



N° d'ordre :

N° de série :

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Hamma Lakhder d'El-Oued
Institut des Sciences et Technologie
Département D'électronique

Mémoire de fin d'études présenté
Pour l'obtention du diplôme de

Master ACADEMIQUE

Domaine : **Sciences et techniques**

Filière : **Electronique**

Spécialité : **Télécommunications**

Présenté par : **Zouad Thabet, Lemouchi Dalal**

Turbo décodage à bruit impulsif α –stable symétrique

Soutenu le : 07 Juin 2015

Devant le jury composé de :

M.	Touhami Redha	MAA	Président
M.	Khalil Abdellatif	MAA	Examineur
M.	Chemsa Ali	MAA	Encadreur

2014-2015

Résumé

Ce travail étudie le turbo décodage pour un bruit impulsif symétrique α –stable (SaS). En raison de l'inexistence d'une forme fermée de la PDF du bruit α –stables sauf pour $\alpha = 1$ (bruit de **Cauchy**) ou pour $\alpha = 2$ (bruit de **Gauss**), et comme l'algorithme MAP nécessite une telle forme de la PDF du bruit pour calculer les métriques de transition des branches, nous avons pris le cas $\alpha = 1$ pour modéliser le bruit impulsif. Comme la variance de ce bruit est infinie, la définition du SNR n'est pas valide dans ce cas et nous avons utilisé à la place le paramètre de dispersion du bruit γ . Dans ce contexte, une méthode originale d'estimation du paramètre de dispersion du bruit impulsif de **Cauchy** a été présentée dans ce travail. Cette méthode a deux avantages primordiaux. La première est la bonne précision d'estimation, ce qui permet d'augmenter remarquablement les performances du turbo décodeur MAP. La deuxième est que cette méthode n'a pas besoin de séparer le bruit du signal utile ; contrairement, elle fait l'estimation à partir du signal reçu uniquement.

Mots clefs : CTC, Bruit impulsif α –stable SaS , BPSK, MAP, FLOS, BER, Paramètre de dispersion.

Abstract

This work investigates the decoding of turbo codes in impulsive symmetric α –stable (SaS) noise. Due to the nonexistence of a closed-form expression for the PDF of α –stable noises except for $\alpha = 1$ (**Cauchy's** noise) or $\alpha = 2$ (**Gauss's** noise), and as the MAP algorithm requires such closed-form to calculate the branch transition metrics, we have taken the case $\alpha = 1$ to model the impulsive noise. As the variance of this noise is infinite, the definition of the SNR is not valid in this case and we used instead the dispersion parameter γ of the noise. In this context, this study presents an original method for estimating the dispersion parameter of the **Cauchy's** impulse noise. This method has two main advantages. The first is the good estimation accuracy, which increases remarkably the performance of the MAP turbo decoder. The second is that this method does not need to separate the noise of the useful signal; unlike, it use only the received signal to estimate this parameter.

Key words : CTC, Impulsive noise α –stable SaS , BPSK, MAP, FLOS, BER, Dispersion parameter.

ملخص

ندرس في هذا العمل الفك النفاث من أجل تشويش نبضي متناظر α –مستقر (SaS). نظرا لعدم وجود شكل صريح لكثافة احتمال هذا التشويش النبضي باستثناء الحالة $\alpha = 1$ (تشويش كوشي) أو $\alpha = 2$ (تشويش غوس)، وكما أن خوارزمية MAP تتطلب هذا الشكل الصريح لحساب مقاييس العبور للفرع، فإننا أخذنا الحالة $\alpha = 1$ لنمذجة التشويش النبضي. بما أن تباین هذا التشويش غير منته فإن تعريف SNR غير صالح في هذا الوضع واستخدمنا بدلا منه معامل التشتت γ الخاص بتشويش كوشي. في هذا السياق، تم تقديم طريقة جديدة و أصلية في هذا العمل لتقدير معامل التشتت للتشويش النبضي. هذه الطريقة لها فائدتان أساسيتان. الأولى هي دقة التقدير، مما يزيد و بشكل ملحوظ في تحسين خصائص الفكك النفاث MAP. أما الثانية فهي أن هذه الطريقة لا تحتاج لفصل التشويش عن الإشارة المرسله، بل على العكس فالتقدير يعتمد على الإشارة الواردة فقط.

المصطلحات الأساسية : CTC، التشويش النبضي α –مستقر SaS ، BPSK، MAP، FLOS، BER، معامل التشتت.

الإهداء

نهدي هذا العمل
المتواضع إلى سيدنا
وحبيبنا وقرّة أعيننا
رسول الله صلى الله عليه
وسلم

REMERCIEMENTS



*Nous rendons grâce à **ALLAH** de nous avoir
donné le courage et la volonté ainsi que
la conscience d'avoir pu terminer nos études.*

*Nous tenons à remercier vivement notre promoteur **Mr. Chems Afi**
pour ses conseils précieux et pour toutes les commodités et aises
qu'il nous a apportées durant notre étude et réalisation de ce projet.*

*Nos remerciements les plus vifs s'adressent aussi aux Messieurs le
président et les membres de jury d'avoir accepté d'examiner et d'évaluer notre
travail.*

*Nous exprimons également notre gratitude à tous les professeurs et
enseignants qui ont collaboré à notre formation depuis nos premiers cycles d'étude
jusqu'à la fin de nos masters.*

*Et enfin, que nos chers parents et familles, et bien avant tout, trouvent ici
l'expression de nos remerciements les plus sincères et les plus profonds en
reconnaissance de leurs sacrifices, aides, soutien et encouragement afin de nous
assurer cette formation de master dans les meilleures conditions.*



ZOUED, LEMOUCHI

Glossaire

ADSL	A symmetric D igital S ubscriber L ine
AM	A mplitude M odulation
APP	A P osteriori P robability
ASK	A mplitude S hift K eying
AWGN	A dditif W hite G aussian N oise
BER	B it E rror R ate
BJRC	B ahl C ocke J elinek and R aviv
BM	B ranch M etric
BPSK	B inary P hase S hift K eying
BSM	B ackward S tate M etric
CTC	C onvolutional T urbo C ode
DEC	D ecoder
FFT	F ast F ourier T ransform
FLOS	F ractional L ower O rders S tatistics
FM	F requency M odulation
FSK	F requency S hift K eying
FSM	F orward S tate M etric
GGD	G eneralized G aussian D istribution
I.I.d	I ndependent and I dentically D istributed

JD	J oint D istribution
JPL	J et P ropulsion L aboratory
LLR	L ogarithm L ikelihood R atio
LOS	L ower O rders S tatistics
MAP	M aximum A Posteriori
ML	M aximum L ikelihood
NRNSC	N on R ecursive N on S ystematic C ode
NSC	N on S ystematic C ode
PDF	P robability D ensity F unction
PLC	P ower L ine C ommunication
PM	P hase M odulation
PSK	P hase S hift K eying
QAM	Q uadrature A mplitude M odulation
RSC	R ecursive S ystematic C ode
RMSE	R oot M ean S quare E rror
SC	S ystematic C ode
SCN	S ystème de C ommunication N umérique
SNR	S ignal N oise R ate
SOVA	S oft O utput V iterbi A lgorithm
<i>S</i>α<i>S</i>	S ymmetric α – S tale with localization parameter $\delta = 0$
UWB	U ltra W ide B and

Liste des figures et tableaux

Figure 1.1	Architecture générale d'un système de communication numérique.....	04
Figure 1.2	Turbo code parallèle.....	10
Figure 1.3	Ecriture-Lecture dans un entrelaceur par blocs.....	12
Figure 1.4	Entrelacement hélicoïdal pour $m = 5$ et $n = 4$	13
Figure 1.5	Schéma de principe d'un entrelaceur convolutif avec $k = 4$ et $B = 1$	13
Figure 1.6	Entrelacement à décalage circulaire pour $L = 8$, $a = 3$ et $s = 0$	15
Figure 1.7	Entrelacement S –aléatoire $N = 12$, $S = 2$	18
Figure 1.8	Diagramme de constellation pour BPSK.	21
Figure 1.9	La probabilité d'erreur en fonction du rapport ρ	22
Figure 2.1	Courbes de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = \delta = 0$ et $\gamma = 1$	31
Figure 2.2	Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = -1$, $\delta = 0$ et $\gamma = 1$	32
Figure 2.3	Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 1$, $\delta = 0$ et $\gamma = 1$...	32
Figure 2.4	Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = \delta = 0$ et $\gamma = 2$	33
Figure 2.5	Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.5\}$, $\beta = \delta = 0$ et $\gamma = 5$	33
Figure 2.6	Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0$, $\delta = -3$ et $\gamma = 1$	34
Figure 2.7	Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0$, $\delta = 3$ et $\gamma = 1$...	34
Figure 2.8	Le tracé de 5000 échantillons d'un bruit $S\alpha S$ (a) $\alpha = 2$ (b) $\alpha = 1.5$ (c) $\alpha = 1$	37
Figure 2.9	Le tracé de $N = 5000$ échantillons d'un bruit $S\alpha S$ (a) $\alpha = 2$ (b) $\alpha = 1.5$ (c) $\alpha = 1$	38
Figure 3.1	Représentation graphique de la métrique d'état en avant	50
Figure 3.2	Représentation graphique de la métrique d'état en arrière.....	51
Figure 3.3	Schéma bloc d'un décodeur itératif	53
Figure 4.1	Modèle de simulation.....	55

Figure 4.2.a	Code convolutif récursif systématique (5,7).....	58
Figure 4.2.b	Code convolutif récursif systématique (13,15).....	58
Figure 4.3.a	Turbo codes convolutifs parallèles (5,7)	58
Figure 4.3.b	Turbo codes convolutifs parallèles (13,15)	59
Figure 4.4	La PDF de Cauchy pour différentes valeurs de γ	60
Figure 4.5	Le tracé de 16384 échantillons d'un bruit impulsif <i>SaS</i> de Cauchy : (a) $\gamma = 0.1$ (b) $\gamma = 10$	61
Figure 4.6	La probabilité ρ en fonction du paramètre γ	63
Figure 4.7	Modèle de simulation adopté pour estimer le paramètre γ	65
Figure 4.8	La RMSE relative en fonction de la valeur exacte de γ	65
Figure 4.9	Schéma fonctionnel d'un décodeur itératif MAP	66
Figure 4.10	Le BER en fonction de $1/\gamma$ d'un CTC parallèle à 4 états pour 6 itérations (6#) et 12 itérations (12#).....	68
Figure 4.11	Le BER en fonction de $1/\gamma$ d'un CTC parallèle à 4 états pour 6 itérations (6#) et 12 itérations (12#).....	69
Tableau 1.1	Tableau de nombres premier avec 8	15
Tableau 1.2	Tableau de nombres premier avec 8	16
Tableau 2.1	Moyenne et variance théorique d'une loi stable	29

Sommaire

Introduction générale	01
-----------------------------	----

1 Concepts de base sur les SCN

1.1 Introduction	03
1.2 Architecture générale d'un SCN	03
1.3 Modélisation du bruit dans un canal	07
1.4 Types de codes.....	09
1.4.1 Les turbo codes convolutifs CTC parallèles	10
1.4.2 Types d'entrelaceurs	11
1.4.2.1 Entrelaceurs par blocs	11
1.4.2.2 Entrelaceurs hélicoïdal	12
1.4.2.3 Entrelaceurs convolutif.....	13
1.4.2.4 Entrelaceurs à décalage circulaire	14
1.4.2.5 Entrelaceurs aléatoire	15
1.4.2.6 Entrelaceurs S –aléatoire.....	17
1.5 Techniques de modulation.....	19
1.6 Conclusion.....	22

2 Les lois α –stables

2.1 Introduction	24
2.2 Les lois stables	25
2.2.1 Lois stables univariées.....	25
2.2.1.1 Inconvénients des lois stables.....	26
2.2.1.2 Diverses propriétés.....	26

2.2.1.3 Propriétés arithmétiques	30
2.2.1.4 Interprétation des paramètres de la fonction caractéristique d'une loi stable	30
2.2.2 Algorithme de simulation.....	35
2.2.3 Statistiques sur les lois symétrique α – stables	36
2.3 Conclusion.....	39

3 Les turbo codes convolutifs CTC et l'algorithme MAP

3.1 Introduction	40
3.2 Le décodage itératif et l'Algorithme MAP	41
3.2.1 La métrique de Vraisemblance Logarithmique ou de corrélation	41
3.2.2 Le principe du Maximum A Posteriori MAP	43
3.2.3 Notations et règle de décision.....	45
3.3 L'Algorithme MAP utilisé.....	47
3.3.1 Expression de la probabilité conjointe	47
3.3.2 Expression de la métrique d'état en avant FSM	49
3.3.3 Expression de la métrique d'état en arrière BSM	50
3.3.4 Calcul de la métrique de branche BM.....	51
3.4 Principe du décodeur itératif MAP.....	52
3.5 Conclusion.....	53

4 Simulation, résultats et interprétations

4.1 Introduction	54
4.2 Modèle de simulation.....	55
4.2.1 Source d'information.....	55
4.2.2 Turbo code convolutif CTC parallèle.....	56
4.2.3 Modulation utilisée	59
4.2.4 Source de bruit	59
4.2.4.1 La distribution de Cauchy	59
4.2.4.2 Simulation d'un bruit impulsif $S\alpha S$ de Cauchy	61
4.2.5 Décodeur itératif MAP.....	61

4.2.5.1 Nouvelle approche pour estimer le paramètre γ	62
4.2.5.2 Les étapes de l'implémentation de l'algorithme MAP	66
4.2.5.3 Schéma fonctionnel d'un décodeur itératif MAP	66
4.2.6 Calcul du taux d'erreurs binaire BER.....	67
4.3 Résultats et Commentaires	67
4.3.1 La performance en termes de BER	67
4.4 Conclusion.....	69
Conclusion générale et perspectives	71
Bibliographies	73

Introduction générale

Les turbo codes inventés en 1993 [01] ont de nombreuses applications dans divers domaines [02][03]. Cependant, ils ont été développés pour de modèles à bruit blanc gaussien additif malgré que de nombreux systèmes réels souffrent d'apparition fréquente de bruits impulsifs. Des exemples de ces systèmes de transmission à bruit impulsif comprennent les chaînes de radio [04], les canaux de communication par courants porteurs en ligne PLC [05], et des lignes numériques d'abonné DSL [06]. Comme les systèmes optimisés pour les bruits gaussiens donnent de résultats médiocres dans le cas des canaux à bruit impulsif [07][08], il est important d'étudier et d'améliorer la performance des turbo codes en présence de tel type de bruit.

En conséquence, il est mieux pratiquement d'étudier l'efficacité des turbo codes, avec un algorithme de décodage optimal bien choisi, dans un canal à bruit impulsif. Deux modèles de distribution du bruit impulsif couramment utilisés sont la distribution gaussienne généralisée symétrique GGD et celle α -stable symétrique ($S\alpha S$), dont la distribution gaussienne est un cas particulier. Ces deux types de distributions possèdent quelques propriétés souhaitables, mais la fonction de densité de probabilité (PDF) du deuxième type $S\alpha S$ n'a pas une forme ferme sauf pour $\alpha = 1$ (bruit de Cauchy) ou $\alpha = 2$ (bruit gaussien).

Pour ces raisons notre travail est consacré sur ce type de codes avec l'algorithme de maximum a posteriori MAP dans un canal à bruit impulsif. Ce type de bruit est présenté dans ce travail par une distribution α -stable symétrique ($S\alpha S$) ($0 < \alpha \leq 2$) [09].

Il est avéré que les performances d'un turbo décodeur appelé aussi décodeur itératif avec l'algorithme MAP dépendent étroitement des probabilités de transition de branche qui nécessitent elles-mêmes la connaissance exacte de la formule de la PDF du bruit. Ceci pose un problème sérieux pour un bruit α -stable $S\alpha S$ parce que la formule de la PDF d'un tel bruit n'existe plus dans la plupart des cas (sauf pour $\alpha = 1$ ou 2) [10]!

Mais, depuis l'implantation de la transformée de Fourier rapide FFT (Fast Fourier Transform), les PDF stables deviennent plus ou moins faciles à calculer.

Dans notre travail nous considérons l'étude des turbo codes pour un bruit impulsif de type α -stable symétrique ($S\alpha S$) pour $\alpha = 1$. Dans ce cas la PDF du bruit possède une formule ferme, ce qui facilite amplement le calcul des probabilités de transition de branche pour l'algorithme MAP.

Ce mémoire est organisé comme suit. Le premier chapitre consiste en des généralités sur les systèmes de communication numérique y compris la modélisation d'un canal. Dans ce chapitre les notions de la distribution gaussienne généralisée et du bruit impulsif sont présentées.

Le second chapitre est dédié à la modélisation des bruits impulsifs par des variables aléatoires qui suivent des α –stable symétrique (*SaS*). Ce chapitre donne une vue d'ensemble des lois α –stables. La caractéristique la plus importante de ces lois est leur index de stabilité α . Encore appelé exposant caractéristique, ce paramètre compris entre 0 et 2 indique la vitesse de décroissance des queues de distributions. La loi normale, par exemple, correspond à un index de stabilité $\alpha = 2$. C'est la seule loi stable qui ne soit pas à queue lourde.

Le troisième chapitre est réservé à l'algorithme de maximum a posteriori MAP. Une étude bien détaillée sur l'algorithme MAP et son application, dans le cas d'un bruit impulsif α –stable *SaS* où $\alpha = 1$, est exposée dans ce chapitre. Le principe du décodage itératif avec MAP et la notion de l'information extrinsèque dans le cas d'un bruit impulsif sont aussi présenté dans cette partie.

Le quatrième chapitre est voué à la simulation et l'évaluation de la performance en termes de BER d'un turbo code pour un canal à bruit additif et impulsif α –stable symétrique (*SaS*) et avec une modulation BPSK. Sous l'effet de la non-existence d'une expression déterminée de la PDF du bruit α –stables sauf pour $\alpha = 1$ (bruit impulsif de **Cauchy**) ou pour $\alpha = 2$ (bruit blanc gaussien) [11], et comme l'algorithme MAP nécessite une telle expression de la PDF du bruit pour déterminer les métriques de transition des branches, nous avons pris le cas $\alpha = 1$ pour modéliser le bruit impulsif. Comme la variance de ce bruit est infinie, la définition du *SNR* n'est pas valide dans ce cas et nous avons utilisé à la place le paramètre de dispersion du bruit γ . Dans ce contexte, nous avons présenté une méthode originale d'estimation du paramètre de dispersion du bruit impulsif de **Cauchy**. Cette méthode possède deux avantages prépondérants. La première consiste à la précision et l'exactitude d'estimation hors de pairs, ce qui permet d'améliorer considérablement les performances du turbo décodeur MAP. La deuxième est que l'estimation du paramètre de dispersion du bruit est effectuée en utilisant le signal reçu uniquement. Ce dernier est un mélange du bruit et du signal utile. Cet avantage semble prodigieux car la méthode proposée n'a pas besoin de séparer le bruit du signal utile pour faire l'estimation.

Enfin, une conclusion générale dresse un bilan de ce travail et propose quelques perspectives de recherche.

Chapitre 1

Concepts de base sur les SCN

1.1 Introduction

Le rôle d'un système de communication est de transmettre à distance des informations d'un émetteur à un ou plusieurs récepteurs au travers d'un canal de manière fiable et à coût réduit. Dans un système de transmission numérique, une suite finie de symboles représente l'information. Celle-ci est transmise sur le canal de transmission par un signal «réel» ou analogique. Ce signal peut prendre une infinité de valeurs différentes et est ainsi soumis à différentes formes de perturbations et d'interférences, pouvant conduire à des erreurs d'interprétations du signal recueilli par le récepteur. Le rôle de l'ingénieur en télécommunications est donc de s'assurer que le récepteur pourra recevoir le message émis par l'émetteur sans aucune erreur, par un dimensionnement judicieux du canal de transmission et par la mise en place de techniques le rendant plus robuste, autrement dit, il faut qu'il adapte le signal initial au canal envisagé, afin de transmettre l'information le plus fidèlement possible tout en optimisant l'utilisation du canal.

Dans le paragraphe suivant de ce chapitre, nous définirons les concepts de base d'un système numérique de communication.

1.2 Architecture générale d'un SCN

L'architecture générale d'un SCN est présentée dans la figure 1.1. [12].

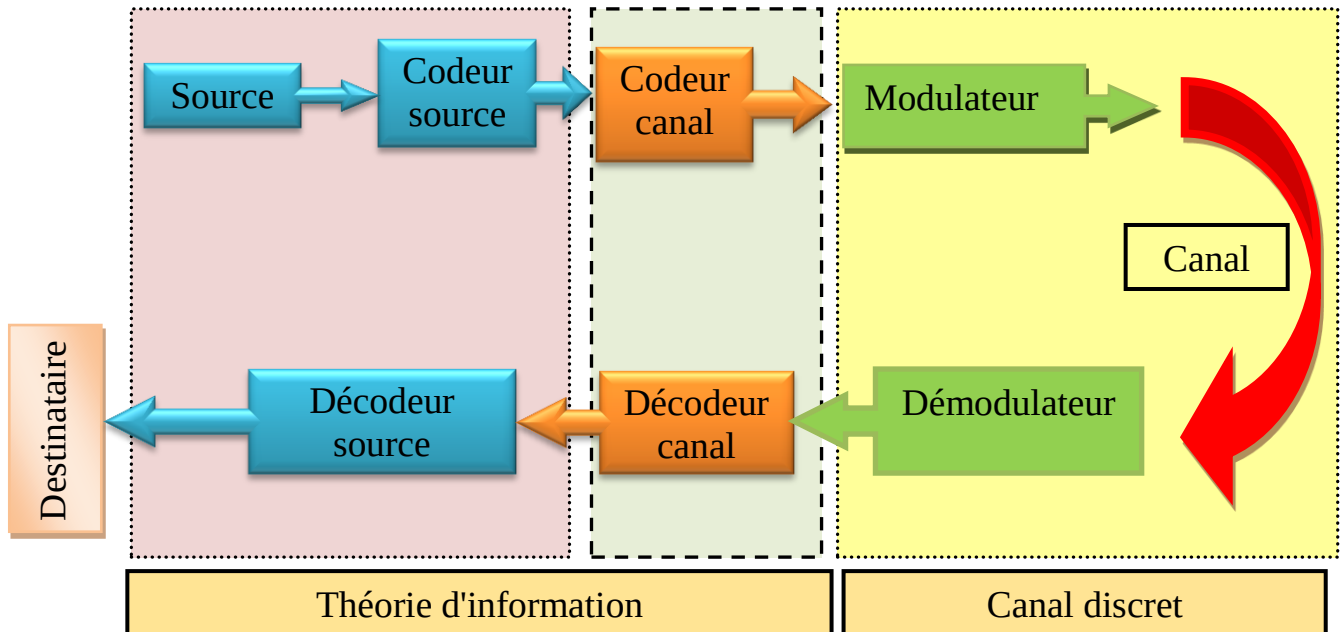


Figure 1.1 Architecture générale d'un système de communication numérique.

Le système se compose de trois grandes parties : un émetteur, un récepteur et le milieu de propagation appelé « canal de propagation ». L'émetteur a pour objectif d'adapter l'information de la source avant de transmettre à travers un canal de propagation. En effet, à partir du signal reçu, le récepteur fait l'opération pour récupérer l'information de la source. Sur la figure 1.1, les opérations duales effectuées à l'émetteur et au récepteur sont le codage et le décodage de source, le codage et le décodage de canal, la modulation et la démodulation. Nous allons maintenant décrire brièvement les différentes fonctions mises en œuvre dans la chaîne de transmission.

Notons que ces fonctions peuvent traiter l'information de façon individuelle ou collective qui dépendent de la conception appliquée.

Pour réaliser une transmission numérique, le message à transmettre doit être mis en forme numérique. Si la source délivre un message analogique tel que le signal de parole ou le signal d'image, il est indispensable de le numériser. Lorsque le message analogique est échantillonné, le théorème de Shannon doit être respecté. Chaque échantillon est quantifié et puis converti sur des éléments binaires. Ensuite, des données

en sortie peuvent être considérées comme des données binaires aléatoires qui restent a priori inconnues au destinataire [13].

Cette séquence des données contient des éléments binaires bruts avec une redondance inutile. Le codage de source permet de compresser l'information moins utile sans perte d'information de façon à réduire le coût de la bande passante de la communication. Cette séquence binaire compressée de sortie du codeur de source est appelée séquence d'information. Dans ce qui suit, nous supposons que la séquence d'information est une suite d'éléments binaires indépendants et identiquement distribués sur l'alphabet $\{0,1\}$. Le nombre de données binaires émis par unité de temps est défini comme son débit binaire $D = 1/T_b$ bit/s où T_b est la durée d'un bit en seconde. La quantité d'information, l'entropie, se mesure à partir de la probabilité respective des éléments binaires. Notons qu'il existe l'opération inverse de ce codage de source utilisé au récepteur.

Après la numérisation et le codage de source, il existe une fonction spécifique pour des transmissions numériques qui permet d'améliorer la qualité de transmission. Cette fonction est appelée « codage de canal » qui a pour objectif de détecter et/ou corriger des erreurs en réception. Le codage de canal insère des éléments binaires dans la séquence d'information. Des éléments binaires ajoutés sont appelés « une redondance ». Cette opération mène alors à une réduction du débit utile binaire de la transmission. Le décodage de canal doit connaître le type de codage de canal utilisé à l'émission pour corriger des erreurs correctement. Sinon, il est possible de créer des erreurs en plus lors de la sortie du décodage du canal. Notons que certains types de codage ne sont pas très utilisés car ils augmentent la complexité en émission et en réception.

Pour transmettre le message numérique en tant qu'une séquence d'éléments binaires, il est nécessaire de le convertir en un signal analogique. Cette opération est généralement appelée « une modulation ». La modulation consiste à un ensemble de n éléments binaires issu du message. Cet ensemble d'éléments binaires correspond à un signal $S_i(t), i = 1, \dots, M$ de durée $T_s = nT_b$, parmi $M = 2^n$ signaux en fonction de la réalisation de n éléments binaires. La séquence binaire de débit D est modulée en la séquence à M –aire états avec la rapidité de modulation R exprimée en Bauds. La rapidité de modulation peut s'exprimer par [13]

$$R = 1/T_s = D/\log_2 M \quad (1.1)$$

Quelques exemples des modulations linéaires sans mémoire sont la modulation par déplacement de phase (PSK), la modulation par déplacement d'amplitude (ASK) et la modulation d'amplitude en quadrature (QAM). Parmi les traitements effectués par l'émetteur, le filtrage du signal modulé est exploité pour limiter sa bande, ce qui permet de partager le même milieu de transmission à plusieurs utilisateurs sans problème d'interférence.

Dans le cadre de transmission sans fil, une antenne émettrice ou un réseau d'antennes émettrices permet de transformer les signaux analogiques en ondes électromagnétiques à l'émetteur. Le signal émis se propage à travers le milieu de propagation qui subit des perturbations et du bruit. Le milieu de propagation est caractérisé par plusieurs effets qui sont présentés ultérieurement. Au récepteur, une autre antenne ou un réseau d'antennes consiste à convertir les ondes électromagnétiques en des signaux analogiques.

Le bruit est considéré comme une perturbation aléatoire qui vient de l'extérieur et l'intérieur du récepteur. Le bruit externe est causé par des systèmes industriels (moteur, ligne à haute tension, etc...) ou par les interférences des signaux des autres utilisateurs dans le cas d'un système multi-utilisateurs. Le bruit thermique (bruit interne) vient de l'agitation thermique des électrons des dispositifs électroniques, dont la puissance est proportionnelle à la température. Une autre source de bruit particulière est le bruit de quantification qui apparaît au niveau de l'opération d'échantillonnage-quantification du fait de la transformation du signal analogique en un signal numérique.

Le bruit considéré ici est un bruit impulsif modélisé par un processus aléatoire α –stable symétrique ($S\alpha S$) à moyenne nulle, indépendant du signal émis. Les bruits de ce type se présentent généralement sous forme d'impulsions élémentaires de forte amplitude, superposées à un bruit de fond homogène plus faible. Ce dernier, souvent dû aux effets thermiques (composants électroniques), est considéré comme un bruit blanc gaussien [14]. De cette constatation, les principales caractéristiques du bruit impulsif α –stable sont décrites scrupuleusement dans le chapitre 2.

