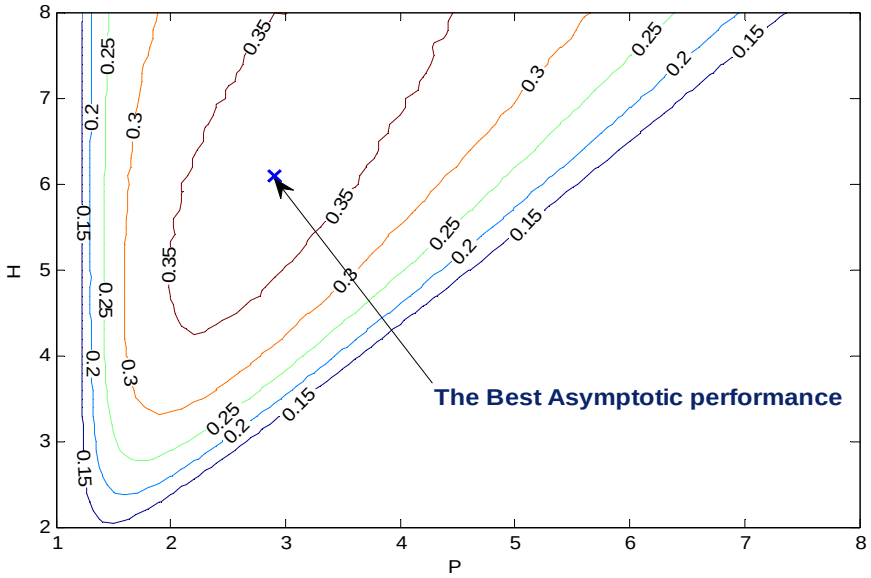


Fig.3.23: Comportement des performances asymptotiques dans différents environnements AISαSN : (a) α=1.8 (b) α=1.2

La différence majeure avec le clipping est que la valeur du seuil est plus grande (bien que restant assez proche de l'amplitude du signal utile puisque dans le cas α = 1.2, le HP annule les valeurs environs deux fois supérieures à cette amplitude) alors que la valeur de pente reste équivalente. Ceci s'explique par le compromis à trouver entre ne plus tenir compte d'une valeur qui apparaît comme être trop bruitée et ne pas faire reposer notre décision sur trop peu de valeurs. Sur les figures 3.24 et 3.25 nous représentons le comportement des paramètres P et H trouvés par évolution densité en fonction de la variation de α.

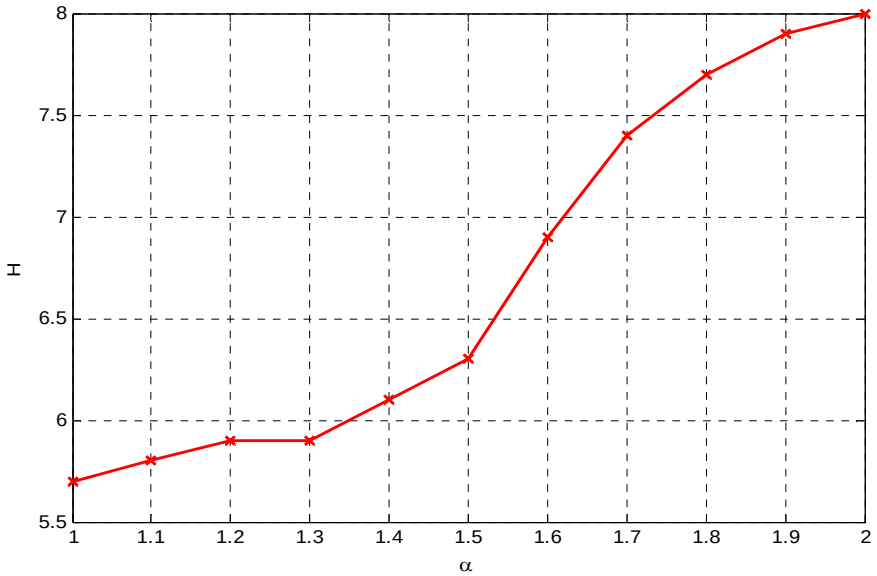


Fig.3.24 : comportement du seuil de retour en zéro H en fonction de variation de α

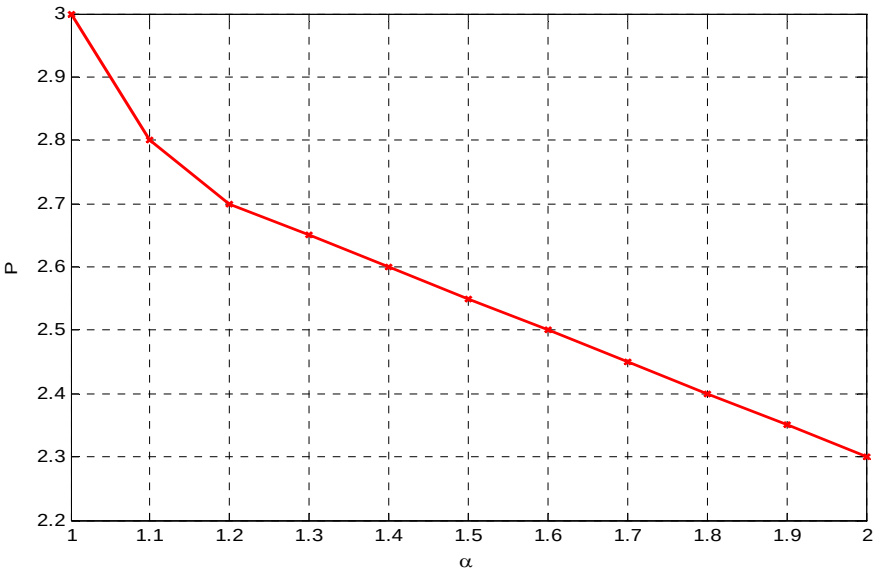
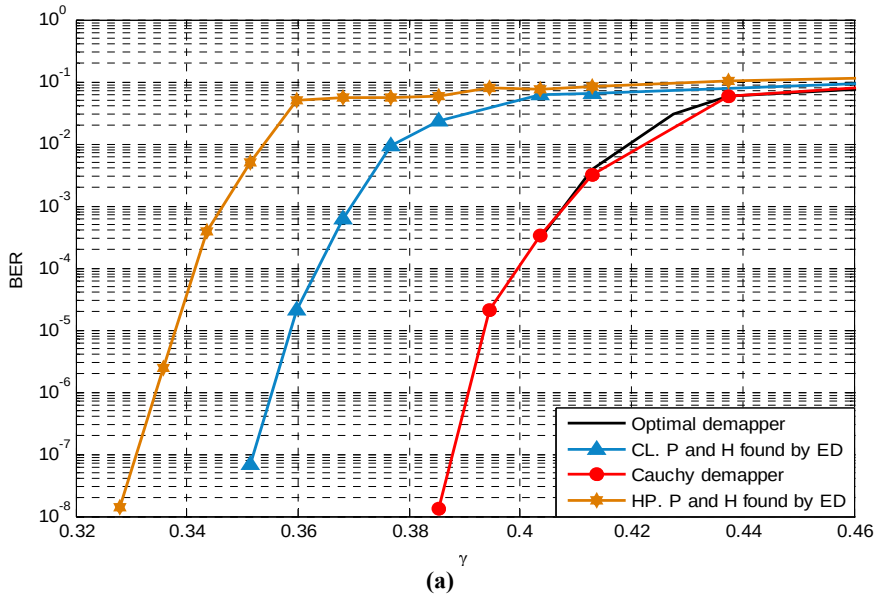


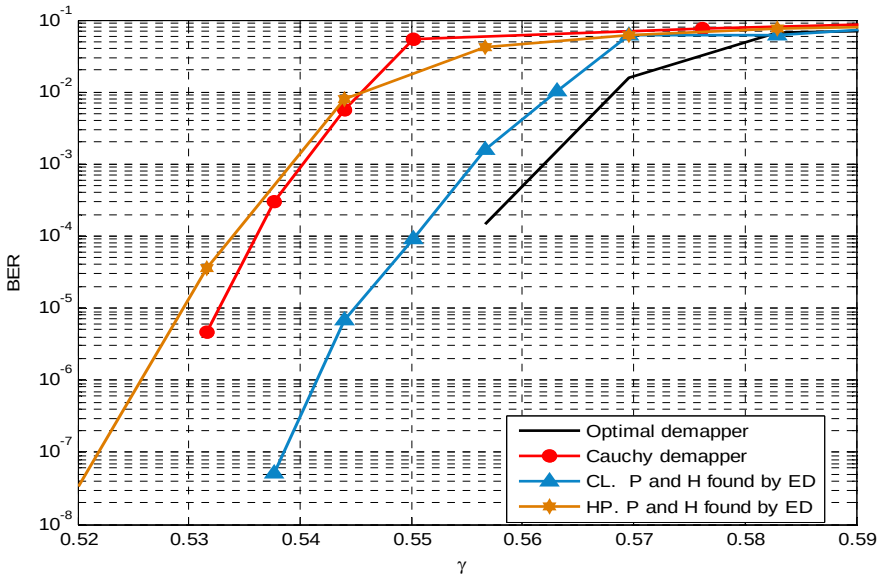
Fig.3.25 : comportement de la pente P en fonction de variation de α

Notons que la variation de la pente (figure 3.25) est assez faible, elle varie entre 2.3 et 3 quand l'impulsivité augmente. Par contre les valeurs du seuil H sont importantes en augmentant quand l'impulsivité de cet environnement diminue (α tendant vers 2). Ce démappeur se comporte comme un démappeur linéaire dans ces environnements AISαSN à cause de son seuil qui prend des grandes valeurs.

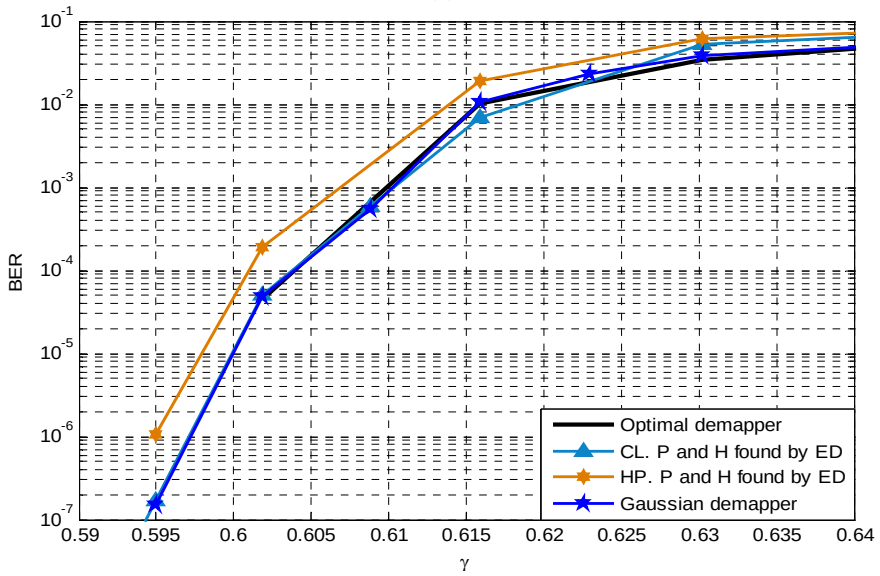
Les valeurs obtenus sur ces figures seront tabulées et utilisés pour obtenir les performances des codes LDPC. Il apparaît clairement que le seuil est réduit quand l'impulsivité augmente, réduisant ainsi le nombre de valeurs qui permettront de prendre les décisions. Les performances de ce démappeur sont présentées sur la figure 3.26.



