

N° d'ordre :

N° de série :

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Mohamed Khider de Biskra

Département de Génie Electrique



Thèse présentée en vue de l'obtention du diplôme de

Doctorat en sciences

Spécialité : **Génie Electrique**

Option : **Automatique**

Présentée par : **CHEMSA Ali**

Intitulée

Modulation avec codage itératif pour un canal à bruits non-gaussiens

Soutenue publiquement le : 11 Avril 2016

Devant le jury composé de

BENCHOUIA Mohamed Toufik	Professeur	Université de Biskra	Président
SAIGAA Djamel	Professeur	Université de M'Sila	Rapporteur
TALEB-AHMED Abdelmalik	Professeur	Université de Valenciennes-France	Co-Rapporteur
ZIAT Lahcen	M.C.A	Université de Sétif	Examineur
SAADOUNE Achour	M.C.A	Université de Biskra	Examineur
MECHGOUG Raihane	M.C.A	Université de Biskra	Examineur

2015 - 2016



الإهداء



أهدي هذا العمل المتواضع

إلى سيدي وحيبي وقرّة

عيني رسول الله صلى الله

عليه وسلم

العبد الفقير الفاني علي شمسة بن البشير التجاني



REMERCIEMENTS

Je rends grâce à **ALLAH** le Tout Miséricordieux, le Très Miséricordieux, qui m'a donné cette capacité de réfléchir, cette capacité de comprendre, cette capacité d'analyser les choses que l'on appelle « al-'aql ». Je rends grâce à **LUI**, qui a permis la réalisation de ce travail. Que le salut d'**ALLAH** soit sur notre **Maître Mohamed** (PSL).

Ce document est le reflet de l'essentiel des idées et arguments que j'ai développés pendant ces années de thèse. Je veux remercier ici, tous ceux qui ont permis que ce travail aboutisse, en espérant n'oublier personne...

Je souhaite adresser mes plus sincères remerciements à mon encadreur **Monsieur SAIGAA Djamel**, Professeur à l'université de M'Sila, qui a accepté de diriger cette thèse. Je le remercie du temps qu'il a consacré à examiner les questions que ce travail soulevait et à s'engager dans les discussions qui s'en sont suivies. Ses conseils, ses suggestions, ses encouragements, et son aide précieuse pour l'organisation des idées, ainsi que la correction de la langue ont été indispensables pour l'aboutissement de ce travail.

Je tiens à remercier vivement mon co-encadreur **Monsieur TALEB-AHMED Abdelmalik**, Professeur à l'université de Valenciennes – Lille en France pour ses conseils précieux et pour toutes les commodités et aisances qu'il m'a apportées durant mon stage longue durée dans son laboratoire LAMIH en France.

Je tiens particulièrement à remercier **Monsieur BENCHOUIA Mohamed Toufik**, Professeur à l'université de Biskra, qui m'a fait l'honneur de présider le jury de ma thèse.

Mes remerciements les plus vifs s'adressent aussi aux membres du jury : **Monsieur ZIAT Lahcen**, Maître de conférences « A » de l'université de Sétif, **Monsieur SAADOUNE Achour**, Maître de conférences « A » de l'université de Biskra et **Madame MECHGOUG Raihane**, Maître de conférences « A » de l'université de Biskra, qui m'ont fait l'honneur d'accepter d'examiner et d'évaluer ce travail.

Je exprime également ma gratitude à tous les **enseignants** qui ont collaboré à ma formation depuis ma première année primaire jusqu'à la fin de mes études supérieures, surtout **Monsieur TOUAHRI Taher** et tous les **enseignants** de l'Ecole Nationale Polytechnique d'Alger ENP.

Enorme merci à **Sidi Abdellah**, mon amie, d'avoir été à mes côtés, grâce à vous ces années de thèse se sont aussi déroulées sans ergonomie dans la joie de partager vos événements !

Je remercie mon pays « **L'Algérie** » qui m'a donné l'opportunité de poursuivre mes études pour finaliser ma thèse en France et bien sûr, plus particulièrement, **l'université d'El-Oued**.

La fin d'une thèse constitue une étape dans la vie de son auteur, aussi il est naturel que mes pensées les plus fortes aillent vers **ma famille**. Je tiens à remercier de tout coeur **mes parents**, mon père **Bachir** et ma mère **Fatma**, qui m'ont toujours soutenus dans cette aventure et n'ont reculé devant aucune épreuve. Je leur serai toujours reconnaissant pour tout ce qu'ils ont bien voulu faire de moi. Je salue ici toutes leurs déterminations, leurs efforts et leurs sens du sacrifice dans tous les moments où je me suis sentie « like a rolling stone »... Aussi, **Amel** ma sœur, **Ahmed**, **Boudiaf**, **Mohamed Taher**, **Laïd**, **Abdellatif** et **Mohamed ben faredj**, mes frères. Je leur dois beaucoup et plusieurs parts de ma personnalité. Je les remercie pour leurs encouragements, leur soutien.

Je remercie **ma femme** qui m'a soutenu et encouragé de tous les instants, je la dis : « je t'aime ».

Ali CHEMSA

Sommaire

Résumé	1
Liste des figures et tableaux.....	2
Glossaire	8
Introduction générale.....	10
Chapitre 1 Concepts de base sur les DCS	16
1.1 Introduction.....	16
1.2 Architecture générale d'un DCS.....	16
1.3 Modélisation du canal	20
1.3.1 Canal discret	20
1.3.2 Canal continu	21
1.4 Capacité du canal.....	22
1.4.1 Canal discret stationnaire et sans mémoire	22
1.4.2 Canal continu stationnaire et sans mémoire.....	23
1.5 Théorèmes fondamentaux sur le codage du canal	25
1.5.1 Théorème de codage de Shannon	25
1.5.2 Théorème de Nyquist	26
1.6 Types de codes	26
1.6.1 Les turbo codes convolutifs CTC parallèles	28
1.6.2 Types d'entrelaceurs	30
1.6.2.1 Entrelaceur par blocs.....	30
1.6.2.2 Entrelaceur hélicoïdal	31
1.6.2.3 Entrelaceur convolutif.....	32
1.6.2.4 Entrelaceur à décalage circulaire	33
1.6.2.5 Entrelaceur aléatoire.....	34
1.6.2.6 Entrelaceur S –aléatoire	36
1.7 Techniques de modulation.....	38
1.8 Conclusion	41

Chapitre 2 Distributions des bruits non gaussiens : GG et α –stables	42
2.1 Introduction.....	42
2.2 Les distributions gaussiennes généralisées GGD.....	43
2.2.1 La densité gaussienne généralisée.....	44
2.2.2 Simulation de la distribution gaussienne généralisée	46
2.2.2.1 Simulation d'un bruit gaussien généralisé avec $\alpha = 1/3$	48
2.2.2.2 Les résultats numériques ($\alpha = 1/3$).....	49
2.2.3 Simulation d'un bruit gaussien généralisé avec $\alpha = 1/2$	50
2.2.3.1 La fonction W de Lambert	50
2.2.3.2 Bruit gaussien généralisé pour $\alpha = 1/2$	51
2.2.3.3 Evaluation numérique de W_{-1}	52
2.2.3.3.1 Evaluation de précision arbitraire.....	53
2.2.3.3.2 Approximation rapide.....	54
2.2.3.4 Génération du bruit gaussien généralisé de paramètre $\alpha = 1/2$	56
2.3 Les lois α –stables.....	57
2.3.1 Rudiments des lois α –stables.....	58
2.3.1.1 Définitions.....	58
2.3.1.2 Définition et interprétation des paramètres d'une loi $S_\alpha(\beta, \gamma, \delta)$	60
2.3.2 Diverses propriétés	62
2.3.3 Puissance géométrique	65
2.3.3.1 Définitions.....	65
2.3.3.2 Propriétés	66
2.3.3.3 Puissance géométrique d'une distribution $S_\alpha S$	67
2.3.3.4 Le rapport signal sur bruit géométrique	67
2.3.4 La simulation d'un bruit α –stable	67
2.3.5 Statistiques sur les lois symétriques α –stables	69
2.4 Conclusion	71
Chapitre 3 Le décodage itératif avec les algorithmes SISO : SOVA et MAP	74
3.1 Introduction.....	74
3.2 La métrique de vraisemblance logarithmique (de corrélation).....	75
3.3 Les algorithmes de décodage SISO	76
3.3.1 Algorithme de Viterbi à sortie douce SOVA	77
3.3.1.1 Algorithme de Viterbi classique avec la métrique de corrélation.....	77

3.3.1.2	Algorithme de Viterbi à sortie douce SOVA	82
3.3.2	Algorithme de maximum a posteriori MAP (ou BCJR)	88
3.3.2.1	Notations et règle de décision	90
3.3.2.2	L'Algorithme MAP utilisé.....	92
3.3.2.2.1	Expression de la probabilité conjointe.....	92
3.3.2.2.2	Expression de la métrique d'état en avant FSM	93
3.3.2.2.3	Expression de la métrique d'état en arrière BSM.....	94
3.3.2.2.4	Calcul de la métrique de branche BM	95
3.4	Schéma de principe d'un décodeur itératif.....	101
3.5	Conclusion	102
Chapitre 4	Nouvelles contributions et simulation numérique.....	105
4.1	Introduction.....	105
4.2	Modèle de simulation	106
4.2.1	Source d'information.....	107
4.2.2	Les CTC parallèles utilisés.....	108
4.2.3	Modulation utilisée	112
4.2.4	Source de bruit	113
4.2.4.1	Simulation d'un bruit gaussien AWGN	113
4.2.4.2	Simulation d'un bruit GG.....	114
4.2.4.3	Simulation d'un bruit $S\alpha S$ de Cauchy	115
4.2.5	Décodeur itératif avec algorithme SISO	116
4.2.5.1	L'implémentation de l'algorithme SOVA.....	117
4.2.5.2	L'implémentation de l'algorithme MAP	121
4.2.6	Comparateur (Calcul du taux d'erreurs binaire BER)	122
4.3	Nouvelles contributions à l'amélioration des décodeurs itératifs	126
4.3.1	Les FLOS d'un bruit impulsif et la probabilité ρ	126
4.3.1.1	Les FLOS d'un bruit impulsif GG	127
4.3.1.2	Les FLOS d'un bruit impulsif $S\alpha S$	129
4.3.1.3	La notion et le calcul de la probabilité ρ	131
4.3.2	Approche 1 : Nouvelle approche pour optimiser un décodeur itératif pour un canal AWGN.....	134
4.3.2.1	Approche proposée et décodeur itératif modifié MID.....	136
4.3.2.2	Résultats et discussions.....	138