Le récepteur a pour objectif de reconstituer au mieux le message émis à partir du signal reçu. Il comprend des circuits d'amplification, de changement de fréquence et de démodulation. Une antenne ou plusieurs antennes en réception induit des signaux. La démodulation fait des opérations d'estimation et de détection avec un échantillon toutes

les T_s secondes en sortie du filtre adapté de réception où $T_s = T_b \cdot \log_2 M$. Chaque échantillon est converti en données binaires à partir d'un circuit de décision. Le décodeur de canal exploite la redondance introduite par le codeur de canal pour détecter et puis corriger (si possible) les erreurs de transmission. Le décodeur de source reçoit une séquence de données provenant du décodeur de canal. En connaissant le traitement réalisé par le codeur de source, il reconstitue le message original. En présence possible d'erreurs en sortie du décodeur de canal, le message décodé n'est pas exactement identique au message délivré par la source.

1.3 Modélisation du bruit dans un canal

Un canal de transmission n'est pas simplement composé du support de transmission, mais aussi de l'ensemble des dispositifs qui permettent d'adapter le signal à transmettre au canal et de minimiser les erreurs de réception dues aux perturbations.

Afin de résister aux perturbations induites par le support de transmission, un signal à transmettre subit en général des opérations de codage de source, de codage de canal, de modulation, de mise en forme, etc.... Il subit les opérations inverses en réception.

Lors de la transmission à travers le canal, le signal subit les atténuations et les déformations propres au canal, ainsi que le bruit provenant de perturbateurs externes. En outre, les émetteurs et récepteurs du canal contribue à générer des perturbations qui dégradent le signal.

Les défauts du canal de transmission et les perturbations externes vont limiter la quantité d'information qui peut passer à travers le canal et affecter la qualité du signal. Alors, en général le récepteur reçoit un signal faible, bruité et distordu. Il doit être en mesure de le reconstruire puis de l'interpréter afin de retrouver le signal d'origine.

Les transmissions d'informations se font en général par liaison filaire (câble électrique ou fibre optique) ou par liaison hertzienne (ou sans fils). Cette dernière est la plus sensible aux perturbations externes et dont l'environnement de propagation est le plus difficile à modéliser.

Un modèle de canal reste une vision simplifiée de la réalité, sa modélisation complète pourrait atteindre une très grande complexité (par exemple un réseau cellulaire dans un milieu urbain). Dans ce mémoire, on prend la notion de canal au sens large du

terme, c'est-à-dire qu'on y inclut le médium de transmission mais aussi les équipements de transmission et de réception. Voici quelques définitions [15]

- Canal discret: l'ensemble des symboles reçus après le passage dans le canal est fini. L'information est donc numérique.
- Canal à temps discret: l'échelle des temps est discrète.
- Canal sans mémoire: le symbole reçu à un instant donné dépend uniquement du symbole émis au même instant t (en considérant le retard de transmission nul). Si le canal a une mémoire, la sortie dépend aussi des symboles présents en $t', t' < t$.
- Canal stationnaire: ses caractéristiques sont fixes au cours du temps. Une fibre optique est un canal stationnaire, ses caractéristiques étant quasi invariantes au cours du temps, alors qu'une liaison radio correspond à un canal non stationnaire puisque ses caractéristiques dépendent de nombreux facteurs tels que les objets environnants, les conditions atmosphériques ou les perturbations électromagnétiques.
- Canal sélectif: le signal à transmettre a des composantes fréquentielles qui sont atténuées différemment par le canal de propagation. Il introduit donc une distorsion dans le signal transmis.

Le bruit blanc gaussien est un modèle de bruit largement utilisé dans de nombreux domaines. En effet, lorsqu'un phénomène correspond à la somme d'un grand nombre de variables aléatoires, il est possible de démontrer par le théorème de la limite centrale que la distribution statistique de ce phénomène suit une distribution gaussienne. Un bruit gaussien suit la loi normale, caractérisée par une moyenne et une variance finies.

Malheureusement, un tel modèle devient inefficace lorsque l'on désire étudier des phénomènes présentant de nombreuses valeurs extrêmes, qui ne peuvent être considérées en tant que des valeurs aberrantes, comme le cas des bruits impulsifs. Les lois stables non-gaussiennes sont une alternative toute naturelle car elles sont une généralisation de la loi normale et prennent en compte des queues lourdes. C'est ainsi que s'est développée l'analyse statistique des processus linéaires dans lesquels les innovations sont distribuées suivant des lois α – stables symétriques ($S\alpha S$) [16][17].

Puisque nous sommes intéressés au bruit impulsif, **tout au long de ce travail, nous considérons un bruit impulsif qui suit une loi α –stable symétrique (SaS) où $\alpha = 1$** . Le chapitre 2 présente bien la notion de ces lois.

1.4 Types de codes

Les codes de canal appelés aussi codes correcteurs d'erreurs sont répartis en trois catégories: Les codes en blocs linéaires, les codes convolutifs et les codes concaténés ou les turbo codes.

Le codage en blocs consiste à associer, à chaque bloc de k bits d'information, un bloc de n bits ($n > k$) contenant $n - k$ bits de redondance. Les 2^k blocs de n bits délivrés par un codeur sont appelés les mots-code. Le rapport k/n est appelé le rendement du code.

Les opérations de codage et de décodage dans les codes en blocs se font à l'aide d'addition et de multiplications sur des éléments binaires. Ces dernières correspondant, respectivement, aux opérations logiques «OU» et «ET» exclusif.

Il existe un type spécial de code en blocs linéaires appelé codes cycliques. Ce sont des codes en blocs linéaires vérifiant quelques propriétés supplémentaires [18], [19].

Les codes convolutifs forment une classe extrêmement souple et efficace de codes correcteurs d'erreur. Ce sont les codes les plus utilisés dans les communications fixes et mobiles. Les codes convolutifs ont les mêmes caractéristiques que les codes en bloc sauf qu'ils s'appliquent à des séquences infinies de symboles d'information et génèrent des séquences infinies de symboles de code [19].

Un turbo codeur classique résulte de l'association de deux (ou plus) codeurs. Il s'agit souvent de codeurs convolutifs récurrents systématiques RSC car leur récursivité apporte des propriétés pseudo-aléatoires intéressantes (voir figure 1.2).

Le principe des turbo codes, comme tout code correcteur d'erreur, est d'introduire une redondance dans le message afin de le rendre moins sensible aux bruits et perturbations subies lors de la transmission. Le codage consiste à utiliser deux codeurs simples, dont les entrées ont été entrelacées; ainsi, chaque codeur voit une série d'informations différentes à son entrée.

Le décodage est une collaboration entre les décodeurs, chacun donnant son « avis » (notion de confiance) sur chaque bit décodé. Cette information est ensuite fournie à l'entrée du prochain décodeur, et ainsi de suite. D'où l'appellation « turbo » [20].

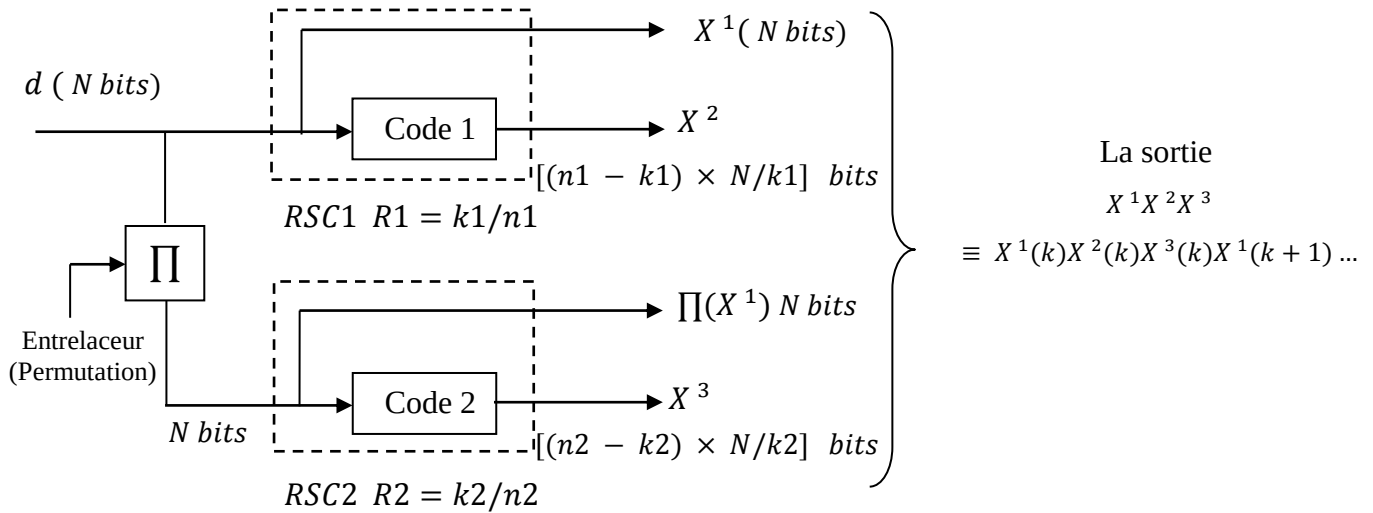


Figure1.2 Turbo code parallèle.

1.4.1 Les turbo codes convolutifs CTC parallèles

L'idée primordiale du CTC parallèle est d'utiliser deux codes convolutifs systématiques récurrents RSC, généralement identiques, concaténés en parallèle via un entrelaceur entre les deux (figure 1.2).

Le but de l'entrelaceur (Interleaver) dans un CTC est de faire la permutation des données émises afin de casser les paquets d'erreur surviennent lors de la transmission, et ce pour rendre leur distribution plus uniforme, ce qui donne au CTC une grande efficacité de corriger les erreurs [21].

Le décodeur associé à un CTC est un décodeur souple ou doux, car ce type de décodeurs est très performant qu'un décodeur dur ou ferme pour un canal à bruit impulsif, et très facile à mettre en œuvre pour tous les codes acceptant des représentations en treillis, par la raison qu'il utilise un algorithme Maximum A Posteriori MAP.

Un tel décodage permet d'extraire de l'information sur chacun des décodeurs constitutifs et de l'échanger entre ces deux décodeurs. L'information extraire d'un décodeur est dite « extrinsèque » et est réinjectée à l'itération suivante dans un autre décodeur composant afin de bénéficier de la diversité de codage. Le processus pouvant se répéter plusieurs fois, on parle de décodage itératif. L'appellation « Turbo » provient justement de la réinjection des informations extrinsèques pour rappeler l'image des moteurs turbo qui réutilisent une partie de l'énergie gaspillée par le moteur pour donner plus de puissance au moteur.

Si les codes convolutifs peuvent coder un flux continu d'information, les turbo codes ne peuvent coder l'information que par paquets de longueur finie égale à celle de l'entrelaceur. Ainsi, après chaque bloc d'information, le codeur est forcé à un état connu, généralement l'état 00. Alors, les bits de terminaison sont placés à la suite des bits d'information avant d'effectuer le décodage [22].

1.4.2 Types d'entrelaceurs

1.4.2.1 Entrelaceurs par blocs

Ce type d'entrelaceurs est utilisé plus couramment dans les systèmes de communication [23]. En particulier, **Ramsey** [23] prouve que nous pouvons construire un entrelaceur optimal avec une largeur de $N = m \times n$, où N est la longueur du bloc d'information, en créant un entrelaceur de m lignes et n colonnes, dans lequel l'information s'écrit en entrée ligne par ligne et se lit en sortie colonne par colonne comme montre la figure 1.3.

L'ordre des éléments ligne par ligne de la matrice de la figure 1.3 est $i = (k - 1) \cdot n + l$, $1 \leq k \leq m$, $1 \leq l \leq n$ avec k et l sont les indices des lignes et des colonnes respectivement et $1 \leq i \leq m \cdot n$. Alors,

$$i - 1 = (k - 1) \cdot n + (l - 1), \quad 1 \leq k \leq m, \quad 0 \leq l - 1 \leq n - 1 \quad (1.2)$$

ce qui donne

$$l - 1 = (i - 1) \bmod [n] \quad (1.2.a)$$

$$k - 1 = \lfloor (i - 1)/n \rfloor \quad (1.2.b)$$

où $\lfloor x \rfloor$ est la partie entière de x .

L'ordre des mêmes éléments colonne par colonne est $j = (l - 1) \cdot m + k$, $1 \leq l \leq n$, $1 \leq k \leq m$ avec k et l sont les indices des lignes et des colonnes respectivement et $1 \leq j \leq m \cdot n$. La fonction d'entrelacement du tel entrelaceur, qui transforme la lecture ligne par ligne en lecture colonne par colonne, est alors

$$\pi(i) = j = (l - 1) \cdot m + k$$

et en substituant (2.2) dans cette dernière équation nous trouvons

$$\pi(i) = ((i - 1) \bmod [n]) \cdot m + \left\lfloor \frac{i - 1}{n} \right\rfloor + 1, \quad 1 \leq i \leq m \cdot n \quad (1.3)$$

La fonction de désentrelacement est

$$\pi^{-1}(i) = ((i - 1) \bmod [m]) \cdot n + \left\lfloor \frac{i - 1}{m} \right\rfloor + 1, \quad 1 \leq i \leq m \cdot n \quad (1.4)$$

Dans le cas d'un entrelaceur à matrice carrée ($m = n$) on obtient $\pi(i) = \pi^{-1}(i)$.

Afin de réordonner le bloc entrelacé, on effectue l'opération inverse. Ce type d'entrelacement est efficace quand le vecteur pénalisant se situe dans une seule colonne.

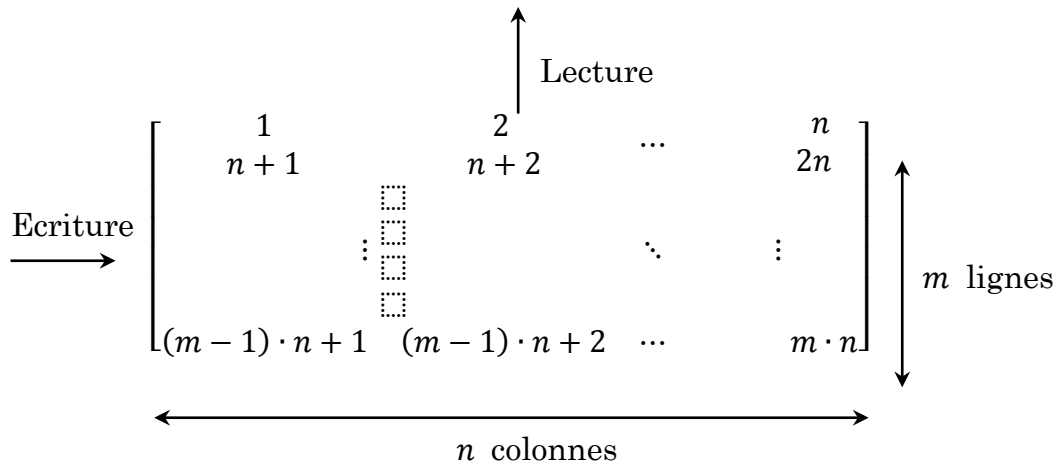


Figure 1.3 Ecriture-Lecture dans un entrelaceur par blocs.

1.4.2.2 Entrelaceur hélicoïdal

Ce type d'entrelaceurs, qui est une version modifiée de l'entrelaceur par blocs, a été proposé par **Barbulescu** [24]. Dans un entrelaceur de m lignes et n colonnes, l'information s'écrit ligne par ligne et se lit diagonalement, en commençant par l'élément en bas à gauche, selon les formules

$$\begin{cases} \pi(i) = i_r \cdot n + j_r, & 1 \leq i \leq m \cdot n \\ j_r = (i - 1) \bmod [n] + 1 \\ i_r = \left(m - \left\lfloor \frac{i-1}{n} \right\rfloor - j_r\right) \bmod [m] \end{cases} \quad (1.5)$$

En substituant les expressions de j_r et i_r dans celle de $\pi(i)$ nous trouvons

$$\begin{cases} \pi(i) = (m \cdot n - i - (n - 2)j_r - 1) \bmod [m \cdot n] + 1, & 1 \leq i \leq m \cdot n \\ j_r = (i - 1) \bmod [n] + 1 \end{cases} \quad (1.6)$$

Nous pouvons vérifier facilement que la fonction de désentrelacement de cet entrelaceur est $\pi^{-1}(i) = \pi(i)$. La figure 1.2 montre la fonction d'entrelacement hélicoïdale pour $m = 5$ et $n = 4$.

$$M = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 \end{bmatrix} \longrightarrow \pi(M) = \begin{bmatrix} 17 & 14 & 11 & 8 \\ 13 & 10 & 7 & 4 \\ 9 & 6 & 3 & 20 \\ 5 & 2 & 19 & 16 \\ 1 & 18 & 15 & 12 \end{bmatrix}$$

Figure 1.4 Entrelacement hélicoïdale pour $m = 5$ et $n = 4$.

1.4.2.3 Entrelaceur convolutif

L'entrelaceur convolutif a été enchâssé par **Ramsey** [23]. Il est constitué par des amas des registres à décalages et par deux commutateurs comme l'indique la figure 1.5 [25].

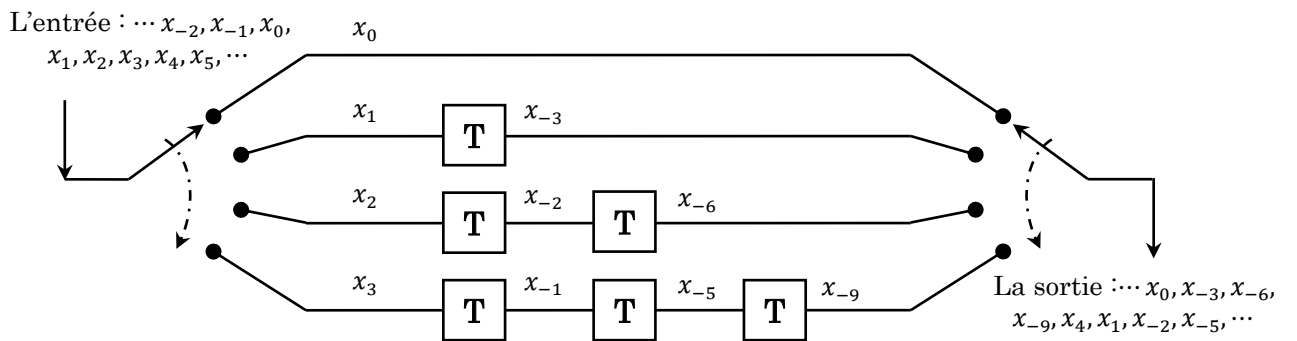


Figure 1.5 Schéma de principe d'un entrelaceur convolutif avec $K = 4$ et $B = 1$.

Cet entrelaceur est totalement caractérisé par deux paramètres: le nombre de branches K (appelé aussi le nombre d'étages) et la profondeur B du registre à décalage de base. Ce dernier, il possède B flip-flops (ou bascules synchrones) et se trouvait

toujours dans la deuxième branche. La première branche ne comprend aucun registre. En général, la branche numéro i ($i = 1, \dots, K$) est constituée de $(i - 1)$ registres à décalage de base, dès lors elle contient $(i - 1) \cdot B$ flip-flops. Au total, cet entrelaceur contient $K \cdot (k - 1) \cdot B/2$ flip-flops.

La séquence d'information est regroupée en blocs de K bits. Les symboles de chaque bloc de K bits sont aiguillés à travers la chaîne des registres à décalage. La séquence entrelacée est ensuite formée par le multiplexage des sorties de chaque registre à décalage [26]. La fonction d'entrelacement convolutif est donnée par [27].

$$\pi(i) = i - [(i - 1) \bmod [K]] \cdot K \cdot B \quad (1.7)$$

Le schéma d'un désentrelaceur convolutif possède la même structure que celui de l'entrelaceur mais avec un ordre inversé pour la distribution des registres à décalage de base sur les branches. Autrement dit, la branche numéro i ($i = 1, \dots, K$) est constituée de $(K - i)$ registres à décalage de base. La fonction de désentrelacement convolutif est alors

$$\pi^{-1}(i) = i + [(i - 1) \bmod [K]] \cdot K \cdot B \quad (1.8)$$

1.4.2.4 Entrelaceur à décalage circulaire

Ce type d'entrelacement est un stratagème dérivé des décalages circulaires. La formule de base pour générer les permutations circulaires est donnée par

$$\pi(i) = (a \cdot i + s) \bmod [L] \quad (1.9)$$

où L est la taille de l'entrelaceur, i est l'indice de l'élément à entrelacé, a est le pas d'itération, et s est une constante de compensation [28]. a et s sont toujours inférieurs à L . a est premier avec L .

Dans le cas où $\sqrt{2L}$ est un entier, on préconise de prendre la valeur $a = \sqrt{2L} - 1$ [28]. Si d_1 représente la distance entre deux 1 d'une séquence de poids 2 avant l'entrelacement et d_2 celle de la séquence après l'entrelacement alors $d_1 + d_2 \geq \sqrt{2L}$. Néanmoins cette valeur du pas d'itération a n'est pas unique, tout coefficient de la forme $a = m \cdot \sqrt{2L} \pm 1$ avec m entier positif et $m < \sqrt{L/2}$, permet d'atteindre cette limite.

La fonction de désentrelacement de cet entrelaceur est

$$\pi^{-1}(i) = \beta(i - s) \bmod [L]$$

où β est un entier qui vérifie $\beta a = 1 \bmod [L]$. Dans le cas où $a = \sqrt{2L} - 1$ on peut prendre $\beta = L - (a + 2) = L - (\sqrt{2L} + 1)$.

La figure 1.6 montre un entrelacement à décalage circulaire avec $L = 8$, $a = 3$, et $s = 0$.

$$\begin{array}{c}
 M = [0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7] \\
 \Downarrow \\
 \pi(M) = [0 \quad 3 \quad 6 \quad 1 \quad 4 \quad 7 \quad 2 \quad 5]
 \end{array}$$

Figure 1.6 Entrelacement à décalage circulaire pour $L = 8$, $a = 3$ et $s = 0$.

1.4.2.5 Entrelacement aléatoire

Dans la pratique on essaye de concevoir des systèmes qui approchent le plus possible le comportement des entrelaceurs aléatoires uniformes. Citons deux exemples

Entrelaceur de Berrou-Glavieux

La taille T de l'entrelaceur est choisie comme étant une puissance de 2 : $T = M \times N = 2^m \times 2^n$. De plus, un tableau de nombres premiers avec 8, est utilisé (voir tableau 1.1) [29].

v	1	2	3	4	5	6	7	8
p_v	17	37	19	29	41	23	13	7

Tableau 1.1 Tableau de nombres premiers avec 8.

Soient i et j les indices ligne et colonne respectivement de l'écriture dans l'entrelaceur, et soient i_r et j_r les indices ligne et colonne respectivement de la lecture. La position d'un élément dans l'entrelaceur est définie soit par (i, j) ou par $k = (i - 1)N + j$ où pour tout $i \in \{1, 2, \dots, M\}$ on a $1 \leq j \leq N$.

L'entrelacement (ou la permutation) est obtenu en appliquant la formule (1.10).

$$\pi(k) = i_r \cdot N + j_r \quad (1.10)$$

avec

$$i_r = (p_{v+1} \cdot i - 1) \bmod [M],$$

$$j_r = \left(\left(\frac{M}{2} + 1 \right) \cdot (i + j - 2) \right) \bmod [N] + 1,$$

$$\text{et } j = (k - 1) \bmod [N] + 1, i = \frac{(k - j)}{N} + 1, v = (i + j - 2) \bmod [8].$$

Si l'indice de la position d'un élément dans le désentrelaceur est k ($1 \leq k \leq M \cdot N$), alors la fonction de désentrelacement est donnée par

$$\pi^{-1}(k) = i \cdot N + j \quad (1.11)$$

avec,

$$i = (\alpha \cdot j_r - 1) \bmod [M]$$

$$j = (\beta \cdot j_r - i) \bmod [N] + 1$$

$$j_r = k \bmod [N]$$

$$i_r = \frac{k - j_r}{N} + 1,$$

où α et β sont deux entiers naturels tels que

$$\alpha \cdot p_{v+1} = 1 \bmod [M]$$

$$\beta \cdot \left(\frac{M}{2} + 1\right) = 1 \bmod [N]$$

$$v = \beta \cdot j_r \bmod [8]$$

Entrelaceurs JPL (Jet Propulsion Laboratory)

Quels que soient M et N (N est pair), la taille de l'entrelaceur est alors $M \times N$, et l'entrelacement est obtenu en appliquant la formule (1.12) pour tout i tel que $0 \leq i \leq M \cdot N - 1$.

v	1	2	3	4	5	6	7	8
p_v	31	37	43	47	53	59	61	67

Tableau 1.2 Tableau de nombres premiers avec 8.

$$\pi(i) = c(i) \cdot N + 2 \cdot r(i) - m(i) + 1 \quad (1.12)$$

avec

$$m(i) = i \bmod [2]$$

$$c_0 = \frac{(i - m(i))}{2} \bmod [M]$$

$$r_0 = \left(\frac{(i - m(i))}{2} - c_0\right) / M$$

$$r(i) = (19 \cdot r_0 + 1) \bmod [N/2]$$

$$v = r(i) \bmod [8]$$

$$c(i) = (p_{v+1} \cdot c_0 + 21 \cdot m(i)) \bmod [M]$$

Cet entrelaceur a été proposé pour l'utiliser dans un standard CCSDS [29]. On peut démontrer que la fonction de désentrelacement est donnée par

$$\pi^{-1}(i) = 2 \cdot (r_0 \cdot M + c_0) + m \quad (1.13)$$

avec

$$m = (i - 1) \bmod [2]$$

$$r = \frac{i+m-1}{2} \bmod [N/2]$$

$$c = \frac{i-(2 \cdot r - m + 1)}{N}$$

$$v = r \bmod [8]$$

$$r_0 = \frac{u_r \cdot (r-1)}{g_r} \bmod [N/2]$$

$$c_0 = \frac{u_c \cdot (c-21 \cdot m)}{g_c} \bmod [M]$$

$$u_r \cdot 19 + v_r \cdot N/2 = g_r = PGCD(19, N/2) \text{ (Théorème de Bézout)}$$

$$u_c \cdot P_{v+1} + v_c \cdot M = g_c = PGCD(P_{v+1}, M) .$$

1.4.2.6 Entrelaceurs S –aléatoire

Les entrelaceurs pseudo-aléatoires sont les plus performants pour les turbo codes dans le cas des blocs d'information de tailles moyenne et grande [30]. Leurs inconvénients par rapport aux entrelaceurs cités précédemment résident dans leur complexité.

Ce type d'entrelaceurs donne de très bons résultats pour les turbo codes [31], parce qu'il est classé parmi les entrelaceurs optimaux [32]. Pour cette raison, nous allons le présenter avec détaille.

Pratiquement, toutes les techniques de conception d'entrelacement des turbo codes dans la littérature sont basés sur l'algorithme de génération d'entrelacement S –aléatoire proposé par l'équipe du JPL [33]. Cet algorithme s'appuie essentiellement sur le choix au hasard d'un entrelaceur avec une limitation sur son étalement. La lecture de la séquence dans tel entrelaceur S –aléatoire s'effectue suivant un ordre donné par un générateur pseudo-aléatoire. L'opération inverse consiste à écrire dans le désentrelaceur suivant le même ordre pseudo-aléatoire puis à lire la séquence selon l'ordre croissant [26].

Pour un entrelaceur S –aléatoire de taille N , la fonction d'entrelacement π est faite en respectant la condition suivante [34][35]

$$\forall i, j: 1 \leq i, j \leq N, 0 < |i - j| \leq S \Rightarrow |\pi(i) - \pi(j)| \geq S \quad (1.14)$$

Les entrelaceurs S –aléatoires permettent de changer aléatoirement l'indexation d'un bloc de telle sorte que leurs éléments restent espacés d'au moins une distance S . Autrement dit, chaque position d'un élément sélectionné sera comparée aux S positions déjà sélectionnées précédemment. Si cette position est distante d'au moins $\pm S$ par rapport aux S positions précédentes alors l'élément est retenu, sinon il est rejeté. Cela se fait jusqu'à la sélection de tous les éléments. Ainsi, pour $S = 1$ on retrouve l'entrelaceur pseudo-aléatoire classique.