(a)



(b)



(c)

Fig.3.26 : Comparaison des performances des démappeurs de Clipping et Hole puncher pour des P et H trouvés par évolution densité (a) α =1.2, (b) α =1.8, (c) α =2,

Si le HP se comporte relativement bien dans l'ensemble des environnements, il reste toujours moins bon que le clipping. Quand le bruit est peu impulsif, il a tendance à annuler des valeurs qui sont en fait porteuses de bonne information. La tendance à accroître nettement le seuil n'empêche pas ce phénomène. Quand l'impulsivité augmente, s'il réduit bien l'impact des fortes valeurs, il a tendance à éliminer aussi de nombreuses valeurs qui étaient fiables. Son comportement est probablement un peu trop radical. En revenant au LLR exact, le fait de couper élimine une partie des valeurs avec un fort LLR. De plus les grandes valeurs ne sont pas exactement égales 0 alors que c'est le cas du HP. Il semblerait alors opportun d'approcher un peu plus finement l'allure du LLR optimal.

3.4 Forme approchée du démappeur

Les démappeurs de Clipping et Hole Puncher ne résolvent pas entièrement le problème d'impulsivité des environnements AISαSN. Leurs performances se dégradent quand l'impulsivité augmente. Le démappeur de Clipping ramène les grandes valeurs à un seuil H alors que le Hole Puncher annule ces valeurs.

Nous proposons alors un autre type de démappeur qui présente l'intermédiaire entre ces deux derniers et tente une meilleure approximation de la courbe du LLR. Nous le nommons démappeur Approché et décrit par un simple LLR dont l'expression :

$$LLR_{ApprochZero} = \begin{cases} H & quand \quad H/P \le |y| < 3H/P \\ P.y & quand \quad 0 < |y| < H/P \\ 0 & quand \quad 3H/P \ge |y| \end{cases} \qquad (3.10)$$

Ce démappeur, représenté figure 3.27, est défini par 2 paramètres : la pente P et le seuil H. Un troisième paramètre est nécessaire : la largeur du plateau. Cependant, nous avons fixé de façon empirique l'intervalle du plateau entre H/P et 3H/P, par souci de simplification de la recherche des paramètres optimaux. Cette valeur semble donner de bons résultats d'après quelques simulations mais nécessiterait probablement une étude plus poussée.

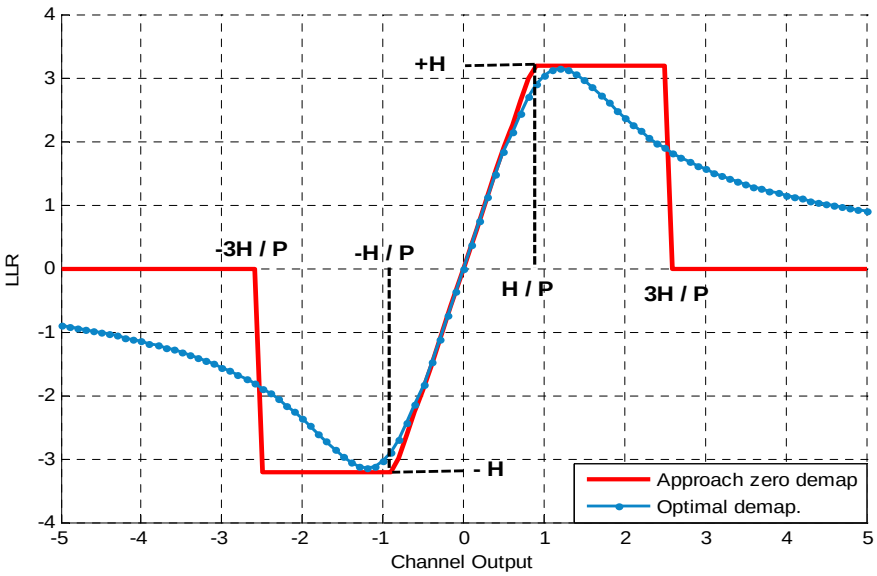


Fig. 3.27 : Comportement du démappeur approche zéro en fonction de la variation du sortie de la canal y.

Pour déterminer ces paramètres, nous avons repris l'étude qui nous a semblé la meilleure dans le cas du démappeur de clipping et que nous avons déjà appliqué pour le Hole Puncher : l'évolution de densité. Sur la figure 3.28, nous présentons les performances asymptotiques du démappeur approché en fonction des valeurs de P et de H pour α = 1.5.

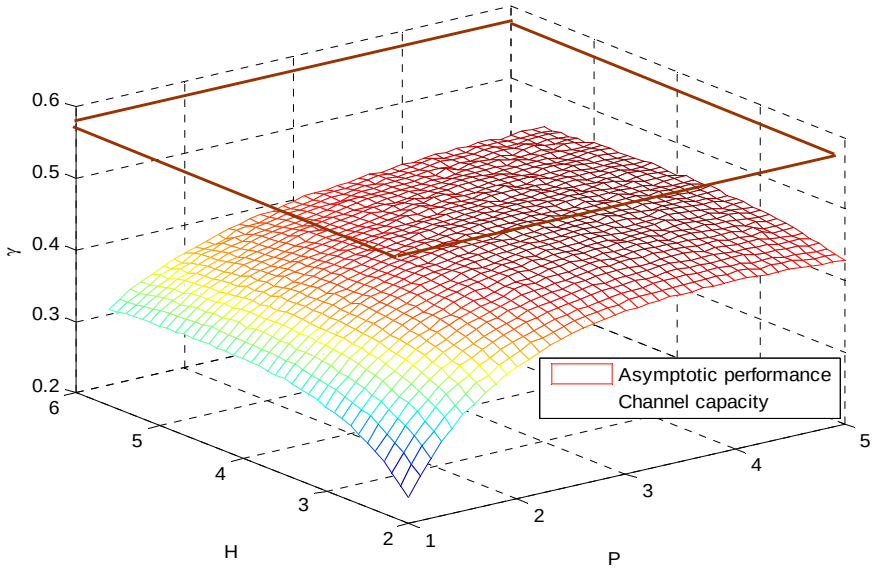
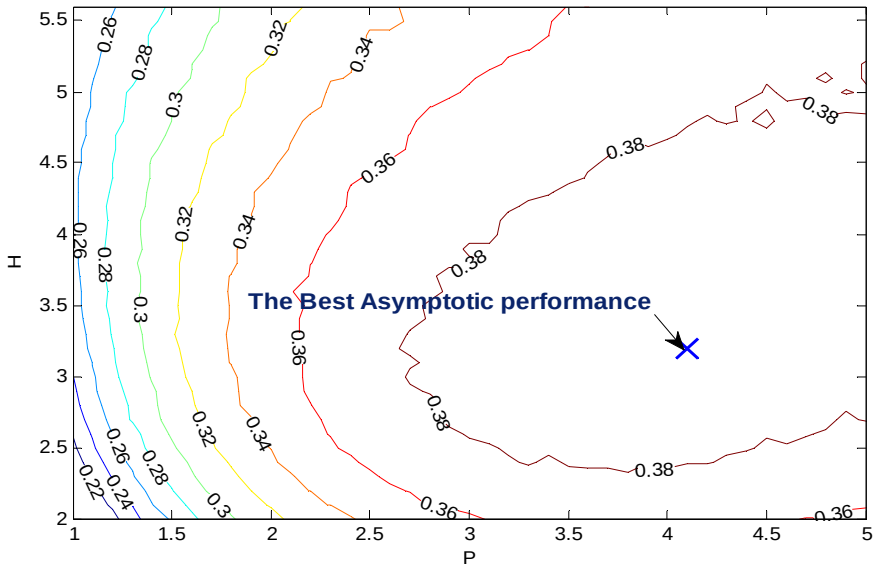
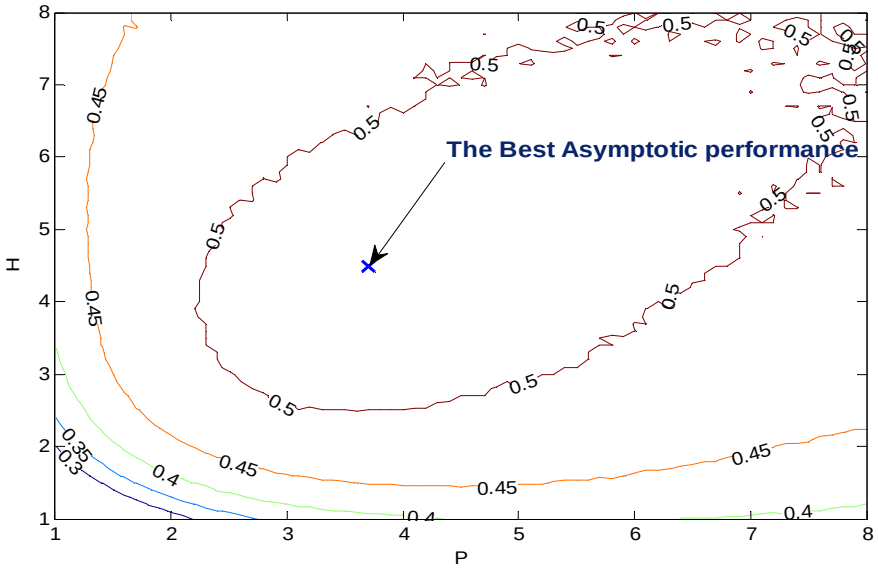


Fig.3.28 : Comportement des performances asymptotiques du démappeur Approch zero en fonction des variations de P et H dans un environnement AISαSN (α=1.5)

Comme dans les cas du clipping et du hole puncher, il apparaît qu'il existe un couple (H, P) de paramètres optimal pour ce démappeur. Nous représentons les contours de la courbe 3D (figure 3.28) pour α = 1.2 et pour α = 1.8 (figure 3.29).



(a)



(b)

Fig.3.29 : Comportement des performances asymptotiques dans différents environnements AISαSN : (a) α=1.2 (b) α=1.8

Nous remarquons à partir des résultats de la figure 3.29 que l'influence de la pente sur les performances asymptotiques est plus significative quand l'impulsivité diminue. Le seuil H dans ce cas ne varie pas beaucoup entre ces différents environnements. Sur les figures 3.30 et 3.31 nous représentons le comportement des paramètres P et H trouvés par évolution densité en fonction de la variation de α.