4.3.3 Approche 2 : Nouvelle approche semi-aveugle pour le turbo décodage dans les canaux à bruit impulsif GG.....	146
4.3.3.1 L'effet de la disparité des paramètres du bruit sur les performances du décodeur	147
4.3.3.2 Estimation semi-aveugle du paramètre de forme utilisant HOS	149
4.3.3.3 Estimation semi-aveugle du paramètre de forme utilisant FLOS	150
4.3.3.3.1 Méthodes d'estimation du paramètre de forme d'une GGD	150
4.3.3.3.2 Approche proposée.....	152
4.3.3.3.3 Résultats et discussions	154
4.3.4 Approche 3 : Nouvelle approche semi-aveugle pour le turbo décodage dans les canaux à bruit impulsif <i>SαS</i> de Cauchy	159
4.3.4.1 L'effet de la disparité des paramètres du bruit sur les performances du décodeur	160
4.3.4.2 Estimation semi-aveugle du paramètre de dispersion utilisant FLOS	162
4.3.4.2.1 Approche proposée.....	162
4.3.4.2.2 Résultats et discussions	164
4.4 Conclusion	167
Conclusion générale et perspectives	169
Annexe A La GGD et la loi gamma	172
A.1 La variable aléatoire à GGD comme une transformation d'une variable aléatoire gamma	172
A.2 Simulation d'une variable aléatoire gamma	175
Annexe B Les séquences pseudo-aléatoires	177
B.1 Introduction	177
B.2 Les séquences pseudo-aléatoires.....	177
B.2.1 Le registre à décalage.....	177
B.2.2. Séquences pseudo-aléatoires.....	178
B.2.3 Propriétés des séquences pseudo-aléatoires.....	180
B.2.4 La fonction d'autocorrélation	181
B.2.5 La Série.....	181
B.2.6 Séquences pseudo-aléatoires <i>q</i> -aires	182
Bibliographie	183

Résumé

Ce travail étudie le décodage itératif ou le turbo décodage, des turbo codes convolutifs pour un bruit généralement non gaussien, particulièrement le bruit impulsif, à savoir le bruit GG symétrique qui possède une variance finie, et le bruit α -stable symétrique $S\alpha S$ à variance infinie. Pour le dernier type de bruit, la définition du SNR n'est pas valide et nous avons utilisé à la place le paramètre de dispersion du bruit γ . Nous savons que les performances d'un décodeur itératif dépendent étroitement de l'information extrinsèque délivrée par chaque décodeur constitutif SISO, parce que plus cette information est fiable plus le décodeur itératif est performant. Dans ce travail, nous avons proposé trois approches originales qui rendent l'information extrinsèque plus fiable. La première approche est établie pour les deux algorithmes de décodage SOVA et MAP dans le cas d'un canal AWGN. Elle consiste à ajouter une fonction porte (binaire) qui dépend des itérations dans l'entrée systématique du décodeur itératif, afin de rendre les informations extrinsèques délivrées par les décodeurs constitutifs plus fiables, ce qui permet ainsi d'améliorer les performances en termes de BER et de convergence. La deuxième et la troisième approches sont utilisées dans l'algorithme MAP pour les canaux à bruit impulsif GG et $S\alpha S$ respectivement. Ces approches sont deux méthodes d'estimation semi-aveugles basées sur les statistiques d'ordre inférieur fractionnel FLOS. Elles estiment les paramètres du bruit à partir du signal reçu uniquement. Ces paramètres estimés sont utilisés par l'algorithme MAP pour calculer l'information extrinsèque. L'avantage majeur de ces deux approches est qu'elles présentent de bonnes estimations des paramètres, ce qui permet aux décodeurs constitutifs de donner des informations extrinsèques plus fiables.

Mots clés : AWGN, BER, BPSK, Bruit impulsif GG, Bruit impulsif $S\alpha S$ de Cauchy, CTC, Décodeur itératif, FLOS, GSNR, MAP, SISO, SNR, SOVA.

Abstract

This work studies the iterative decoding or the turbo decoding of the convolutional turbo codes for a generally non-gaussian noise, especially impulsive noise, namely the symmetric GG noise which has a finite variance, and the symmetric α -stable noise ($S\alpha S$) with an infinite variance. For the last type of noise, the SNR definition is not valid and we used instead of it the dispersion parameter γ of the noise. We know that the performance of an iterative decoder are closely related to the extrinsic information provided by each constituent SISO decoder, because more this information is reliable more the iterative decoder is performing. In this work, we proposed three original approaches that make the most reliable extrinsic information. The first approach is established for both decoding algorithms MAP and SOVA in the case of an AWGN channel. It is to add a gate function (binary) that depends on the iterations, in the systematic input of the iterative decoder, in order to make the extrinsic informations delivered by the constitutive decoders more reliable, this allows also to improve the performance in terms of BER and convergence. The second and third approaches are used in the MAP algorithm for a channel with GG and $S\alpha S$ impulsive noise respectively. These approaches are two semi-blind estimation methods based on fractional lower order statistics FLOS. They estimate the parameters of the noise from the received signal only. These estimated parameters are used by the MAP algorithm to calculate the extrinsic information. The major advantage of these two approaches is that they have good estimates of the parameters, which allows the constitutive decoders to provide more reliable extrinsic informations.

Key words : AWGN, BER, BPSK, CTC, FLOS, GG impulsive noise, GSNR, Iterative decoder, MAP, $S\alpha S$ impulsive noise of Cauchy, SISO, SNR, SOVA.

ملخص

هذا العمل يدرس الفكاكات التكرارية أو الترميز فكاكات المستعملة في فك رموز الترميز لافافات من أجل تشويش غير غوسي عموماً، يعني التشويش النبضي كالتشويش المتناظر GG والذي لديه تباين محدود، و التشويش α -مستقر $S\alpha S$ المتناظر الذي يملك تباين لانهاضي. بالنسبة للنوع الأخير من التشويش، تعريف SNR غير صالح فيه، واستخدمنا بدلاً منه معامل تشتت التشويش γ . نحن نعلم أن أداء أي فكاك تكراري متعلق تعلقاً وثيقاً بالمعلومات الغير جوهرية المقدمة من طرف كل من الفكاكات المكوّنة له SISO، لأنه كلما كانت هذه المعلومات أكثر موثوقية كان الفكاك التكراري أكثر فعالية. في هذا العمل، اقترحنا ثلاث تقنيات أصلية والتي تجعل المعلومات الغير جوهرية أكثر موثوقية. تم تأسيس التقنية الأولى من أجل خوارزميات الفك SOVA و MAP وهذا في حالة قناة AWGN. هذه التقنية الجديدة تتمثل في إضافة دالة بوابة (ثنائية) إلى المدخل النظامي للفكاك التكراري، وهذا من أجل جعل التحصل على معلومات غير جوهرية من الفكاكات المكوّنة أكثر موثوقية، وبالتالي التحصل على أداء فعال للفكاك التكراري من حيث BER و من حيث التقارب. أما التقنيتين الثانية والثالثة فهما تستخدمان في الخوارزمية MAP من أجل التشويشين GG و $S\alpha S$ على التوالي. هاتين التقنيتين هما طريقتي تقدير شبه عمياوين تعتمدان على الاحصاءات ذات الرتب الصغرى الكسرية FLOS. فهما تقدران وسائط التشويش من الإشارة الواردة على مستوى المستقبل فقط. وتستخدم هذه الوسائط المقدرّة من قبل خوارزمية MAP لحساب المعلومات الغير الجوهرية. والميزة الرئيسية لهاتين التقنيتين هي أن لديهما تقديرات جيدة للوسائط، وهذا ما يسمح لفكاكات التشفير المكوّنة إعطاء معلومات غير جوهرية أكثر موثوقية.

المصطلحات الأساسية : AWGN, BER, BPSK, تشويش انبضي GG, تشويش نبضي $S\alpha S$ لكوشي, CTC, فكاك تكراري, FLOS, GSNR, SOVA, SNR, SISO, MAP.