Par conséquent, l'algorithme de construire un entrelaceur S –aléatoire de taille N et de facteur de dispersion S est donné comme suit [36][28].

1. Considérer le vecteur $I = (1, 2, \dots, N)$;
2. Choisir une valeur entière pour le facteur de dispersion $S < \sqrt{N/2}$ afin que l'algorithme soit réussi.
3. Mettre le compteur à $i = 1$. Répéter les étapes suivantes
 - a. Choisir aléatoirement un nombre k dans le vecteur I . Si le nombre choisi k est distant d'au moins $\pm S$ de n'importe lequel des S précédents nombres stockés dans un vecteur J , stocker le comme élément $J_i = k$ dans $J = \{J_1, J_2, \dots\}$ et marquer le comme non disponible pour la prochaine sélection aléatoire dans I . Sinon, répéter l'étape a.
 - b. Augmenter le compteur i .
 - c. Répéter a. et b. pour $1 \leq i \leq N$.
4. Pour tout k , avec $1 \leq k \leq N$, détermine l'index i tel que $J_i = k$. La fonction d'entrelacement est définie alors $\pi(k) = i$.

L'exemple de la figure 1.7 représente l'entrelaceur S –aléatoire pour un bloc de longueur 12, c'est-à-dire $S = 2$.

$$\begin{array}{c}
 M = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11] \\
 \Downarrow \\
 \pi(M) = [5 \ 2 \ 7 \ 9 \ 4 \ 11 \ 1 \ 6 \ 10 \ 8 \ 0 \ 4]
 \end{array}$$

Figure 1.7 Entrelacement S –aléatoire $N = 12$, $S = 2$.

Le temps de calcul de cet algorithme augmente en fonction de S et la convergence vers une solution est assurée si on choisit $S < \sqrt{N/2}$. Avec la relation (1.14), l'algorithme proposé peut nous conduire à la solution $\pi(i) = i$! ce qui signifie pratiquement qu'il n'y a pas d'entrelacement de la séquence d'entrée. Pour cette raison **H. R. Sadjadpour** et al. [37] a ajouté une condition supplémentaire dans la construction d'un entrelaceur S -aléatoire, qui est

$$\forall i : 1 \leq i \leq N, \quad |i - \pi(i)| \geq S' \quad (1.15)$$

où S' est entier naturel non nul.

Le théorème suivant permet de faciliter la construction d'un entrelaceur S -aléatoire.

Théorème 1.1

Soient α et N deux entiers naturels premiers entre eux tels que $(\alpha - 1)$ divise N . Alors, il existe une permutation $\pi : \{1, 2, \dots, N\} \rightarrow \{1, 2, \dots, N\}$ définie par : $\pi(i) = (\alpha \cdot i + \beta) \bmod [N]$ qui satisfait les conditions (1.14) et (1.15) avec $S = \min \left\{ \alpha, \left\lfloor \frac{N}{\alpha+1} \right\rfloor \right\}$, $S' = \beta = \left\lfloor \frac{\alpha-1}{2} \right\rfloor$, où $\lfloor x \rfloor$ représente la partie entière de x .

Il faut noter qu'à cause de son importance, l'entrelaceur S -aléatoire sera utilisé avec les turbo codes utilisés dans la partie de simulation de notre travail. Les turbo codes utilisés dans ce travail sont les turbo codes convolutifs CTC parallèles.

1.5 Techniques de modulation

La fonction de modulation a pour objectif d'adapter le spectre de signal à émettre au canal de transmission. Lorsqu'il s'agit d'une transmission numérique à travers un canal du type passe bande, il est nécessaire de transposer (ou de traduire) le spectre de fréquence du signal à bande de base. Ceci peut se faire au moyen d'une porteuse qui est en général une fonction sinusoïdale pure. Cette opération s'appelle modulation numérique ou bien modulation analogique discrète car la porteuse est analogique et l'information (signal de base) est discrète.

En tout cas, on peut distinguer dans la modulation analogique discrète [19]

- ASK: modulation par déplacement d'amplitude (Amplitude Shift Keying) ;

- PSK: modulation par déplacement de phase (Phase Shift Keying) ;
- FSK: modulation par déplacement de fréquence (Frequency Shift Keying)
- Ou une modulation composite ou hybride comme la modulation QAM qui est une combinaison des modulations ASK et PSK.

Ces modulations peuvent être considérées comme des cas spéciaux des modulations analogiques AM, PM, FM, et AM-PM respectivement.

Sur un canal à bruit impulsif, le choix d'une modulation se fait en considérant l'occupation spectrale, les performances et la complexité du couple modulateur-démodulateur. Il est à noter que la faible occupation spectrale et les performances du système sont deux contraintes antagonistes, ce qui nécessite en pratique un compromis lors du choix d'une modulation.

Nous allons maintenant présenter un rappel sur le type de modulation utilisée dans notre travail qui est la modulation BPSK.

La Phase-shift keying (ou PSK, soit « modulation par changement de phase ») désigne une famille de formes de modulations numériques qui ont toutes pour principe de véhiculer de l'information binaire via la phase d'un signal de référence (porteuse), et exclusivement par ce biais.

Comme pour toute technique de modulation numérique, la phase en question ne peut prendre qu'un nombre fini de valeurs. Chacune de ces valeurs représente un unique nombre binaire, dont la taille (et donc la quantité d'information transmise) dépend du nombre de valeurs possibles pour la phase. Généralement, pour une modulation PSK donnée, les nombres binaires représentés sont tous de même taille.

BPSK est la forme la plus simple du PSK. Elle utilise deux phases qui sont séparées de 180° ; on l'appelle également 2-PSK. Cette modulation est la plus robuste de toutes les PSK car il faut une grande déformation du signal pour que le démodulateur se trompe sur le symbole reçu. Cependant on ne peut moduler qu'un seul bit par symbole (voir la figure 1.8), ce qui est un inconvénient pour les applications qui nécessitent un débit binaire élevé.

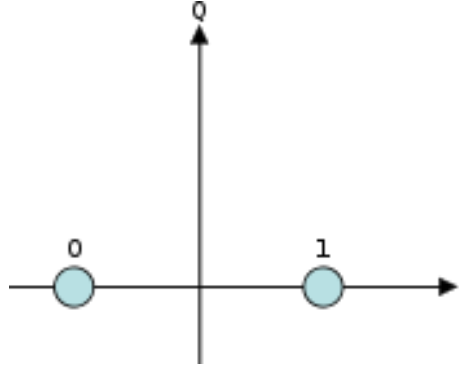


Figure 1.8 Diagramme de constellation pour BPSK.

La probabilité d'erreur binaire du BPSK pour un bruit $S\alpha S$ est la probabilité de décider qu'un 1 avait été transmis alors que c'était un 0 (et inversement). Ecrivons ce qu'est P_e dans ce cas concret

$$P_e = P(0) \times P(1 \text{ est choisit}/0 \text{ est trasmis}) + P(1) \times P(0 \text{ est choisit}/1 \text{ est trasmis})$$

$$\begin{aligned} P_e &= \frac{1}{2} P(1 \text{ est choisit}/0 \text{ est trasmis}) + \frac{1}{2} P(0 \text{ est choisit}/1 \text{ est trasmis}) \\ &= \frac{1}{2} \cdot P(R_k < 0) + \frac{1}{2} \cdot P(R_k > 0) \end{aligned}$$

Au niveau du récepteur le signal reçu est $R_k = Y_k + W_k$, $k = 1, 2, \dots$, $Y_k = \sqrt{E_b}$ si le 0 est transmis et $Y_k = -\sqrt{E_b}$ si le 1 est transmis (où E_b est l'énergie d'un bit pour la modulation BPSK), W_k représente le bruit α -stable $S\alpha S$. Alors

$$P_e = \frac{1}{2} P(R_k < 0/0 \text{ est trasmis}) + \frac{1}{2} P(R_k > 0/1 \text{ est trasmis})$$

$$P_e = \frac{1}{2} P(\sqrt{E_b} + W_k < 0) + \frac{1}{2} P(-\sqrt{E_b} + W_k > 0) = \frac{1}{2} - P(0 \leq W_k \leq \sqrt{E_b})$$

Dans le cas où $\alpha = 1$ (bruit de **Cauchy** (voir chapitre 2)) la PDF du bruit est

$$f(x) = \frac{2\gamma}{\pi((x - \delta)^2 + 4\gamma^2)}$$

$$P_e = \frac{1}{2} - \int_0^{\sqrt{E_b}} \frac{2\gamma}{\pi((x - \delta)^2 + 4\gamma^2)} dx = \frac{1}{2} - \frac{1}{\pi} \arctan\left(\frac{2\gamma\sqrt{E_b}}{4\gamma^2 - \delta(\sqrt{E_b} - \delta)}\right) \quad (1.16)$$

Si le bruit à moyenne nulle ($\delta = 0$), la probabilité d'erreur devient

$$P_e = \frac{1}{2} - \frac{1}{\pi} \arctan\left(\sqrt{\frac{E_b}{4\gamma^2}}\right) \quad (1.17)$$

La figure 1.9 illustre la variation de la probabilité d'erreur de l'équation (1.17) en fonction du rapport $\rho = \sqrt{E_b}/\gamma$ (en dB).

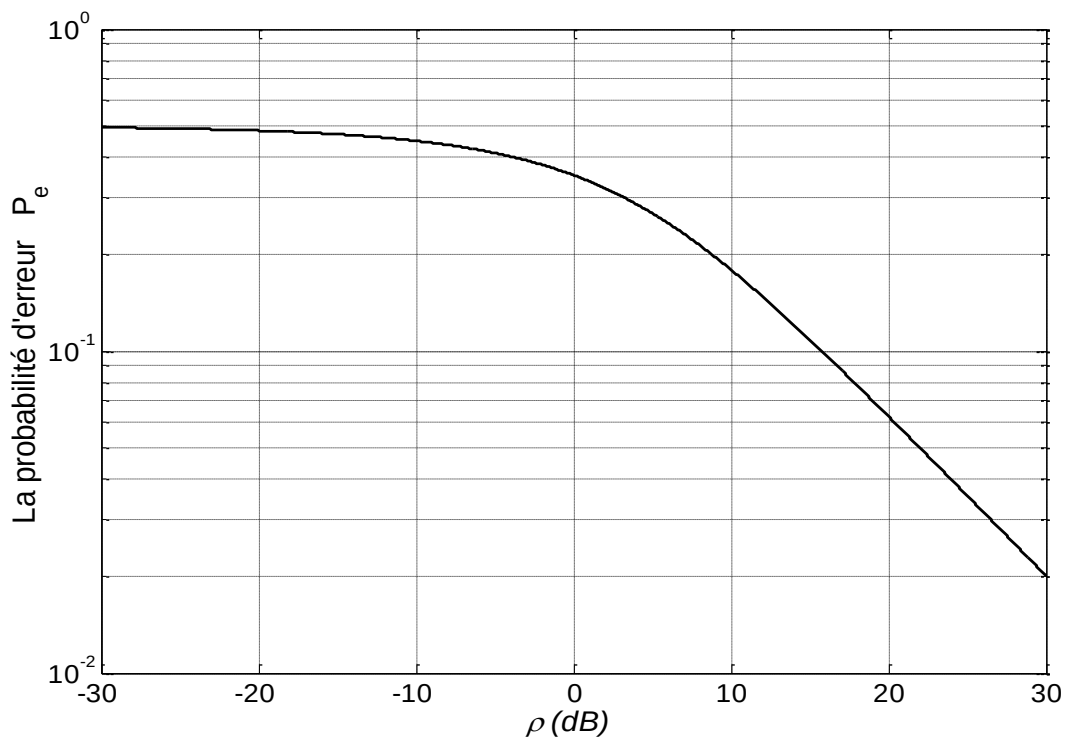


Figure 1.9 La probabilité d'erreur en fonction du rapport ρ .

1.6 Conclusion

On a présenté dans ce chapitre quelques concepts de base sur les SCN et ses parties constituantes, en particulier le canal et les fonctions du codage et de la modulation. Au fur et à mesure de la présentation de ces parties on a focalisé notre élucidation sur les concepts qui seront utilisés dans les chapitres qui suivent, à savoir les turbo codes convolutifs parallèles CTC, les entrelaceurs S -aléatoire, le bruit additif et impulsif à distribution symétrique α -stable ($S\alpha S$) pour $\alpha = 1$ (distribution de **Cauchy**), la modulation BPSK et l'algorithme de décodage MAP.

Toutefois dans cette présentation, on a essayé de donner l'essentielle de ce thème pour ne pas être exhaustif. Dans la section 1.4.2, nous avons présenté les différents types d'entrelacement généralement utilisés. Pour chaque fonction d'entrelacement on a donné la fonction de désentrelacement correspondante, dans le but de faciliter la programmation de cette fonction d'un côté et d'éviter de stocker la matrice d'entrelacement donc de gagner de l'espace mémoire d'un autre côté. Dans le paragraphe 1.5, nous avons calculé et tracé la probabilité d'erreur de la modulation BPSK pour un signal numérique non codé, dans le cas d'un bruit impulsif $S\alpha S$ pour $\alpha = 1$ (bruit de **Cauchy**). Ceci est très utile pour les comparaisons que nous devons faire entre les performances des systèmes avec codage et sans codage.

Dans un nombre croissant d'applications, les bruits rencontrés s'éloignent fortement des modèles classiques qui les modélisent par une gaussienne ou un mélange de gaussiennes. C'est en particulier le cas des bruits impulsifs que nous rencontrons dans plusieurs domaines, notamment celui des télécommunications. Dans ce cas, une modélisation mieux adaptée peut reposer sur les distributions α – stables [38]. Ceci fait l'objet du chapitre suivant.

Chapitre 2

Les lois α – stables

2.1 Introduction

Lorsque nous faisons du traitement de données, en particulier pour les séries chronologiques, il nous arrive fréquemment d'utiliser la variance empirique. Comme le nombre d'observations est fini, cette variance est évidemment finie. Il semble alors logique d'employer des méthodes, de proposer des modèles, de faire des estimations utilisant une variance finie. Dans certains cas cependant, la variance empirique est très grande. La question se pose alors de savoir si la variance théorique est finie ou infinie.

Dans le cadre des distributions à variance infinie sont apparues les lois α – stables, dont le moment d'ordre 2 est infini dès que α est strictement inférieur à 2. Dans les années 60, un nouveau modèle de variation des prix repose sur des lois qui s'inspirent des lois de **Pareto**. Ce sont les lois α – stables ou lois à queue lourde. Les domaines d'application de ces lois sont entre autres la finance, les télécommunications, internet...

Plusieurs livres sont consacrés à ces lois: **Zolotarev** [39] qui a étudié les lois α – stables dans le contexte univarié; **Samorodnitsky** et al. [40] qui ont étudié de manière approfondie beaucoup de propriétés de ces lois dans le cas univarié comme dans le cas multivarié ; **Nikias** et al. [41] qui ont appliqué ces lois dans le domaine du traitement du signal, pour n'en citer que trois.

Dans ce chapitre, le cas univarié est traité. Les lois sont définies et diverses propriétés sont présentées. Puis, sont et nous explicitons le calcul des moments et la simulation.

2.2 Les lois stables

2.2.1 Lois stables univariées

Les lois Pareto-stables, appelées aussi lois Lévy-stables ou tout simplement lois stables, ont été introduites par **Paul Lévy** en 1924. Les définitions et propriétés énoncées dans ce chapitre sont issues de [42] et [40].

Définition 2.1 [43]

On dit qu'une variable aléatoire réelle X qu'elle suit une loi stable si pour tout $n \geq 2$; il existe un réel strictement positif C_n et un réel D_n tels que

$$X_1 + X_2 + \dots + X_n \stackrel{d}{=} C_n X + D_n \quad (2.1)$$

où $X_1, X_2, \dots, X_n$ sont n copies indépendantes de X .

Lorsque $D_n = 0$, on parle de loi strictement stable.

- stable: stable par convolution;
- le réel strictement positif C_n est de la forme $n^{1/\alpha}$, pour un certain $\alpha \in]0, 2] \Rightarrow \alpha$ – stables.

Remarque 2.1

On peut montrer qu'il existe une constante α , $0 < \alpha \leq 2$, telle que $C_n = n^{1/\alpha}$ pour $n \in \mathbb{N}^*$. La démonstration est détaillée dans **Feller** (p.170-171)[44].

Définition 2.2 [17]

On dit qu'une variable aléatoire réelle X qu'elle suit une loi stable de paramètres α, β, γ et δ , que l'on note $X \sim S_\alpha(\beta, \gamma, \delta)$, si et seulement si sa fonction caractéristique est de la forme

$$\psi_X(t) = \exp\{-\gamma^\alpha |t|^\alpha [1 + i \cdot \beta \cdot \text{sign}(t) \cdot \omega(t, \alpha)] + i \cdot \delta \cdot t\} \quad (2.2)$$

où

$$\omega(t, \alpha) = \begin{cases} -\tan\left(\frac{\pi\alpha}{2}\right) & \text{si } \alpha \neq 1 \\ \frac{2}{\pi} \ln|t| & \text{si } \alpha = 1 \end{cases} \quad (2.2.a)$$

$$\text{sign}(t) = \begin{cases} 1 & \text{si } t > 0 \\ 0 & \text{si } t = 0 \\ -1 & \text{si } t < 0 \end{cases} \quad (2.2.b)$$

$$i^2 = -1, \quad \alpha \in]0, 2], \quad \beta \in]-1, 1], \quad \gamma \geq 0 \text{ et } \delta \in \mathbb{R}$$

Les lois stables sont intéressantes car elles ne sont pas trop compliquées. En effet, il suffit de quatre paramètres pour les connaître complètement.

Par convention, $S_\alpha(\beta, \gamma, \delta)$ représentera une loi stable de paramètres α, β, γ et δ et $S_\alpha S(\gamma)$ une loi symétrique α – stable de paramètre γ .

Exemple 2.1

- La loi normale $N(m, \gamma^2)$ est une loi $S_2\left(\beta, \frac{\gamma^2}{2}, m\right)$ (et réciproquement une loi $S_2(\beta, \gamma, \delta)$ est une loi normale $N(\mu, 2\gamma)$).
- La loi de **Cauchy** généralisée de densité $f(x) = \frac{\gamma}{\pi((x-\delta)^2 + \gamma^2)}$ est une loi $S_1(0, \gamma, m)$.
- La loi de **Poisson** $P(\lambda)$ n'est pas stable. En effet, soient X_1 et X_2 deux variables aléatoires suivent une loi de **Poisson**. Supposons que X_1 et X_2 sont stables, alors il existe $C > 0$ et D tels que

$$X_1 + X_2 \cong CX + D \quad (2.3)$$

Par égalité des moyennes et des variances, nous pouvons voir que

$$\begin{cases} 2\lambda = C\lambda + D \\ 2\lambda = C^2\lambda \end{cases} \Rightarrow \begin{cases} D = (2 - \sqrt{2})\lambda \\ C = \sqrt{2} \end{cases} \quad (2.4)$$

ce qui entraîne une contradiction car $(X_1 + X_2)$ à ses valeurs uniquement dans $\mathbb{N}$, alors que $\sqrt{2}X_1 + (2 - \sqrt{2})\lambda$ n'a pas que des valeurs dans $\mathbb{N}$.

2.2.1.1 Inconvénients des lois stables

Le principal inconvénient est que les densités des lois stables sont inconnues sauf dans deux cas [17]

1. La distribution gaussienne $S_2(0, \gamma, \delta)$ où $f(x) = \frac{1}{2\gamma\sqrt{\pi}} \exp\left(-\frac{(x-\delta)^2}{4\gamma^2}\right)$.
2. La distribution de Cauchy $S_1(0, \gamma, \delta)$ où $f(x) = \frac{2\gamma}{\pi((x-\delta)^2 + 4\gamma^2)}$.

Mais, depuis l'implantation de la transformée de Fourier rapide FFT, les densités stables sont faciles à calculer. Nous pouvons approcher par cette méthode la densité

$$f_X(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-itx} \psi_X(t) dt \quad (2.5)$$

2.2.1.2 Diverses propriétés

Nous allons rappeler quelques propriétés importantes des variables aléatoires stables de loi $S_\alpha(\beta, \gamma, \delta)$ que nous utiliserons dans la suite de ce mémoire.

– Densité

Pour la plupart des lois connues, nous avons une forme explicite de la PDF (normale, Cauchy, gamma,...). Pour la loi α – stables, nous n'avons que la forme explicite

de la fonction caractéristique. A l'aide de la transformée inverse de la fonction caractéristique, donnée par équation (2.5) nous pouvons obtenir f d'une loi α – stables sous la forme d'une intégrale comme suit

$$f_X(x) = \frac{1}{\pi} \int_0^{+\infty} e^{-t^\alpha} \cos[xt + \beta t^\alpha \omega(t, \alpha)] dt \quad (2.6)$$

où

$$\omega(t, \alpha) = \begin{cases} \tan\left(\frac{\pi\alpha}{2}\right) & \text{si } \alpha \neq 1 \\ -\frac{2}{\pi} \ln|t| & \text{si } \alpha = 1 \end{cases}$$

– Stabilité

Propriété 2.1

Pour $\alpha \neq 1$, nous avons l'équivalence suivante

$$X \text{ suit une loi } S_\alpha(\beta, \gamma, \delta) \Leftrightarrow Y = \frac{X-\delta}{\gamma^{\frac{1}{\alpha}}} \text{ suit une loi } S_\alpha(\beta, 1, 0)$$

Preuve: Tout d'abord, remarquons que pour $Y = mX + p$ on a

$$\psi_Y(t) = E[e^{it(mX+p)}] = e^{itp} \psi_X(mt)$$

Condition nécessaire

Prenons $m = \frac{1}{\gamma^{\frac{1}{\alpha}}}$ et $p = -\frac{\delta}{\gamma^{\frac{1}{\alpha}}}$. Nous avons alors

$$\psi_Y(t) = \exp\left(-\frac{i\delta t}{\gamma^{\frac{1}{\alpha}}}\right) \psi_X\left(\frac{t}{\gamma^{\frac{1}{\alpha}}}\right) \quad (2.7)$$

$$\psi_Y(t) = \exp\left(-\frac{i\delta t}{\gamma^{\frac{1}{\alpha}}}\right) \exp\left(-\frac{i\delta t}{\gamma^{\frac{1}{\alpha}}} - \gamma \left|\frac{t}{\gamma^{\frac{1}{\alpha}}}\right|^\alpha \left[1 - i\beta \operatorname{sign}\left(\frac{t}{\gamma^{\frac{1}{\alpha}}}\right) \tan\left(\frac{\pi\alpha}{2}\right)\right]\right) \quad (2.8)$$

or $\operatorname{sign}\left(\frac{t}{\gamma^{\frac{1}{\alpha}}}\right) = \operatorname{sign}(t)$ car $\gamma > 0$ donc, $\psi_Y(t) = \exp\left(-|t|^\alpha \left[1 - i\beta \operatorname{sign}(t) \tan\left(\frac{\pi\alpha}{2}\right)\right]\right)$, qui est bien la forme de la fonction caractéristique d'une loi $S_\alpha(0, \beta, 1)$.

Condition suffisante

La démonstration est similaire à la condition nécessaire en prenant $m = \gamma^{\frac{1}{\alpha}}$ et $p = \delta$.

Remarque 2.2

Pour la simulation, il suffit de générer des lois $S_\alpha(\beta, 1, 0)$ et par changement de variables, nous pouvons obtenir des lois $S_\alpha(\beta, \gamma, \delta)$.

– Queues lourdes

Propriété 2.2

Soit X une variable aléatoire $S_\alpha(\beta, \gamma, \delta)$ on a les deux résultats suivants

$$\begin{cases} \lim_{t \rightarrow +\infty} t^\alpha P(X > t) = \gamma C(\alpha) \frac{1 + \beta}{2} \\ \lim_{t \rightarrow +\infty} t^\alpha P(X < -t) = \gamma C(\alpha) \frac{1 - \beta}{2} \end{cases}$$

où

$$C(\alpha) = \left(\int_0^{+\infty} x^{-\alpha} \sin x \, dx \right)^{-1}$$

Remarque 2.3

L'égalité précédente nous fait penser à la caractérisation des lois de **Pareto**. En effet, une variable aléatoire suit une loi du type **Pareto** si

$$P(X \geq x) = x^{-\alpha} h(x)$$

où $h(x)$ est une fonction à variation lente, c'est-à-dire pour tout $t > 0$, $\lim_{x \rightarrow +\infty} \frac{h(tx)}{h(x)} = 1$. **Mandelbrot** [45] a montré que pour les lois α – stables, nous avons

$$P(X \geq x) = x^{-\alpha} [f_1(\alpha) + f_2(\alpha)x^{-\alpha} + o(x^{-2\alpha})] \quad (2.9)$$

Par passage à la limite quand x tend vers $+\infty$, on s'aperçoit que les lois stables sont asymptotiquement parétiennes.

– Calcul des moments

Une conséquence directe de la décroissance "lente" des lois de probabilité α – stables est montrée dans la propriété suivante.

Propriété 2.3

Si X suit une loi $S_\alpha(\beta, \gamma, \delta)$, et $p \in \mathbb{N}^*$, les moments d'ordre p sont comme suivant

1. Si $\alpha = 2$: $\forall p, E(|X|^p) < +\infty$.
2. Si $0 < \alpha < 2$: $\forall 0 \leq p < \alpha, E(|X|^p) < +\infty$ et $\forall p \geq \alpha, E(|X|^p) = +\infty$.

Preuve [17]

En fait, pour la première partie de la propriété 2.3, nous avons vu dans l'exemple 2.1 que le cas $\alpha = 2$ correspond au cas gaussien. Et nous savons que les variables aléatoires gaussiennes ont toutes leurs moments finis. Pour la seconde partie de la propriété 2.3, remarquons tout d'abord que pour toute variable aléatoire positive et intégrable, on a

$$E(Y) = \int_0^{+\infty} P(Y \geq v) dv \quad (2.10)$$

En posant $Y = |X|^\alpha$ et $v = u^p$, nous avons

$$E(Y) = \int_0^{+\infty} P(|X|^\alpha \geq u^p) p u^{p-1} du \quad (2.11)$$

or

$$\lim_{u \rightarrow 0} \frac{P(|X| \geq u) u^{p-1}}{u^{p-1}} = \lim_{u \rightarrow 0} P(|X| \geq u) = 1 \quad (2.12)$$

Donc, $P(|X| \geq u) u^{p-1} \sim u^{p-1}$ au voisinage de 0, et

$$\int_0^M P(|X| > u) u^{p-1} du < +\infty \Leftrightarrow \int_0^M u^{p-1} du < +\infty \Leftrightarrow p > 0$$

or

$$\lim_{u \rightarrow 0} \frac{P(|X| \geq u) u^{p-1}}{u^{p-\alpha-1}} = \lim_{u \rightarrow 0} u^\alpha P(|X| \geq u) = \gamma C(\alpha)$$

donc, $P(|X| \geq u) u^{p-1} \sim u^{p-\alpha-1}$ au voisinage de $+\infty$, et

$$\int_M^{+\infty} P(|X| > u) u^{p-1} du < +\infty \Leftrightarrow \int_M^{+\infty} u^{p-\alpha-1} du < +\infty \Leftrightarrow p > \alpha$$

Conclusion : d'après (2.11), il est facile de voir que

$$E|X|^p < +\infty$$

lorsque $\begin{cases} \int_0^M P(|X| > u) u^{p-1} du < +\infty \\ \int_M^{+\infty} P(|X| > u) u^{p-1} du < +\infty \end{cases}$ c'est-à-dire lorsque $0 < p < \alpha$.

Remarque 2.4

1. Dès que α est strictement inférieur à 2, la variance d'une loi α – stable est infinie.

Dès que α est strictement inférieur à 1, c'est la moyenne qui devient infinie.