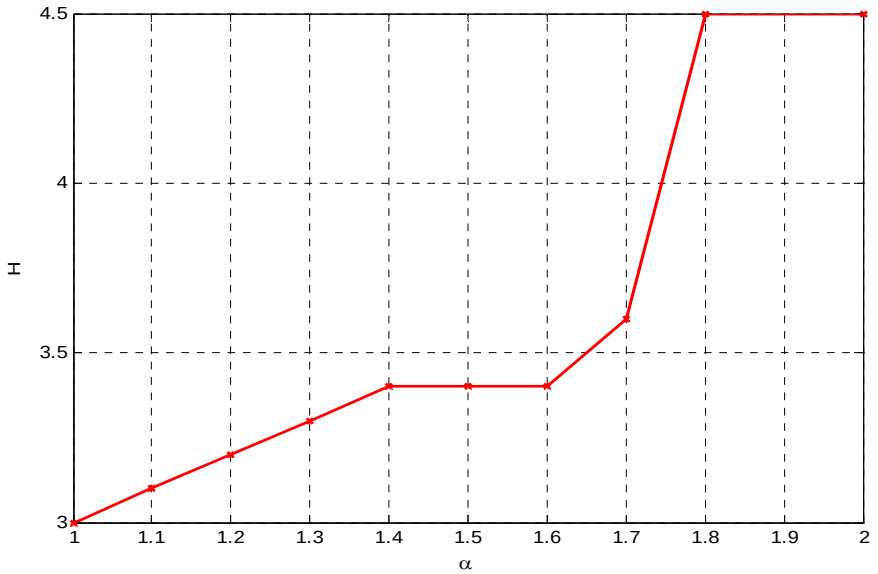


Fig.3.30 : comportement du seuil H en fonction de variation de α

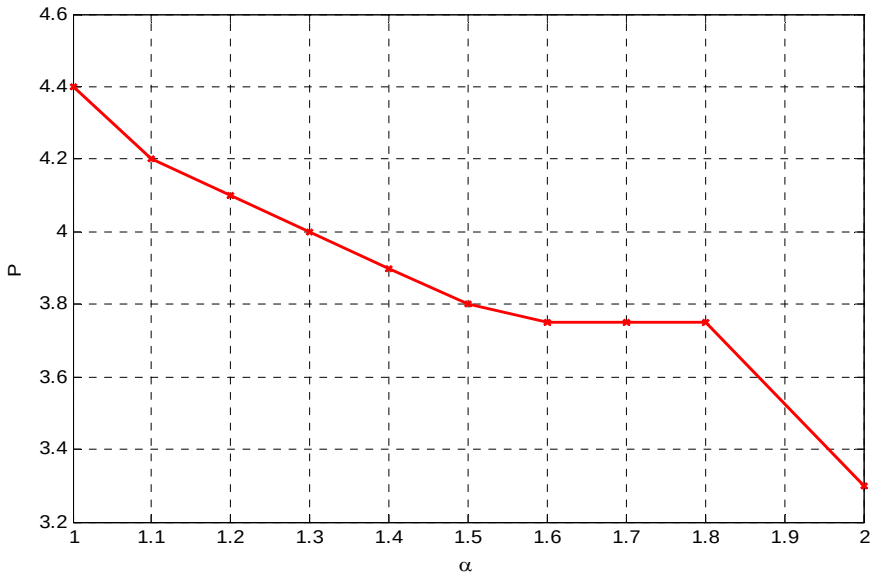
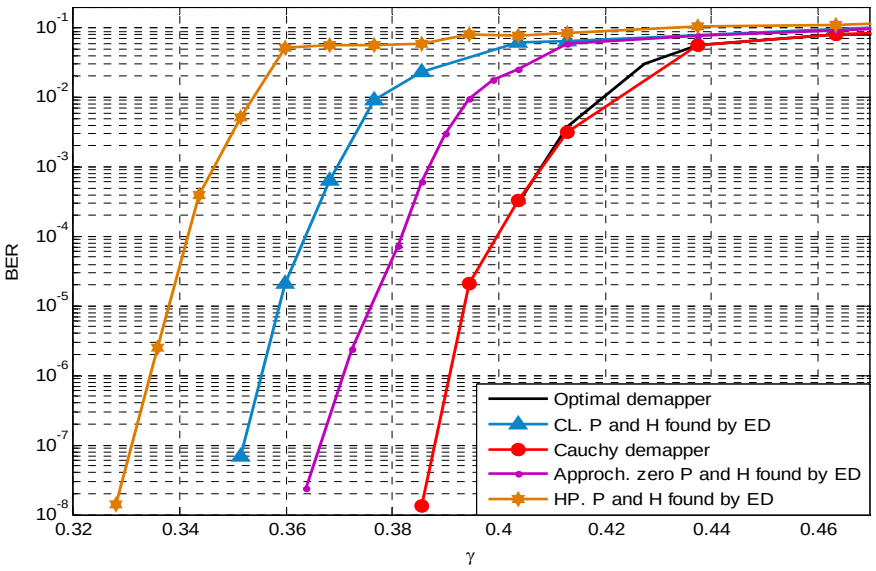


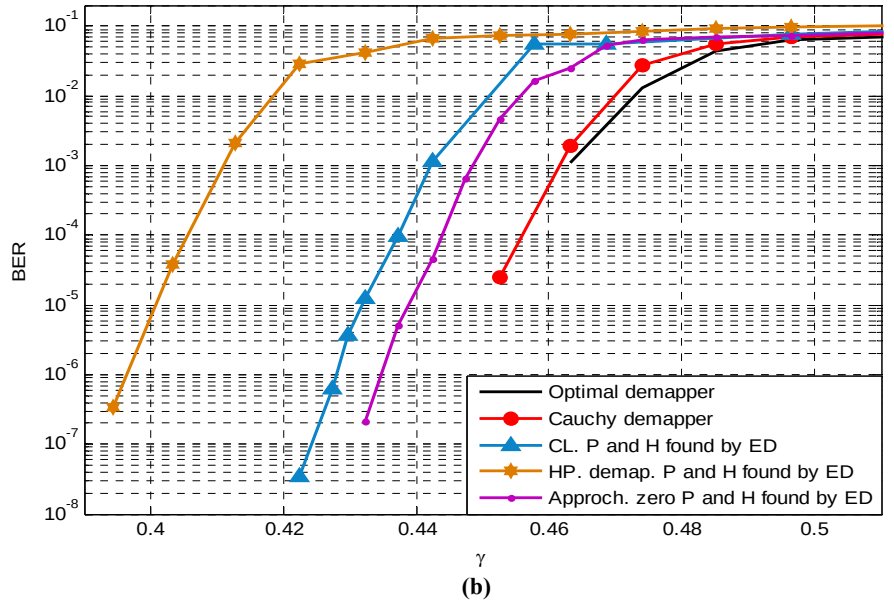
Fig.3.31 : comportement de la pente P en fonction de variation de α

Nous notons que la variation de la pente est relativement faible (figure.3.31). Cette pente prend de plus grandes valeurs quand l'impulsivité augmente. Par contre les valeurs du seuil H (figure.3.30) sont d'autant plus petites que l'impulsivité est importante. Ceci tend à limiter l'impact des fortes valeurs reçus en les limitant sur une [H/P, 3H/P] et en les annulant au delà de 3H/P, lorsque la valeur du bruit est grande et donc la fiabilité de l'échantillon faible.

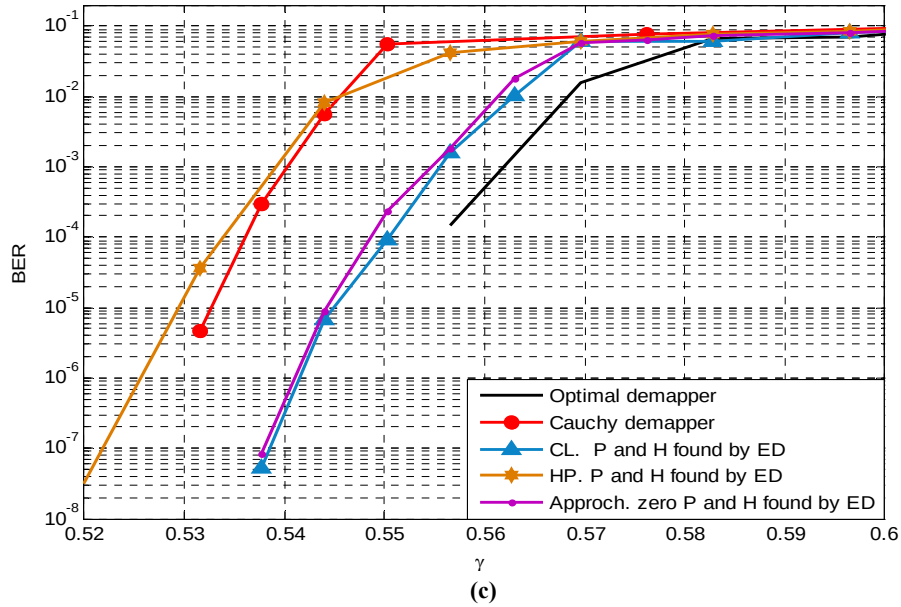
Les performances de ce démappeur avec les paramètres tabulés seront présentées la figure 3.32.



(a)



(b)



(c)

Fig.3.32 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α = 1.2, (b) α = 1.4, (c) α = 1.8

Les résultats de la figure 3.32 montrent que le démappeur approché se comporte aussi bien ou mieux que le démappeur de clipping quel que soit l'environnement. Le gain est significatif quand l'impulsivité est importante, démontrant s'il était encore nécessaire que les valeurs très importantes ont un impact important sur les performances du décodeur et qu'ils faut donc éviter d'en tenir compte.

Le démappeur approché offre donc un bon compromis entre la complexité et la qualité et ceci dans tous les environnements. Cette approche améliore significativement ce que permettent le clipping ou le hole puncher qui réduisent le calcul du logarithme du rapport de vraisemblance à des fonctions trop simples. Il est alors certainement possible d'améliorer les performances en trouvant des fonctions approchant mieux cette vraisemblance comme les fonctions proposées par Zozor dans [5]. Cependant cette étude n'entre pas dans le cadre de ce travail où nous avons voulu nous limiter à des fonctions affines par morceaux.

4 Décodage Min-Sum

Les démappeurs précédents (clipping, hole puncher et approche zéro) nécessitent d'estimer les paramètres du canal (α et γ) afin de fixer leurs paramètres. Cette approche comporte plusieurs inconvénients : d'une part cette estimation n'est pas nécessairement précise et d'autre par les paramètres H et P doivent être calculé et tabulés par avance.

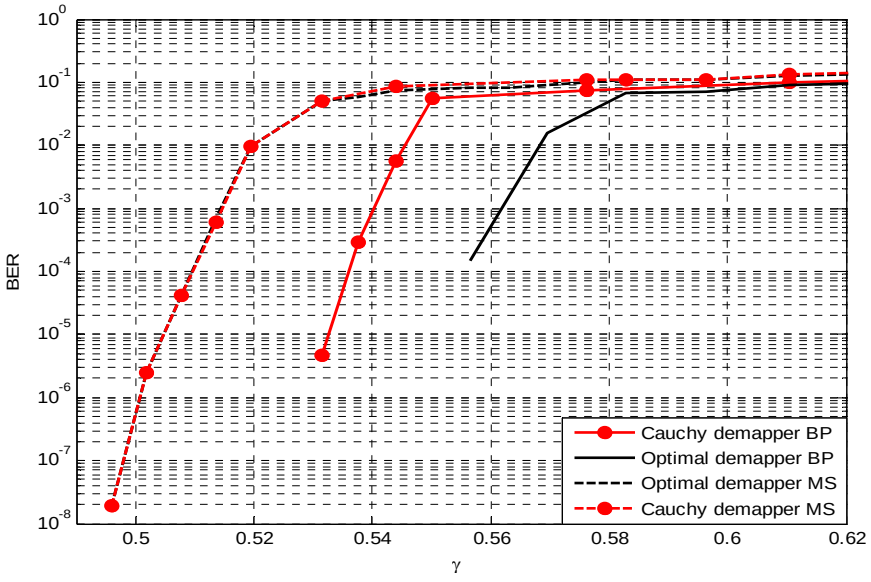
Afin de résoudre ces problèmes, nous proposons dans cette section d'utiliser l'algorithme approché de BP nommé par Min-Sum (MS). Cet algorithme qui est détaillé dans la section 4.3.3 du chapitre 2, utilise comme son nom l'indique que des sommes, des minimisations sur de petits ensembles de valeurs.