Liste des figures

Fig. 1.1 Architecture générale d'un DCS.....	17
Fig. 1.2 Canal discret sans mémoire.	21
Fig. 1.3.a Turbo code série.	27
Fig. 1.3.b Turbo code parallèle.....	27
Fig. 1.4.a Code convolutif récursif systématique (5,7).....	29
Fig. 1.4.b Représentation en treillis du RSC de la figure 1.4.a.....	29
Fig. 1.4.c Turbo code convolutif parallèle (5,7).	29
Fig. 1.5 Ecriture-Lecture dans un entrelaceur par blocs.....	31
Fig. 1.6 Entrelacement hélicoïdale pour $m = 5$ et $n = 4$	32
Fig. 1.7 Schéma de principe d'un entrelaceur convolutif avec $K = 4$ et $B = 1$	32
Fig. 1.8 Entrelacement à décalage circulaire pour $L = 8$, $a = 3$ et $s = 0$	33
Fig. 1.9 Entrelacement S –aléatoire $N = 12$, $S = 2$	37
Fig. 1.10 Diagramme de constellation pour la BPSK.	39
Fig. 1.11 La probabilité d'erreur P_e en fonction du SNR pour un bruit GG.....	40
Fig. 1.12 La probabilité d'erreur P_e en fonction de $1/\gamma$ pour un bruit $S\alpha S$ de Cauchy . .	40
Fig. 2.1 La densité gaussienne généralisée pour différentes valeurs du paramètre α , avec $\mu = 0$ et $\sigma = 1$	46
Fig. 2.2 Le Kurtosis de la GGD avec variance unité et α variable : (a) dans $[0, 2]$ (b) dans $[0, 20]$	46
Fig. 2.3 Les réalisations d'un bruit à GGD avec les paramètres $\alpha = 1/3$, $\mu = 0$, et $\sigma = 1$ ($N = 10\ 000$).	49

Fig. 2.4 Les fonctions de densité de probabilité estimée et théorique d'un bruit à GGD avec $\alpha = 1/3$ et kurtosis $\mathcal{K}_X(1/3) = 107.25$	49
Fig. 2.5 Les deux branches réelles de la fonction W de Lambert. La branche $W_{-1}(z)$ (ligne continue) : définie pour $-1/e \leq z < 0$. La branche principale $W_0(z)$ (ligne pointillée) : définie pour $-1/e \leq z < +\infty$. Les deux branches ont un point commun $(-1/e, -1)$	51
Fig. 2.6 Les temps d'exécution en secondes pour générer un bruit à GGD avec $\alpha = 1/2$ en utilisant directement la fonction de Lambert de Matlab puis l'algorithme d'approximation.	55
Fig. 2.7 Les réalisations d'un bruit à GGD avec les paramètres $\alpha = 1/2$, $\mu = 0$, et $\sigma = 1$ ($N = 10\ 000$).	56
Fig. 2.8 Les fonctions de densité de probabilité estimée et théorique d'un bruit à GGD avec $\alpha = 1/2$	56
Fig. 2.9 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0, \gamma = 1, \delta = 0$	61
Fig. 2.10 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = -1, \gamma = 1, \delta = 0$	61
Fig. 2.11 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 1, \gamma = 1, \delta = 0$	61
Fig. 2.12 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0, \gamma = 2, \delta = 0$	61
Fig. 2.13 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0, \gamma = 5, \delta = 0$	61
Fig. 2.14 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0, \gamma = 1, \delta = -3$	61
Fig. 2.15 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0, \gamma = 1, \delta = 3$	62
Fig. 2.16 Le tracé de 5000 échantillons d'un bruit $S\alpha S$: (a) $\alpha = 2$ (b) $\alpha = 1.5$ (c) $\alpha = 1$	70
Fig. 2.17 Test graphique de la variance pour $N = 5000$ échantillons d'un bruit $S\alpha S$: (a) $\alpha = 2$ (b) $\alpha = 1.5$ (c) $\alpha = 1$	71
Fig. 3.1 Le treillis simplifié du RSC (5,7).	79
Fig. 3.2 L'algorithme de Viterbi classique avec métrique de corrélation pour le RSC (5,7).	81
Fig. 3.3 Calcul des métriques de branches dans les deux sens du treillis.	87

Fig. 3.4 Représentation graphique de la métrique d'état en avant.....	94
Fig. 3.5 Représentation graphique de la métrique d'état en arrière.....	95
Fig. 3.6 La FSM, BSM et BM sur le treillis simplifié du RSC (5,7) pour calculer le $LLR(d_k)$	98
Fig. 3.7 Schéma bloc d'un décodeur itératif associé à un CTC.....	102
Fig. 4.1 Modèle de simulation.....	106
Fig. 4.2 Générateur pseudo-aléatoire.	107
Fig. 4.3.a Code convolutif récursif systématique (5,7).....	109
Fig. 4.3.b Code convolutif récursif systématique (13,15).....	109
Fig. 4.3.c Code convolutif récursif systématique (35,23).	109
Fig. 4.4.a Représentation en treillis du codeur de la figure 4.3.a.	110
Fig. 4.4.b Représentation en treillis du codeur de la figure 4.3.b.	110
Fig. 4.5.a Turbo code convolutif parallèle (5,7).....	111
Fig. 4.5.b Turbo code convolutif parallèle (13,15).....	111
Fig. 4.5.c Turbo code convolutif parallèle (35,23).	112
Fig. 4.6 Le tracé de 5000 échantillons d'un bruit gaussien avec $\mu = 0$ et $\sigma = 1$	114
Fig. 4.7 Le tracé de 5000 échantillons d'un bruit impulsif GG avec $\mu = 0$ et $\sigma = 1$ pour deux valeurs de α : (a) $\alpha = 0.5$ (b) $\alpha = 1$	115
Fig. 4.8 Le tracé de 5000 échantillons d'un bruit impulsif $S\alpha S$ de Cauchy (a) $\gamma = 0.2$ (b) $\gamma = 20$	115
Fig. 4.9 Schéma bloc d'un décodeur itératif avec facteur de réglage de gain.	116
Fig. 4.10.a Fermeture du treillis.....	118
Fig. 4.10.b Organigramme de l'algorithme SOVA.	121
Fig. 4.11 Organigramme de l'implémentation du modèle de la figure 4.1.....	125
Fig. 4.12 La $RMSE$ de l'estimation du moment m_r en fonction de r pour un bruit impulsif GG et pour différentes valeurs de N : (a) $N = 100$ (b) $N = 1000$ (c) $N = 10\ 000$	129

Fig. 4.13 La <i>RMSE</i> de l'estimation du moment m_r en fonction de r pour un bruit impulsif $S\alpha S$ et pour différentes valeurs de N : (a) $N = 100$ (b) $N = 1000$ (c) $N = 10\,000$	130
Fig. 4.14 La probabilité ρ en fonction du paramètre de forme α pour un bruit impulsif GG.....	133
Fig. 4.15 La probabilité ρ en fonction du paramètre de dispersion γ pour un bruit impulsif $S\alpha S$ de Cauchy	133
Fig. 4.16 L'organigramme de l'algorithme de calcul de $p(i)$	137
Fig. 4.17 Schéma bloc d'un décodeur itératif modifié associé à un CTC.....	138
Fig. 4.18 Le <i>BER</i> en fonction du <i>SNR</i> des MID et CID à 4 états et entrelaceur 15 –aléatoire de taille 64×64 pour les deux algorithmes SOVA et MAP : (a) 6 itérations (b) 12 itérations.....	144
Fig. 4.19 Le <i>BER</i> en fonction du <i>SNR</i> des MID et CID à 8 états et entrelaceur 15 –aléatoire de taille 64×64 pour les deux algorithmes SOVA et MAP : (a) 6 itérations (b) 12 itérations.....	144
Fig. 4.20 Le <i>BER</i> en fonction du <i>SNR</i> des MID et CID à 4 états et entrelaceur 15 –aléatoire de taille 128×128 pour les deux algorithmes SOVA et MAP : (a) 6 itérations (b) 12 itérations.....	145
Fig. 4.21 Le <i>BER</i> en fonction du <i>SNR</i> des MID et CID à 8 états et entrelaceur 15 –aléatoire de taille 128×128 pour les deux algorithmes SOVA et MAP : (a) 6 itérations (b) 12 itérations.....	145
Fig. 4.22 L'effet de la disparité du paramètre de forme sur les performances du décodeur itératif. Les valeurs disparates de la valeur exacte $\alpha = 0.9$ sont $\alpha = 0.5$, $\alpha = 1.5$ et $\alpha = 2$. La valeur de σ est exacte.....	149
Fig. 4.23 L'effet de la disparité de l'écart type sur les performances du décodeur itératif. Les valeurs disparates de la valeur exacte σ sont 2σ , 4σ et 6σ . La valeur exacte du paramètre de forme est $\alpha = 0.9$	149
Fig. 4.24 Le modèle de simulation adopté pour calculer la <i>RMSE</i> en fonction de α	154
Fig. 4.25 La <i>RMSE</i> des approches HOS et FLOS en fonction du paramètre de forme pour $N = 1024$. (a) $SNR = -3\text{ dB}$ (b) $SNR = 0\text{ dB}$ (c) $SNR = 3\text{ dB}$	157