2. Si α est plus grand que 1, la moyenne d'une loi α – stable est δ .

En particulier, pour la moyenne $E(X)$ et la variance $Var(X)$, on obtient les résultats dans le tableau suivant

$0 < \alpha \leq 1$		$0 < \alpha < 2$		$\alpha = 2$	
$E(X) = \infty$	$VAR(X) = \infty$	$E(X) = \delta$	$E(X) = \infty$	$E(X) = \delta$	$E(X) = 2\gamma^2$

Tableau 2.1 Moyenne et variance théorique d'une loi stable.

Propriété 2.4

Si X suit une loi $S_\alpha(\beta, \gamma, 0)$ alors on a

$$\forall 0 \leq p < \alpha, E|X|^p = C(\alpha, \beta, p) \gamma^{\frac{p}{\alpha}}$$

où

$$C(\alpha, \beta, p) = \frac{2^{p-1} \Gamma\left(1 - \frac{p}{\alpha}\right)}{p \int_0^{+\infty} u^{-p-1} \sin^2 u \, du} \left(1 + \beta^2 \tan^2\left(\frac{\alpha\pi}{2}\right)\right)^{\frac{p}{2\alpha}} \cos\left(\frac{p}{\alpha} \arctan\left(\beta \tan\left(\frac{\alpha\pi}{2}\right)\right)\right) \quad (2.13)$$

et $\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt, \quad z > 0.$

2.2.1.3 Propriétés arithmétiques**Propriété 2.5**

Soient X_1 et X_2 deux variables aléatoires indépendantes de loi stable $S_\alpha(\beta_1, \gamma_1, \delta_1)$ et $S_\alpha(\beta_2, \gamma_2, \delta_2)$ alors $X_1 + X_2$ suit une loi stable $S_\alpha(\beta, \gamma, \delta)$ avec

$$\gamma = (\gamma_1^\alpha + \gamma_2^\alpha)^{\frac{1}{\alpha}}; \beta = \frac{\beta_1 \gamma_1^\alpha + \beta_2 \gamma_2^\alpha}{\gamma_1^\alpha + \gamma_2^\alpha}; \delta = \delta_1 + \delta_2 \quad (2.14)$$

Notons que si $\beta_1 = \beta_2$ alors $\beta = \beta_1 = \beta_2$.

Cette propriété d'additivité est très intéressante, car deux variables aléatoires ayant la même valeur du paramètre α peuvent être considérées ensemble et la loi qui résultera de cette association conservera la même valeur du paramètre mais les autres paramètres seront modifiés.

$$AX_1 + BX_2 + C = S_\alpha\left(\beta, \gamma(A^\alpha + B^\alpha)^{\frac{1}{\alpha}}, \delta(A^\alpha + B^\alpha)^{\frac{1}{\alpha}} + C\right) \quad (2.15)$$

2.2.1.4 Interprétation des paramètres de la fonction caractéristique d'une loi stable

Une loi stable alors est caractérisée par 4 paramètres

- **$0 < \alpha \leq 2$** : Le paramètre α appelé **exposant caractéristique**, ou indice de stabilité décrit la forme de la distribution ou le degré d'épaisseur de la queue de distribution (Figures 2.1 jusqu'à 2.7). Plus ce paramètre est petit, plus les queues de la distribution sont épaisses, donc plus d'impulsivité au niveau des échantillons de la variable. Une distribution gaussienne a la valeur maximum de α soit $\alpha = 2$.
- **$-1 < \beta < 1$** : Paramètre **d'asymétrie**. Lorsque β vaut zéro alors la distribution est symétrique (Figures 2.1 jusqu'à 2.3). La loi est symétrique par rapport au

paramètre δ . Si de plus, $\delta = 0$, la loi est dite symétrique α – stable ($S\alpha S$) de fonction caractéristique

$$\psi_X(t) = \exp(-\gamma|t|^\alpha) \quad (2.16)$$

- ✓ Lorsque β est positif (resp. négatif), le mode est à gauche (resp. à droite) de la moyenne.
- ✓ Lorsque β est positif (resp. négatif), la queue de distribution est plus épaisse à droite (resp. à gauche).
- $\gamma > 0$: Paramètre de **dispersion** ou **d'échelle**, tel que $\gamma_{Gaussien} = \sqrt{2}\gamma$. Plus γ est grand, plus les données sont volatiles. Le paramètre γ permet de cintrer plus ou moins le corps de la distribution (Figures 2.1, 2.4 et 2.5).
- $-\infty < \delta < +\infty$: Paramètre **de localisation**, pour α supérieur à 1, à la moyenne de la loi de distribution. Si $\beta = 0$ alors δ est la médiane. Dans les autres cas le paramètre δ ne peut pas être interprété (Figures 2.1, 2.6 et 2.7). Lorsque δ est positif (resp. négatif), la courbe décale vers la droite (resp. vers la gauche).

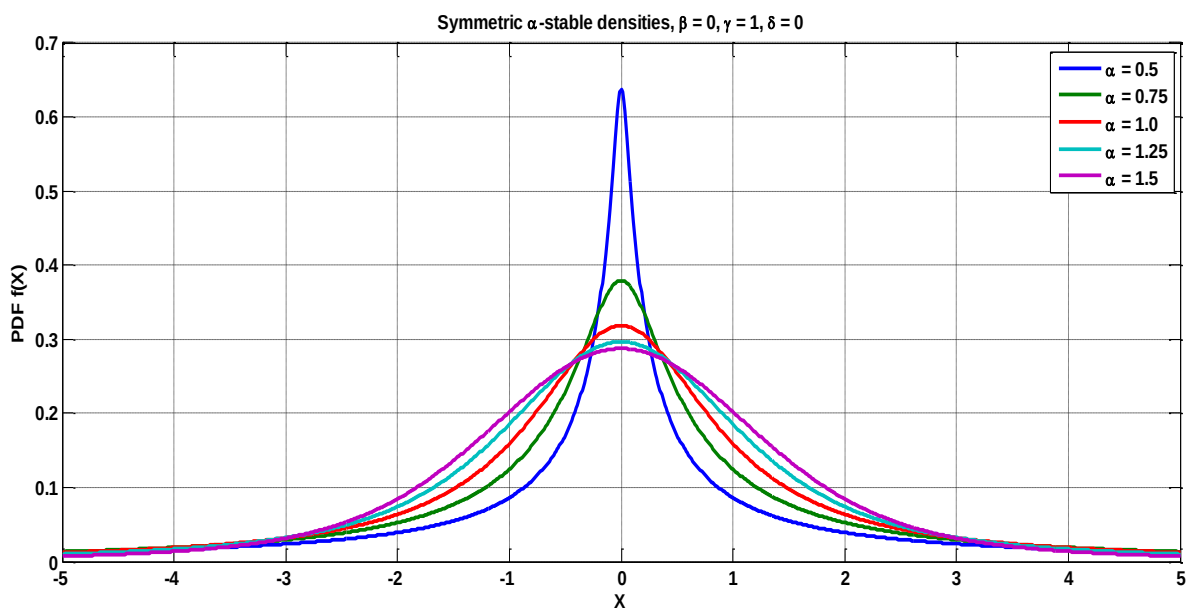


Figure 2.1 Courbes de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = \delta = 0$ et $\gamma = 1$.

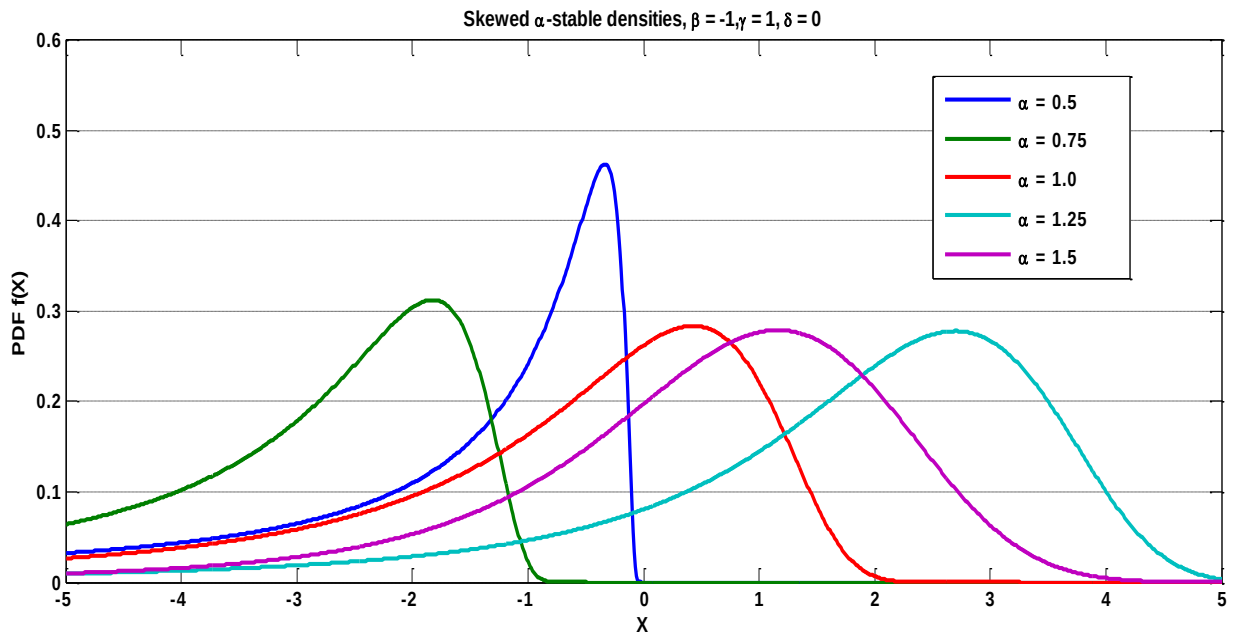


Figure 2.2 Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = -1$, $\delta = 0$ et $\gamma = 1$.

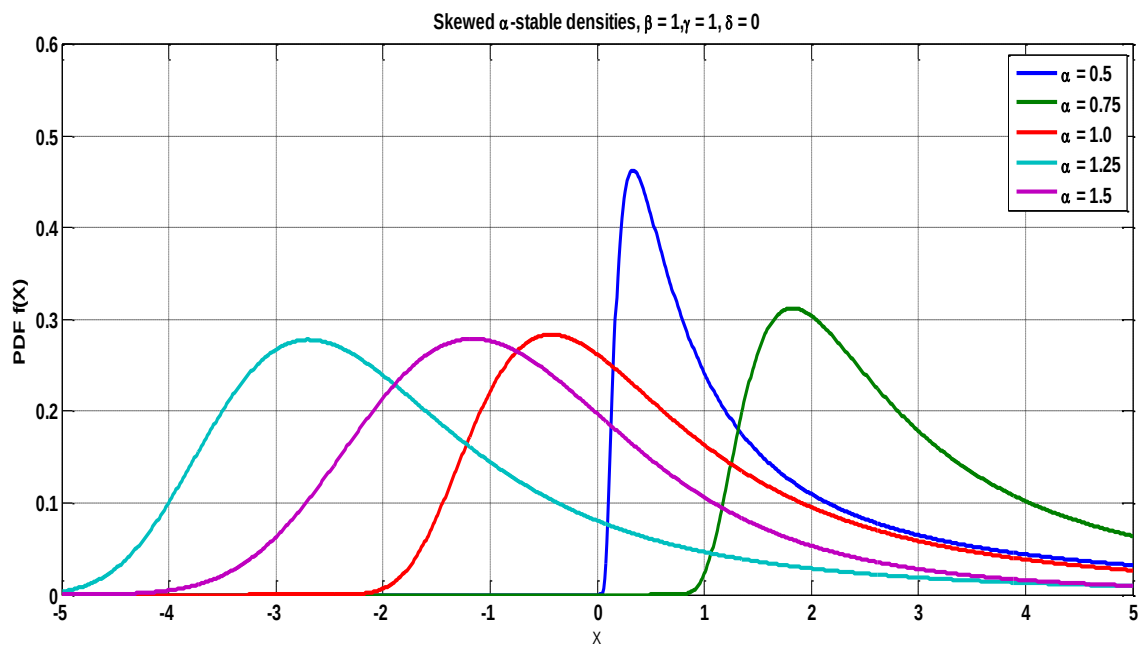


Figure 2.3 Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 1$, $\delta = 0$ et $\gamma = 1$.

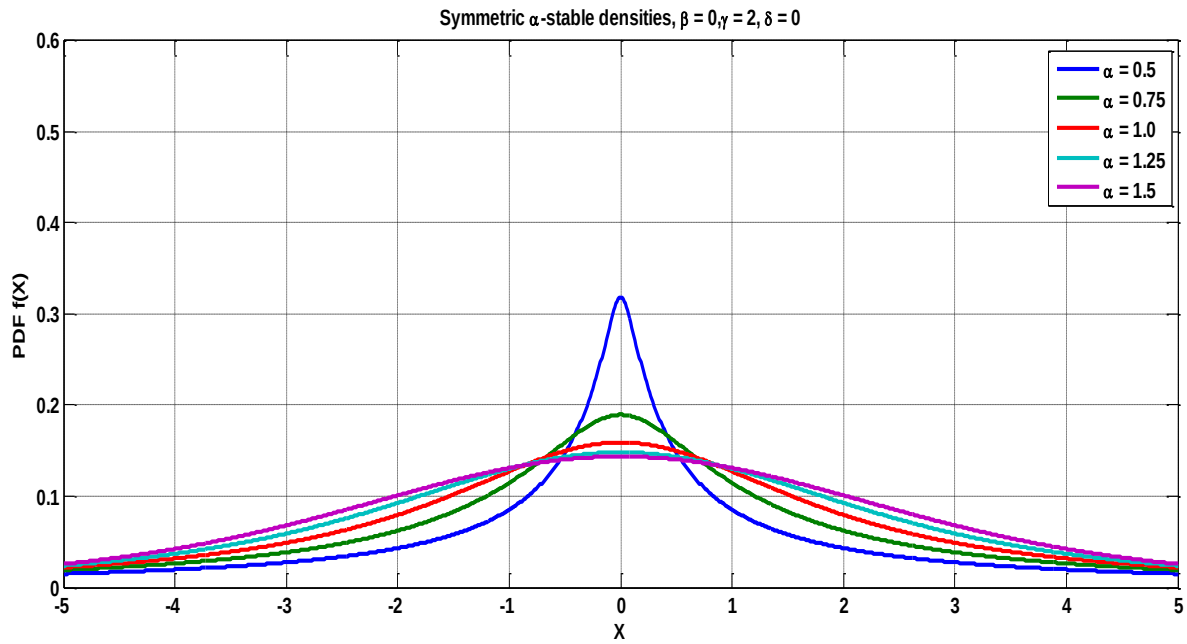


Figure 2.4 Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = \delta = 0$ et $\gamma = 2$.

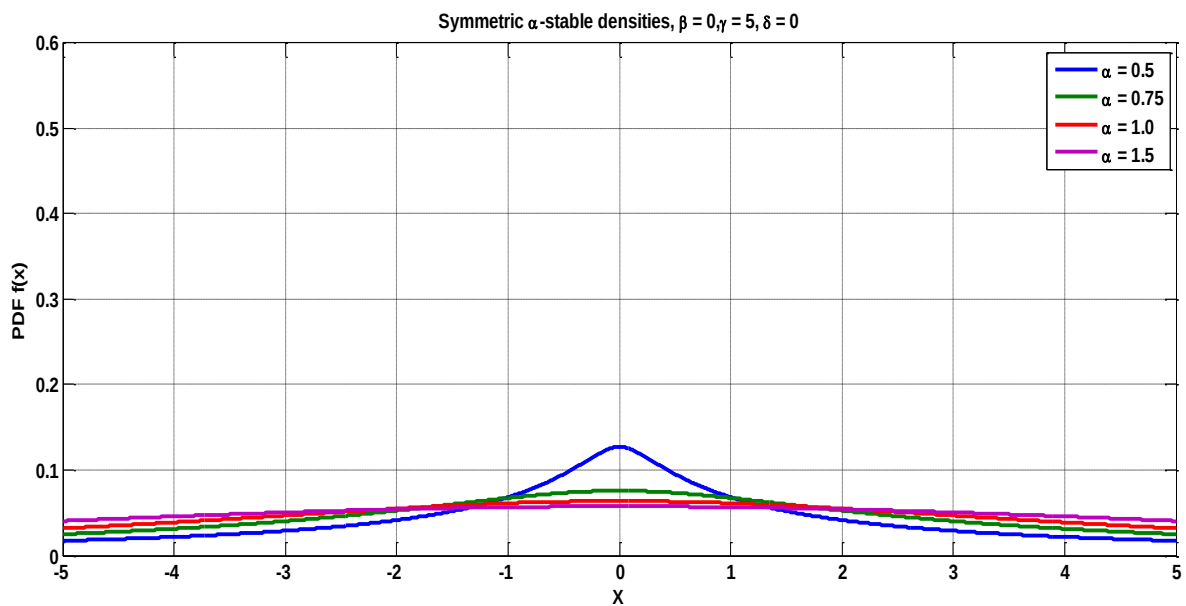


Figure 2.5 Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.5\}$, $\beta = \delta = 0$ et $\gamma = 5$.

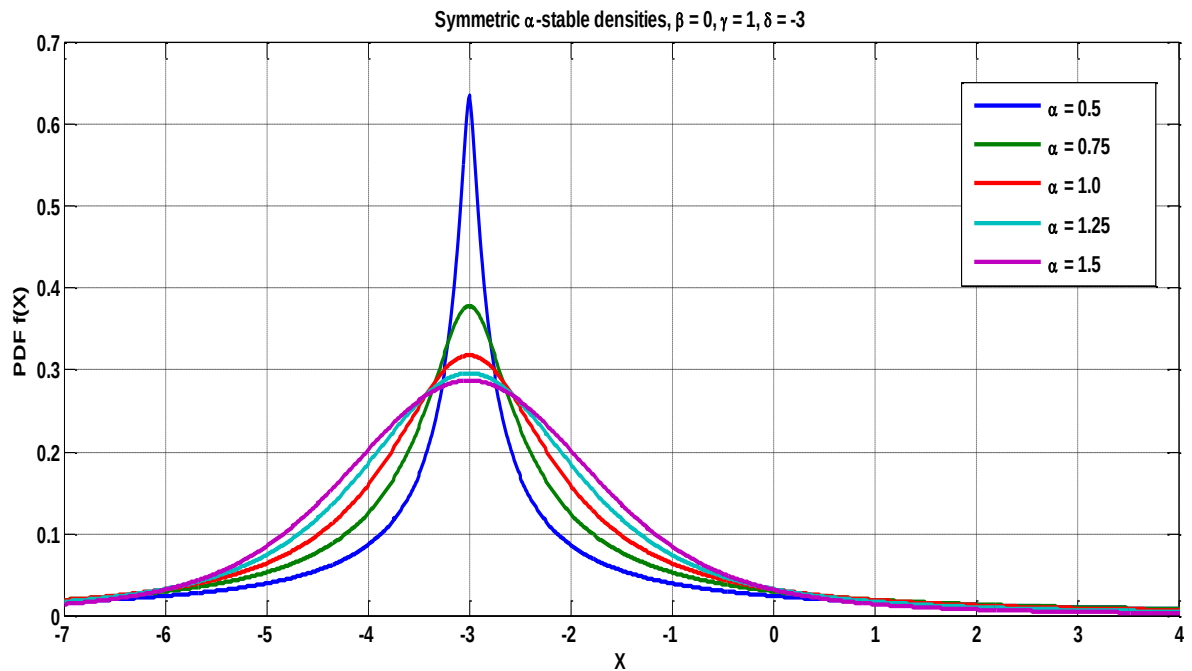


Figure 2.6 Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0$, $\delta = -3$ et $\gamma = 1$.

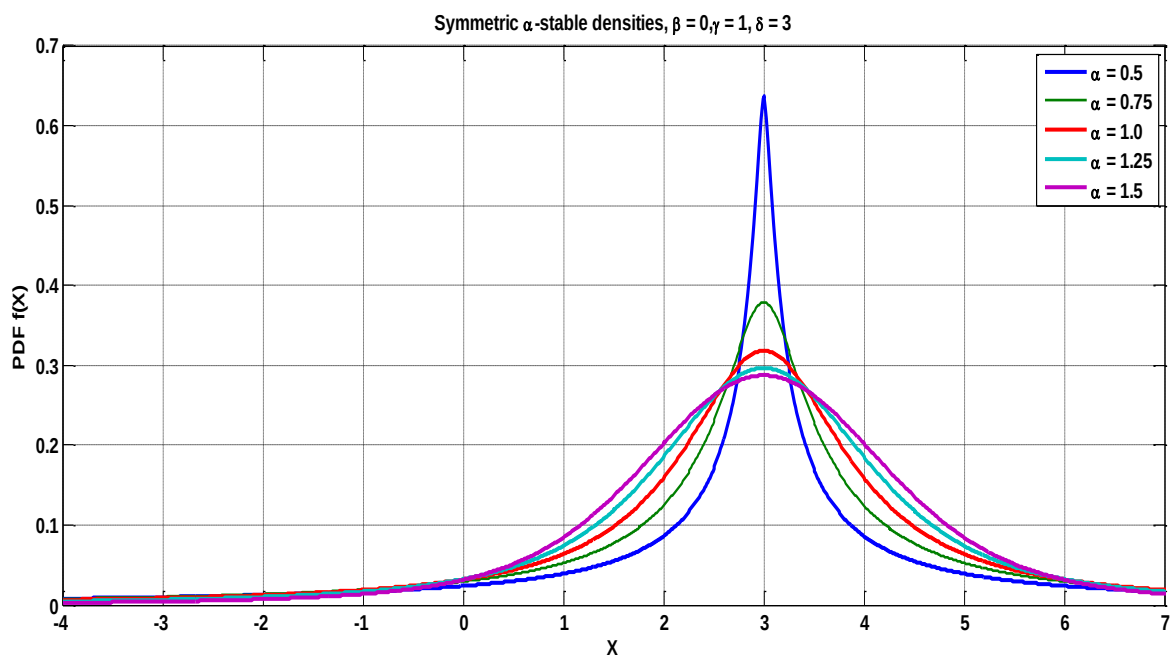


Figure 2.7 Courbe de la densité pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0$, $\delta = 3$ et $\gamma = 1$.

Remarque 2.5

Dans les tracés de la PDF $f(x)$ d'une loi stable (voir les figures 2.2 et 2.3), nous remarquons que $f(-x, \alpha, \beta) = f(x, \alpha, -\beta)$.

2.2.2 Algorithme de simulation

Pour simuler les lois stables, il existe un algorithme développé par **Chambers** et al. [46]. Celui-ci permet de générer une loi $S_\alpha(\beta, 1, 0)$. Pour obtenir une loi $S_\alpha(\beta, \gamma, \delta)$, il suffit de faire un changement de variables (propriété 2.5).

– **Première étape**

Elle consiste à générer une loi Φ uniforme sur $]-\frac{\pi}{2}, \frac{\pi}{2}[$ et une loi W exponentielle de paramètre 1. Pour cela, il faut d'abord générer deux variables aléatoires U_1 et U_2 uniformes sur $]0, 1[$. Puis en utilisant le changement de variables suivant

$$\begin{cases} \Phi = \pi U_1 - \frac{\pi}{2} \\ W = \log(1 - U_2) \end{cases} \quad (2.17)$$

on obtient bien le résultat désiré.

– **Deuxième étape**

Elle consiste à générer une loi Y stable $S_\alpha(\beta, 1, 0)$. Pour obtenir cela, il faut utiliser la proposition suivante

Proposition 2.1 Soit Φ une loi uniforme sur $]-\frac{\pi}{2}, \frac{\pi}{2}[$ et W une loi exponentielle de paramètre 1, si nous posons

– Pour $\alpha \neq 1$

$$Y = \left(\frac{\sin \alpha(\Phi - \Phi_0)}{(\cos \Phi)^{\frac{1}{\alpha}}} \right) \left(\frac{\cos(\Phi - \alpha(\Phi - \Phi_0))}{W} \right)^{\frac{1-\alpha}{\alpha}} \quad (2.18)$$

$$\text{où } \Phi_0 = \left(\frac{\pi\beta}{2} \right) \left(\frac{1-|1-\alpha|}{\alpha} \right)$$

– Pour $\alpha = 1$

$$Y = \frac{2}{\pi} \left(\left(\frac{1}{2}\pi + \beta\Phi \right) \tan \Phi - \beta \log \left(\frac{\frac{1}{2}\pi W \cos(\Phi)}{\frac{1}{2}\pi + \beta\Phi} \right) \right) \quad (2.19)$$

alors la variable aléatoire Y suit une loi $S_\alpha(\beta, 1, 0)$.

Exemple 2.2

– Dans le cas d'une loi $S\alpha S$ (c'est-à-dire $\beta = 0$), nous avons

$$\begin{cases} Y = \frac{\sin(\alpha\Phi)}{(\cos \Phi)^{\frac{1}{\alpha}}} \left(\frac{\cos((1-\alpha)\Phi)}{W} \right)^{\frac{1-\alpha}{\alpha}} & \text{si } \alpha \neq 1 \\ Y = \tan \Phi & \text{si } \alpha = 1 \end{cases} \quad (2.20)$$

- Plus particulièrement, dans le cas où $\alpha = 2$, nous avons

$$Y = \frac{\sin 2\Phi}{\sqrt{\cos \Phi}} \left(\frac{\cos \Phi}{W} \right)^{-\frac{1}{2}} = 2\sqrt{W} \sin \Phi \quad (2.21)$$

- Enfin, dans le cas où $\alpha = 1$, nous avons

$$Y = \tan \Phi \quad (2.22)$$

formule connue, qui permet de simuler une loi de **Cauchy**.

2.2.3 Statistiques sur les lois symétriques α – stables

D'après ce qui précède on peut rappeler de la propriété importante : **pour $0 < \alpha < 2$, une loi α – stable admet des moments d'ordre $p < \alpha$ uniquement.**

En accord avec cette propriété on a

- La variance d'une loi α – stable est infinie car $\alpha < 2$;
- L'espérance d'une loi stable est infinie si $\alpha < 1$;
- Une loi α – stable admet une espérance (une moyenne) ssi $\alpha > 1$ et vaut δ (le paramètre de localisation de la loi).

On suppose qu'on a N observations $(x_1, \dots, x_N)$ d'une variable aléatoire X qui suit une loi symétrique α – stable ($S\alpha S$). Si on veut prendre une idée sur la valeur de α on peut utiliser les points suscités. Par conséquent, on a besoin de tester l'espérance et la variance si elles sont finies ou non. Il y a plusieurs façons pour faire ces tests [17].

Soit à tester la variance par un test graphique [17]. Ce test est très simple et se décompose en 2 étapes

1. Calculer la variance empirique pour différentes valeurs de $n \leq N$

$$s_n^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad \text{où} \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

2. Tracer le graphique (n, s_n^2) . Intuitivement, lorsque n augmente jusqu'à N et lorsque la variance est finie, le tracé doit converger et α est égal à 2. Au contraire, si on est en présence d'une loi à variance infinie, le tracé diverge et α est strictement être inférieur à 2.

La figure 2.8 présente les tracés de $n = 5000$ observations d'une loi SaS pour différentes valeurs de α . On remarque que plus α est petit plus le tracé présente d'impulsivité.

La figure 2.9 montre le test graphique de la variance pour un nombre total d'observations $N = 5000$ et pour diverses valeurs de α .