Il résout le problème de la dynamique des grandes valeurs qui apparaissent à son entrée en ne prenant pas en compte l'unité de grandeur des entrées. En effet cet algorithme est complètement indépendant d'un coefficient de magnification. L'algorithme du BP n'a pas cette propriété à cause de la présence des fonctions tangente hyperbolique et inverse. Par conséquent l'algorithme MS est moins complexe et plus souple. Ces avantages sont toutefois contrebalancés par des performances moindres.

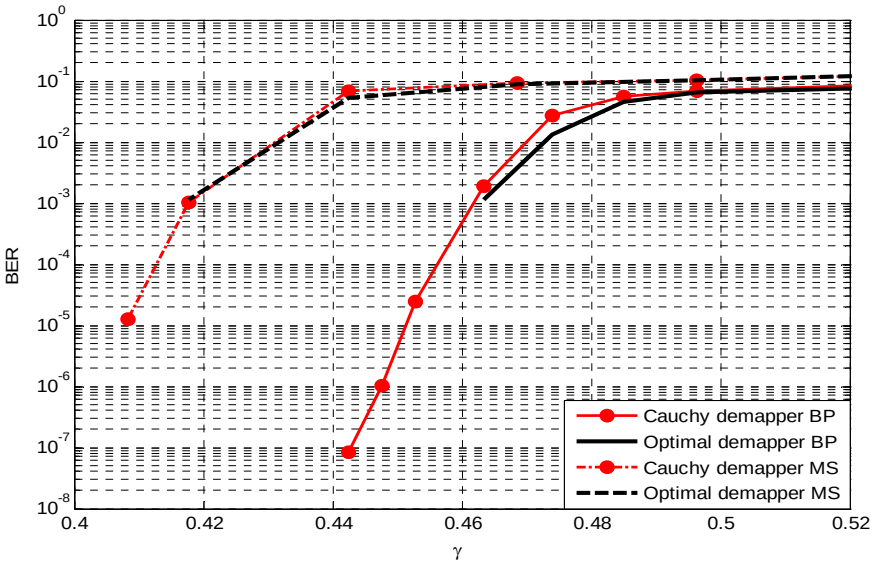
Nous proposons dans cette partie d'étudier au début le comportement des démappeurs LDPC utilisant cet algorithme MS en les comparants aux performances du BP. Dans une deuxième étape nous étudions le comportement du démappeur de clipping avec le MS en proposant différentes approches d'optimisation de son unique paramètre.

4.1 Cas des démappeurs optimal et basés sur la vraisemblance

Dans cette partie nous proposons de faire une étude comparative des performances des codes LDPC avec les algorithmes de décodage BP et MS. L'objectif est de montrer les différences de comportement du gain entre ces deux algorithmes quand la nature de l'environnement varie. Dans ce contexte les démappeurs optimaux et de Cauchy seront utilisés car nous avons vu précédemment qu'ils se comportent bien dans des environnements impulsifs. Les résultats de cette étude sont présentés par la figure 3.33.



(a)



(b)

Fig.3.33 : Comparaison des performances du code LDPC (Mac-20000) utilisant BP et MS dans différents environnements AISαSN : (a) α=1.8, (b) α=1.4

Nous remarquons à partir de ces résultats que, comme attendu, les performances des codes LDPC utilisant le MS se dégradent par rapport au BP et ceci pour les deux types de démappeurs considérés. Ceci nous amènent par la suite à étudier des solutions plus robustes mais qui permettent de conserver de bonnes performances tout en s'adaptant au contexte.

4.2 Cas du démappeur de Clipping

Le bruit impulsif est donc particulièrement nuisible au décodeur Sum-Product ou BP. Un clipping des entrées est nécessaire. Il dépend cependant de deux paramètres: la pente P et le seuil H.

Toutes les fonctions utilisées lors de l'exécution de l'algorithme Min-Sum sont homogènes, c.-à-d. qu'elles vérifient $f(a.x)=a.f(x)$ pour toute facteur d'échelle a. L'homogénéité de l'algorithme Min-Sum et le fait que les décisions binaires ne dépendent que du signe rendent cet algorithme complètement indépendant du facteur d'échelle.

Toutefois les impulsions restent nocives pour le décodage d'où le choix de faire du clipping. L'indépendance vis-à-vis du facteur d'échelle implique que seul le rapport seuil/pente du clipping intervient. Le clipping dans ce cas est caractérisé soit par le seuil H soit par la proportion des échantillons écrêtés.

Dans ce contexte nous proposons d'étudier le comportement des performances asymptotiques des codes LDPC en fonction de variation des paramètres α et γ d'environnement AISαSN. L'évolution densité permet de montrer le comportement des ces performances pour différents seuil ou taux de clipping et qui vont être présentés par les figures 3.34 et 3.35.

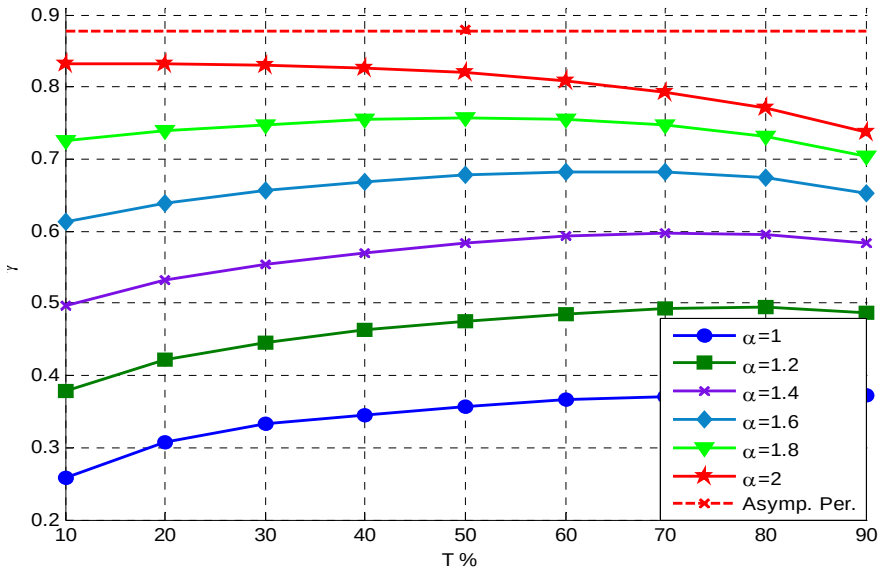


Fig.3.34 : Variation des performances asymptotiques des codes LDPC en fonction de variation du taux de clipping pour différents α

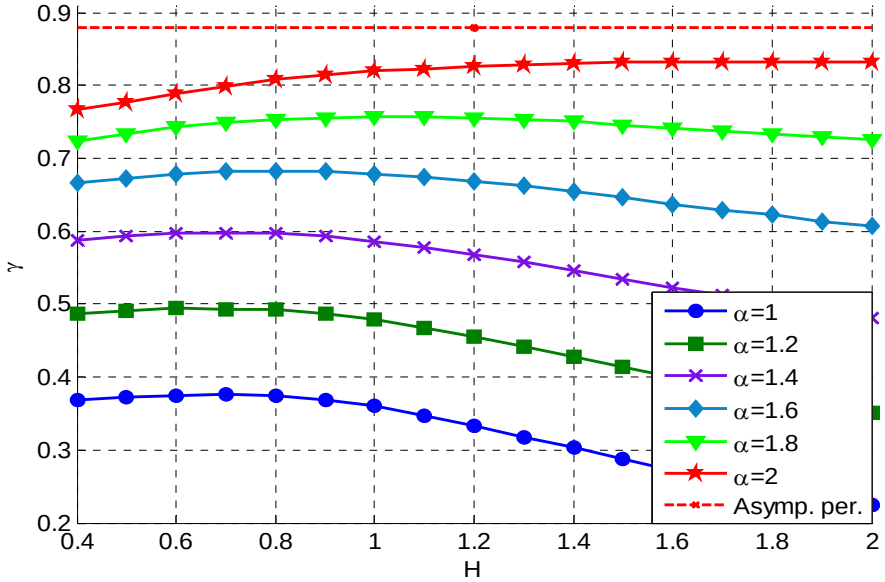


Fig.3.34 : Variation des performances asymptotiques des codes LDPC en fonction de variation du seuil de clipping pour différents α

A partir des résultats des figures 3.34 et 3.35 nous remarquons que la présence d'optimum pour chaque situation. Le taux de clipping où les codes LDPC présente des performances asymptotiques les plus proche de l'optimal, augmente quand l'impulsivité augmente alors que le niveau ou le seuil de clipping diminue dans ce contexte. Ceci prouve que ce démappeur réduit l'impacte des grandes valeurs de ces impulsions en les limitant à des petites valeurs. Ce démappeur prend des seuils de clipping inférieur à l'amplitude du signal dès que α est inférieur à 1.8.

Pour confronter les bornes théoriques de l'évolution de densité à la pratique, nous avons fait des simulations. Ces simulations utilisent le code de MacKay (3, 6) de taille 20000 qui était utilisé pour l'étude utilisant l'algorithme du BP. Seul le paramètre α = 1.2 est utilisé. Cette valeur est suffisamment typique des buits α-stable pour notre étude sans être trop spécifique comme α = 2 ou trop pathologique comme α = 1.

Les figures 3.35 et 3.36 présentent le comportement de ces performances pour différents nivaux de clipping et différents taux de clipping respectivement.

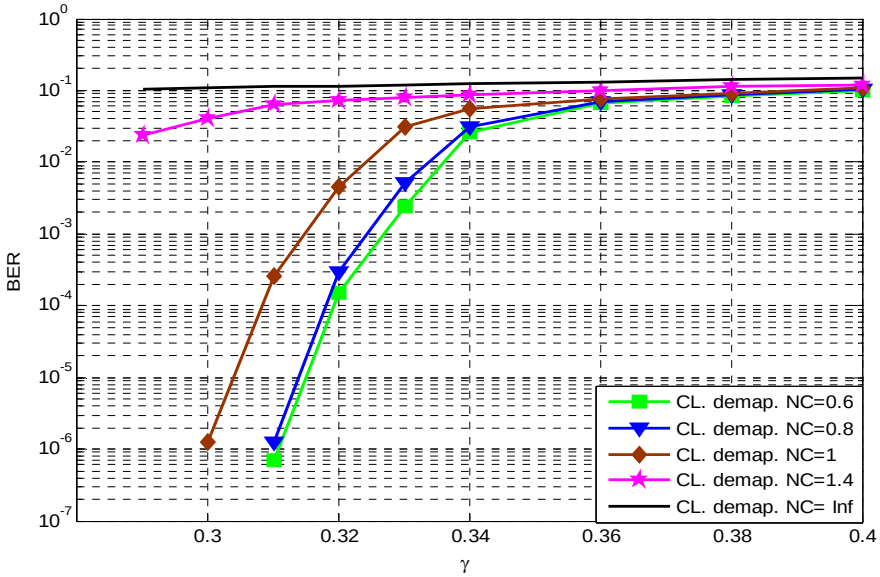


Fig.3.35 : Performances du code LDPC (Mac-20000) utilisant le démappeur de clipping de différents seuils de clipping dans un environnement AISαSN α = 1.2

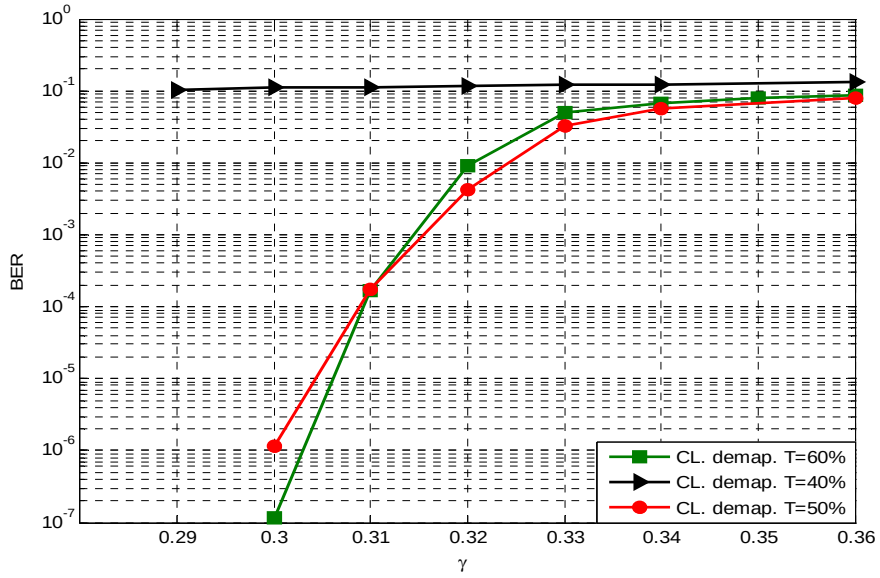


Fig.3.36 : Performances du code LDPC (Mac-20000) utilisant le démappeur de clipping de différents taux de clipping dans un environnement AISαSN α = 1.2

A partir de ces résultats nous constatons que les seuils ou le taux de clipping trouvés par évolution densité correspondent assez bien aux paramètres où les codes LDPC présentent des meilleures performances.