Fig. 4.26 La $RMSE$ des approches HOS et FLOS en fonction du paramètre de forme pour $N = 4096$. (a) $SNR = -3 \text{ dB}$ (b) $SNR = 0 \text{ dB}$ (c) $SNR = 3 \text{ dB}$	157
Fig. 4.27 La $RMSE$ des approches HOS et FLOS en fonction du paramètre de forme pour $N = 16384$. (a) $SNR = -3 \text{ dB}$ (b) $SNR = 0 \text{ dB}$ (c) $SNR = 3 \text{ dB}$	158
Fig. 4.28 Les performances du décodeur itératif pour un bruit impulsif GG dans les trois contextes : (\diamond) L'hypothèse de AWGN (\circ) Paramètres estimés avec l'approche FLOS (*) Paramètres exactes. Deux bruits impulsifs GG sont considérés : (a) $\alpha = 0.7$ (b) $\alpha = 0.3$	158
Fig. 4.29 La sensibilité de $LLR_c(R_k)$ à la disparité de paramètre γ	161
Fig. 4.30 Le modèle de simulation adopté pour calculer l'erreur $RMSE_{rel}$ en fonction de γ	164
Fig. 4.31 La $RMSE_{rel}$ en fonction de la valeur exacte de γ pour trois valeurs du nombre d'échantillons N	164
Fig. 4.32 Les deux erreurs de $LLR_c(R_k)$ en fonction de $1/\gamma$ (en dB) pour différentes estimation $\hat{\gamma}$: $\hat{\gamma}$ calculé par l'approche proposée, $\hat{\gamma} = 2\gamma$, $\hat{\gamma} = 4\gamma$ et $\hat{\gamma} = 6\gamma$. (a) $RMSE$ (b) $RMSE_{rel}$	166
Fig. 4.33 Le BER du décodeur itératif pour un bruit impulsif $S\alpha S$ de Cauchy dans les trois contextes : (\square) γ estimé = 2γ (une mauvaise estimation) (*) γ estimé (par notre approche) (\diamond) γ exacte.	167
Fig. B.1 Registre à décalage avec un feedback correspondant à $h(x) = x^4 + x + 1$	178
Fig. B.2 Registre à décalage défini par une relation de récurrence.	178
Fig. B.3 Fenêtre d'une séquence pseudo-aléatoire.....	180
Fig. B.4 Fonction d'autocorrélation d'une séquence pseudo-aléatoire.	181

Liste des tableaux

Tab. 1.1 Tableau de nombres premiers avec 8.	34
Tab. 1.2 Tableau de nombres premiers avec 8.	35
Tab. 3.1 Tableau des survivants du treillis de la figure 3.2.	81
Tab. 3.2 Les valeurs des BM du treillis du RSC (5,7) pour le message reçu $R = (-4 - 3 - 3 - 2 + 2 + 4 - 3 + 3 - 3 + 1)$ dans le but de calculer le $LLR(d_k)$	99
Tab. 4.1 La queue de séquence du CTC de la figure 4.5.a.	117
Tab. 4.2 La fonction porte $p(i)$ pour différents décodeurs itératifs.	140
Tab. 4.3 Comparaison MID / CID dans la zone des faibles SNR en fonction du nombre d'itérations.	142
Tab. 4.4 Le rapport $RMSE_{HOS}/RMSE_{FLOS}$	156
Tab. B.1 Polynômes primitifs pour $m \leq 40$	179

Glossaire

ADSL	A symmetric D igital S ubscriber L ine
AM	A mplitude M odulation
APP	A Posteriori P robability
ASK	A mplitude S hift K eying
AWGN	A dditif W hite G aussian N oise
BCJR	B ahl C ocke J elinek R aviv (Les noms des fondateurs de cet algorithme)
BER	B inary E rror R ate
BM	B ranch M etric
BPSK	B inary P hase S hift K eying
BSM	B ackward S tate M etric
CDF	C umulative D istribution F unction
CID	C lassical I terative D ecoder
CTC	C onvolutional T urbo C ode
DCS	D igital C ommunication S ystem
DEC	D ecoder
DSL	D igital S ubscriber L ines
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
GG	G eneralized G aussian
GGD	G eneralized G aussian D istribution
GSNR	G eometric S ignal N oise R ate

I.I.d	I ndependent and I dentically D istributed
JD	J oint D istribution
JPL	J et P ropulsion L aboratory
LDPC	L ow D ensity P arity C heck
LLR	L ogarithm L ikelihood R atio
LOS	L ower O der S tatistics
MAP	M aximum A P osteriori
MID	M odified I terative D ecoder
MIMO	M ulti I nput M ulti O utput
ML	M aximum L ikelihood
NRNSC	N on R ecursive N on S ystematic C ode
NSC	N on S ystematic C ode
OFDM	O rthogonal F requency D ivision M ultiplexing
PDF	P robability D ensity F unction
PLC	P ower L ine C ommunication
PM	P hase M odulation
PN	P seudo N oise
PSK	P hase S hift K eying
QAM	Q uadrature A mplitude M odulation
RMSE	R oot M ean S quare E rror
RSC	R ecursive S ystematic C ode
$S\alpha S$	S ymmetric α – S table with localization parameter $\delta = 0$
SC	S ystematic C ode
SISO	S oft I nput – S oft O utput
SNR	S ignal N oise R ate
SOVA	S oft O utput V iterbi A lgorithm
UMTS	U niversal M obile T elecommunications S ystem
UWB	U ltra W ide B and.

Introduction générale

Par définition, une transmission numérique permet d'acheminer des messages de nature numérique entre une source et un destinataire. Cette source délivre, avec un certain rythme, une suite de symboles qui prennent leurs valeurs dans un ensemble fini appelé alphabet. Le caractère numérique de la transmission se manifeste de manière évidente dans le fonctionnement du récepteur lors de l'échantillonnage et de la prise de décision. Il est beaucoup moins apparent au niveau du signal transmis qui présente plus nettement le caractère discret de la source.

Par une convention préalable, le destinataire a la connaissance de l'alphabet utilisé par la source. Il peut donc interpréter l'information qu'il reçoit en fonction de cet alphabet. Il compare les signaux reçus (déformés et perturbés par la transmission dans le canal) à la liste de caractères possibles et il déduit par une « décision » lequel de ces caractères est le plus probablement à l'origine du signal reçu.

La transmission de l'information entre la source et le destinataire ne s'effectue pas sans risques d'erreurs à cause du bruit de distorsion du canal. Afin de remédier à cela et donner une grande fiabilité au système de transmission, il existe plusieurs techniques de protections de l'information se basant toutes sur le principe d'ajouter une certaine redondance au message original avant la transmission à l'aide d'un codeur qui maximise la distance de **Hamming** entre les mots-code. Cette opération qui consiste à l'ajout des bits de redondance est dite « codage de canal ». Parmi les célèbres codes qui ont le pouvoir de combattre les erreurs de transmission, sont les codes convolutifs car leurs prépotences de correction s'accroissent lorsque la longueur du registre de codage augmente. Pour satisfaire les applications les plus courants du codage de canal, une mémoire ν de l'ordre de 30 ou 40 serait nécessaire (donc 2^{30} ou 2^{40} états, c'est à dire plus d'un milliard d'états) puisque à partir d'une certaine longueur de registre et pour un rendement de codage 1/2 la distance minimale de **Hamming** d'un code convolutif de mémoire ν est de l'ordre de grandeur de ν [1]. Si l'on savait décoder aisément un code

convolutif à plus d'un milliard d'états, nous ne parlions plus beaucoup de codage de canal! [1] Une telle proposition ne convienne plus. Pour cela, **C. Berrou** et **A. Glavieux** [2] en 1993, ont inventé des très bons codes qui font parfaitement l'affaire, constitués par une concaténation de deux codes convolutifs via un entrelaceur entre les deux codes. Ces codes concaténés sont non seulement des codes puissants, mais leur structure permet de plus une faible complexité de décodage grâce au décodage itératif. Le décodage itératif consiste à décoder alternativement les codes constitutifs et à passer de l'information entre les décodeurs constitutifs. Afin d'exploiter au mieux l'information produite par chaque décodeur constitutif, il a été établi que passer des décisions douces ou souples plutôt que des décisions fermes peut conduire à d'excellentes performances. Ce décodage itératif basé sur les décisions souples est appelé aussi le turbo décodage. Pour cette raison ces codes sont appelés « turbo codes convolutifs ».

Ces turbo codes sont étudiés et évalués dans l'hypothèse de gaussianité du bruit, puisque l'objectif était d'obtenir des relations simples sur les performances et des mises en œuvre peu coûteuses. Même si cette hypothèse est justifiée dans un large éventail de scénarios de situations réelles, il y a des situations où le théorème central limite ne peut plus être invoqué et où les expériences mettent en exergue un bruit non gaussien à caractère impulsif où la queue de la distribution de son amplitude est lourde.

Parmi ces situations, nous pouvons citer les chaînes de radio [3], les canaux de communication par courants porteurs en ligne PLC [4], et des lignes numériques d'abonné DSL [5]. Le bruit impulsif qui affecte ces systèmes est caractérisé généralement par des trains d'impulsions (isolées ou en rafales) de faibles durées et de forte densité spectrale de puissance.

Par conséquent, l'étude de ce type de bruit est indispensable. Diverses approches de modélisation du bruit impulsif sont couramment utilisées et nous pouvons citer notamment : les modèles basés sur les chaînes de **Markov**, le modèle de bruit de Classe A de **Middleton** ou bien encore le modèle stochastique. Ce dernier modèle utilise une approche statistique des paramètres caractéristiques des impulsions qui sont approchées par des lois de distributions connues. Les plus utilisées dans la littérature parmi ces distributions connues sont les distributions gaussiennes généralisées GGD et les distributions α –stables [6][7].

Les GGD constituent un modèle approprié pour décrire le phénomène des bruits impulsifs ou le phénomène de la queue lourde, elles ont une densité de probabilité bien

définie analytiquement. Cette densité possède trois paramètres : la moyenne μ , l'écart type σ et un paramètre de forme $\alpha > 0$. Il s'agit d'une extension de la densité gaussienne avec un paramètre supplémentaire, dit de forme, qui est une mesure d'aplatissement de la distribution. Cela permet d'englober en plus : la distribution gaussienne ($\alpha = 2$), la distribution de **Laplace** ($\alpha = 1$) et la distribution uniforme ($\alpha \rightarrow +\infty$) [8].