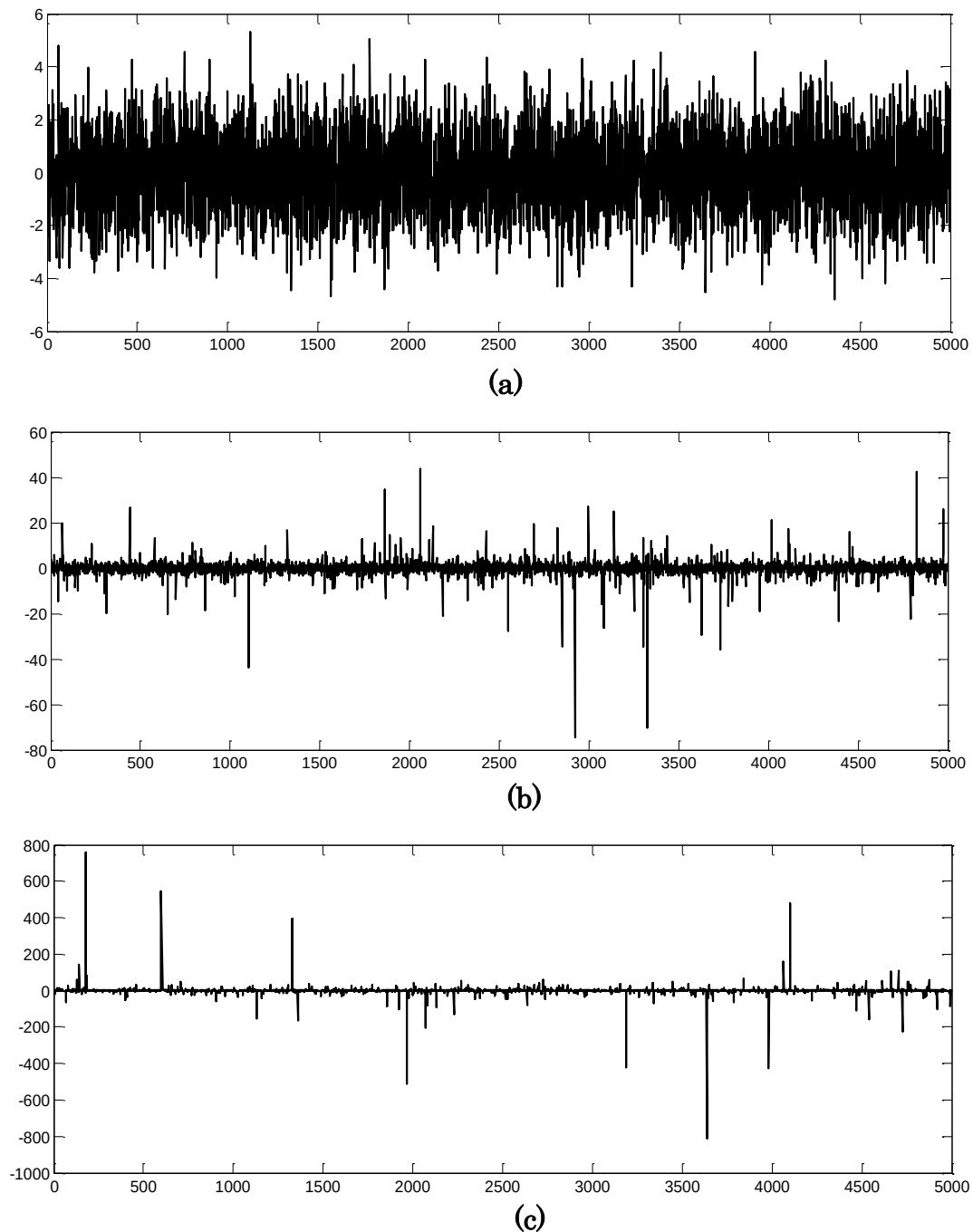


Figure 2.8 Le tracé de 5000 échantillons d'un bruit SaS (a) $\alpha = 2$ (b) $\alpha = 1.5$ (c) $\alpha = 1$.

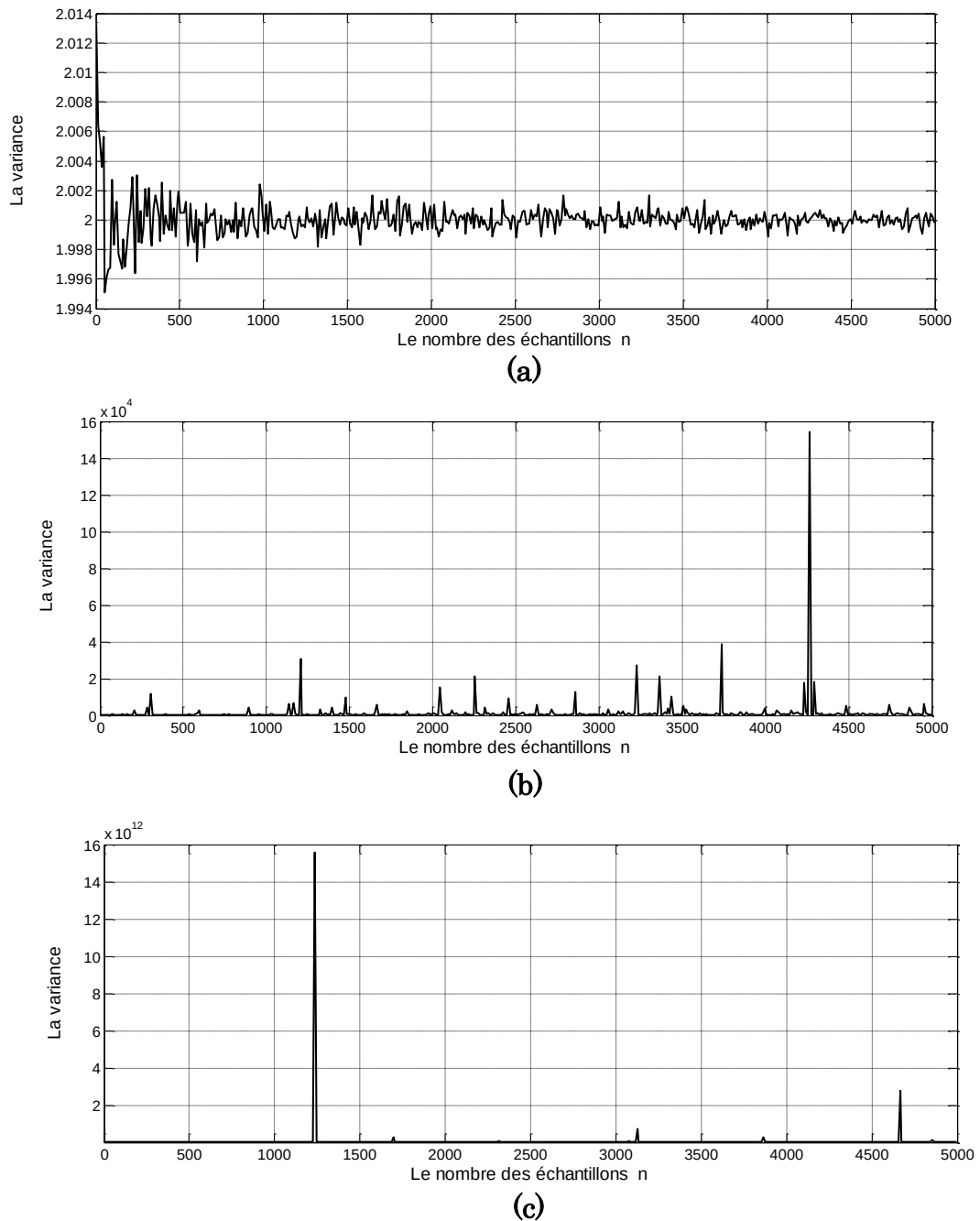


Figure 2.9 Test graphique de la variance pour $N = 5000$ échantillons d'un bruit SaS
 (a) $\alpha = 2$ (b) $\alpha = 1.5$ (c) $\alpha = 1$.

2.3 Conclusion

Dans ce chapitre, nous avons présenté une classe particulière de lois statistiques : les lois α – stables. Ce sont des lois permettant de prendre en compte les notions d'asymétrie et de queue lourde. Les lois de **Gauss** ($\alpha = 2$) et de **Cauchy** ($\alpha = 1$) sont des cas particuliers de lois α – stables. Pour une variable aléatoire α – stable nous avons vu que plus α est petit plus la queue de la loi est lourde, ce qui traduit par la présence de fortes impulsions au niveau des observations de la variable aléatoire.

Dans la section 2.2.2 nous avons exhibé un algorithme de simulation d'une variable aléatoire α – stable qui est développé par **Chambers** et al. [46].

Une classe particulière des lois α – stables est les lois symétriques ($\beta = 0$) avec un paramètre de localisation nul ($\delta = 0$). Par convention, cette classe est notée $S\alpha S$.

L'un des inconvénients des lois α – stables réside dans l'expression de sa PDF. En effet, elle est calculée à partir d'un calcul d'intégrale aux bornes infinies d'une fonction complexe comme le montre l'équation (2.5), sauf pour $\alpha = 2$ et $\alpha = 1$.

L'algorithme de décodage MAP, par exemple, exige la forme de la PDF du bruit. Un tel algorithme est alors obligé de passer aux traitements numériques à l'aide de la FFT pour surmonter le problème de la PDF du bruit sauf pour $\alpha = 2$ et $\alpha = 1$.

Le turbo décodage et l'algorithme MAP avec un bruit impulsif additif $S\alpha S$ pour $\alpha = 1$ (bruit de **Cauchy**) font l'objet du chapitre suivant.

Chapitre 3

Le turbo décodage et l'algorithme MAP

3.1 Introduction

Le théorème de **Shannon** affirme qu'il existe au moins un code de canal qui nous permet de transmettre de l'information sous forme d'une série de symboles discrets avec une probabilité d'erreur faible que l'on veut à condition que le débit d'information soit inférieur à la capacité de canal.

Après un long tâtonnement **C. Berrou**, **A. Glavieux** et **P. Thitimajshima**, en 1993 [47] ont trouvé un très bon code correcteur d'erreurs fonctionnant à moins de 0.5 dB de la limite de **Shannon** pour un canal à bruit gaussien [48]. Cette rupture technologique dans le domaine du codage de canal a tout d'abord surpris la communauté scientifique, mais les résultats ont été très rapidement confirmés. La solution proposée, qui consiste en une concaténation parallèle ou série de deux codes convolutifs, généralement récurrents systématiques identiques [49], au travers d'un entrelaceur, est connue sous le nom « Turbo Codes Convolutifs CTC ».

Le but de l'entrelaceur (Interleaver) dans un CTC est de faire la permutation des données émises afin de casser les paquets d'erreur surviennent lors de la transmission, et ce pour rendre leur distribution plus uniforme, ce qui donne au CTC une grande efficacité de corriger les erreurs [21].

Le décodeur associé à un CTC est un décodeur souple ou doux, car ce type de décodeurs est très performant qu'un décodeur dur ou ferme pour un canal à bruit impulsif, et très facile à mettre en œuvre pour tous les codes acceptant des représentations en treillis, par la raison qu'il utilise un algorithme Maximum A Posteriori MAP.

Un tel décodage permet d'extraire de l'information sur chacun des décodeurs constitutifs et de l'échanger entre ces deux décodeurs. L'information extraire d'un décodeur est dite « extrinsèque » et est réinjectée à l'itération suivante dans un autre décodeur composant afin de bénéficier de la diversité de codage. Le processus pouvant se répéter plusieurs fois, on parle de décodage itératif ou de turbo décodage. L'appellation « Turbo » provient justement de la réinjection des informations extrinsèques pour rappeler l'image des moteurs turbo qui réutilisent une partie de l'énergie gaspillée par le moteur pour donner plus de puissance au moteur.

3.2 Le décodage itératif et l'Algorithme MAP

Dans un système de communication numérique sur un canal bruité, un codeur est souvent utilisé à l'émetteur avant l'étape de modulation, afin de corriger les erreurs de transmission au récepteur. Lorsque la structure du code s'y prête, le processus de décodage peut être exécuté en plusieurs étapes ou itérations simples, d'où le nom de décodage itératif.

3.2.1 La métrique de Vraisemblance Logarithmique ou de corrélation

Une fois codés, les bits d'information sont transmis sur le canal. A la réception, partant des mots-code reçus et comportant, à cause du bruit, des erreurs, il faut trouver une règle de décision qui optimise un critère comme par exemple celui de maximiser la probabilité de décision correcte.

Dans un système de communication numérique le canal est caractérisé par un alphabet d'entrée, un alphabet de sortie et une loi de probabilité de la sortie conditionnellement à l'entrée (probabilité de transition).

Dans notre étude on s'intéresse au canal analogique à bruit impulsif additif de type $S\alpha S$ où $\alpha = 1$ (bruit de **Cauchy**). La sortie se représente par une variable aléatoire continue R_k dite l'observation du canal.

$$R_k = Y_k + W_k \quad (3.1)$$

où Y_k est le symbole émis et W_k est une variable aléatoire $S\alpha S$, de paramètre de dispersion γ , correspondant au bruit du canal.

Pour une modulation BPSK l'alphabet du symbole Y_k est $\{+\sqrt{E_b}, -\sqrt{E_b}\}$ tel que la fonction de modulation $m(X_k)$ fait correspondre à un bit $X_k = 0$ le symbole $Y_k = +\sqrt{E_b}$ et à

un bit $X_k = 1$ le symbole $Y_k = -\sqrt{E_b}$ où X_k représente la sortie binaire du CTC et E_b est l'énergie du signal par bit. Sans perdre de généralité, on peut prendre $\sqrt{E_b} = 1$ et la fonction de modulation devient

$$Y_k = m(X_k) = \begin{cases} +1 & \text{pour un bit } X_k = 0 \\ -1 & \text{pour un bit } X_k = 1 \end{cases} \quad (3.2)$$

ou aussi $Y_k = 1 - 2X_k = (-1)^{X_k}$

Les densités de probabilité de transition du canal s'écrivent

$$P(R_k/Y_k \text{ émis}) = \frac{2\gamma}{\pi((R_k - Y_k)^2 + 4\gamma^2)} \quad (3.3)$$

Pour tirer profit de l'information portée par le symbole reçu R_k dans la sortie du canal, on doit étudier avec un soin méticuleux les statistiques du bruit impulsif W_k , du coup, il faut prendre toutes les valeurs possibles de la sortie du canal. On parle donc d'un canal à décision douce, souple ou soft.

Au niveau du récepteur on doit trouver à partir de l'observation R_k une estimation du symbole Y_k de sorte que la probabilité moyenne de l'erreur de décision soit minimisée, donc la probabilité conditionnelle de la relation (3.3) soit maximisée. Le principe de Maximum de Vraisemblance (Maximum Likelihood ML) donne la solution cherchée à ce problème. Le ML cherche à maximiser la probabilité $P(R_k/Y_k \text{ émis})$ au sens d'**Euclid**, ça veut dire il maximise le logarithme népérien de cette probabilité. On élimine toutes les constantes dans l'expression de $\ln(P(R_k/Y_k \text{ émis}))$ on trouve l'expression à maximiser qui est $-\ln|R_k - Y_k|$. Maximiser la dernière expression revient à minimiser la quantité $|R_k - Y_k|$. Pour que cette dernière quantité soit minimale il faut que R_k soit de même signe que Y_k , autrement dit, il faut que le produit $R_k \times Y_k$ soit positif. On peut donc définir la **métrique de Vraisemblance Logarithmique ou de corrélation** comme le produit $M = R_k \times Y_k$ qui est une mesure de la probabilité $P(R_k/Y_k \text{ émis})$.

Pour calculer la métrique M on utilise la règle suivante « si $X_k = 0$, alors $Y_k = 1$, la métrique $M = R_k$ (pas de changement de signe de l'observation R_k), et si $X_k = 1$, alors $Y_k = -1$, la métrique $M = -R_k$ (il y a un changement de signe de R_k) » [50].

3.2.2 Le principe du Maximum A Posteriori MAP

L'algorithme MAP consiste à minimiser la probabilité d'erreur par symbole ou par bit. Dans le cas du MAP on parlera de la règle du maximum a posteriori c.à.d. de prendre le maximum des probabilités a posteriori (APP) du symbole ou bit envoyé dans le canal. Le calcul de l'APP se base sur l'observation de la séquence reçue, d'où la nomination « a posteriori » qui signifie en latin "en partant de ce qui vient après". Afin d'éviter la confusion des termes qui sont utilisés dans la littérature, nous rappelons que la notion Algorithme MAP la signifie que la règle de maximum a posteriori associée à un symbole a été appliquée. Nous pouvons alors parler de l'algorithme MAP symbole-par-symbole. La règle de maximum a posteriori peut être appliquée à un symbole ou une séquence. Dans le cas où cette règle est appliquée à une séquence (ou un mot de code par exemple) et que la distribution des mots de source est uniforme, ceci revient exactement à appliquer le principe ML. En effet, il suffit d'appliquer la règle de **Bayes** afin de prouver l'identité entre les deux règles. Avec le MAP à la sortie du décodeur, les bits décodés ne sont pas comparées à un seuil donné, les valeurs (des bits) obtenues ne sont pas modifiées (par large approximation ou quantification). Dans ce cas, nous pouvons parler d'une décision douce (Soft) en opposition à une décision dure où la valeur du bit à la sortie du décodeur est affectée soit à la valeur 0 soit à 1.

Soient les événements A_i , $i = 1, 2, \dots, M$, l'ensemble des messages transmis dans un intervalle de temps donné, $P(A_i)$ représente la probabilité a priori de ces événements. Soit B le signal reçu qui n'est qu'un des A_i corrompu par du bruit. B est une variable aléatoire continue de densité de probabilité $p(B)$. L'expression $P(A_i|B)$ est l'APP de l'événement A_i sachant que le signal reçu B a été observé. Le détecteur MAP cherche à maximiser l'APP $P(A_i|B)$. Cette dernière s'exprime sous la forme [51]

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{p(B)} \quad (3.4)$$

où $P(B|A_i)$ est la fonction de densité de probabilité (PDF) conditionnelle du vecteur observé B , et $P(A_i)$ est la probabilité a priori du $i^{\text{ème}}$ signal transmis. Le dénominateur de l'équation (3.4) est donné par

$$p(B) = \sum_{i=1}^M p(B|A_i)P(A_i) \quad (3.5)$$

Notons que $p(B|A_i)$ est appelé aussi la fonction de vraisemblance. Le critère de

décision qui consiste à prendre le maximum de $p(B|A_i)$ parmi les M signaux est le principe du ML si les signaux sont équiprobables.

Si la distribution des signaux A_i est uniforme (c.à.d les signaux A_i sont équiprobables), or $p(B)$ est indépendant du signal A_i transmis, donc maximiser $P(A_i|B)$ revient à maximiser l'expression $p(B|A_i)$. Donc le détecteur basé sur le principe MAP fera la même décision que celui basé sur le principe ML.

Dans notre étude les événements A_i appartiennent au champ de Galois $GF(2)$ dont les éléments sont $\{0, 1\}$ ou 0 est l'élément neutre sous l'addition $\oplus$, donc $M = 2$. Soit R la séquence reçue sur toutes les classes de signaux. Alors l'équation (3.4) devient

$$P(d_k = i|R) = \frac{P(R|d_k = i)P(d_k = i)}{p(R)} \quad i = 0, 1 \quad (3.6)$$

La règle de décision MAP consiste à comparer les probabilités APP $P(d_k = 0|R)$ et $P(d_k = 1|R)$, ensuite à en prendre le maximum, ceci peut se résumer dans l'équation suivante

$$P(d_k = 1|R) \underset{H_1}{\overset{H_0}{\leq}} P(d_k = 0|R) \quad (3.7)$$

Dans l'équation (3.7) l'hypothèse H_1 signifie que le détecteur MAP assigne d_k à la valeur 1 dans le cas où $P(d_k = 1|R)$ est supérieur à $P(d_k = 0|R)$. Dans le cas contraire le détecteur MAP choisit ($d_k = 0$) ce qui correspond à l'hypothèse H_0 .

La décision ferme optimale donne $\hat{d}_k$ au sens du MAP est alors la suivante

$$\hat{d}_k = \begin{cases} 0 & \text{si } P(d_k = 0|R) > P(d_k = 1|R) \\ 1 & \text{sinon.} \end{cases} \quad (3.8)$$

La décision souple est définie par le rapport de vraisemblance logarithmique LLR

$$LLR(d_k) \triangleq \ln \left[\frac{P(d_k = 1|R)}{P(d_k = 0|R)} \right] \quad (3.9)$$

En utilisant (3.6) la décision souple peut être décomposée en

$$LLR(d_k) = \Lambda_c(R) + \Lambda_a(d_k) \quad (3.10)$$

où $\Lambda_c(R) = \ln \left[\frac{p(R|d_k=1)}{p(R|d_k=0)} \right]$ est le rapport de vraisemblance logarithmique du canal et $\Lambda_a(d_k) = \ln \left[\frac{P(d_k=1)}{P(d_k=0)} \right]$ est le rapport de vraisemblance logarithmique a priori. Le critère de décision au sens du MAP peut alors se s'écrire

$$\hat{d}_k = \begin{cases} 1 & \text{si } \Lambda(d_k) > 0 \\ 0 & \text{sinon.} \end{cases} \quad (3.11)$$

Il s'ensuit que le signe de la décision souple détermine la décision ferme et que la valeur absolue de la décision souple détermine la fiabilité de cette décision. Il est important à signaler que le LLR obtenu est valable pour un seul bit ou symbole.

En particulier, si l'on suppose le bruit impulsif de **Cauchy SaS**, le rapport de vraisemblance logarithmique du canal a pour expression

$$\Lambda_c(R) = \ln \left[\frac{\frac{2\gamma}{\pi((R_k+1)^2 + 4\gamma^2)}}{\frac{2\gamma}{\pi((R_k-1)^2 + 4\gamma^2)}} \right] = \ln \left[\frac{(R_k - 1)^2 + 4\gamma^2}{(R_k + 1)^2 + 4\gamma^2} \right] \quad (3.12)$$

Jusqu'à présent nous avons considéré un système de transmission non-codé. Pour un système de transmission codé, nous pouvons montrer que si l'information a priori est distribuée de manière indépendante, la sortie souple d'un décodeur peut s'écrire sous la forme [52]

$$LLR(d_k) = \Lambda_c(R) + \Lambda_a(d_k) + \Lambda_e(d_k) = \Lambda_c(R) + \Psi(d_k) \quad (3.13)$$

où $\Lambda_e(d_k)$ est le rapport de vraisemblance logarithmique représentant la connaissance acquise grâce au processus de décodage. La quantité $\Psi(d_k) = \Lambda_a(d_k) + \Lambda_e(d_k)$ est appelée **information extrinsèque**, indépendante de R .

3.2.3 Notations et règle de décision

Afin d'appliquer le principe MAP aux turbo codes, considérons un (RSC) de taux de codage $R = 1/2$ (voir chapitre 4, figure 4.1.a). Nous allons adopter les notations suivantes

- $d_k = i$, $i = 0, 1$ le bit d'information présent à l'entrée du codeur à l'instant k ;
- $M = (K - 1)$ bits, est la mémoire du codeur ;
- $S_k = m$, $m = 0, 1, \dots, 2^M$ est l'état du codeur à l'instant k ;
- N est la taille de la séquence à coder (ou encore la longueur du bloc exprimée en bits) ;
- $(d_1, \dots, d_k, \dots, d_N) = (X_1^s, \dots, X_k^s, \dots, X_N^s)$ est la séquence à coder ;
- $(X_1^s, \dots, X_k^s, \dots, X_N^s)$ est appelée la séquence systématique ;
- $(X_1^p, \dots, X_k^p, \dots, X_N^p)$ est la séquence de parité à la sortie du codeur RSC ;
- $R_k = (r_k^s, r_k^p)$ est la version bruitée de (X_k^s, X_k^p) à l'instant k ;
- $R_1^N = (R_1, R_2, \dots, R_N)$ est la séquence reçue après passage dans un canal à bruit impulsif SaS.

En supposant que la séquence reçue est R_1^N , l'expression du LLR devient

$$LLR(d_k) = \ln \left[\frac{P(d_k = 1 | R_1^N)}{P(d_k = 0 | R_1^N)} \right] \quad (3.14)$$

Le calcul de l'APP $P(d_k = i | R_1^N)$, $i = 0, 1$ s'avère complexe d'où l'idée d'introduire la probabilité conjointe (JD) définie par

$$\lambda_k^i(m) = P(d_k = i, S_k = m | R_1^N) \quad (3.15)$$

donc l'APP d'un bit décodé d_k , peut s'exprimer en fonction de la JD et est égale à

$$P(d_k = i | R_1^N) = \sum_{m=0}^{2^M-1} \lambda_k^i(m) \quad (3.16)$$

où $i = 0, 1$ et la sommation est sur les 2^M états du codeur.

En remplaçant l'expression de la probabilité conjointe dans l'équation (3.14), nous obtenons

$$LLR(d_k) = \ln \left[\frac{\sum_{m=0}^{2^M-1} \lambda_k^1(m)}{\sum_{m=0}^{2^M-1} \lambda_k^0(m)} \right] \quad (3.17)$$

Le décodeur MAP peut faire la décision sur le bit décodé en comparant le $LLR(d_k)$ à un seuil égal à zéro

$$\text{Si } LLR(d_k) \geq 0 \text{ le bit décodé est } 1 \quad (3.18)$$

$$\text{Si } LLR(d_k) < 0 \text{ le bit décodé est } 0 \quad (3.19)$$

Notons que l'expression de la probabilité conjointe (JD) diffère d'un auteur à un autre. Dans l'algorithme d'origine BJRC [53], JD a été définie sous forme de deux équations

$$\delta_k(m) = P(S_k = m | R_1^N) \quad (3.20)$$

$$\sigma_k(m', m) = P(S_{k-1} = m', S_k = m, R_1^N) \quad (3.21)$$

En analysant, la deuxième équation (3.21), nous remarquons que ceci revient à calculer la JD en la multipliant par une constante. En effet, nous travaillons dans un espace binaire donc le passage d'un état à un autre (nous supposons que le codeur est représenté sous forme de treillis) est provoqué par un bit d'entrée. En supposant que le

passage de l'état $S_{k-1} = m'$ à l'état $S_k = m$ est provoqué par le bit $d_{k-1} = i$ et en appliquant ceci à σ_k nous obtenons

$$\begin{aligned}
 \sigma_k(m', m) &= P(S_{k-1} = m', S_k = m, R_1^N) \\
 &= P(S_{k-1} = m', S_k = m | R_1^N) \times P(R_1^N) \\
 &= P(d_{k-1} = i, S_k = m | R_1^N) \times P(R_1^N) \\
 &= \lambda_{k-1}^i(m) \times P(R_1^N)
 \end{aligned} \tag{3.22}$$

La majorité des auteurs ont adopté la définition de la probabilité conjointe que nous avons présenté en premier (équation (3.15)). En revanche le développement de la JD afin de simplifier le LLR est beaucoup plus diversifié. Pour notre part, nous avons adopté le dernier développement utilisé par [54]. Évidemment, la façon dont l'expression est développée affectera le nombre d'opérations ou plus c.à.d. la complexité de l'algorithme MAP. Nous allons essayer d'analyser les versions de l'algorithme MAP les plus utilisées. A l'origine du MAP est l'article phare paru en 1993 [47]. Les auteurs de ce dernier article ont adopté une version modifiée de l'algorithme original BJRC. Cette version a été appelée BJRC modifiée du fait qu'elle a été adoptée en particulier pour le codes récurrents convolutifs (RSC).

Dans un premier temps, nous allons décrire la dernière version de l'algorithme MAP que nous avons utilisée. Ensuite, nous allons présenter brièvement les quelques versions les plus utilisées de l'algorithme tout en essayant de faire un parallèle avec la version que nous avons adoptée.

3.3 L'Algorithme MAP utilisé

3.3.1 Expression de la probabilité conjointe

En notant que les événements R_k sont indépendants et en utilisant des relations les plus élémentaires en probabilité, nous pouvons développer l'expression de la probabilité conjointe sous la forme suivante

$$\begin{aligned}
 \lambda_k^i(m) &= P(d_k = i, S_k = m | R_1^N) \\
 &= \frac{P(d_k = i, S_k = m, R_1^{k-1}, R_k^N)}{P(R_1^N)}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{P(R_1^{k-1} | d_k = i, S_k = m, R_k^N) \times P(d_k = i, S_k = m, R_k^N)}{P(R_1^N)} \\
&= \frac{P(R_1^{k-1} | d_k = i, S_k = m, R_k^N) \times P(d_k = i, S_k = m, R_k, R_{k+1}^N)}{P(R_1^N)} \\
&= \frac{P(R_1^{k-1} | d_k = i, S_k = m, R_k^N) \times P(R_{k+1}^N | d_k = i, S_k = m, R_k) \times P(d_k = i, S_k = m, R_k)}{P(R_1^N)} \quad (3.23)
\end{aligned}$$

Nous allons définir la métrique d'état en avant FSM par

$$\alpha_k(m) = P(R_1^{k-1} | d_k = i, S_k = m, R_k^N) \quad (3.24)$$

or les événements produits après l'instant k n'influencent pas les événements qui les ont précédés, donc l'équation précédente se réduit à

$$\alpha_k(m) = P(R_1^{k-1} | S_k = m) \quad (3.25)$$

De même nous définissons aussi la métrique d'état en arrière BSM comme

$$\beta_k(m) = P(R_k^N | S_k = m) \quad (3.26)$$

Nous définissons la métrique de branche (BM) par

$$\delta_k^i(m) = P(d_k = i, S_k = m, R_k) \quad (3.27)$$

En remplaçant les expressions des FSM, BSM et BM dans l'équation (3.23)

$$\begin{aligned}
\lambda_k^i(m) &= \frac{\alpha_k(m) \times P(R_{k+1}^N | d_k = i, S_k = m, R_k) \times \delta_k^i(m)}{P(R_1^N)} \\
&= \frac{\alpha_k(m) \times P(R_{k+1}^N | S_{k+1} = f(i, m)) \times \delta_k^i(m)}{P(R_1^N)} \\
&= \frac{\alpha_k(m) \times \beta_{k+1}(f(i, m)) \times \delta_k^i(m)}{P(R_1^N)} \quad (3.28)
\end{aligned}$$

En connaissant l'état du codeur $S_k = m$ et le bit d'entrée $d_k = i$ à l'instant k , l'état successeur S_{k+1} peut se noter sous la forme $S_{k+1} = f(i, m)$. L'égalité précédente signifie que l'état précédent de S_{k+1} est l'état m et que le bit d'entrée à l'instant k est $d_k = i$.