Afin de bien montrer la sensibilité de ces performances aux variations du taux de clipping ainsi que le seuil de clipping, nous avons fixé la valeur de la dispersion $\gamma = 0.3$ et de l'exposant caractéristique $\alpha = 1.2$ et nous avons regardé le comportement de ces performances dont les figures 3.37 et 3.38 les présentent. Le choix de γ repose sur celui d'un taux d'erreur binaire cible de l'ordre de 10^{-6} pour le meilleur taux de clipping.

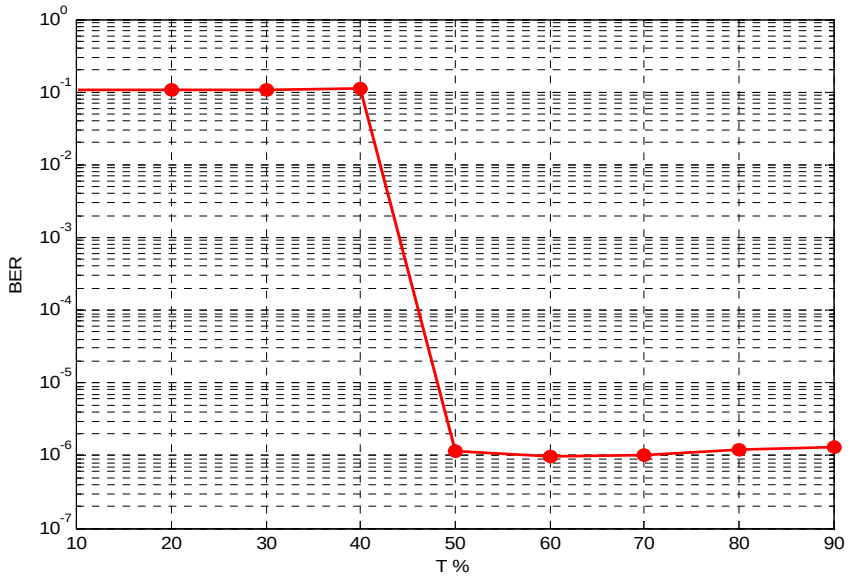


Fig.3.36 : Variation des performances du code LDPC (Mac-20000) en fonction du taux de clipping pour γ= 0.3 et α = 1.2

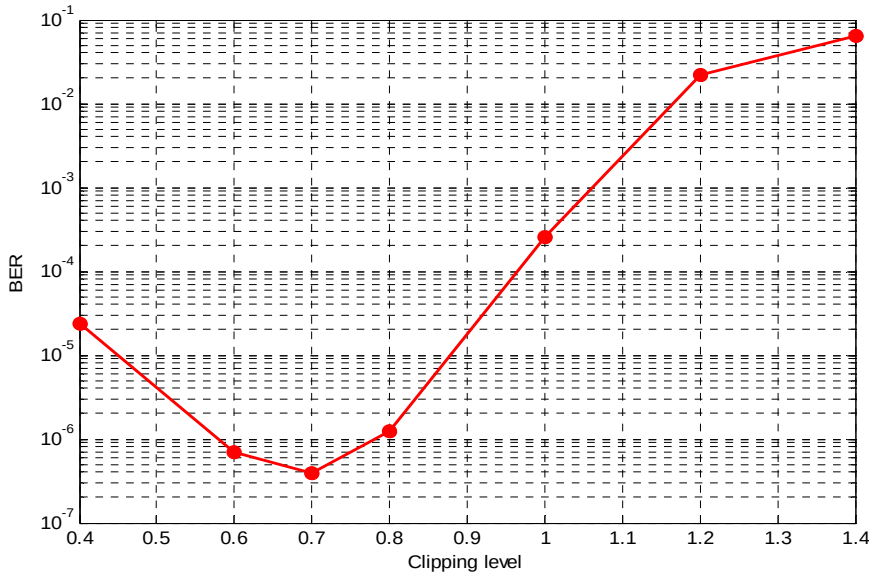


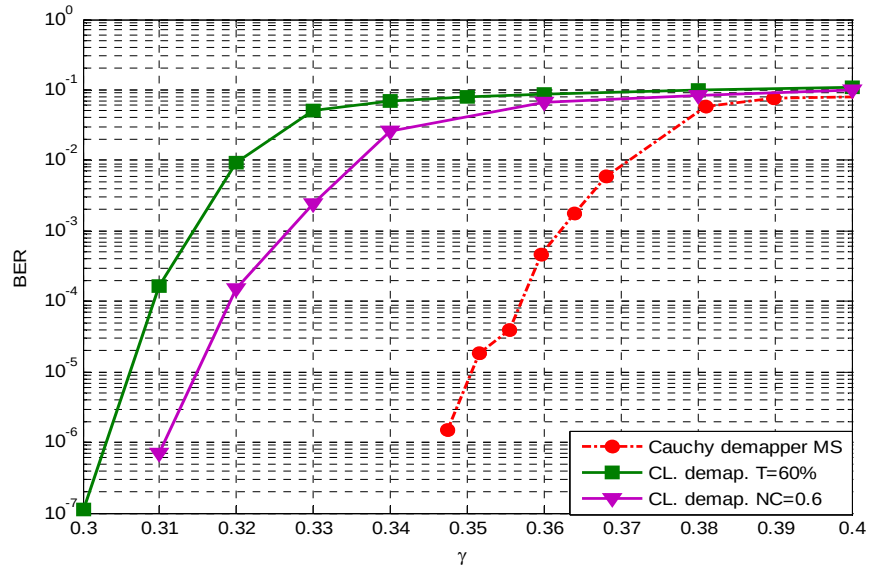
Fig.3.36 : Variation des performances du code LDPC (Mac-20000) en fonction du seuil de clipping pour γ= 0.3 et α = 1.2

Nous remarquons à partir des résultats de la figure 3.35 qu'il deux modes de fonctionnement bien distincts : soit le décodage est performant soit il ne l'est pas du tout. Cette hypothèse nous donne une idée sur le choix du taux de clipping pour chaque cas d'environnement AISαSN (valeur de α).

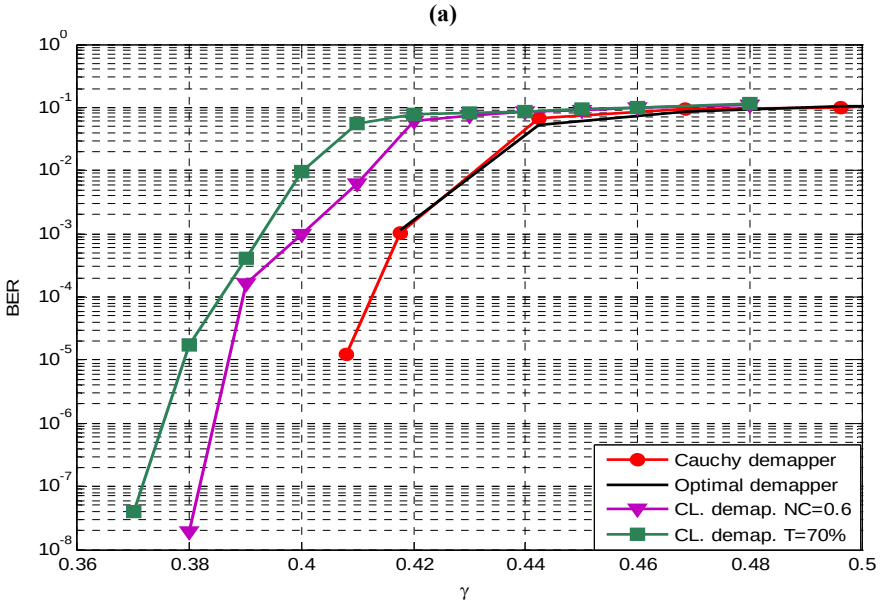
De même, la figure 3.36 nous montre qu'il existe des zones du seuil de clipping où ces codes présentent des meilleures performances. Ces zones diffèrent selon les environnements AISαSN.

Du point de vue de la robustesse il semblerait que l'utilisation du taux de clipping soit plus judicieuse que celui du niveau de clipping. En effet il y a un effet de seuil qui peut se révéler pertinent pour une mise en œuvre.

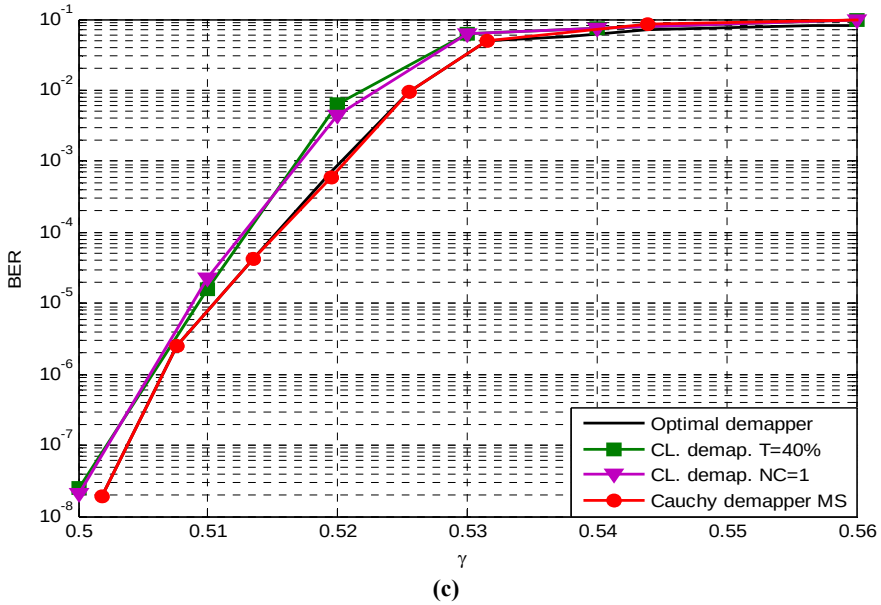
Le démappeur de clipping avec ces paramètres (taux ou seuil de clipping) tabulés par évolution densité présente des performances qui vont être comparées à celles des démappeurs optimales et de Cauchy. Ceci est présenté par la figure 3.37.