Les GGD sont utilisées pour modéliser plusieurs processus, qui sont observés dans des domaines variés dont le traitement de la parole, l'audio, ou le signal vidéo, les images, ou bien la turbulence [6].

Quant aux distributions α -stables, elles présentent un autre modèle pour décrire le phénomène des bruits impulsifs. Elles ont été introduites pour la première fois pendant les années 1920 dans les travaux de **Paul Lévy** [9] dans le théorème central limite généralisé. Par la suite, en 1970, trois chercheurs de **Bell Labs** (**Chambers**, **Mallow** et **Stuck**) ont prouvé que ce modèle peut servir à modéliser le bruit dans les lignes téléphoniques [10] ; elle a eu aussi du succès dans diverses applications en sciences de l'ingénieur comprenant le traitement des signaux de radar [11], la modélisation en télétrafic [12], etc.

Les distributions α -stables sont caractérisées par quatre paramètres : paramètre de forme $\alpha \in]0, 2]$, appelé aussi exposant caractéristique, paramètre d'asymétrie $\beta \in [-1, 1]$, paramètre de dispersion ou d'échelle $\gamma > 0$ et paramètre de localisation $\delta \in \mathbb{R}$. Dans le cas des distributions α -stables symétriques notées $S\alpha S$, les deux paramètres β et δ sont nuls. Si le paramètre α prend ses valeurs dans l'intervalle ouvert $]0, 2[$, ces distributions ont une densité de probabilité non-gaussienne, caractérisée par des queues lourdes. Par exemple, si $\alpha = 1$, les distributions α -stables ont une densité de **Cauchy** ; mais si $\alpha = 2$, les distributions sont gaussiennes.

Contrairement aux distributions GG, les distributions α -stables présentent des limites : d'une part elles ne permettent pas un accès analytique à la densité de probabilité sauf pour $\alpha = 2$ (cas gaussien) et $\alpha = 1$ (cas de **Cauchy**) ; d'autre part, si $\alpha \neq 2$, les statistiques d'ordre supérieur, y compris la variance, ne sont pas définies, ce qui reste un problème dans la recherche [13]. Pour cette raison, la définition de la puissance d'une variable aléatoire n'est pas valide pour ce type des distributions. **Gonzalez** dans [14] et [15] a proposé une autre forme de puissance, nommée puissance géométrique, où il a proposé et utilisé la théorie des statistiques d'ordre zéro, qui est basée sur les moments d'ordre logarithmique, afin de caractériser ce type des distributions.

Pour un bruit impulsif modélisé par une variable aléatoire à GGD ou à distribution α -stable, les décodeurs itératifs des turbo codes, qui sont optimisés au début pour les bruits gaussiens, donnent de performances médiocres [16][17].

Jusqu'à présent, peu de recherches ont été consacrées à l'optimisation des décodeurs itératifs pour un canal à bruit non-gaussien de type impulsif.

Pour ces raisons, notre travail est dédié au décodage itératif des turbo codes convolutifs CTC dans un canal à bruit généralement non-gaussien de type impulsif et avec modulation BPSK, où nous utilisons les deux algorithmes de décodage à entrées-sorties douces SISO, à savoir, l'algorithme de **Viterbi** à sortie douce SOVA et l'algorithme de maximum a posteriori MAP. Bien entendu, le bruit impulsif est représenté dans cette thèse par les deux distributions symétriques à queue lourde : la GGD et la distribution *SaS* de **Cauchy** où $\alpha = 1$.

Cette thèse est structurée en quatre chapitres.

Le premier chapitre est consacré à quelques concepts de base sur les systèmes de communication numérique DCS et leurs parties constituantes, en particulier le canal et les fonctions du codage et de la modulation. Cependant, pour ne pas être exhaustif, nous allons essayer de donner l'essentielle de ce thème. Nous allons calculer et analyser le comportement de la capacité du canal pour les trois canaux continus : canal gaussien, canal GGD et canal *SaS* de **Cauchy**. La notion des turbo codes convolutifs CTC est bien présentée dans ce chapitre, ainsi que les différents types d'entrelaceurs généralement utilisés dans les CTC.

Le second chapitre est dédié à l'étude des deux distributions des bruits non-gaussiens, à savoir, la GGD et la distribution α -stable. Ce chapitre consiste en des analyses théoriques et des algorithmes de simulation de ces bruits.

Dans le cas d'un bruit gaussien généralisé GG, nous présentons un algorithme rapide, et en un seul coup, pour synthétiser des réalisations effectives dans le cas où $\alpha = 1/2$ [6]. Nous exposons explicitement une méthode qui s'applique directement à l'évaluation numérique de la fonction de **Lambert**, cette dernière nous sert à résoudre le problème de synthèse du bruit GG de paramètre de forme $\alpha = 1/2$ [6].

Dans ce chapitre, nous verrons qu'un bruit GG a tous ses moments finis. Le kurtosis d'un tel bruit, qui caractérise l'aplatissement de la densité de probabilité du bruit par rapport à celui d'un bruit gaussien, est le rapport entre le moment d'ordre 4 et

le moment d'ordre 2 à la puissance 2 du bruit. Un bruit GG présente une queue lourde si et seulement si son kurtosis est strictement supérieur à celui d'un bruit gaussien qui est égal à 3, ce qui correspond à un paramètre de forme $\alpha < 2$.

Pour les bruits α -stables présentés dans ce deuxième chapitre, le paramètre de forme α qui est compris entre 0 et 2 est le plus important, car il indique la vitesse de décroissance des queues de la distribution du bruit. En plus, si $\alpha < 2$, tout moment d'ordre strictement supérieur à α est infini. Le seul bruit α -stable a tous ses moments finis est le bruit gaussien qui correspond à $\alpha = 2$. C'est le seul bruit α -stable qui ne soit pas à queue lourde. Dès lors, dans le cas où $\alpha \neq 2$, la notion du rapport signal sur bruit *SNR* pour caractériser l'intensité d'un bruit α -stable n'est plus valide. Pour cette raison, nous définirons dans ce chapitre un nouvel rapport dit le rapport signal sur bruit géométrique *GSNR* établi par **Gonzalez** [14].

Le troisième chapitre est réservé au décodage itératif avec les deux algorithmes SISO : SOVA et MAP. Nous présenterons finement ces deux algorithmes de décodage en définissant les notions suivantes : la décision ferme optimale au sens du MAP, la fiabilité de décision présentée par le rapport logarithmique de vraisemblance LLR et l'information extrinsèque. Pour chaque algorithme, nous donnerons un exemple numérique bien détaillé qui illustre comment estimer l'information envoyé et calculer la fiabilité de décision LLR. Pour l'algorithme MAP, nous exprimerons la métrique de branche BM pour un bruit AWGN, un bruit GG et un bruit *SaS* de **Cauchy**. Le principe du décodage itératif avec un algorithme SISO qui est basé sur l'information extrinsèque est aussi présenté dans ce chapitre.

Le quatrième chapitre est voué aux nouvelles contributions et à la simulation numérique. Dans ce chapitre, nous proposerons trois nouvelles approches pour optimiser un décodeur itératif selon le type du canal utilisé pour une modulation BPSK. La première approche est établie pour un canal AWGN. Elle permet au décodeur itératif SISO de travailler avec des informations extrinsèques plus fiables en incorporant une fonction porte (binaire), qui varie en fonction des itérations, dans son entrée systématique. La deuxième approche est utilisée dans l'algorithme MAP pour un canal bruit impulsif à GGD. Cette approche qui représente une méthode d'estimation est qualifiée de semi-aveugles, car elle estime le paramètre de forme du bruit envisagé à partir du signal reçu uniquement. Ce paramètre estimé par cette approche est donné par

une formule explicite en fonction de moments du signal reçu. Ces moments sont des statistiques d'ordre inférieur fractionnel FLOS. Quant à la troisième approche, elle possède exactement le même principe et les mêmes hypothèses que la deuxième approche, sauf qu'elle est créée pour un bruit *SaS* de **Cauchy**. Le paramètre à estimer par cette approche est celui de dispersion. Contrairement à la deuxième approche, le paramètre estimé ici est donné par une formule implicite, ou une équation homogène non linéaire en fonction de moments du signal reçu.

Les paramètres estimés par ces deux approches semi-aveugles sont utilisés par l'algorithme MAP pour calculer les informations extrinsèques des décodeurs constitutifs du décodeur itératif. L'avantage majeur de ces deux approches est qu'elles présentent de bonnes estimations des paramètres, ce qui permet aux décodeurs constitutifs de donner des informations extrinsèques plus fiables.

Nous verrons que la simulation numérique montre que l'utilisation de ces trois approches dans un décodeur itératif améliore d'une façon certaine et considérable ses performances en termes de taux d'erreurs binaire *BER*.

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

Les annexes sont divisées en deux : A et B. Dans l'annexe A, nous montrerons comment à partir de la simulation d'une variable aléatoire à distribution gamma nous pouvons simuler une variable aléatoire à GGD, puis nous rappellerons la technique de la simulation d'une variable aléatoire à distribution gamma. L'annexe B contenant les détails de la théorie des séquences pseudo-aléatoires, qui sont utilisées dans notre simulation pour simuler l'information binaire utile à envoyer.

Chapitre 1

Concepts de base sur les DCS

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 décrivons d'une façon non approfondie les différents éléments constituant d'un DCS.

1.2 Architecture générale d'un DCS

L'architecture générale d'un DCS est présentée dans la figure 1.1 [18]. 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 la 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 dépend de la conception appliquée.