Finalement, en remplaçant l'expression de la probabilité conjointe dans l'équation (3.17), nous obtenons

$$LLR(d_k) = \ln \left[\frac{\sum_{m=0}^{2^M-1} \alpha_k(m) \times \beta_{k+1}(f(1, m)) \times \delta_k^1(m)}{\sum_{m=0}^{2^M-1} \alpha_k(m) \times \beta_{k+1}(f(0, m)) \times \delta_k^0(m)} \right] \quad (3.29)$$

D'après l'équation (3.29), le LLR est en fonction des trois expressions définies précédemment: FSM, BSM et BM. Donc, il suffit de calculer ces trois expressions afin de fournir une valeur pour le LLR. Nous allons d'abord simplifier l'expression de la FSM ensuite celle de la BSM et finalement celle de la métrique de branche BM.

3.3.2 Expression de la métrique d'état en avant FSM

Nous pouvons remarquer que la FSM à l'instant k dépend de la séquence reçue R_1^k qui précède l'instant k (ou au plus jusqu'à l'instant k). En utilisant la règle de **Bayes**, nous pouvons exprimer la métrique d'état en avant sous la forme

$$\begin{aligned} \alpha_k(m) &= P(R_1^{k-1} | S_k = m) \\ &= \sum_{m'=0}^{2^M-1} \sum_{j=0}^1 P(d_{k-1} = j, S_{k-1} = m', R_1^{k-1} | S_k = m) \\ &= \sum_{m'=0}^{2^M-1} \sum_{j=0}^1 P(R_1^{k-2} | S_k = m, d_{k-1} = j, S_{k-1} = m', R_{k-1}) \times P(d_{k-1} = j, S_{k-1} = m', R_{k-1} | S_k = m) \end{aligned}$$

Sachant que le passage de l'état $S_{k-1} = m'$ a été provoqué par le bit $d_{k-1} = j$, nous pouvons compacter ces informations sous la forme $S_{k-1} = b(j, m)$. Ceci signifie que l'état S_{k-1} est l'état qui précède l'état $S_k = m$ sachant que le bit d'entrée au codeur à l'instant $(k-1)$ est égale j .

$$\begin{aligned} \alpha_k(m) &= \sum_{m'=0}^{2^M-1} \sum_{j=0}^1 P(R_1^{k-2} | S_{k-1} = b(j, m)) \times P(d_{k-1} = j, S_{k-1} = b(j, m), R_{k-1}) \\ &= \sum_{j=0}^1 \alpha_{k-1}(b(j, m)) \times \delta_{k-1}^j(b(j, m)) \end{aligned} \quad (3.30)$$

Nous pouvons déduire que la métrique d'état à l'instant k s'exprime en fonction de la FSM à l'instant $k-1$ d'où la nomination « en avant ». En effet, il faut connaître la valeur initiale à l'instant $k=0$ afin d'avancer aux instants suivants.

La métrique d'état en avant peut être schématisée sous forme graphique comme à la figure 3.1.

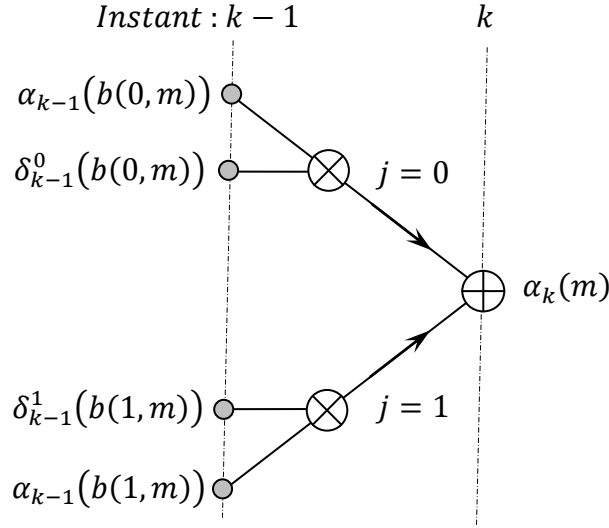


Figure 3.1 Représentation graphique de la métrique d'état en avant.

3.3.3 Expression de la métrique d'état en arrière BSM

Contrairement à la FSM, la métrique d'état en arrière à l'instant k dépend de la séquence reçue après cet instant k (ou à partir de l'instant k) et jusqu'à recevoir la totalité de la séquence (instant N). En utilisant encore la règle de **Bayes**, nous pouvons exprimer la métrique d'état en arrière BSM sous la forme

$$\begin{aligned}
 \beta_k(m) &= P(R_k^N | S_k = m) \\
 &= \sum_{m'=0}^{2^M-1} \sum_{j=0}^1 P(d_k = j, S_{k+1} = m', R_k^N | S_k = m) \\
 &= \sum_{m'=0}^{2^M-1} \sum_{j=0}^1 P(R_{k+1}^N | S_k = m, d_k = j, S_{k+1} = m', R_k) \times P(d_k = j, S_{k+1} = m', R_k | S_k = m) \\
 \beta_k(m) &= \sum_{j=0}^1 P(R_{k+1}^N | S_{k+1} = f(j, m)) \times P(d_k = j, S_k = m, R_k) \\
 &= \sum_{j=0}^1 \beta_{k+1}(f(j, m)) \times \delta_k^j(m)
 \end{aligned} \tag{3.31}$$

Nous pouvons déduire que la BSM à l'instant k s'exprime en fonction de la BSM à l'instant $k+1$ d'où la nomination « en arrière ». En effet, il faut connaître la valeur initiale à l'instant $k=N$ afin de reculer dans le treillis et de calculer les BSM précédentes. La métrique d'état en arrière peut illustrée être aussi schématisée sous forme graphique à la figure 3.2.

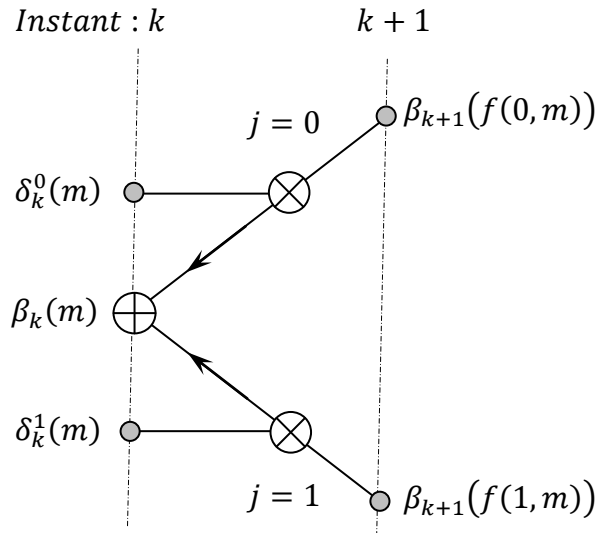


Figure 3.2 Représentation graphique de la métrique d'état en arrière.

3.3.4 Calcul de la métrique de branche BM

Comme nous l'avons indiqué précédemment le calcul de la métrique de branche est nécessaire afin de calculer le logarithme du rapport de vraisemblance. D'après l'expression de δ_k^i (équation (3.27)), la BM à l'instant k dépend seulement de l'instant même et non plus des instants précédents ou suivants d'où la nomination métrique de branche.

$$\begin{aligned}\delta_k^i(m) &= P(d_k = i, S_k = m, R_k) = P(R_k | d_k = i, S_k = m) \times P(d_k = i, S_k = m) \\ &= P(R_k | d_k = i, S_k = m) \times P(S_k = m | d_k = i) \times P(d_k = i)\end{aligned}\quad (3.32)$$

En remplaçant $R_k = (r_k^s, r_k^p)$ dans l'équation précédente nous obtenons

$$\delta_k^i(m) = P(r_k^s | d_k = i, S_k = m) \times P(r_k^p | d_k = i, S_k = m) \times P(S_k = m | d_k = i) \times P(d_k = i) \quad (3.33)$$

Les deux premiers termes de la métrique de branche dépendent des symboles recueillis à la sortie du canal. Donc le type canal utilisé affecte l'expression de la BM. Nous allons exprimer la BM dans un canal à bruit blanc additif impulsif à GGD.

Nous pouvons exprimer respectivement la densité de probabilité de l'information reçue et celle de l'information de parité reçue comme suit

$$P(r_k^s | d_k = i, S_k = m) = \frac{2\gamma}{\pi((r_k^s - (1 - 2i))^2 + 4\gamma^2)} \quad (3.34)$$

et

$$P(r_k^p | d_k = i, S_k = m) = \frac{2\gamma}{\pi(r_k^p - (1 - 2c_m^i))^2 + 4\gamma^2)} \quad (3.35)$$

où c_m^i est le bit codé sachant $d_k = i$ et $S_k = m$. En remplaçant les deux précédentes expressions dans l'équation (2.35), la métrique de branche devient

$$\begin{aligned}
\delta_k^i(m) &= \frac{2\gamma}{\pi((r_k^s - (1 - 2i))^2 + 4\gamma^2)} \times \frac{2\gamma}{\pi(r_k^p - (1 - 2c_m^i))^2 + 4\gamma^2)} \times \frac{1}{2^M} \times P(d_k = i) \\
&= \frac{1}{2^M} \times \frac{4\gamma^2}{\pi^2((r_k^s - (1 - 2i))^2 + 4\gamma^2) \times (r_k^p - (1 - 2c_m^i))^2 + 4\gamma^2)} \times P(d_k = i) \\
&= C(M, \gamma) \times P(d_k = i) \times \frac{1}{((r_k^s - (1 - 2i))^2 + 4\gamma^2) \times (r_k^p - (1 - 2c_m^i))^2 + 4\gamma^2)} \quad (3.36)
\end{aligned}$$

où la variable $C(\cdot)$ ne dépend pas de la variable i et de l'état m . $C(\cdot)$ peut être supprimée car en faisant le LLR, la constante $C(\cdot)$ disparaîtra.

3.4 Principe du décodeur itératif MAP

Le décodage des turbo codes peut être réalisé à partir de deux décodeurs DEC1 et DEC2 associés selon le principe représenté sur la figure 3.3. Les symboles de redondance bruités $R_k^{(2)}, R_k^{(3)}$ sont envoyés vers les décodeurs DEC1 et DEC2 respectivement lorsque la redondance est produit par les codeurs RSC1 et RSC2 respectivement. Le décodeur DEC1 reçoit des symboles bruités $R_k^{(1)}$ et $R_k^{(2)}$ issus du démodulateur et produit une décision relative à chaque symbole d_k . Un entrelaceur (Π), place entre les deux décodeurs élémentaires permet d'éclater les paquets d'erreurs produit par le décodeur DEC1. Les décodeurs DEC1 et DEC2 travaillant à partir de décisions pondérées ou souples, le décodeur DEC1 associe à chaque symbole décodé d_k une mesure de fiabilité sous forme du logarithme de son rapport de vraisemblance $\text{LLR}_1(d_k)$, et DEC2 associe au même symbole d_k une autre mesure de fiabilité $\text{LLR}_2(d_k)$. L'information extrinsèque $\Psi_1(d_k)$ extraire du décodeur DEC1 est réinjectée à l'itération suivante dans le décodeur DEC2 afin de bénéficier de la diversité de codage, et l'information extrinsèque $\Psi_2(d_k)$ extraire du décodeur DEC2 passe par le désentrelaceur puis elle est réinjectée dans le décodeur DEC1. Le processus pouvant se répéter plusieurs fois (ce qui justifie l'appellation décodage itératif). Les deux décodeurs DEC1 et DEC2 doivent converger vers la même solution.

Après un nombre précisé des itérations, une décision ferme du décodeur DEC2 (ou DEC1) donne la séquence estimée $\hat{d}$ de la séquence originale d en basant sur la relation (3.11).

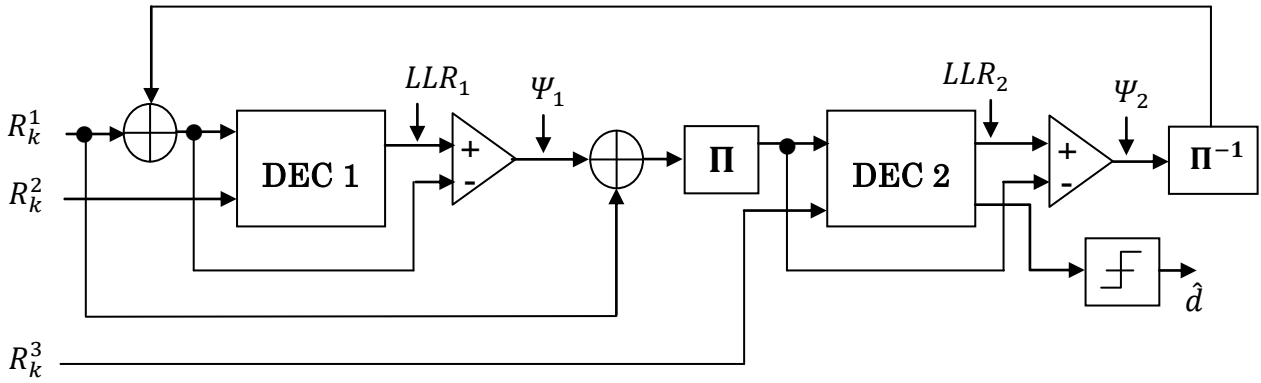


Figure 3.3 Schéma bloc d'un décodeur itératif.

3.5 Conclusion

Dans ce chapitre on a défini la métrique dite de corrélation, dans le cas d'un bruit impulsif *SaS* pour $\alpha = 1$, en basant sur la maximisation de la probabilité de transition du canal au sens du maximum de vraisemblance. Ce principe de vraisemblance coïncide avec le MAP lorsque les bits 0 et 1 sont équiprobables.

L'algorithme de MAP a été exposé avec subtilité, où on a défini la métrique d'état en avant FSM, la métrique d'état en arrière BSM et la métrique de branche BM pour calculer le rapport de vraisemblance logarithmique LLR. Ce rapport est une mesure très opportune de la fiabilité de décision. Pour n'être pas épuisant, on a essayé de donner l'essentiel dans le développement des formules qui apportent le LLR.

Enfin, on a présenté le principe du décodage itératif d'un CTC qui consiste à laisser les deux décodeurs constitutifs travaillent conjointement, c'est à dire que le décodeur 1 profite de la deuxième entrée du décodeur 2 (R_k^3) et que le décodeur 2 profite de la deuxième entrée du décodeur 1 (R_k^2), en utilisant ce qu'on appelle l'information extrinsèque.

Comme on a, le MAP dépend étroitement du paramètre de dispersion γ pour le bruit de **Cauchy SaS**, alors au niveau du récepteur, il faut trouver un moyen pour estimer ce paramètre à partir du signal reçu. Dans le prochain chapitre nous allons présenter une méthode originale et efficace qui fait parfaitement l'affaire. Ce même chapitre est consacré à la mise en œuvre d'une simulation afin d'évaluer les performances des turbo codes pour un bruit impulsif de **Cauchy SaS**.

Chapitre 4

Simulation, résultats et interprétations

4.1 Introduction

La simulation est un outil utilisé par le chercheur, l'ingénieur, le militaire etc... pour étudier les réponses d'un système sans réaliser l'expérience sur le système réel.

Lorsque l'outil de simulation utilise un ordinateur on parle de simulation numérique. Les chercheurs, les ingénieurs, les militaires et bien d'autres professionnels se posent souvent la question : quel est le résultat que j'obtiens si j'exerce telle action sur un système ? Le moyen le plus simple serait de tenter l'expérience, c'est-à-dire d'exercer l'action souhaitée sur le système en cause pour pouvoir observer ou mesurer le résultat. Dans de nombreux cas l'expérience est irréalisable, trop chère ou contraire à l'éthique. On a alors recours à la simulation : rechercher un modèle qui réagit d'une manière semblable à celui que l'on veut étudier et qui permettra de déduire les résultats.

On appelle modèle un système, analogique ou numérique, dont le comportement vis-à-vis d'un phénomène est similaire à celui du système à étudier. Les sorties sont les éléments que l'on veut étudier. Les entrées, paramètres et contraintes sont les éléments dont la variation influe sur le comportement du modèle ; on appelle entrée ceux qui sont commandés par l'expérimentateur, paramètres ceux que l'opérateur choisit de fixer et contraintes ceux qui dépendent d'éléments extérieurs. On appelle simulation l'ensemble constitué par un modèle, les ordres d'entrée, les paramètres et contraintes, et les résultats obtenus. Les équations sont des simulations numériques. Aujourd'hui ce terme s'applique essentiellement aux modèles et simulations réalisés sur ordinateur.

Dans le cas d'un SCN l'évaluation des performances se fait, au niveau de la simulation, en calculant le taux d'erreur binaire BER pour un bruit donné.

4.2 Modèle de simulation

Le modèle utilisé lors de notre simulation est représenté par la figure 4.1. Il comporte

- Une source d'information binaire
- Un turbo codeur convolutif (CTC) parallèle
- Un modulateur BPSK
- Un canal de **Cauchy** supposé idéal (sans distorsions)
- Une source de bruit impulsif *SaS* (avec $\alpha = 1$)
- Un turbo décodeur (ou décodeur itératif) MAP.
- Un comparateur pour calculer le nombre de bits erronés après le décodage afin d'estimer la probabilité d'erreurs binaire BER.

Le décodage itératif MAP est détaillé dans le chapitre précédent.

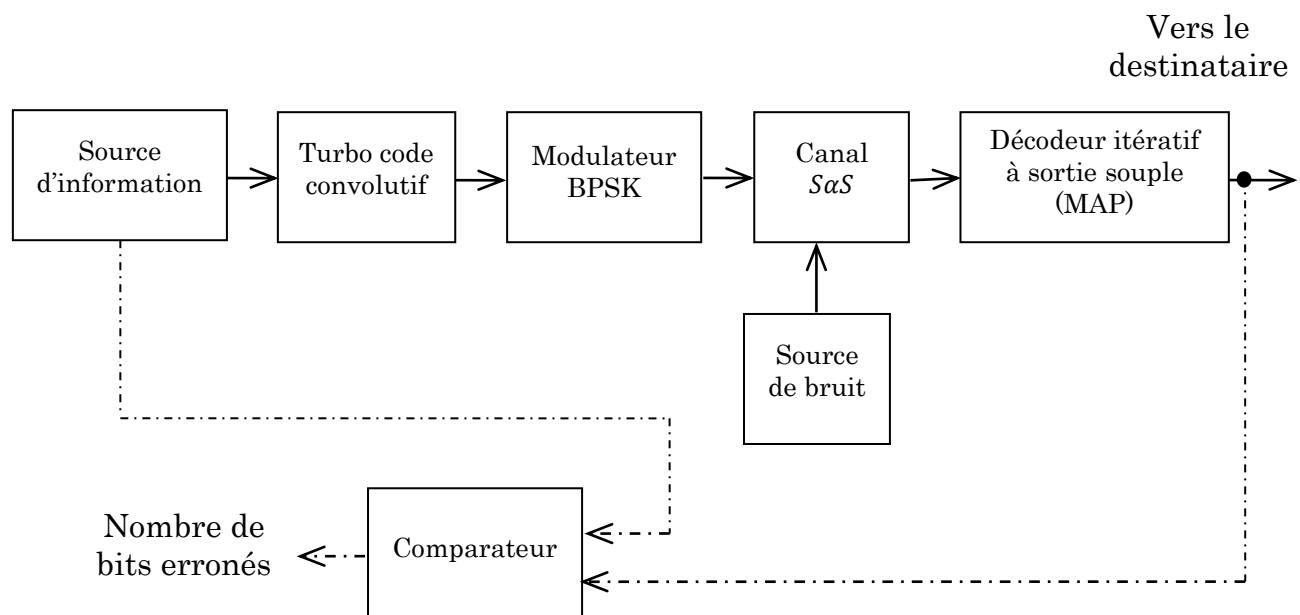


Figure 4.1 Modèle de simulation.

4.2.1 Source d'information

Pour que l'évaluation soit effectuée dans des conditions proches de ce qui est rencontré en exploitation, on doit choisir une séquence d'information qui simule le mieux possible le trafic réel tout en permettant une mesure simple.

La méthode universellement utilisée dans toutes les simulations, consiste à utiliser une séquence dite pseudo-aléatoire, séquence périodique dont les propriétés

statistiques sont «voisines» de celles d'un trafic réel «aléatoire» qui est généralement le résultat d'un brassage. Le générateur d'une telle séquence est constitué d'un registre à décalage comportant m étages, et des additionneurs modulo-2 [55]. Le choix de m détermine la période de la séquence, qui est égale à $2^m - 1$.

Les connexions aux additionneurs modulo-2 sont déterminées par des polynômes primitifs de degré m ayant la forme [55]

$$h(x) = \sum_{j=0}^m h_j x^j \quad \text{avec} \quad \begin{cases} h_0 = h_m = 1 \\ h_j = 0 \text{ ou } 1 \end{cases} \text{ pour } j = 1, \dots, m-1.$$

Un polynôme primitif de degré m est un polynôme irréductible qui divise $x^{2^m-1} + 1$ et ne divise pas $x^n + 1$ pour $n < 2^m - 1$.

Chaque coefficient h_j égal à 1 correspond à une connexion.

La séquence générée comporte 2^{m-1} éléments égaux à 1 et $2^{m-1} - 1$ à égaux à 0 ce qui donne une répartition presque uniforme des 1 et des 0.

Les principales propriétés de ces séquences sont bien détaillées dans [55] [56].

• Choix de la longueur de la séquence utilisée

Le choix de la longueur de la séquence à utiliser dans la simulation, c'est-à-dire la longueur du registre qui l'engendre, est basé sur le fait qu'avec de faibles longueurs de registre (3 ou 4) on risque d'obtenir des résultats différents de la réalité, car une séquence courte ne représente pas un nombre statistiquement suffisant de configurations, donc il est recommandé d'utiliser des séquences relativement longues, dont leurs nombres des registres soient supérieurs à 15 [57]. A cause du facteur de temps et de la mémoire limitée de l'ordinateur utilisé, nous avons choisi $m = 16$ qui nous donne une séquence de longueur $2^m - 1 = 65536$.

4.2.2 Turbo code convolutif CTC parallèle

Un turbo code convolutif parallèle est constitué de deux codes convolutifs RSC via un entrelaceur entre les deux.

- **Codes convolutifs utilisés**

On a utilisé dans notre simulation deux codes RSC très employés par **C. Berrou** [59] illustrés dans les figures 4.2. Le RSC de la figure 4.2.a contient quatre états et de rendement $1/2$. Le turbo code parallèle associé à ce RSC est alors à quatre états et de rendement $1/3$ (Figure 4.3.a). Le RSC de la figure 4.2.b contient huit états et de rendement $1/2$. Le turbo code parallèle associé à ce RSC est alors à huit états et de rendement $1/3$ (Figure 4.3.b).

- **Entrelaceurs utilisés**

On remémore que le but du procédé d'entrelacement est de permettre l'utilisation des codes destinés à corriger des erreurs indépendantes dans le cas des canaux à paquets d'erreurs.

L'idée maîtresse de ce procédé est de transmettre les symboles d'un mot code entrelacés à des symboles d'autres mots code, de sorte que deux symboles successifs du même mot se trouvent, l'un par rapport à l'autre, à une distance plus grande que la longueur du paquet d'erreurs. De cette sorte, un paquet d'erreurs ne peut affecter plusieurs symboles du mot code et par conséquent les symboles du mot code seront affectés par des paquets différents (donc indépendants) et donc leur effet sera celui produit par les erreurs indépendantes.

Le type le plus performant à utiliser dans les turbo codes est celui d'un entrelaceur S –aléatoire (voir chapitre 1). Dans notre simulation on a utilisé ce type d'entrelaceurs avec une taille moyenne 128×128 pour éviter les grands temps d'exécution. Pour cette taille, le meilleur entrelaceur S –aléatoire choisit correspond à $S = 15$ (voir théorème 1.1 du paragraphe 1.4.2.6 dans le chapitre 1).

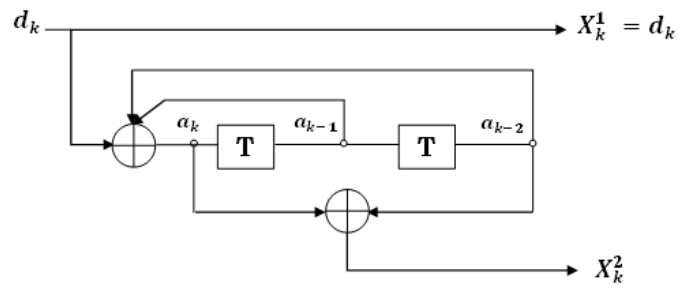


Figure 4.2.a Code convolutif récursif systématique (5,7).

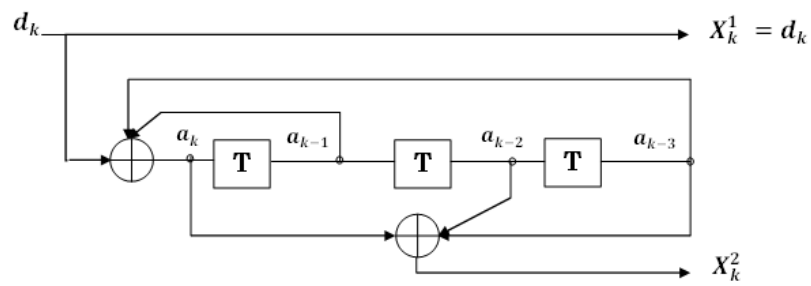


Figure 4.2.b Code convolutif récursif systématique (13,15).

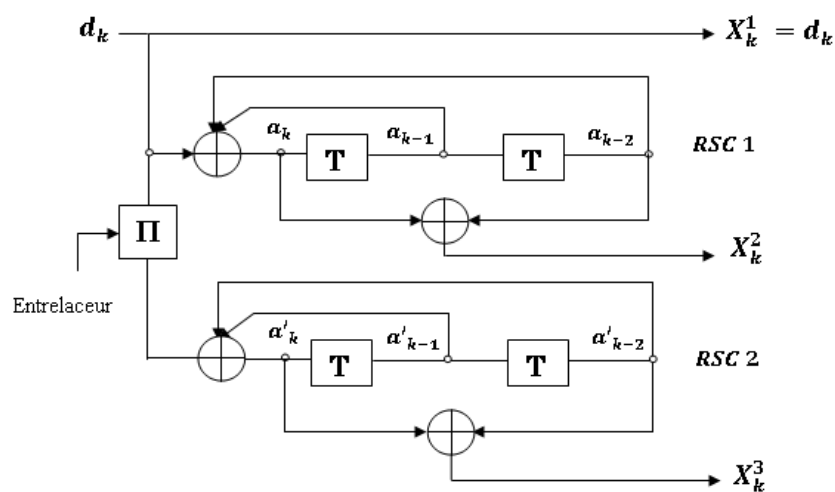


Figure 4.3.a Turbo code convolutif parallèle (5,7).