(a)



(b)



(c)

Fig.3.37 : Performances du code LDPC (Mac-20000) dans différents environnements AISαSN : (a)α = 1.2, (b) α = 1.4, (c) α = 1.8

Nous remarquons à partir des résultats de la figure.3.37 que les performances de ce démappeur utilisant ces paramètres se dégradent quand l'impulsivité augmente ($\alpha < 1.8$). Le démappeur de clipping utilisant le seuil de clipping comme paramètre se comporte mieux que celui utilisant le taux de clipping dans les environnements AISαSN.

Au regard des simulations précédentes, fixer un seuil de clipping semble plus performant que l'utilisation du taux de clipping. Toutefois en pratique le taux de clipping est intéressant. En effet, l'utilisation du seuil de clipping suppose que la pente est fixée à 1. Par conséquent la sortie du démappeur est fortement dépendante du facteur de magnification de la sortie du canal. Le taux de clipping quant à lui est simple à mettre en œuvre : il fixe la proportion des valeurs des entrées les plus grandes qui doivent être écrêtés. Un algorithme de k-sélection de complexité linéaire suffit à cette tâche.

Un second avantage du taux de clipping est la grande plage de performances acceptables. Il n'est alors pas nécessaire de faire une estimation des paramètres α et γ du canal mais de faire directement un seuillage selon une proportion qui peut être estimée directement et mis à jour au fil du temps. Cette approche évite en plus l'utilisation de l'algorithme de sélection.

5 Conclusion

Dans la première partie de ce chapitre nous avons montré que les performances des codes LDPC sont sensibles à la nature de démappeur. Il n'existe pas de démappeur à la fois un simple et robuste qui s'adaptent facilement aux environnements AISαSN. Ceci nous a amené en deuxième partie de ce chapitre à étudier d'autres types de démappeurs. Nous avons proposé différents démappeurs dont chacun présente des avantages et des inconvénients :

– le démappeur linéaire avec la forme la forme simple de son LLR qui ne dépend que de la pente s'adapte parfaitement à un canal AWGN mais ces performances se dégradent et devient très mauvaises quand l'impulsivité augmente ($\alpha < 2$). Ce démappeur donne aux grandes valeurs de

ces processus une grande importance et, surtout, laisse croire qu'elles sont très fiables alors, qu'en réalité, elles sont le résultat d'une impulsion de bruit ;

- le démappeur de clipping avec la forme simple de son LLR qui dépend de 2 paramètres, P et H, résout certains problèmes du démappeur linéaire en limitant l'impacte des grandes valeurs de bruit qui masquent l'information en les seuillant. Ce démappeur présente des performances proches de l'optimal dans l'environnement AWGN et ses performances ne se dégradent pas beaucoup par rapport à l'optimal quand cet environnement devient impulsive ($\alpha < 2$). Deux approches d'optimisation des P et H de ce démappeur ont permis de s'adapter facilement à tous les environnements AISαSN. La première approche est basée sur l'évolution de densité et elle permet de tabuler les P et H en fonction de la nature d'environnements AISαSN (valeur de α) alors que la deuxième approche qui est basée sur la minimisation de la puissance de bruit à la sortie du démappeur permet de trouver un seuil H en fonction de α et γ de cet environnement pour une pente fixe (dans notre cas à P = 3.25). Ces différents approches montrent que l'estimation de l'exposant caractéristique α est plus critique que celle de la dispersion γ afin d'optimiser les P et H de ce démappeur : la mauvaise estimation de α n'a pas un effet aussi importante sur les performances. Cette approche qui optimise les P et H en fonction de α a été appliquée aux autres types de démappeur ;

- contrairement aux dernier démappeur, le démappeur hole puncher annulent les grandes valeurs des entrées du décodeur. Il dépend aussi de deux paramètres : la pente P et seuil H à partir duquel les valeurs sont annulées. Tout comme pour le démappeur de clipping, l'évolution de densité a été utilisée afin d'optimiser les paramètres P et H de ce démappeur. Les codes LDPC avec ce démappeur s'adaptent cependant difficilement aux différents environnements AISαSN. Il a tendance à éliminer des valeurs qui sont pourtant porteuse d'information. Ceci nous a permis de construire par la suite un démappeur qui se rapproche de l'allure du LLR de démappeur optimal ;

- le démappeur approché représente un intermédiaire de faible complexité entre le clipping et hole puncher. Il dépend aussi de deux paramètres P et H. Ce démappeur avec ses paramètres optimisés par évolution de densité en fonction de α s'adaptent bien à tous les environnements AISαSN particulièrement les canaux très impulsifs ($\alpha = 1.2$). Il présente des performances les plus proches de l'optimal si nous les comparons aux derniers démappeurs.

En troisième partie de ce chapitre, afin de résoudre les problèmes de complexité numérique et de mauvaise adaptation d'algorithme de décodage BP surtout dans les environnements très impulsifs ($\alpha < 1.2$) nous avons proposé d'utiliser un autre algorithme approché de BP nommé par Min-Sum. Cet algorithme simplifie énormément la complexité du décodage mais il est moins performant que le BP avec les différents démappeurs et résolut la mauvaise adaptation de BP dans les environnements très impulsifs.

Le démappeur de clipping avec cet algorithme de décodage MS présente une complexité réduite par rapport au BP. Les décisions rendues par l'algorithme du Min-Sum est complètement indépendant du facteur d'échelle. Par contre, il reste sensible aux impulsions de bruit à son entrée. Par conséquent l'utilisation du démappeur de clipping reste d'actualité Le démappeur de clipping dans ce cas est caractérisé par un seul paramètre : soit un seuil H, soit la proportion des échantillons étêtés. Ces paramètres ont été optimisés par évolution de densité en fonction du paramètre α du canal. Les

résultats montrent que ce démappeur s'adapte bien dans les environnements moins impulsifs et ces performances se dégradent quand l'impulsivité augmente.

Conclusions et Perspectives

L'objectif de cette thèse est de concevoir une stratégie simple et robuste de décodage des codes LDPC dans des environnements impulsifs.

Nous avons dans le premier chapitre étudié différents modèles permettant de représenter les processus de nature impulsive qui apparaissent dans les environnements de transmission. Notre choix s'est porté sur la distribution symétrique α-stable qui, du fait de sa propriété de stabilité et de sa queue lourde, permet une justification physique et une justification mathématique de ce choix. Elle est définie par deux paramètres : l'exposant caractéristique α et la dispersion γ. Plusieurs articles ont montré la validité de ce choix dans des réseaux *ad hoc*. Nous avons alors défini le canal de transmission à bruit additif symétrique α-stable à échantillons indépendants noté par AISαSN et en avons déterminé la capacité. En dehors de l'absence d'expression analytique de la densité de probabilité (à quelques exceptions près), l'une des difficultés rencontrées est la variance infinie dés que α est strictement inférieur à 2. Ceci implique que le rapport signal à bruit tel qu'il est habituellement défini est nul. Pour remédier à cette difficulté, il est possible d'utiliser la théorie des statistiques d'ordre zéro et de définir, à partir des moments d'ordre logarithmique, la puissance géométrique et la qualité d'une transmission par le SNR géométrique. Une seconde approche que nous avons également utilisée est d'évaluer les performances (taux d'erreurs binaires) directement en fonction de la dispersion pour un exposant caractéristique fixé.

Le second chapitre présente les codes LDPC et décrit les différents algorithmes de décodages et les outils d'analyse des performances asymptotiques. Ces études nous ont en particulier permis d'évaluer l'impact de l'entrée du décodeur. Cette entrée est obtenue (par le démappeur) en calculant le logarithme du rapport de vraisemblance. Sa forme exacte est cependant difficile à obtenir du fait de l'absence d'expression analytique des densités de probabilité. Des calculs numériques (à partir de la fonction caractéristique) sont possibles mais évidemment difficile à mettre en œuvre dans un environnement réel. Nous avons donc étudié les performances de différents démappeurs et noté que le démappeur linéaire (faisant l'hypothèse d'un bruit gaussien) a de très mauvaises performances. Le gain potentiel en adaptant le démappeur s'avère très significatif c'est pourquoi nous consacrons la suite du travail à la recherche d'une stratégie à la fois assez simple à mettre en œuvre mais donnant des performances proches de celles que donneraient l'expression exacte du LLR.

La troisième partie est consacrée à l'étude des performances d'un code LDPC spécifique afin (1) de confirmer l'importance du démappeur et (2) de trouver un démappeur performant. Les démappeurs étudiés sont les démappeurs linéaire, clipping, hole puncher et une fonction approchant l'allure du LLR.

- Le démappeur linéaire est optimal dans l'environnement AWGN. Ses performances se dégradent rapidement quand l'impulsivité augmente ($\alpha < 2$). Ce démappeur donne aux grandes valeurs une grande importance et laisse ainsi croire qu'elles sont très fiables alors, qu'en réalité, elles ne le sont pas. Il est alors incapable de bien se comporter dans un environnement impulsif.

- Le démappeur de Clipping tente de résoudre cette difficulté en limitant l'impact des grandes valeurs. La fonction du démappeur dépend de 2 paramètres P et H. Nous avons proposé plusieurs approches pour déterminer les paramètres P et H optimaux, à partir de l'allure exacte du LLR, de la puissance de bruit en sortie du démappeur ou en utilisant directement l'évolution de densité. Cette dernière approche présente l'avantage de ne faire reposer les paramètres que sur la valeur de α. Si le démappeur de clipping a de bonnes performances pour l'ensemble des valeurs de α testées (entre 1 et 2) il reste un gap non négligeable par rapport au démappeur optimal quand l'impulsivité augmente.

- Le démappeur Hole puncher enlève toute influence aux grandes valeurs reçues en les annulant avant le décodage. L'évolution densité a été utilisée afin d'obtenir les paramètres du démappeur : la pente P et le seuil au delà duquel les valeurs sont remises à 0. Nous avons constaté que cette approche est moins bonne que le clipping, supprimant une information parfois significative. Le clipping apparaît comme une approche plus robuste dans un environnement impulsif.

- Le démappeur approché est un intermédiaire entre le clipping et le hole puncher. Les valeurs au delà d'un premier seuil (H) sont remplacées par H mais celles au delà d'un second seuil (que nous avons pris égale à 3H) sont annulées. les paramètres de ce démappeur ont été obtenus par évolution densité. Il s'adapte bien aux différents environnements AISαSN et permet un gain significatif en comparaison du clipping pour les environnements les plus impulsifs ($\alpha = 1.2$).

La dernière partie de ce chapitre propose d'utiliser l'algorithme de décodage min-sum plutôt que la propagation de croyance utilisée jusque là. Cette approche permet en particulier de réduire la complexité numérique et de ramener la définition du démappeur à un unique paramètre (le seuil) au prix d'une perte de performance. Nous proposons alors de baser le choix du seuil sur la valeur de α et le taux de clipping, ce qui évite d'estimer la dispersion. Il s'avère cependant que cet algorithme de décodage est moins robuste quand l'impulsivité augmente.

Le travail accompli durant cette thèse constitue un point de départ à toute une série de travaux possibles :

- L'amélioration de la fonction permettant d'approcher le LLR tout en conservant une forme analytique relativement simple à mettre en œuvre. Zozor dans [102] a proposé plusieurs solutions et les a appliquées à la détection dans un bruit α-stable.