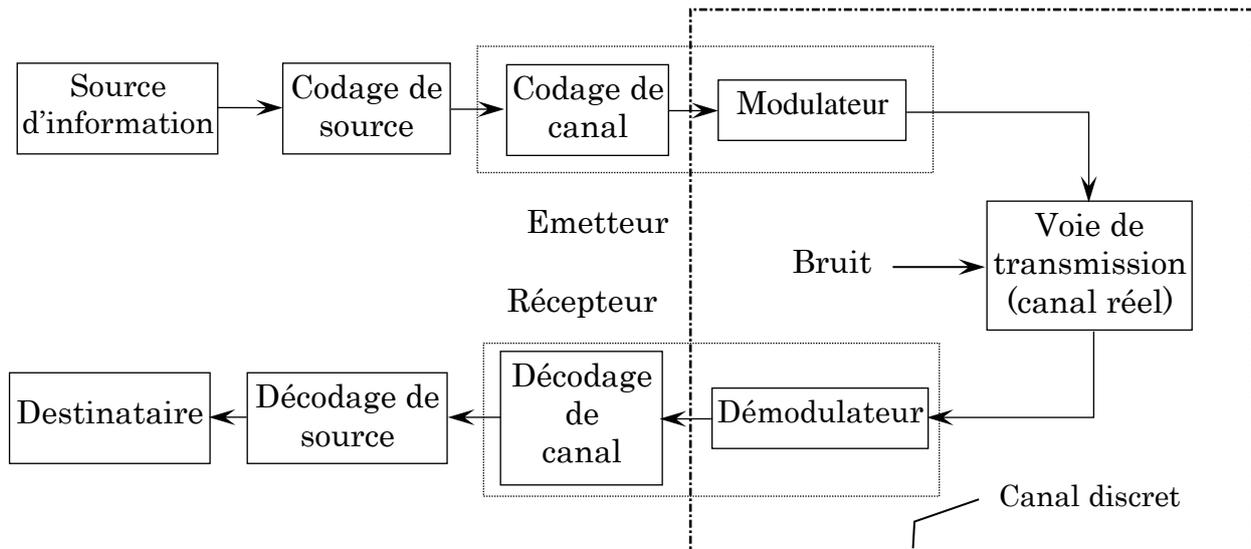


Fig. 1.1 Architecture générale d'un DCS.

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 [19].

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 et l'entropie, se mesurent 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 [19]

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

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.

Les bruits considérés dans ce qui suit sont le bruit gaussien et le bruit impulsif modélisé par un processus aléatoire soit à distribution gaussienne généralisée GGD symétrique soit à distribution α -stable symétrique $S\alpha S$, 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 [20]. De cette constatation, les principales caractéristiques du bruit impulsif 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 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).

Deux modèles de canal sont souvent adoptés selon que la sortie du canal sera prise après ou avant le circuit de décision du démodulateur et selon le type de bruit affectant le canal.

Dans le cas où l'entrée et la sortie du canal seront discrètes, alors nous parlerons de *canal discret* (*canal discret = modulateur + canal réel + démodulateur*). Dans le cas où l'entrée est discrète et la sortie est continue, nous parlerons de *canal analogique* ou *continu* (*canal continu = canal réel si le bruit est une variable aléatoire continue*) [21].

1.3.1 Canal discret

Un canal discret peut être défini par ses probabilités de transition p_{ij}^k où

$$p_{ij}^k = \Pr\{R_k = r_j / Y_k = y_i\}, \quad \sum_{j=1}^m p_{ij}^k = 1, \forall i \quad (1.1)$$

où Y_k et R_k représentent respectivement l'entrée et la sortie du canal à l'instant k et prennent leurs valeurs dans les ensembles : $[y_1, \dots, y_i, \dots, y_n]$ et $[r_1, \dots, r_j, \dots, r_m]$, p_{ij}^k est la probabilité d'avoir r_j en sortie lorsque y_i est introduit à l'entrée (est transmis) (voir la figure 1.2).

Lorsque les probabilités de transition sont indépendantes du temps, le canal est *stationnaire*. Si un symbole à la sortie du canal à l'instant $t = kT$ (où T est la période d'échantillonnage) ne dépend que du symbole à l'entrée du canal à l'instant $t = kT$, le canal est dit *sans mémoire*.

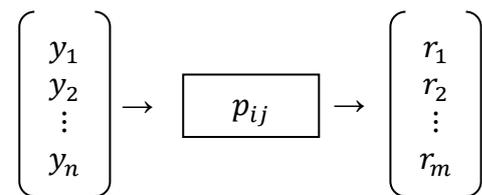


Fig. 1.2 Canal discret sans mémoire.

Lorsque les symboles à l'entrée et à la sortie du canal sont binaires et que les probabilités de transition sont symétriques, c'est-à-dire

$$p_{ij} = p_{ji}, \forall i, j \in \{0,1\} \quad (1.2)$$

alors le canal est appelé *canal binaire symétrique*, et il est entièrement défini par sa probabilité d'erreur p d'un bit.

$$p_{01} = p_{10} = p, \quad p_{00} = p_{11} = 1 - p \quad (1.3)$$

1.3.2 Canal continu

Il s'agit d'un canal à entrée binaire et sortie analogique. La sortie se représente par une variable aléatoire continue R_k

$$R_k = Y_k + W_k \quad (1.4)$$

où Y_k est le symbole binaire émis et W_k est une variable aléatoire continue correspond au bruit du canal de densité de probabilité f_{W_k} . Les densités de probabilité de transition du canal s'écrivent

$$P(R_k/Y_k \text{ émis}) = f_{W_k}(R_k - Y_k) \quad (1.5)$$

Si le canal est stationnaire et sans mémoire, les densités de probabilité sont indépendantes du temps et la sortie du canal à l'instant $t = kT$ (où T est la période d'échantillonnage, généralement prise égale à l'unité du temps) ne dépend que du symbole d'entrée à l'instant $t = kT$.

1.4 Capacité du canal

1.4.1 Canal discret stationnaire et sans mémoire

Pour définir la capacité d'un canal discret, nous introduisons la notion d'information mutuelle entre deux variables aléatoires. Considérons deux variables aléatoires dépendantes Y_k et R_k qui prennent, respectivement, leurs valeurs dans les ensembles suivants : $Y_k \in \{y_1, \dots, y_i, \dots, y_n\}$ et $R_k \in \{r_1, \dots, r_j, \dots, r_m\}$. Lorsqu'il y a une certaine dépendance entre y_i et r_j , la réalisation de l'événement $Y_k = y_i$ apporte une certaine information sur la réalisation de l'événement $R_k = r_j$ que nous pouvons la mesurer par la quantité [22]

$$i(y_i, r_j) = \log_2 \frac{P\{R_k = r_j / Y_k = y_i\}}{P\{R_k = r_j\}} \quad (1.6)$$

appelée information mutuelle entre y_i et r_j .

L'information mutuelle moyenne $I(Y_k, R_k)$ est définie comme la moyenne de la quantité $i(y_i, r_j)$ [22]

$$I(Y_k, R_k) = \sum_{i=1}^n \sum_{j=1}^m P\{Y_k = y_i, R_k = r_j\} \log_2 \frac{P\{R_k = r_j / Y_k = y_i\}}{P\{R_k = r_j\}} \quad (1.7)$$

L'information mutuelle moyenne est positive et s'annule lorsque les variables Y_k et R_k sont indépendantes. Nous pouvons montrer qu'elle peut s'exprimer en fonction de $H(Y_k)$ et de $H(Y_k/R_k)$ ou en fonction de $H(R_k)$ et de $H(R_k/Y_k)$ [22]

$$I(Y_k, R_k) = H(Y_k) - H(Y_k/R_k) = H(R_k) - H(R_k/Y_k) \quad (1.8)$$

$H(Y_k) = -\sum_{i=1}^n P(y_i) \log_2 P(y_i)$ est l'entropie de l'alphabet à l'entrée du canal ;

$H(Y_k/R_k) = -\sum_{i=1}^n \sum_{j=1}^m P(y_i, r_j) \log_2 P(y_i/r_j)$ est l'équivoque ;

$H(R_k) = -\sum_{j=1}^m P(r_j) \log_2 P(r_j)$ est l'entropie de l'alphabet à la sortie du canal ;

$H(R_k/Y_k) = -\sum_{j=1}^m \sum_{i=1}^n P(y_i, r_j) \log_2 P(r_j/y_i)$ est l'erreur moyenne.

La capacité C d'un canal discret est définie comme la plus grande quantité d'information mutuelle moyenne qu'il peut transmettre.

$$C = \max_{P(Y_k=y_i)} I(Y_k, R_k) \quad (1.9)$$

Cette capacité d'un canal s'exprime en bit par seconde. Le maximum d'information mutuelle est à prendre par rapport à la distribution de probabilité $\{P_1, \dots, P_n\}$ où $P_i = P(Y_k = y_i)$.

Pour un canal symétrique binaire, le maximum d'information mutuelle $I(Y_k, R_k)$ est obtenu lorsque les symboles 0 et 1 à l'entrée du canal sont équiprobables et sa capacité est égale à [22][23]

$$C = 1 + (1 - p) \cdot \log_2(1 - p) + p \cdot \log_2 p \quad (1.10)$$

où p est la probabilité d'avoir un 0 ou un 1.

1.4.2 Canal continu stationnaire et sans mémoire

Dans le cas d'un canal continu, l'entrée du canal Y_k n'est autre que la sortie du modulateur. Pour une source binaire et une modulation BPSK nous trouvons $Y_k \in \{-1, +1\}$.

Nous supposons que les deux symboles $Y_k = -1$ et $Y_k = +1$ sont équiprobables, et que la densité de probabilité $f_{W_k}(w)$ du bruit est paire (canal continu symétrique).