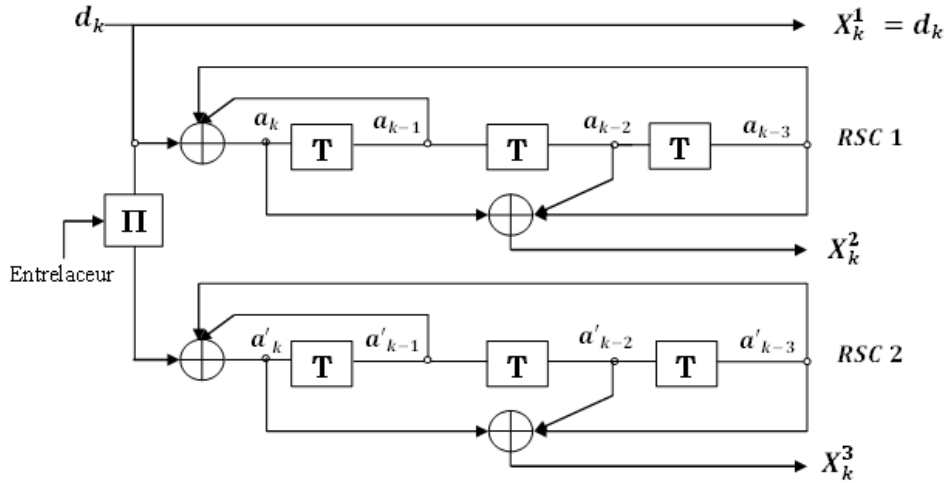


Figure 4.3.b Turbo code convolutif parallèle (13,15).

4.2.3 Modulation utilisée

Pour la modulation utilisée dans la simulation, notre choix reste limité à la modulation BPSK pour tous les turbo codes employés. La modulation des bits de sortie du turbo code est alors (voir chapitre 1)

$$Y = \begin{cases} +\sqrt{E_b} & \text{pour un bit de sortie } X = 0 \\ -\sqrt{E_b} & \text{pour un bit de sortie } X = 1 \end{cases} \quad (4.1)$$

Pour simplifier les choses, on a pris $\sqrt{E_b} = 1$. La modulation devient tout simplement

$$Y = \begin{cases} +1 & \text{pour un bit } X = 0 \\ -1 & \text{pour un bit } X = 1 \end{cases} \quad \text{ou} \quad Y = 1 - 2X \quad (4.2)$$

4.2.4 Source de bruit

Le bruit envisagé est impulsif additif et à distribution $S\alpha S$ où $\alpha = 1$, ce qui correspond au bruit de **Cauchy**.

4.2.4.1 La distribution de Cauchy

La PDF d'une variable aléatoire X suit la loi $S\alpha S$ pour $\alpha = 1$, avec un paramètre de localisation $\delta = 0$ et un paramètre de dispersion γ est définie comme

$$f(x) = \frac{2\gamma}{\pi(x^2 + 4\gamma^2)} \quad (4.3)$$

La figure 4.4 illustre la courbe de cette PDF pour différentes valeurs du paramètre de dispersion γ . On remarque que plus γ est petit plus la queue de la PDF est lourde, par conséquent, plus d'impulsivité au niveau des échantillons de la variable X .

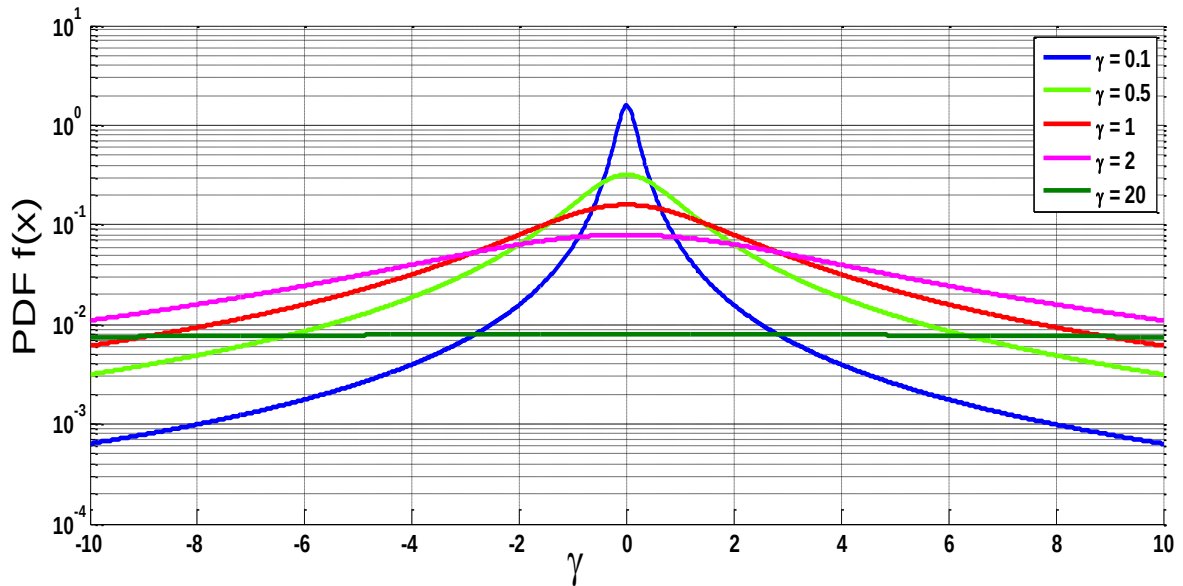


Figure 4.4 La PDF de Cauchy pour différentes valeurs de γ .

D'après la propriété 2.3 dans le chapitre 2 la distribution de **Cauchy** ne possède que des moments d'ordre $p < 1$, c'est ce qu'on appelle les statistiques d'ordre inférieur LOS. Soit à calculer le moment m_p d'ordre $p < 1$.

On a

$$m_p = E(|X|^p) = \int_{-\infty}^{+\infty} |x|^p f(x) dx$$

$$m_p = \frac{2}{\pi} \cdot \int_0^{+\infty} \frac{2\gamma x^p}{(x^2 + 4\gamma^2)} dx$$

$$m_p = \frac{2}{\pi} \cdot \int_0^{+\infty} \frac{1}{2\gamma} \cdot \frac{x^p}{\left(\frac{x}{2\gamma}\right)^2 + 1} dx$$

Avec le changement de variable $z = x/2\gamma$ et avec $p = 1/n$ (dans ce cas des statistiques d'ordre inférieur fractionnel FLOS) on trouve

$$m_{\frac{1}{n}} = \frac{2}{\pi} \cdot \frac{1}{2\gamma} \cdot (2\gamma)^{\frac{1}{n}+1} \int_0^{+\infty} \frac{z^{\frac{1}{n}}}{z^2 + 1} dz$$

Si on pose encore $z^{\frac{1}{n}} = x$ on obtient

$$m_{\frac{1}{n}} = \left(\frac{2^{\frac{n+1}{n}}}{\pi} \cdot n \cdot \int_0^{+\infty} \frac{x^n}{x^{2n} + 1} dx \right) \cdot \gamma^{\frac{1}{n}} = a_n \gamma^{\frac{1}{n}} \quad (4.4)$$

avec

$$a_n = \left(\frac{2^{\frac{n+1}{n}}}{\pi} \cdot n \cdot \int_0^{+\infty} \frac{x^n}{x^{2n} + 1} dx \right) \quad (4.5)$$

4.2.4.2 Simulation d'un bruit impulsif $S\alpha S$ de Cauchy

Soient F la fonction de répartition d'une variable aléatoire X et F^{-1} son inverse. Il est bien connu que si F^{-1} peut être directement évaluée, un grand nombre de réalisations de X peuvent être obtenues par $x_i = F^{-1}(u_i)$, où u_i ($i = 1, 2, \dots, n$) sont de nombres aléatoires distribués uniformément dans $[0, 1]$. Si F^{-1} a une expression analytique, une telle méthode peut être appliquée de manière efficace.

A partir de l'équation (4.3) on peut évaluer directement la fonction de répartition F et on trouve

$$F(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{2\gamma}\right) \quad (4.6)$$

et facilement on ôte F^{-1} comme suit

$$x = F^{-1}(u) = 2\gamma \cdot \tan(\pi(u - 1/2)) \quad (4.7)$$

Par conséquent, si u est une variable aléatoire distribuée uniformément dans $[0, 1]$, alors X suit la loi de **Cauchy**.

La figure 4.4 montre le tracé de 16384 échantillons d'un bruit impulsif $S\alpha S$ de **Cauchy** générés à l'aide de l'équation (4.7) pour deux valeurs de γ . Un simple coup d'œil sur cette figure montre que plus on augmente le paramètre γ , plus la dispersion des échantillons augmente aussi, et inversement.

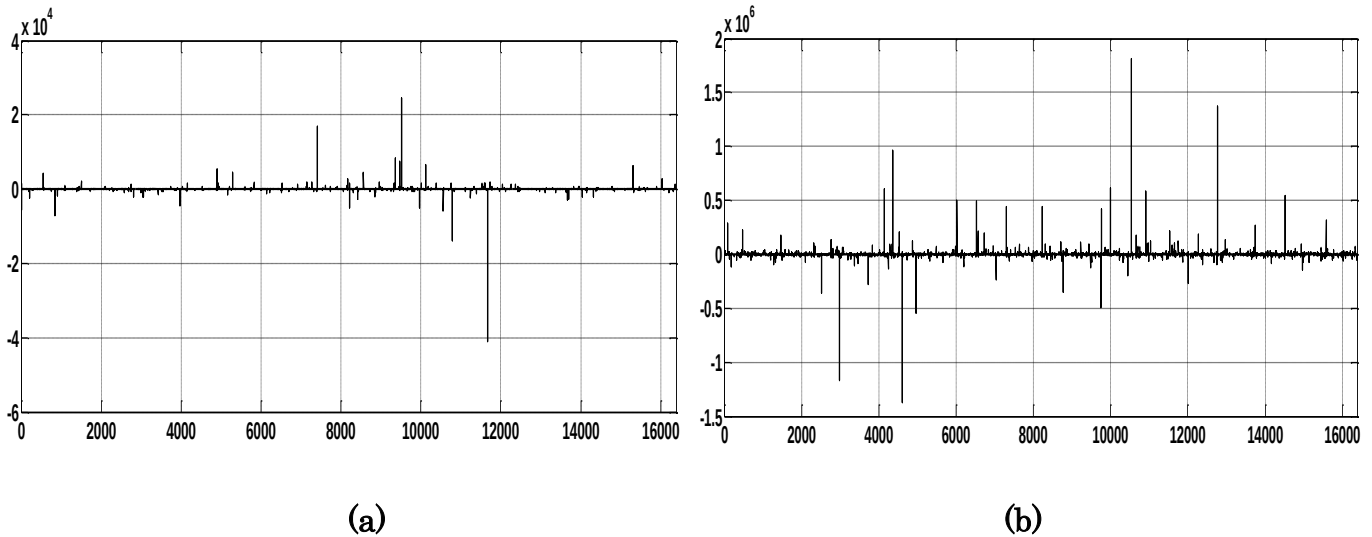


Figure 4.5 Le tracé de 16384 échantillons d'un bruit impulsif $S\alpha S$ de Cauchy:
(a) $\gamma = 0.1$ (b) $\gamma = 10$.

4.2.5 Décodeur itératif MAP

On a vu dans le chapitre 3 que l'algorithme MAP nécessite la connaissance du paramètre de dispersion γ du bruit impulsif de **Cauchy** affecte le signal émis, mais au niveau de la réception nous n'avons que le signal reçu, qui est une superposition du

signal émis et le bruit ! Pour cette raison nous sommes obligés de faire une estimation pour ce paramètre.

4.2.5.1 Nouvelle approche pour estimer le paramètre γ

Au niveau du récepteur le signal reçu est

$$R_k = Y_k + W_k, \quad k = 1, 2, \dots, 3N \quad (4.8)$$

où $3N$ est le nombre des symboles reçus, $Y_k \in \{-1, +1\}$ sont les symboles émis et W_k sont des variables aléatoires i.i.d de **Cauchy**, de paramètre dispersion γ , qui représentent le bruit impulsif. Tant que le décodeur itératif traite les symboles systématiques reçus par blocs de tailles égales à celle de l'entrelaceur, on peut prendre alors $3N$ égale à 3 fois la taille de l'entrelaceur, et dans notre cas $3N$ devient égal à 3×16384 , c'est-à-dire $N = 16384$.

D'après le paragraphe 3.2.1 dans le chapitre 3, les densités de probabilité de transition de notre canal peuvent s'écrire sous la forme

$$P(R_k/Y_k \text{ émis}) = \frac{2\gamma}{\pi((R_k - Y_k)^2 + 4\gamma^2)} \quad (4.9)$$

On sait bien qu'au niveau du récepteur on doit trouver à partir de l'observation R_k une estimation du symbole Y_k de sorte que la probabilité moyenne de l'erreur de décision soit minimisée, donc la probabilité conditionnelle de la relation (4.9) soit maximisée. Dans le même paragraphe 3.2.1, on a montré, à l'aide du principe de ML sens d'**Euclid**, que la probabilité $P(R_k/Y_k \text{ émis})$ est maximale si et seulement si R_k est de même signe que Y_k , autrement dit, si le produit $R_k \cdot Y_k$ soit positif. D'où vient l'idée de calculer la probabilité $\rho = P(R_k \cdot Y_k > 0)$. On a

$$\rho = P(R_k \cdot Y_k > 0)$$

$$\rho = P(1 + Y_k \cdot W_k > 0) = P(Y_k \cdot W_k > -1) \quad \text{car } Y_k \in \{-1, +1\} \text{ donc } Y_k^2 = 1$$

$$\rho = P(R_k \cdot Y_k > 0) = P(1 + Y_k \cdot W_k > 0) = P(Y_k \cdot W_k > -1)$$

$$\rho = P(Y_k \cdot W_k > -1/Y_k = -1) \cdot P(Y_k = -1) + P(Y_k \cdot W_k > -1/Y_k = +1) \cdot P(Y_k = +1)$$

Avec $P(Y_k = -1) = P(Y_k = +1) = 1/2$, on trouve

$$\rho = \frac{1}{2} \cdot (P(W_k < +1) + P(W_k > -1)) = \frac{1}{2} \cdot (1 + P(-1 < W_k < +1))$$

$$\rho = \frac{1}{2} \cdot \left(1 + \int_{-1}^1 f(w) dw\right) = \frac{1}{2} \cdot \left(1 + \int_{-1}^1 \frac{2\gamma}{\pi(w^2 + 4\gamma^2)} dw\right)$$

enfin

$$\rho = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{1}{2\gamma}\right) \quad (4.10)$$

On remarque que cette probabilité est toujours supérieure à 1/2. La Figure 4.6 montre la variation de ρ en fonction de γ .

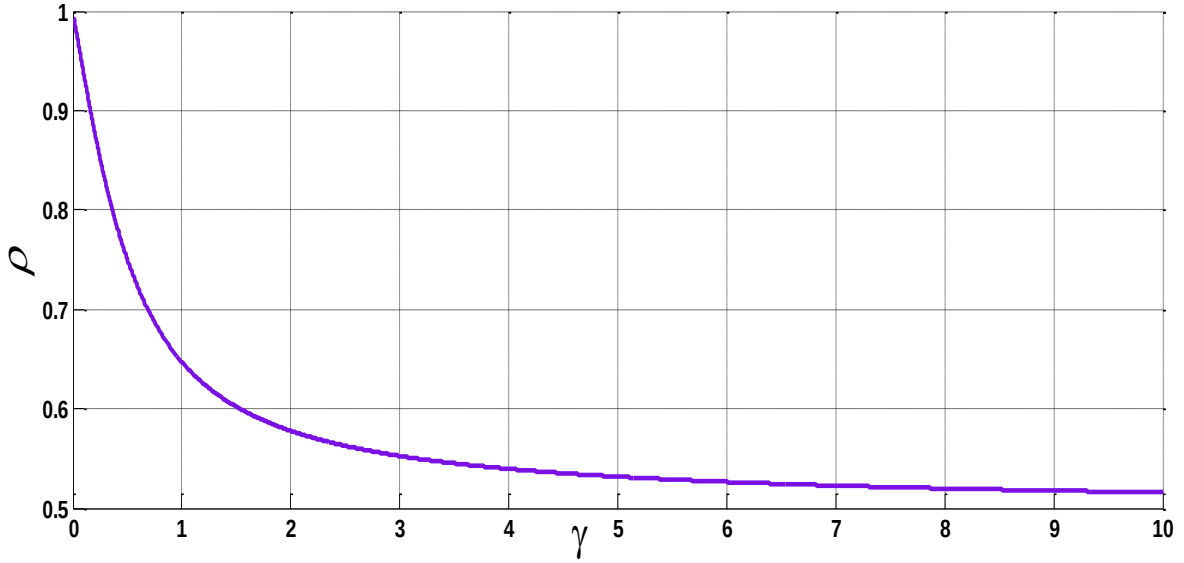


Figure 4.6 La probabilité ρ en fonction du paramètre γ .

La figure 4.6 nous permet de conclure que la probabilité pour que R_k et Y_k soient de même signe varie inversement avec la dispersion bruit. Néanmoins, cette probabilité reste strictement supérieur à 1/2 quelque soit la dispersion du bruit.

Après le calcul de ρ on peut écrire

$$Y_k = \begin{cases} \text{sign}(R_k) & \text{avec probabilité } \rho \\ -\text{sign}(R_k) & \text{avec probabilité } 1 - \rho \end{cases} \quad (4.11)$$

du coup, pour tout $p < 1$

$$|W_k|^p = |R_k - Y_k|^p = \begin{cases} |R_k - \text{sign}(R_k)|^p & \text{avec probabilité } \rho \\ |R_k + \text{sign}(R_k)|^p & \text{avec probabilité } 1 - \rho \end{cases} \quad (4.12)$$

et on peut approximer $|W_k|^p$ par sa valeur moyenne comme suit

$$|W_k|^p \cong \rho \cdot |R_k - \text{sign}(R_k)|^p + (1 - \rho) \cdot |R_k + \text{sign}(R_k)|^p \quad (4.13)$$

Donc, le moment m_p d'ordre $p < 1$ est approximé par

$$m_p = E(|W_k|^p) \cong \rho \cdot E(|R_k - \text{sign}(R_k)|^p) + (1 - \rho) \cdot E(|R_k + \text{sign}(R_k)|^p) \quad (4.14)$$

Pour $p = 1/n$ avec $n \geq 2$ on écrit, le moment FLOS $m_{1/n}$, comme suit

$$m_{1/n} = E(|W_k|^{1/n}) \cong \rho \cdot E\left(|R_k - \text{sign}(R_k)|^{\frac{1}{n}}\right) + (1 - \rho) \cdot E\left(|R_k + \text{sign}(R_k)|^{\frac{1}{n}}\right) \quad (4.15)$$

ou sous la forme simplifiée suivante

$$m_{1/n} \cong \rho \cdot b_n + (1 - \rho) \cdot c_n = \rho \cdot (b_n - c_n) + c_n \quad (4.16)$$

$$\text{où} \quad b_n = E\left(|R_k - \text{sign}(R_k)|^{\frac{1}{n}}\right) \quad \text{et} \quad c_n = E\left(|R_k + \text{sign}(R_k)|^{\frac{1}{n}}\right) \quad (4.17)$$

En remplaçant ρ par son expression (4.10) on trouve

$$m_{1/n} \cong \frac{b_n - c_n}{\pi} \cdot \arctan\left(\frac{1}{2\gamma}\right) + \frac{b_n + c_n}{2} \quad (4.18)$$

L'indentification de (4.18) et (4.4) nous permet d'écrire l'approximation

$$\varphi(\gamma) = \frac{b_n - c_n}{\pi} \cdot \arctan\left(\frac{1}{2\gamma}\right) + \frac{b_n + c_n}{2} - a_n \gamma^{\frac{1}{n}} \cong 0 \quad (4.19)$$

avec,

$$\begin{cases} a_n = \frac{2^{\frac{n+1}{n}}}{\pi} \cdot n \cdot \int_0^{+\infty} \frac{x^n}{x^{2n} + 1} dx \\ b_n = E\left(|R_k - \text{sign}(R_k)|^{\frac{1}{n}}\right) \\ c_n = E\left(|R_k + \text{sign}(R_k)|^{\frac{1}{n}}\right) \end{cases} \quad (4.20)$$

Il faut noter que l'espérance $E(\cdot)$ dans le système (4.20), est calculée comme étant la moyenne empirique (arithmétique).

Mathématiquement on a $\lim_{n \rightarrow +\infty} (b_n - c_n) = 0^-$, et on peut montrer par simulation que si $n \geq 10$ la quantité $(b_n - c_n)$ est toujours négative et relativement petite. Or la fonction $\frac{1}{\pi} \cdot \arctan\left(\frac{1}{2\gamma}\right)$ est bornée entre $-1/2$ et $1/2$, alors la dérivée de la fonction $\varphi(\gamma)$ est $\varphi'(\gamma) \cong \frac{-a_n}{n} \gamma^{\frac{1}{n}-1}$, et $\varphi(\gamma)$ est donc strictement décroissante pour tout $\gamma > 0$. En tenant compte que $\lim_{\gamma \rightarrow 0^+} \varphi(\gamma) = b_n > 0$ et $\lim_{\gamma \rightarrow +\infty} \varphi(\gamma) = -\infty$, alors le théorème des valeurs moyennes implique l'existence d'une solution unique notée $\hat{\gamma}$ de l'équation (4.19).

Notre nouvelle approche s'appuyait alors sur deux points, à savoir

- Le calcul des moments FLOS b_n et c_n de l'équation (4.20) à partir des observations du signal reçu R_k ;
- La résolution numérique de l'équation (4.19) sachant qu'elle possède une solution unique $\hat{\gamma}$.

Pour toiser la précision (donc la qualité) de notre nouvelle approche on utilise la racine carrée de l'erreur quadratique moyenne $RMSE$ donnée par [60]

$$RMSE = \sqrt{\frac{1}{K} \sum_{k=1}^K (\gamma - \hat{\gamma}_k)^2} \quad (4.21)$$

ou aussi la $RMSE$ relative

$$RMSE_{rel} = \frac{RMSE}{\gamma} = \sqrt{\frac{1}{K} \sum_{k=1}^K (1 - \hat{\gamma}_k / \gamma)^2} \quad (4.22)$$

où,

K est la taille totale de la simulation de Monte Carlo ($K = 10\,000$ dans notre simulation) ;

γ est la valeur exacte du paramètre de dispersion ;

$\hat{\gamma}_k$ est la valeur estimée au $k^{ème}$ exécution.

Pour tracer la $RMSE_{rel}$ en fonction de γ , on a adopté le modèle simulation illustré dans la figure 4.7.

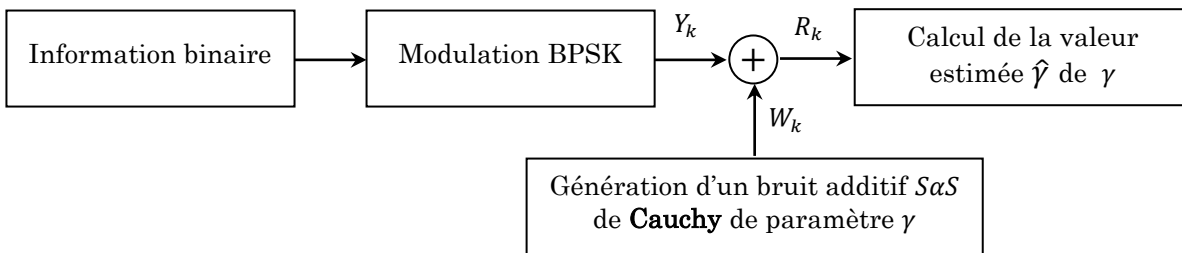


Figure 4.7 Modèle de simulation adopté pour estimer le paramètre γ .

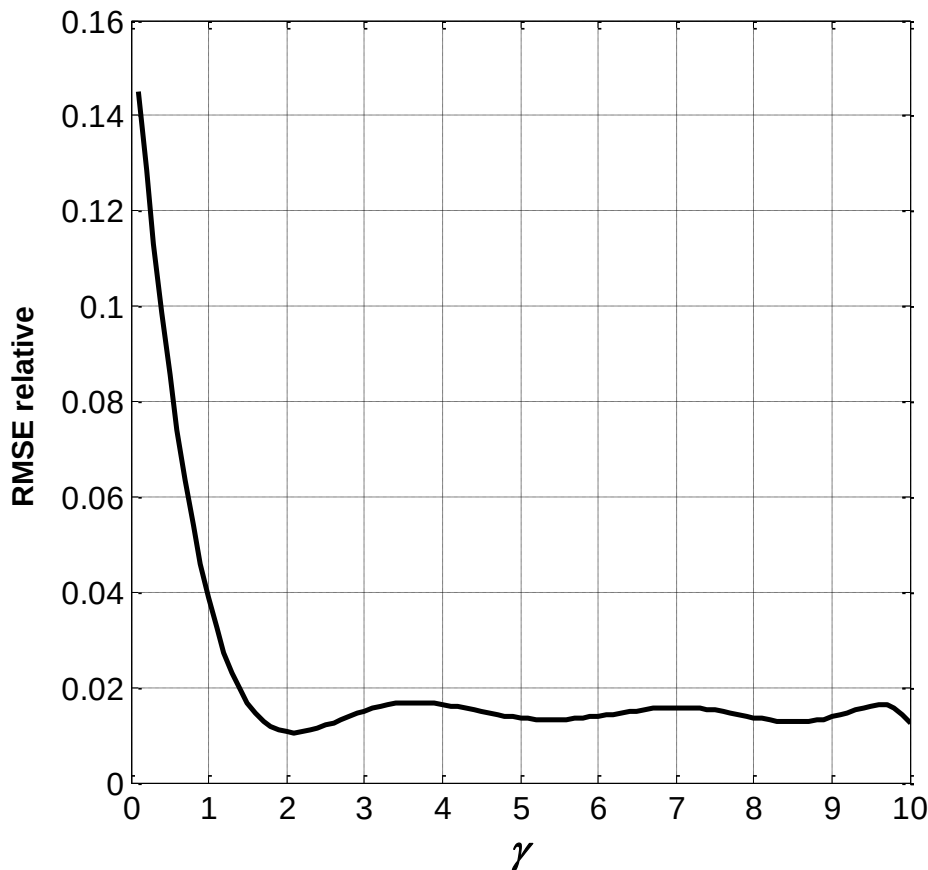


Figure 4. 8 La RMSE relative en fonction de la valeur exacte de γ .

La figure 4.8 montre la variation de la *RMSE* relative en fonction de la valeur exacte de γ . L'analyse de cette figure montre que si $\gamma \leq 2$, la *RMSE* relative varie entre 2% et 14% , et si $\gamma > 2$, la *RMSE* relative reste toujours inférieure à 2%.

4.2.5.2 Les étapes de l'implémentation de l'algorithme MAP

Voici les principales étapes de l'algorithme de décodage

- Étape 1:** Déclarer le signal reçu $R_k = (R_k^s, R_k^p)$ et estimer la valeur du paramètre de dispersion, en utilisant les relations (4.19) et (4.20), avec $R = R_k^s$ et $k = 1, 2, \dots, N = 16384$.
- Étape 2:** A partir de l'instant $k = 1$ pour $i = 0, 1$ et $m = 0, \dots, 2^M - 1$, où M est la mémoire du CTC, calculer la métrique de branche $\delta_k^i(m)$ suivant l'expression (3.36) pour tous les symboles reçus et les sauvegarder dans un tableau.
- Étape 3:** Initialiser la variable $\alpha_k(m)$ telle que $\alpha_1(0) = 1$ et $\alpha_1(m) = 0$, pour tout $m \neq 0$, et calculer la métrique d'état en avant $\alpha_k(m) = 0$ à partir de l'instant $k = 2$ et pour tout les états ($m = 0, \dots, 2^M - 1$) en utilisant l'équation (3.30).
- Étape 4:** Initialiser la variable $\beta_k(m)$ telle que $\beta_N(0) = 1$ et $\beta_N(m) = 0$, pour tout $m \neq 0$, et calculer la métrique d'état en arrière $\beta_k(m) = 0$ à partir de l'instant $k = N - 1$ et pour tout les états ($m = 0, \dots, 2^M - 1$) en utilisant l'équation (3.31).