- Il serait également intéressant d'étudier ce que pourrait apporter l'analyse des performances asymptotiques des codes LDPC par les EXIT charts, ce qui permettrait également de simplifier la complexité numérique par rapport à l'évolution densité.

- A l'interférence s'ajoute un bruit thermique dans les récepteurs. Si nous l'avons considéré négligeable, il serait pertinent de considérer la somme d'une vraiable aléatoire α-stable et d'un bruit gaussien, comme le fait par exemple Beaulieu dans [103], [104], Le cadre de notre étude s'appliquerait alors assez directement car il est possible d'écrire simplement la fonction caractéristique de cette somme.

Bibliographie

[1] L. Jeongkeun and L. Sung-Ju, "***Understanding Interference and Carrier Sensing in Wireless Mesh Networks***". IEEE Commu. Mag., vol 7, pp. 102-109, Jul. 2009

[2] M. Chiani, Moe Z. Win, H. Shin, "***MIMO Networks: The Effects of Interference***". IEEE Trans. Inf. Theory, vol. 56, pp. 336-349, Jan. 2010

[3] E. Del Re, R. Fantacci, S. Morosi, A. Pugi, "***Advanced Blind Adaptive Multi-User Detector for Communications in Nonstationary Multipath Fading Channel***". IEEE Trans. on veh. Tech., vol. 50, pp. 1497-1506, Nov. 2001

[4] S. V. Vaseghi,"***Advanced Digital Signal Processing and Noise Reduction***". Third Edition John Wiley & Sons Ltd, 2006.

[5] B. S. Everitt, A. Skrondal, "***The Cambridge dictionary of Statistics***". Fourth Edition Cambridge university press, 2010.

[6] W. Feller, "***An Introduction to Probability Theory and its Applications***, ***volume I of Wiley Series in Probability and Mathematical Statistics***". John Wiley & Sons, New York, 1970.

[7] R. Durrett, "***Probability: Theory and Examples***". 4th edition Cambridge Series in Statistical and Probabilistic Mathematics, 2010.

[8] M. K. Simon. "***Probability Distributions Involving Gaussian Random Variables: A Handbook for Engineers, Scientists and Mathematicians***". Springer-Verlag GmbH, 2007.

[9] M. Zimmermann and K. Dostert, "***Analysis and modeling of impulsive noise in broad-band powerline communications***". IEEE Trans. Electromagn. Compat., vol. 44, pp. 249–258, Jun. 2002.

[10] J. W. Cook, "***Wideband impulsive noise survey of the access network***". BT tech. journal, vol. 11, pp. 155–162, 1993

[11] M. G. Sanchez, A. V. Alejos, and I. Cuinas, "***Urban wideband measurement of the UMTS electromagnetic environment***". IEEE Trans. Veh. Tech., vol. 53, pp. 1014–1022, Jul. 2004.

[12] P.C Pinto, C-C. Chong, A.Giorgetti, M. Chiani and M. Z. Win. "***Narrowband Communication in a Poisson Field of Ultrawideband Interfers***". In the IEEE Inter. Conf. on Ultra-Wideband (ICUWB), pp. 387-392, Sep, 2006.

[13] J. Ilow and D.Hatzinakos, "***Analytic alpha-stable noise modeling in a Poisson field of interferers or scatterers***". IEEE. Trans. on Sig. Process., vol.46, pp.1601-1611, Aout 1997.

[14] E.Sousa, "***Performance of a spread spectrum packet radio network link in a Poisson field of interferers***". IEEE Trans. Info. Theory, vol.38, pp.1743-1754. Nov. 1992.

[15] S. A. Kassam, "***Signal Detection in Non-Gaussian Noise***". Springer-Verlag, 1988.

[16] J. H. Miller and J. B. Thomas, "***The detection of signals in impulsive noise modeled as a mixture process***", IEEE Trans. on Comm., vol.24, pp 559-563, May 1976.

[17] D. F. Nahimana, "***Impact des multi trajets sur les performances des systèmes de navigation par satellite : Contribution à l'amélioration de la précision de localisation par modélisation bayèsienne***". Ph.D. dissertation, Ecole centrale Lille, 19 Feb. 2009.

[18] T. Erseghe, V. Cellini, and G. Don, "***On UWB Impulse Radio Receivers Derived by Modeling MAI as a Gaussian Mixture Process***". IEEE Trans. On Wireless Comm., vol. 7, No. 6, Jun. 2008.

[19] A. P. Dempster, N. M. Laird and D. B. Rubin, "***Maximum Likelihood from Incomplete Data via the EM Algorithm***". Journal of the Royal Statistical Society, vol.39, pp.1–38, 1977.

[20] R. Fisher and L. Tippet, "***Limiting Forms of the Frequency Distribution of the Largest or Smallest Member of a Sample***". Proceedings of the Cambridge Philo. Society, vol.24, pp.180-190, 1928.

[21] R. Fisher and L. Tippet, "***On the tails of extreme event distributions in hydrology***". Proceedings Elsevier Journal of Hydro., vol. 355, pp. 180-190, 2008.

[22] P. Hubert and H. Ben djoudi, "***Introduction à l'étude des longues séries pluviométriques***". In: Journées hyrologiques de l'ORSTOM, Montpellier, p. 20, 1996.

[23] P. S. Laplace, "***Mémoire sur la probabilité des causes par les événements***". Mémoires de Mathématique et de Physique, vol.6, pp.621-656, 1774.

[24] M. Abramowitz and I. A. Stegun, "***Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables***", 9th printing. New York: Dover, pp. 930, 1972

[25] N. C. Beaulieu and B. Hu, "***A Soft-Limiting Receiver Structure for Time-Hopping UWB in Multiple Access Interference***". IEEE Ninth Inter. Symp. on Spread Spect. Tech. and App., pp 417-421, Feb. 2006

[26] M. N. Desai and R. S. Mangoubi, "***Robust Subspace Learning and Detection in Laplacian Noise and Interference***". IEEE. Trans. on Sig. Process., vol. 55, no. 7, Jul. 2007.

[27] C. Goodall, "***M-estimators of location: An outline of the theory***". In D. C. Hoaglin et al, editor, Understanding Robust and Exploratory Data Analysis, chapter 11, pp. 339-403, John Wiley and Sons, New York, 1983.

[28] A. B. Varanasi, "***Parametric Generalized Gaussian Density Estimation***". J. Acoust. Soc. Am., vol.86, pp. 1404-1405, Oct. 1989.

[29] M. Abramowitz and I. Stegum, "***Handbook of mathematical Function***". New York: Dover, 1970.

[30] J.H. Miller and J. B. Thomas, "***Detectors for discrete time signals in non Gaussian noise***". IEEE Trans. on Info. Theory, vol.18, pp. 241-250, Mar. 1972.

[31] L. Sharifi and L. Garcia. "***Estimation of Shape parameter for Generlized Gaussian distribution in Subband Decomposition of video***". IEEE trans. On cir. and sys. for vid. Tech., vol. 5, pp. 52-56, Feb. 1995.

[32] N. C. Beaulieu and D. J. Young. "***Designing time-hopping ultrawide bandwidth receivers for multiuser interference environments***". Proceedings of the IEEE., vol. 97, pp. 255–284, Feb. 2009.

[33] J. Fiorina, "***A simple IR-UWB receiver adapted to multi-user interferences***". in Proc. IEEE Global Telecom. Conf., pp. 1-4, Nov. 2006.

[34] H. El ghannudi, L. Clavier, P.A. Rolland, "***Modélisation du MAI d'un Système IR-UWB Transposè à 60 GHz pour des Réseaux Ad hoc***". Colloque GRETSI, 11-14 septembre 2007, Troyes.

[35] A. Zoubir and R. Brcich, "***Multiuser detection in non-gaussian channels***". Digital Sig. Proces., vol12, pp 262-273, 2002.

[36] R. Adler. R Feldman, and M. S. Taqqu, "***editors. A Practical Guide to Heavy Tails: Statistical Techniques for Analyzing Heavy-Tailed Distributions***". Birkhauser, Boston, 1997.

[37] C. L. Nikias, and M. Shao, "***Signal Processing with Alpha-Stable Distributions and Applications***". John Wiley & Sons, New York. (1995).

[38] E. J. Wegman, S. G. Schwartz, and J. Thomas. "***Topics in non-Gaussian Signal Processing***", Academic Press", New York, 1989.

[39] G. Samorodnitsky and M. Taqqu, "***Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance***". Chapman & Hall, New York. 1994.

[40] K. G. Hagstroem, "***Remarks on Pareto distributions***". Skandinavisk Aktuarietidskrift, pp 59-71, 1960.

[41] V. Pareto, "***Cours d'Économie Politique***". Rouge and Cie, Lausanne and Paris 1997.

[42] J. R. M. Hosking, J. R. Wallis, "***Parameter and Quantile Estimation for the Generalized Pareto Distribution***", Technometrics, Vol. 29, No. 3, pp.339-349, 1987.

[43] H. M. Hall, "***A new model for impulsive phenomena: Application to atmospheric-noise communication channels***". Technical Reports 3412-8 and 7050-7, Stanford Electronics Laboratories, Stanford University, Stanford, California, August 1966.

[44] P. Lévy. "***Théorie des erreurs. la loi de gauss et les lois exceptionnelles***". Bulletin de la Société Mathématique de France, vol 52, pp. 49–85, 1924.

[45] F. Bassi, P. Embrechts and M. Kafetzaki, "***Risk management and quantile estimation***". In Adler, Feldman and Taqqu (Eds.), A practical guide to heavy tails, pp 111–130, 1998, Birkhäuser Boston.

[46] A. Bestravos, M. Crovella and M. Taqqu, "***Heavy-tailed probability distributions in the world wide web***". In Adler , Feldman et Taqqu (Eds.), A Practical Guide to Heavy Tails, pp. 3–25, 1998, Birkhäuser Boston.

[47] A. Karasaridis and D. Hatzinakos, "***Network heavy traffic modeling using alpha-stable self-similar processes***". IEEE Trans. on Comm., vol 49, pp.1203–1214, 2001.

[48] V. M. Zolotarez. "***One dimensional Stable Distribution***". American Mathematical society. 1986.

[49] E. Sousa, "***Performance of a spread spectrum packet radio network in a poisson field of interferers***," IEEE Trans. Info. Theory, vol. 38, pp. 1743–1754, Nov.1992.

[50] J. Ilow and D. Hatzinakos, "***Impulsive noise modeling with stable distributions fading environments***". in 8th IEEE Sig. Process. Workshop on Stat. Sig. and Array Process., pp. 140–143, Jun. 1996.

[51] P. Pinto, C.-C. Chong, A. Giorgetti, M. Chiani, and M. Win, "***Narrowband Communication in a Poisson Field of Ultrawideband Iinterferers***". in The IEEE 2006 Inter. Conf. on Ultra-Wideband (ICUWB), pp. 387–392, Sept. 2006.

[52] M.Z. Win, P.C. Pinto and L.A. Shepp, "***A Mathematical Theory of Network Interference and Its Applications***", Proceedings of the IEEE, Vol. 97, No. 2, February 2009

[53] H. El Ghannudi, L. Clavier, N. Azzaoui, F. Septier, and P.A. Rolland. "***Alpha-stable interference modeling and Cauchy receiver for an IR-UWB ad hoc network***" in IEEE Trans. on Com., Vol 58, n° 6, pp 1748-1757, Jun. 2010.