La capacité d'un canal continu, garde la même définition que celle d'un canal discret, sauf que la somme sur toutes les valeurs possible r_j de R_k est remplacée par une intégrale sur toutes les valeurs de R_k de $-\infty$ jusqu'à $+\infty$. Nous avons

$$C = \max_{P(Y_k=y_i)} I(Y_k, R_k) = H(R_k) - H(R_k/Y_k) \text{ avec } P(Y_k = -1) = P(Y_k = +1) = 1/2$$

(les symboles d'entrée du canal Y_k sont équiprobables). Alors

$$\begin{aligned} C &= \int_{-\infty}^{+\infty} \left(-P(R_k) \log_2 P(R_k) + \sum_{i=1}^2 P(y_i) P(R_k/y_i) \log_2 P(R_k/y_i) \right) dR_k \\ &= \int_{-\infty}^{+\infty} \frac{1}{2} \sum_{i=1}^2 P(R_k/y_i) \log_2 \frac{P(R_k/y_i)}{\frac{1}{2}(P(R_k/Y_k = -1) + P(R_k/Y_k = +1))} dR_k \\ &= \int_{-\infty}^{+\infty} P(R_k/Y_k = +1) \log_2 \frac{P(R_k/Y_k = +1)}{\frac{1}{2}(P(R_k/Y_k = -1) + P(R_k/Y_k = +1))} dR_k \\ &= \int_{-\infty}^{+\infty} f_{W_k}(R_k - 1) \log_2 \frac{f_{W_k}(R_k - 1)}{\frac{1}{2}(f_{W_k}(R_k + 1) + f_{W_k}(R_k - 1))} dR_k \end{aligned} \quad (1.11)$$

Nous posons

$$LLR_c(R_k) = \ln \left(\frac{f_{W_k}(R_k + 1)}{f_{W_k}(R_k - 1)} \right) \quad (1.12)$$

Ce rapport est appelé rapport logarithmique de vraisemblance LLR du canal (voir le paragraphe 3.3.1.2 du chapitre 3). La capacité s'écrit alors [7]

$$C = \int_{-\infty}^{+\infty} f_{W_k}(R_k - 1)(1 - \log_2(1 + e^{LLR_c(R_k)})) dR_k = 1 - E(\log_2(1 + e^{LLR_c(R_k)})) \quad (1.13)$$

où $E(\cdot)$ est l'espérance mathématique.

L'expression $LLR_c(R_k)$ est une variable aléatoire qui permet de calculer la capacité du canal directement sous la forme d'une moyenne, comme la montre l'équation (1.13). Cette variable aléatoire possède une densité de probabilité f_{LLR_c} qui peut être facilement estimée par la méthode de Monte Carlo. Par conséquent, afin d'estimer la capacité du canal, nous pouvons appliquer la méthode de Monte Carlo sur N échantillons de la variable aléatoire LLR_c . La capacité estimée est donnée par [7]

$$C \cong 1 - \frac{1}{N} \sum_{n=1}^N \log_2(1 + e^{LLR_{cn}}) \quad (1.14)$$

La relation (1.14) (ou 1.13) montre que la capacité du canal dépend étroitement des paramètres du bruit envisagé, car elle dépend de la variable aléatoire LLR_c qui dépend à son tour de la densité de probabilité du bruit.

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 distributions gaussiennes généralisées GGD et α -stables, qui sont deux distributions non-gaussiennes, sont deux alternatives toute naturelle car elles sont une généralisation de la loi normale et prennent en compte des queues lourdes [7][24][25].

Par conséquent, tout au long de ce travail, le bruit impulsif est modélisé par deux distributions symétriques : gaussienne généralisée et α -stable SaS de **Cauchy** où $\alpha = 1$ (ou SaS de **Cauchy**) (voir le chapitre 2). Dans ces conditions, nous pouvons évaluer la capacité d'un canal continu pour les trois cas suivant

- Pour un bruit gaussien généralisé symétrique de paramètre de forme α et de variance σ^2 , nous trouvons

$$LLR_c(R_k) = \left(\sigma^2 \frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)} \right)^{-\alpha/2} \cdot (|R_k - 1|^\alpha - |R_k + 1|^\alpha) \quad (1.15)$$

et la capacité d'un canal gaussien généralisé symétrique est donc

$$C_{GGD} = 1 - E \left(\log_2 \left(1 + e^{\left(\sigma^2 \frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)} \right)^{-\alpha/2} \cdot (|R_k - 1|^\alpha - |R_k + 1|^\alpha)} \right) \right) \quad (1.16)$$

- Pour un bruit gaussien de moyenne nulle et de variance σ^2 , il suffit de mettre $\alpha = 2$ dans l'équation (1.15) pour trouver le $LLR_c(R_k)$ du canal gaussien, car la distribution gaussienne est un cas particulier de la GGD.

$$LLR_c(R_k) = -\frac{2}{\sigma^2} R_k \quad (1.17)$$

et la capacité d'un canal gaussien est alors

$$C_{AWGN} = 1 - E \left(\log_2 \left(1 + e^{-\frac{2}{\sigma^2} R_k} \right) \right) \quad (1.18)$$

- Pour un bruit impulsif $S\alpha S$ de **Cauchy** de paramètre de dispersion γ , nous obtenons

$$LLR_c(R_k) = \ln \left(\frac{(R_k - 1)^2 + \gamma^2}{(R_k + 1)^2 + \gamma^2} \right) \quad (1.19)$$

et la capacité d'un canal $S\alpha S$ de **Cauchy** s'écrit

$$C_{cauchy} = 1 - E \left(\log_2 \left(1 + \frac{(R_k - 1)^2 + \gamma^2}{(R_k + 1)^2 + \gamma^2} \right) \right) \quad (1.20)$$

Une analyse grossière des formules déjà obtenues, montre que pour un canal gaussien généralisé (y compris le canal gaussien), la capacité se dégrade au fur et à mesure de l'augmentation de la variance ou de l'impulsivité du bruit ($\alpha \rightarrow 0$) [26]. Pour un bruit $S\alpha S$ de **Cauchy**, la capacité se dégrade aussi quand la valeur de la dispersion γ augmente [7].

1.5 Théorèmes fondamentaux sur le codage du canal

1.5.1 Théorème de codage de Shannon

Il est possible, en utilisant un codage de canal approprié, de transmettre de l'information avec un débit \mathcal{R} sous forme d'une suite de symboles discrets avec une probabilité d'erreur aussi faible que l'on veut si \mathcal{R} est inférieur à la capacité du canal C .

Le théorème de **Shannon** affirme l'existence de codes dont la probabilité d'erreur est arbitrairement faible mais ne montre pas comment ces codes peuvent être construits.

Ce théorème affirme une chose toute à fait surprenante à savoir que, quel que soit le niveau des perturbations d'un canal, nous pouvons toujours y passer des messages codés d'une manière appropriée avec une probabilité d'erreur aussi faible que l'on veut, c'est la raison pour laquelle ce théorème était la cause de l'énorme développement de la théorie des codes.

En pratique, dans tous les cas où $\mathcal{R} < (1/2)C$, il existe des codes qui réalisent une probabilité d'erreur très faible, où \mathcal{R} représente le débit d'information et C est la capacité du canal [22].

1.5.2 Théorème de Nyquist [22]

Le théorème de **Nyquist** affirme qu'à travers un canal équivalent à un filtre passe-bas idéal avec une fréquence de coupure B , il est possible de transmettre des signaux binaires (impulsions) indépendants avec un débit de moments $\mathcal{R}_s \leq 2B$ impulsions /seconde sans interférences entre symboles, où \mathcal{R}_s est donné par définition

$$\mathcal{R}_s = \frac{\text{nombre d'impulsions dans l'intervalle } T}{T}$$

1.6 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* » exclusifs.

Il existe un type spécial de code en blocs linéaires appelé codes cycliques. Ce sont des codes en blocs linéaires vérifiant quelques propriétés supplémentaires [23][27].

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 [27].

Un turbo code classique résulte de l'association de deux (ou plus) codeurs, en série ou en parallèle (voir figures 1.3.a et 1.3.b respectivement). Il s'agit souvent de codeurs convolutifs récurrents systématiques RSC car leur récursivité apporte des propriétés pseudo-aléatoires intéressantes [28].

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 » [28].

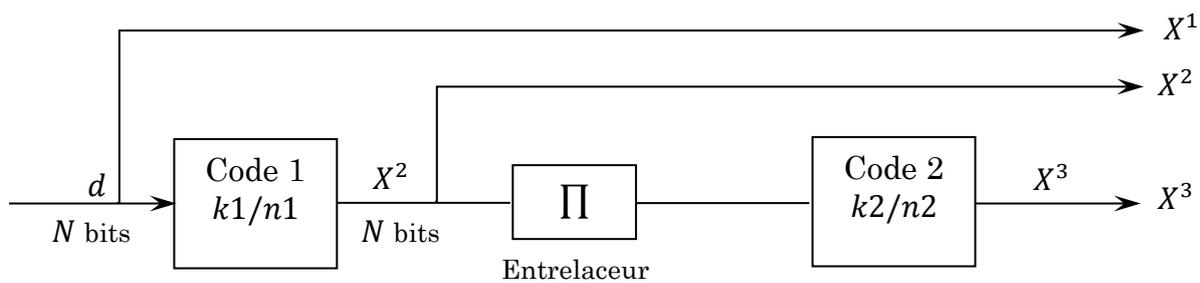


Fig. 1.3.a Turbo code série.