4.2.5.3 Schéma fonctionnel d'un décodeur itératif MAP

Le décodeur itératif de la figure 3.3 dans le chapitre 3 converge lentement lorsque le bruit est trop dispersé. Pour remédier à cela, on multiplie l'information extrinsèque calculée de chaque décodeur constitutif par un facteur de feedback β avant de la réinjectée dans l'autre décodeur constitutif (voir la figure 4.9) [20]. L'utilisation de ce facteur diminue le nombre des itérations presque de la moitié. Notre simulation montre que ce facteur est dans l'ordre de 0.1.

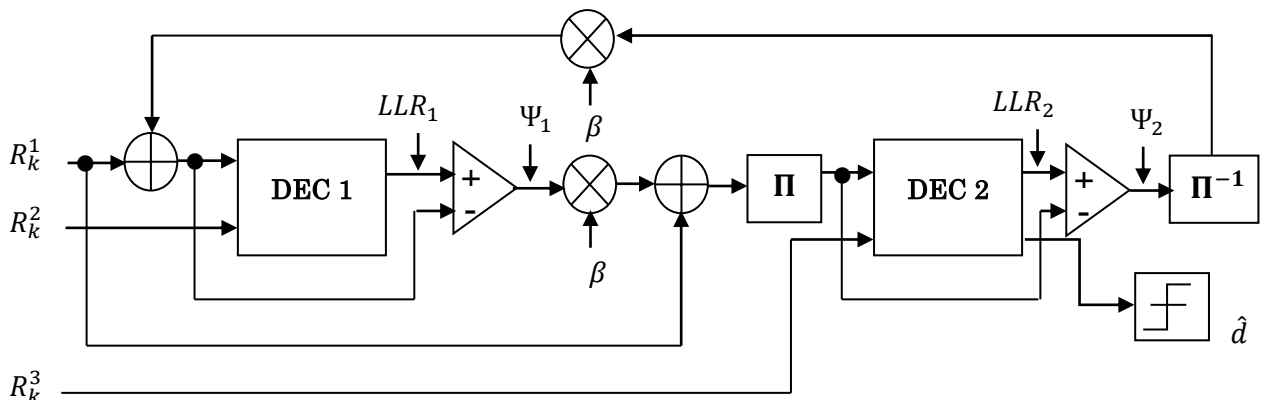


Figure 4.9 Schéma fonctionnel d'un décodeur itératif MAP.

4.2.6 Calcul du taux d'erreurs binaire BER

La séquence d'information pseudo-aléatoire générée par la source est de longueur $2^{16} = 65536$ bits, qui est un multiple de la taille de l'entrelaceur 128×128 .

Le décodeur itératif décode la séquence reçue par paquets. Chaque paquet est de taille égale à celle de l'entrelaceur. Donc, dans notre cas on a $65536/(128 \times 128) = 4$ paquets.

Comme la variance du bruit envisagé est infinie, la définition du SNR n'est pas valide dans ce cas et nous avons utilisé à la place le rapport $1/\gamma$ [61]. Pour un rapport $1/\gamma$ et un nombre d'itérations donnés, le décodeur itératif compte le nombre des bits erronés par le truchement d'une comparaison entre la vraie séquence d et la séquence estimée $\hat{d}$ délivrée par l'un des décodeurs constitutifs (le deuxième décodeur par exemple). Le taux d'erreurs binaire *BER* est le rapport entre ce nombre des bits erronés et la longueur totale de la séquence.

4.3 Résultats et Commentaires

Le taux d'erreurs binaire BER mesure la performance d'un système de communication numérique. Notre simulation a permis de calculer ce paramètre pour l'ensemble des systèmes étudiés et pour différentes valeurs du rapport $1/\gamma$.

Les figures 4.10, 4.11 montrent les BER pour deux systèmes à bruit impulsif *SaS*, l'un est sans codage et l'autre avec turbo codage à MAP, en fonction du rapport du rapport $1/\gamma$. Pour le système BPSK sans codage son BER est donné par la relation (1.17) (voir chapitre 1) et représenté dans toutes les figures par la courbe supérieure.

L'analyse des figures 4.10 et 4.11 permet de faire les constatations et les remarques suivantes

- Un système avec codage est plus performant que celui sans codage.
- Le BER est inversement proportionnelle au rapport $1/\gamma$, ce qui confirme davantage la théorie, puisque pour un rapport $1/\gamma$ grand la séquence émise n'est affectée que peu par le bruit car ce dernier est faiblement dispersé (γ est petit). Donc, le décodeur itératif MAP aura à son entrée une séquence avec un nombre faible de bits erronés, d'où une probabilité faible, et vice versa.

- Le BER décroît rapidement avec l'augmentation du rapport $1/\gamma$ et surtout pour un grand nombre d'itérations, ce qui prouve l'importance de l'information extrinsèque dans la performance du décodeur itératif.

Performance en fonction du nombre d'états

Les figures 4.10 et 4.11 montrent les BER du turbo code à 4 états et à 8 états respectivement. On peut dire, d'après ces deux figures et pour un bruit impulsif $S\alpha S$ de **Cauchy**, qu'un turbo code à 8 états est plus qualifié en termes de BER qu'un turbo code à 4 états de même rendement, surtout pour les forts rapports $1/\gamma$. Ce résultat est prévisible, parce que le treillis d'un codeur à 8 états contient plus des branches que celui d'un codeur à 4 états, donc plus de capacité de correction des erreurs.

Performance en fonction du nombre d'itérations

Les figures 4.10 et 4.11 montrent que le décodeur itératif MAP donne une très bonne performance si on augmente le nombre d'itérations même pour les faibles rapport $1/\gamma$, et ceci revient à l'exploitation de l'information extrinsèque tirée dans chaque itération.

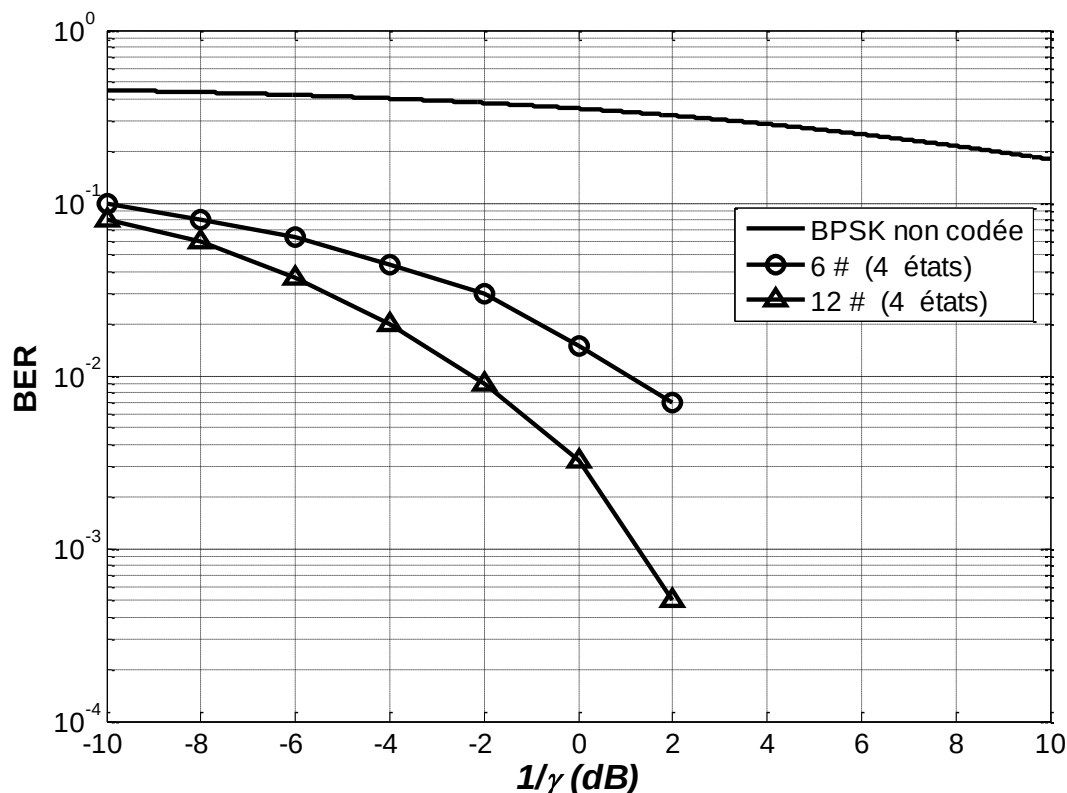


Figure 4.10 Le BER en fonction de $1/\gamma$ d'un CTC parallèle à 4 états pour 6 itérations (6 #) et 12 itérations (12 #).

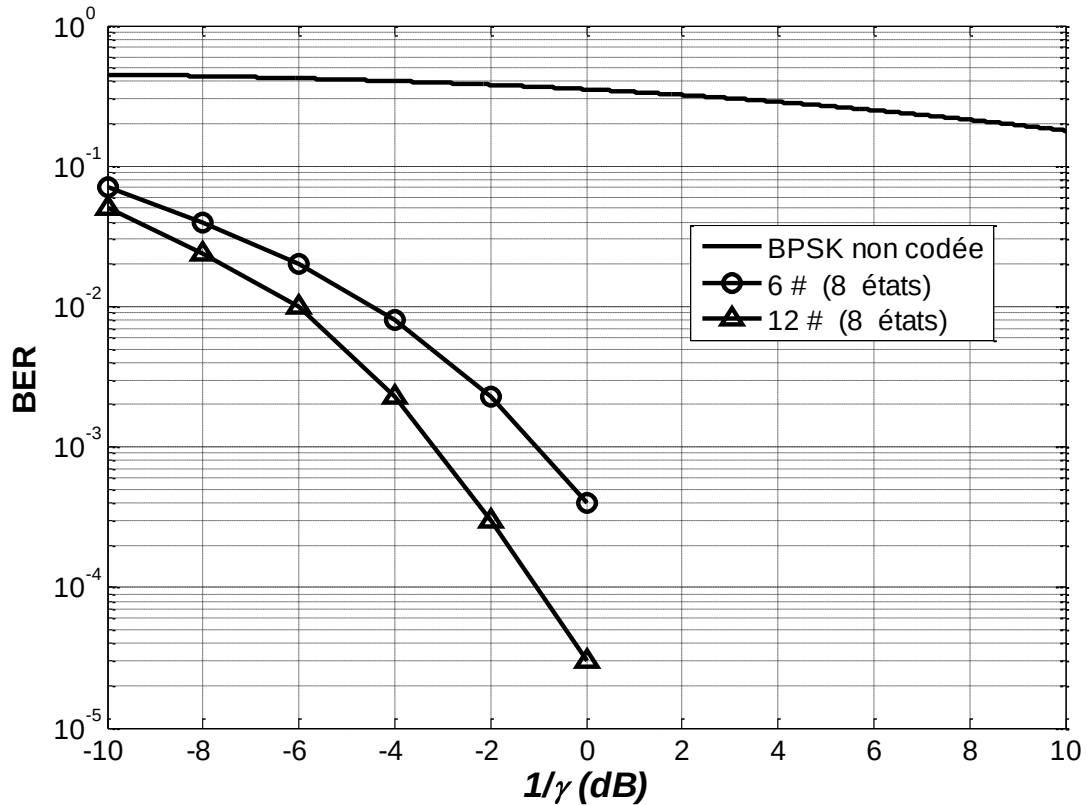


Figure 4.11 Le BER en fonction de $1/\gamma$ d'un CTC parallèle à 8 états pour 6 itérations (6 #) et 12 itérations (12 #).

4.4 Conclusion

On a étudié dans ce chapitre les performances d'un CTC avec l'algorithme MAP pour un bruit impulsif *SaS* de **Cauchy** où l'exposant caractéristique $\alpha = 1$.

La distribution *SaS* de **Cauchy** est bien présentée ici, où on a présenté la méthode de la fonction de répartition pour générer des échantillons i.i.d suivent une telle distribution, et on a vu graphiquement en traçant les échantillons générés que, plus le paramètre γ est proche de zéro plus le bruit est moins dispersé et inversement.

En raison que le MAP nécessite la connaissance des paramètres du bruit, on a présenté dans ce chapitre une méthode originale très efficace pour estimer le paramètre de dispersion du bruit impulsif, à partir du signal reçu uniquement, en basant surtout sur le FLOS. Notre méthode a été évaluée à l'aide de la RMSE relative, et on vu que cette erreur relative ne dépasse pas généralement 2%.

L'implémentation de l'algorithme MAP qui est basée sur quatre étapes a été présenté dans ce chapitre.

Notre simulation a permis de calculer les performances des CTC avec l'algorithme MAP. Les résultats obtenus sont en conformité avec la théorie, et montre que les CTC restent parmi les meilleurs codes existants pour les SCN à bruits impulsifs.

Notre travail s'est limité bien sûr aux turbo codes avec bruit impulsif $S\alpha S$ de **Cauchy**, en utilisant le MAP. Notre étude peut être étendue à des bruits impulsifs $S\alpha S$ où $\alpha \neq 1$. Dans ce cas la PDF du bruit n'a pas une forme explicite, donc l'acrobatie numérique on se sert de la FFT peut résoudre ce problème. Cette problématique fait un bon thème de recherche dans le domaine des turbo codes.

Conclusion générale et perspectives

Selon le cahier de charge exposé, notre travail concernant l'étude du turbo décodage à bruit impulsif α -stable symétrique en utilisant le MAP, a englobé deux parties : une partie théorique et l'autre une simulation.

La partie théorique a pour but de dégager les fondements théoriques des ces codeurs de canaux, à s'avoir : les théorèmes fondamentaux du codage de canal, le principe des turbo codes convolutifs, la notion de l'information extrinsèque et le principe de décodage itératif, la notion de l'algorithme MAP, et de dégager aussi les autres fonctions d'un système de communications numérique, comme la modulation BPSK et la notion du bruit impulsif α -stable symétrique. Les deux premiers chapitres ont été consacrés à cette partie.

Le deuxième partie était la mise en œuvre d'une simulation permettant d'évaluer les performances des turbo décodeurs, et cela en estimant le taux d'erreurs binaire BER en fonction du rapport $1/\gamma$. Le bruit envisagé dans notre travail est le bruit impulsif $S\alpha S$ de **Cauchy** de paramètre de dispersion γ . Le décodage étant réalisé à l'aide d'un décodeur itératif basé sur le MAP. Une nouvelle méthode d'estimation du paramètre γ du bruit impulsif envisagé, à partir du signal reçu uniquement, est incorporée dans l'implémentation de l'algorithme MAP. Nous avons vu que la RMSE relative de notre méthode, dans la plupart des cas, ne dépasse pas 2%. Ce qui confirme la bonne précision de cette approche.

Nos moyens informatiques disponibles ont limité la génération de la séquence d'information pseudo-aléatoire à une longueur de $2^{16} = 65536$ **bits**, et les calculs des intervalles d'incertitudes dans les courbes obtenus (temps d'exécution des programmes est très grand).

Notre simulation a permis de calculer le BER qui est le paramètre d'évaluation des CTC. L'allure des courbes obtenues correspond bien à la théorie, puisque le BER diminue avec l'augmentation du rapport $1/\gamma$, converge rapidement si on augmente le nombre d'itérations, et ceci grâce à l'optimalité de l'algorithme MAP.

Notre travail s'est limité bien sûr aux turbo codes avec bruit impulsif $S\alpha S$ de **Cauchy**, en utilisant le MAP. Notre étude peut être étendue à des bruits impulsifs $S\alpha S$ où $\alpha \neq 1$. Comme on a vu dans ce travail, la PDF du bruit dans le cas où $\alpha \neq 1$ ne possède pas une expression fermée ce qui pose un problème pour l'algorithme MAP. L'utilisation de la FFT permet de surmonter ce problème [11]. Comme perspectives on propose alors l'étude de cette problématique on se sert de la FFT.

Bibliographies

- [01] C. Berrou, A. Glavieux, and P. Thitimajshima, « Near SHANNON limit error-correcting coding and encoding: turbo-codes, » in Proceedings of the IEEE International Conference on Communications, pp. 1064–1070, Geneva, Switzerland, 1993.
- [02] C. P. Shah, C. C. Tsimenidis, B. S. Sharif, and J. A. Neasham, « Low complexity iterative receiver design for shallow water acoustic channels, » EURASIP Journal on Advances in Signal Processing, vol. 2010, Article ID 590458, 13 pages, 2010. [View at Publisher](#) · [View at Google Scholar](#).
- [03] L. Ye and A. Burr, « Separate turbo code and single turbo code adaptive OFDM transmissions, » EURASIP Journal on Wireless Communications and Networking, vol. 2009, Article ID 609386, 7 pages, 2009. [View at Publisher](#) · [View at Google Scholar](#).
- [04] R. J. Barton and M. Daniel, « Performance evaluation of non-Gaussian channel estimation techniques on ultra wide band impulse radio channels, » International Journal of Communication Systems, vol. 20, no. 6, pp. 723–741, 2007. [View at Publisher](#) · [View at Google Scholar](#) · [View at Scopus](#).
- [05] M. Zimmermann and K. Dostert, « Analysis and modeling of impulsive noise in broad-band powerline communications, » IEEE Transactions on Electromagnetic Compatibility, vol. 44, no. 1, pp. 249–258, 2002. [View at Publisher](#) · [View at Google Scholar](#) · [View at Scopus](#).
- [06] D. B. Levey and S. McLaughlin, « The statistical nature of impulse noise interarrival times in digital subscriber loop systems, » Signal Processing, vol. 82, no. 3, pp. 329–351, 2002. [View at Publisher](#) · [View at Google Scholar](#) · [View at Scopus](#)

- [07] S. A. Kassam, *Signal Detection in Non-Gaussian Noise*, Springer, New York, NY, USA, 1988.
- [08] V. Kontorovich and V. Lyandres, « Impulsive noise: a nontraditional approach, » *Signal Processing*, vol. 51, no. 2, pp. 121–132, 1996. [View at Publisher](#) · [View at Google Scholar](#) · [View at Scopus](#)
- [09] P. G. Georgiou, « Alpha-stable modeling of noise and robust time delay estimation in the presence of impulsive noise, » *IEEE Transactions on Multimedia*, vol. 1, no. 3, pp. 291–301, 1999. [View at Publisher](#) · [View at Google Scholar](#) · [View at Scopus](#).
- [10] T. Majoul, F. Raouafi, and M. Jaïdane, « Semi-blind turbo decoding in impulsive noise channels ». *IEEE , ISCCSP 2008*. Malta, 12-14 March 2008, pp. 810-813.
- [11] M. Shafieipour, H. S. Lim, and T. C. Chuah, « Decoding of turbo codes in symmetric alpha-stable noise, » *ISRN Signal Process*, vol. 2011, no. 683972, pp. 1-7, Jan., 2011.
- [12] D. Roviras, *Communications numériques 1* (version 10),. CNAM, Département EASY Electronique, Automatique et Systèmes, 2014-2015. Paris.
- [13] N. Prayongpun, *Modélisation et étude de la capacité du canal pour un système multi-antennes avancé exploitant la diversité de polarisation*, Thèse de doctorat, l'université Joseph Fourier, Grenoble. Soutenu le 29 avril 2009.
- [14] S. Riffi, M. Karmouche and M. Lamotte, « Estimation d'un signal entaché de bruit impulsif, » *J. Phys. IV France Volume 04, Number C5*, Mai 1994 3ème Congrès français d'acoustique, France.
- [15] A. Boyer « Canaux de transmissions bruités, » Support de cours et énoncé de travaux dirigés 4^{ème} IR, Institut National des Sciences Appliquées de Toulouse, Sep 2011.
- [16] K. Banzouzi, *Mesures de dépendance pour une modélisation alpha-stable. Application aux séries chronologiques stables*, Thèse Doctorat en statistique, Aoû 2011, Université Toulouse III - Paul Sabatier, Toulouse, France.

- [17] A. Ludovic D'Estampes, Traitement statistique des processus alpha-stables: mesures de dépendance et identification des ar stables. Test séquentiels tronqués. Mathematics. Institut National Polytechnique de Toulouse - INPT, 2003. French.
- [18] F. Flitti et Y. Omarouyoub, *codes convolutifs : Etude, Simulation et valuation*, Projet de fin d'étude, ENP, département d'électronique, Sept. 1996.
- [19] S. Haykin, *Digital Communication*, John Wiley & Sons, New York.1988.
- [20] C. Berrou, *codes et turbocodes*, Springer Paris, octobre 2007.
- [21] Y. Mori, Codage de source et de canal, Vol 6, Ed. LAVOISIER, Paris-France, 2006.
- [22] V. T. Zarasoa, « Codage de sources distribué : codage symétrique et adaptatif en débit de sources corrélées, » Rapport du stage, université de RENNES 1, cedex, France, Sep. 2007.
- [23] J. L. Ramsey. « Realization of optimum interleavers, » IEEE Trans. Inform. Theory. vol.16. no. 3, pp. 338-345, May. 1970.
- [24] S. A. Barbulescu, and S.S. Pietrobon, (1994), Interleaver Design for Turbo codes. Electronic Letters, 30(25), 2107-2108.
- [25] G. D. Forney, « Burst-correcting codes for the class bursty channel, » IEEE Trans. On Commun. Technol. Vol. COM-19, No. 5, October 1971, pp. 772-780.
- [26] Pierre Csillag, Introduction aux codes correcteurs, Edition Marketing, 1990.
- [27] L. Jusif, H. Weinrichter, W. Henkel, and J. Sayir, *Interleaver Design for Minimum Latency Turbo coding*, Master's thesis, Insitut für Nachrichten- und Hochfrequenztechnik, Wien, 2003.
- [28] S. Dolinar, and D. Divsalar, « Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations, » JPL TDA Progress Report 42-122, Aug. 15, 1995.
- [29] Chris Heegard and Stephen Wicker, « Turbo Coding, » ISBN: 0-7923-8378-8, Kluwer Academic Press, January 1999.

- [30] A. H. Osseiran, *Sur le décodage des codes turbo*, Mémoire de maîtrise des sciences appliquées (Génie électrique), Oct 1999, Université de Montréal, Ecole Polytechnique de Montréal, Canada.
- [31] G. Royer, *Evaluation des entrelaceurs au sein des Codes Turbo par simulations*, Mémoire de Maîtrise des sciences appliquée, Ecole polytechnique de Montréal, Nov 2000.
- [32] J. Briffa, *Interleavers for Turbo Codes*, Master dissertation of Philosophy of the University of Malta, Faculty of Engineering Oct 1999.
- [33] D. Divsalar and F. Pollara, « Turbo Codes for Deep-Space Communications, » The Telecommunications and Data Acquisition Progress Report 42-121, Jet Propulsion Laboratory, Pasadena, California Institute of technology, pp. 66-77, May 15, 1995.
- [34] L. Dinoi and S. Benedetto, « Design of prunable S-random interleavers, » in Proc. International Symposium on Turbo codes, Brest, France, 2003, pp. 279–282.
- [35] M. Ferrari, F. Scalise, and S. Bellini, « Prunable S-random interleavers, » in Proc. IEEE Intl. Commun. Conference (ICC), vol. 3, 2002, pp.1711 –1715.
- [36] L. Jusif, H. Weinrichter, W. Henkel, and J. Sayir, *Interleaver Design for Minimum Latency Turbo coding*, Master's thesis, Institut für Nachrichten- und Hochfrequenztechnik, Wien, 2003.
- [37] H. R. Sadjadpour, N. J. A. Sloane, M. Salehi, G. Nebe, « Interleaver design for turbo codes, » IEEE Journal on Selected Areas in Communications 19(5): 831-837 (2001).
- [38] N. Jaoua, *Estimation Bayésienne non Paramétrique de Systèmes Dynamiques en présence de Bruit Alpha-stables*, Thèse doctorat, 06 Jui 2013, Laboratoire d'Automatique, Génie Informatique et Signal UMR CNRS 8219, Université de Lille, France.
- [39] V. M. Zolotarev (1986). One-dimensional stable distributions, volume 65 of Translations of Mathematical Monographs. American Mathematical Society, Providence.

- [40] G. Samorodnitsky et M. S. Taqqu (1994). *Stable non-Gaussian random processes*, Stochastic Modeling. Chapman & Hall, New York-London.
- [41] C. Nikias et M. Shao (1995). *Signal processing with alpha-stable distributions and applications*. Wiley-Interscience.
- [42] A. Janicki and A. Weron, « Can one see α –Stable Variables and Processes, » *Statistical Science* Vol. 9, No. 1 (Feb., 1994), pp. 109-126.
- [43] M. Laurence, (4 mars 2010). *Lois α –stables*, d'après Nikias et Shao.
- [44] W. Feller (1971). *An introduction to probability theory and its applications*. Vol. II. John Wiley & Sons Inc., 2 éd.
- [45] B. Mandelbrot (1962). Sur certains prix spéculatifs : faits empiriques et modèle basé sur les processus stables additifs non gaussiens de Paul Lévy. *Comptes-rendus à l'Académie des Sciences*, vol. 254. pages 3968–3970.
- [46] C. J. M. Hambers, C. L. Mallows et B. W. Stuck (1976), A method for simulating stable random variables. *Journal of the American Statistical Association*, vol. 71, n°354. pages 340–344.
- [47] C. Berrou, A. Glavieux, P. Thitimajshima : « Near Shannon limit error-correcting coding and decoding : turbo-codes, » *Proceedings of ICC'93*, Genève, PP. 1064-1070, Mai 1993.
- [48] J. B. Doré, *Optimisation conjointe de codes LDPC et de leurs architectures de décodage et mise en œuvre sur FPGA*, thèse de doctorat, INSA de Rennes, France, octobre 2007.
- [49] B. Ferhat et A. Guemari, *Les turbocodes convolutifs avec modulation BPSK pour un canal AWGN*, Projet de fin d'étude, Université de Kasdi Merbah, département de mécanique et d'électronique, Ouaregla, Juin 2008.
- [50] H. Robert Morelos-Zaragoza, *The Art of Error Correcting Coding* : Second Edition, JOHN WILEY & SONS, LTD, ISBN: 0-470-01558-6, England, 2006.
- [51] J. G. Proakis, *Digital Communication*. Third edition, Mc Graw-Hill, 1995.

- [52] Frédéric LEHMANN, *Les Systèmes de Décodage Itératif et leurs Applications aux Modems Filaires et Non-filaires*, thèse de doctorat, INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE, France, décembre 2002.
- [53] L. Bahl, J. Cocke, F. Jeineck, and J. Raviv, « Optimal decoding of linear codes for minimizing symbol error rate, » IEEE Trans. Inf. Theory, pp. 284-287, Mar. 1974.
- [54] S. S. Pietrobon, « Implementation and performance of a turbo/map decoder ,» Int. J. Satellite Commun., Feb. 1997.
- [55] F. J. Mac Williams and N.J.A. Sloane, « Pseudo-Random Sequences and Arrays, » Proceedings of the IEEE, Vol, IT-64, PP. 1715-1728, Decem. 1989.
- [56] A. Spataru, *Fondements de la théorie de la Transmission de l'information*, Presse Polytechnique Romandes. Lausanne, Suisse, 1987.
- [57] J. C. Bie. Duponteil, J.C. Lmbeaux, *Elements de Communications Numériques : Transmission sur Fréquence Porteuse*, Tome II, Bordas et C.E.N.T.E.N.S.T, Paris, 1986.
- [58] T. Majoul, F. Raouafi and M. Jaïdane, « Semi-blind turbo decoding in impulsive noise channels, » IEEE , ISCCSP 2008, Malta, 12-14 March 2008, pp. 810-813.
- [59] C. Berrou, A. Glavieux, « *turbo codes : principe et application*, » département électronique, département signal et communications, ENST, France. 1995.
- [60] C. Yunfei, and C. B. Norman, , « Novel low-complexity estimators for the shape parameter of the generalized gaussian distribution ». IEEE Transactions on Vehicular Technology, Vol.58, No.4, May 2009, pp 2067-2071.
- [61] M. Nassar and M. DeYoung, « Receivers for Digital Communication in Symmetric Alpha Stable Noise, ». Wireless Communication Lab Final Project, Vol 7, Dec 2007.