[54] A. N. Shiryayev, "***Probability. In Graduate Texts in Mathematics***". volume 95. Springer-Verlag, 1984.

[55] Z. Sun, C. Han. "***Parameter estimation of positive alpha-stable distribution based on negative moments***". in The IEEE ICASSP 2007.

[56] J. G. Gonzalez, J. L. Paredes, and G. R. Arce, "***Zero-order statistics: A mathematical framework for the processing and characterization of very impulsive signals***". IEEE Trans. Sig. Process., vol. 54, no. 10, pp. 3839–3851, Oct. 2006.

[57] J. G. Gonzalez, "***Robust techniques for wireless communications in non-Gaussian environments***". Ph.D. dissertation, Dept. Elec. Eng., Univ. of Delaware, Newark, DE, 1997.

[58] R. Durrett. "***Probability Theory and Examples***". Duxbury Press, Belmont, second edition, 1996

[60] T. Richardson and R. Urbanke, "***Modern coding theory***". Cambridge University Press, Oct. 2007.

[61] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "***Design of capacity-approaching irregular low-density parity-check codes***", IEEE Trans. on Info. Theory, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[62] T. Richardson, and R. L. Urbranke "***The capacity of Low Density Parity-Check codes Under Message-Passing Decoding***". IEEE Trans. Info. Theory, vol47, no.02, Feb. 2001.

[63] H. Luediger & B. Kull ."***UWB Multi-User Interference Signal Considerations***" Whyless communication.

[64] Kr. Gomadam, V. R. Cadambe, S. A. Jafar, " ***Approaching the Capacity of Wireless Networks through Distributed Interference Alignment***". in The IEEE GLOBECOM 2008 proceedings.

[65] C. Berrou, A. Glavieux, and P. Thitimajshima,*" **Near shannon limit error-correcting coding and decoding :turbo-codes***". IEEE Inter. Conf. on Comm., vol. 2, may 1993

[66] R. G. Gallager, *"**Low-density parity-check codes***". M.I.T press, 1963.

[67] *" **ETSI. Digital Video Broadcasting (DVB), Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Application***". EN 302 307 V1.1.1, June 2004

[68] *" **Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands***". IEEE P802.16e-2005, October 2005.

[69] S. Xiaonan, S. Yoshizawa and Y. Miyanaga, *," **Performance evaluation of quasi-cyclic LDPC codes for IEEE802.11n based MIMO-OFDM systems***". In the 3rd International Symposium on Comm. Control and Sig. Proces., Mar. 2008.

[70] N. Wiberg, "***Codes and decoding on general graphs***". PhD thesis, Linköping university, Sweden, 1996.

[71] C. Berrou, A. Glavieux. "***Near optimum error correcting coding and decoding : turbo-codes***". IEEE Trans. on Communications, vol. 44, pp.1261–1271, 1996.

[72] D. J. C. MacKay, "***Good error-correcting codes based on very sparse matrices***," IEEE Trans. on Info. Theory, vol.45, pp.399–431, Mar. 1999.

[73] M. C. Davey and D. J. C. MacKay, "***Low density parity check codes over gf(q)***". IEEE Comm. Letters, vol.2, pp. 165–167, Jun. 1998.

[74] D. Declercq and M. Fossorier, "***Decoding algorithms for nonbinary LDPC codes over GF(q)***". IEEE Trans. on Comm., vol.55, pp.633–643, Apr. 2007.

[75] A. Goupil, M. Colas, G. Gelle, and D. Declercq, "***FFT-based BP decoding of general LDPC codes over Abelian groups***". IEEE Trans. on Comm., vol.55, pp. 644–649, Apr. 2007.

[76] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "***Regular and irregular progressive edge-growth Tanner graphs***". IEEE Trans. on Info. Theory, vol.51, pp.386–398, Jan. 2005.

[77] H. Xiao and A. H. Banihashemi, "***Improved progressiveedge-growth (PEG) construction of irregular LDPC codes***". IEEE Com. Letters, vol.8, pp.715–717, Dec. 2004.

[78] Y. Kou, S. Lin, and M. P. C. Fossorier. "***Low-density parity check codes based on finite geometries : a rediscovery and new results***". IEEE Trans. on Info. Theory, vol 47, pp. 2711–2736, Nov. 2001.

[79] H. Tang, J. Xu, S. Lin, and K. A. S. Abdel-Ghaffar. "***Codes on finite geometries***". IEEE Trans. on Info. Theory, vol 51, pp 572–596, Feb. 2005.

[80] D. J. C. MacKay and M. J. Postol," ***Weaknesses of Margulis and Ramanujan–Margulis low-density parity-check codes***". In Proceedings of MFCSIT2002, Galway, vol 74 of Electronic Notes in Theoretical Computer Science. Elsevier, 2003.

[81] G. Liva, S. Song, L. Lan, Y. Zhang, S. Lin, and W. E. Ryan. "***Design of LDPC codes : a survey and new results***". J. Comm. Software and Systems, 2006.

[82] T. J. Richardson and R. L. Urbanke. "***Efficient encoding of low-density parity-check codes***". IEEE Trans. on Info. Theory, vol.47(2), pp.638–656, Feb. 2001.

[83] Y. Kaji, M.P.C. Fossorier, and S. Lin. ***"Encoding LDPC codes using the triangular factorization***". In Proceedings of ISITA' 04, Parma, Italia, October 2004.

[84] R. M. Tanner. "***A recursive approach to low complexity codes***", IEEE Trans. on Info. Theory, vol. 27, pp 533–547, Sep. 1981.

[85] H.-A. Loeliger. "***An introduction to factor graphs***". IEEE Sig. Process. Mag., vol 21, pp.28–41, Jan. 2004.

[86] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "***Factor graphs and the sum-product algorithm***". IEEE Trans. on Info. Theory, vol 47, pp. 498–519, Feb. 2001.

[87] R. J. McEliece and M. Yildirim. "***Belief propagation on partially ordered sets***". In Mathematical Theory of Systems and Networks, August 2002.

[88] G. David Forney, Jr. "***Codes on graphs: normal realizations.***" IEEE Trans. on Information Theory, 47(2) :520–548, February 2001

[89] Jr. Forney, G. David and Pascal O. Vontobel. "***Partition functions of normal factor graphs***". In Info. Theory and Applications Workshop, Feb. 2011

[90] T. Etzion, A. Trachtenberg, and A. Vardy. "***Which codes have cycle-free Tanner graphs***". IEEE Trans. on Info. Theory, vol. 45, pp. 2173–2181, Sep. 1999.

[91] A. Amraoui, A. Montanari, T. J. Richardson, and R. Urbanke. "***Finite-length scaling and finite-length shift for low-density parity-check codes.***" ArXiv, Oct. 2004.

[92] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke. "***Finite-length analysis of low-density parity-check codes on the binary erasure channel.***" IEEE Trans. on Info. Theory, 48(6) :1570–1579, June 2002.

[93] J. Zhang and A. Orlitsky. "***Finite-length analysis of LDPC codes with large left degrees.***" In Proceedings of ISIT'02, Lausanne, Switzerland, June 2002.

[94] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke. "***Analysis of sum-product decoding of low-density paritycheck codes using a gaussian approximation.***" IEEE Trans. on Info. Theory, vol.47, pp 657–670, Feb. 2001.

[95] S. ten Brink. "***Convergence behavior of iteratively decoded parallel concatenated codes.***" IEEE Trans. on Comm., vol. 49, pp. 1727–1737, Oct. 2001.

[96] M. Luby, M. Mitzenmacher, and A. Shokrollahi. "***Analysis of random processes via And-Or tree evaluation.***" Technical Report TR-97-042, ICSI Berkeley, Nov. 1997.

[97] T.C. Chuah, "***Robust iterative decoding of turbo codes in heavy-tailed noise*",** IEE Proc.-Commun., Vol. 152, No. 1, Feb. 2005

[98] T.C. Chuah, "***Decoding of low-density parity-check codes in non-Gaussian channels",*** IEE Proc.-Commun., Vol. 152, No. 6, December 2005

[99] T.C. Chuah, "***Distance metric for soft-decision decoding in non-Gaussian channels*",** IEE Proc.-Commun., *Vol. 39 No. **74,** Jul 2003*

[100] V. Zyablov, and M. Pinsker "***Estimation of Error-Correction Complexity of Gallager Low density Codes*",** Problemy Peredachi Informatsii, *Vol. 11 pp. 23 26, 1975.*

[101] M. Shm and C. L. Nikias, "***Signal Detection in Impulsive Noise Based on Stable Distributions*",** IEEE Proc. Sig. Sys and Comp., Nov.1993

[102] S. Zozor, J. Brossier, and P.-O. Amblard*" **A Parametric Approach to Suboptimal Signal Detection in α-Stable Noise"*** IEEE Trans. On sig. process., vol. 54, no. 12, Dec 2006

[103] N. C. Beaulieu, and S. Niranjayan*" **UWB Receiver Designs Based on a Gaussian-Laplacian Noise-Plus-MAI Model"*** IEEE Trans. On comm., vol. 58, no. 3, Mar. 2010.

[104] N. C. Beaulieu, and David J. Young*" **Designing Time-Hopping Ultrawide Bandwidth Receivers for Multiuser Interference Environments"*** Proceedings of the IEEE, Vol. 97, No. 2, Feb. 2009

Résumé

L'objectif de cette thèse est d'étudier le comportement des codes LDPC dans un environnement où l'interférence générée par un réseau n'est pas de nature gaussienne mais présente un caractère impulsif. Un premier constat rapide montre que sans précaution, les performances de ces codes se dégradent très significativement. Nous étudions tout d'abord les différentes solutions possibles pour modéliser les bruits impulsifs. Dans le cas des interférences d'accès multiples qui apparaissent dans les réseaux *ad hoc* et les réseaux de capteurs, il nous semble approprié de choisir les distributions α-stables. Généralisation de la gaussienne, stables par convolution, elles peuvent être validées théoriquement dans plusieurs situations.

Nous déterminons alors la capacité de l'environnement α-stable et montrons par une approche asymptotique que les codes LDPC dans cet environnement sont bons mais qu'une simple opération linéaire à l'entrée du décodeur ne permet pas d'obtenir de bonnes performances. Nous avons donc proposé différentes façon de calculer la vraisemblance en entrée du décodeur. L'approche optimale est très complexe à mettre en œuvre. Nous avons étudié plusieurs approches différentes et en particulier le clipping dont nous avons cherché les paramètres optimaux.

Mots clefs : codes LDPC, distributions α-stables, interférences, bruits impulsifs, capacité du canal.

Abstract

The goal of this PhD is to study the performance of LDPC codes in an environment where interference, generated by the network, has not a Gaussian nature but presents an impulsive behavior. A rapid study shows that, if we do not take care, the codes' performance significantly degrades. In a first step, we study different approaches for impulsive noise modeling. In the case of multiple access interference that disturb communications in *ad hoc* or sensor networks, the choice of □-stable distributions is appropriate. They generalize Gaussian distributions, are stable by convolution and can be theoretically justified in several contexts.

We then determine the capacity if the α-stable environment and show using an asymptotic method that LDPC codes in such an environment are efficient but that a simple linear operation on the received samples at the decoder input does not allow to obtain the expected good performance. Consequently we propose several methods to obtain the likelihood ratio necessary at the decoder input. The optimal solution is highly complex to implement. We have studied several other approaches and especially the clipping for which we proposed several approaches to determine the optimal parameters.

Keywords: LDPC codes, α-stable distributions, interference, impulsive noise, channel capacity.