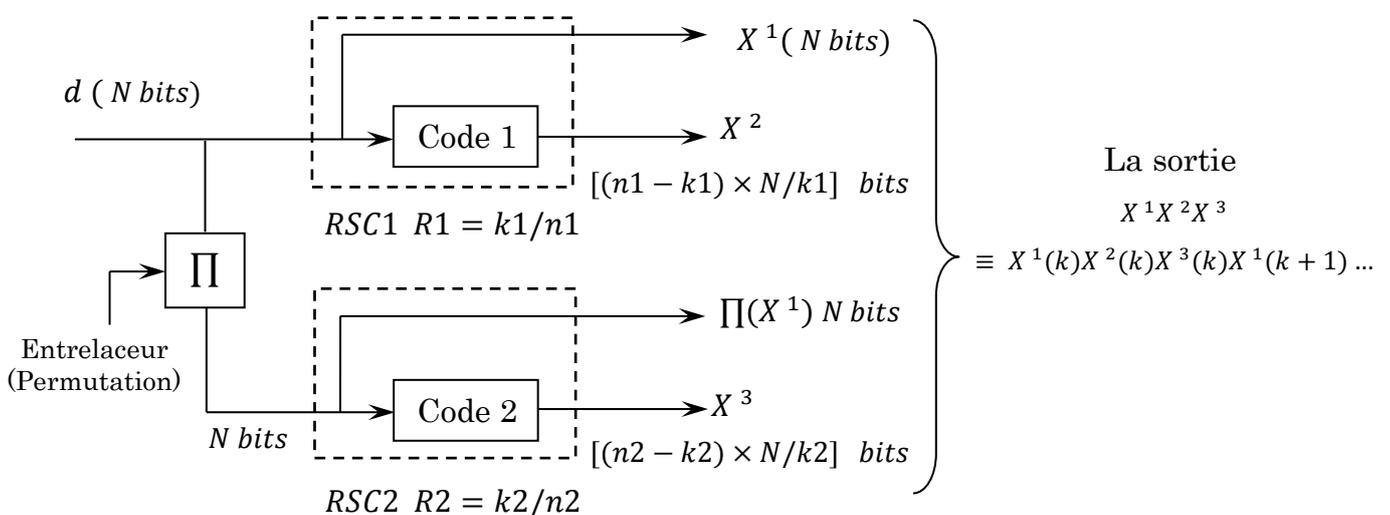


Fig. 1.3.b Turbo code parallèle.

1.6.1 Les turbo codes convolutifs CTC parallèles

L'idée primordiale du CTC parallèle est d'utiliser deux codes convolutifs systématiques récurrents RSC, généralement identiques ($RSC1 \equiv RSC2$), concaténés en parallèle via un entrelaceur entre les deux (figure 1.3.b).

Le but de l'entrelaceur 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 [29].

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 [30]. Cette opération est appelée fermeture du treillis.

Le turbo code parallèle est alors un code linéaire binaire qui donne pour une séquence d'entrée à N bits une séquence de sortie à J bits (figure 1.3.b).

D'après la figure 1.3.b, les N bits d'information d sont codés par le codeur RSC1, entrelacés à travers Π et ensuite codés par le codeur RSC2. Chaque code RSC génère $(n_i - k_i) \cdot N/k_i$ bits de parité ou de contrôle noté X^i . Les bits d'information $d = X^1$ et $\Pi(X^1)$ étant les mêmes à une permutation près, nous ne transmettrons pas l'entrée du code RSC2 afin d'augmenter le rendement du CTC.

La longueur de la fenêtre de terminaison du treillis est $T = (L - 1)/K$ branches, où L est la longueur de contrainte du RSC et K est le nombre des sorties de parité d'un RSC. Si en négligeant la fermeture des deux treillis, le rendement R du CTC s'obtient par l'expression simple

$$R = \frac{N}{J} = \frac{R1R2}{R1 + R2 - R1R2} = \frac{R1}{2 - R1} \text{ pour } R1 = R2 \quad (1.21)$$

La figure 1.4.a illustre un code convolutif RSC de rendement 1/2, très employés par **C. Berrou** [2], de polynôme générateur (5, 7) à 4 états. Son diagramme en treillis est montré dans la figure 1.4.b. Le CTC associé à ce RSC est illustré dans la figure 1.4.c. Son rendement est calculé à partir de la relation (1.21) avec $R1 = R2 = 1/2$, c'est-à-dire son rendement est égal à 1/3.

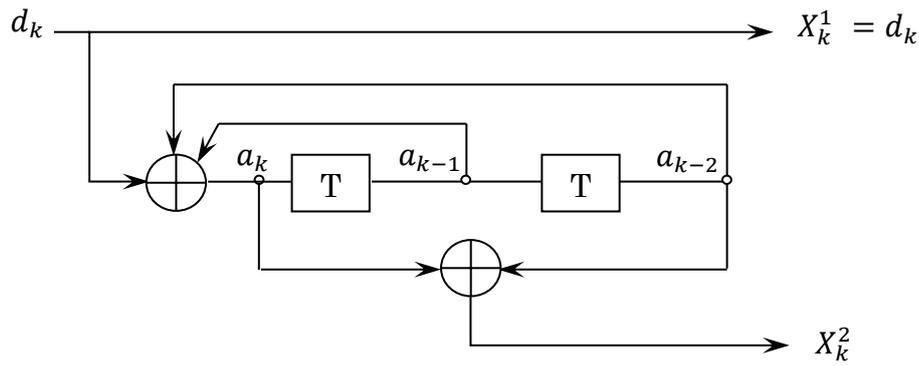


Fig. 1.4.a Code convolutif récursif systématique (5,7).

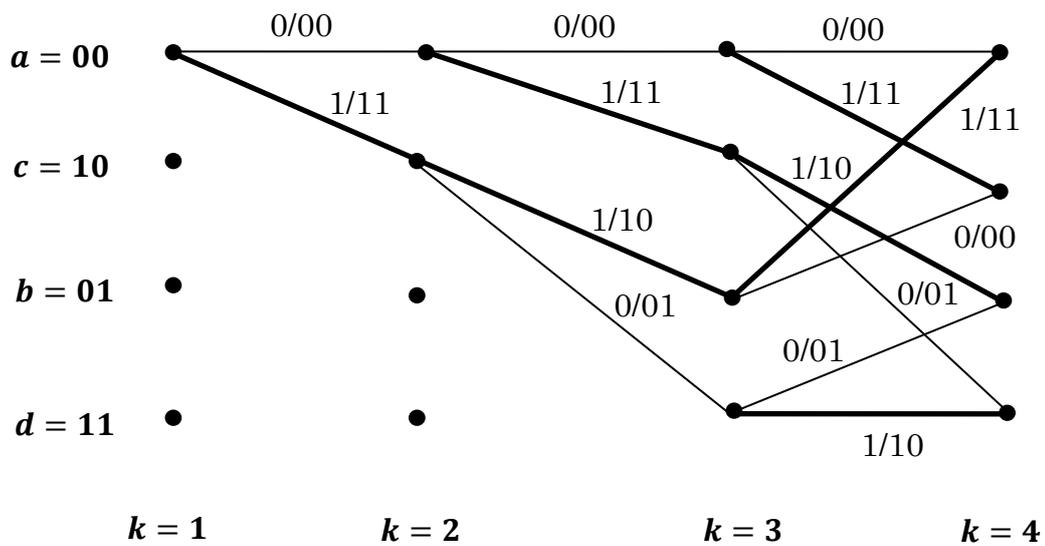


Fig. 1.4.b Représentation en treillis du RSC de la figure 1.4.a.

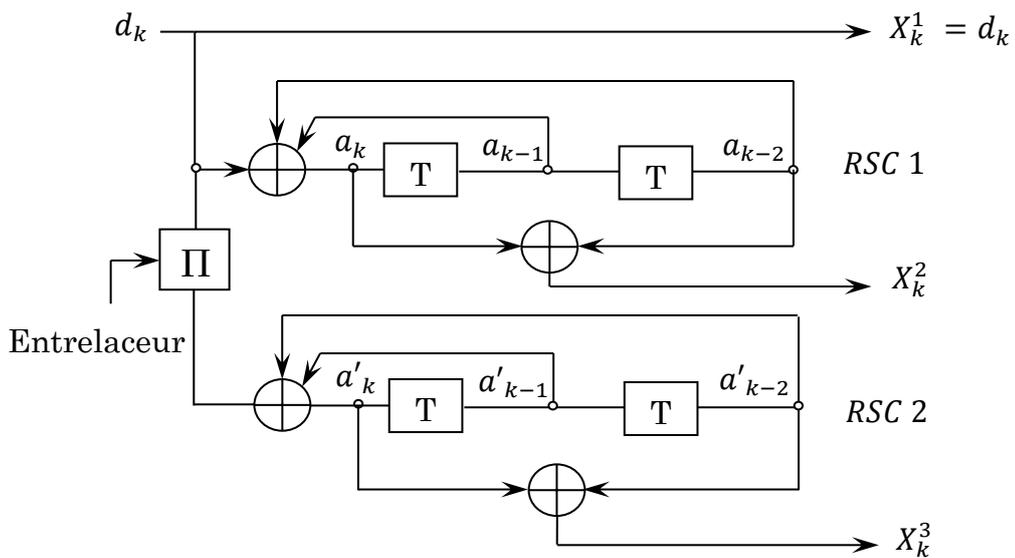


Fig. 1.4.c Turbo code convolutif parallèle (5,7).

Le décodeur associé à un CTC est un décodeur doux ou souple, car ce type de décodeurs est très performant qu'un décodeur dur ou ferme et très facile à mettre en œuvre pour tous les codes acceptant des représentations en treillis, par la raison qu'il utilise un algorithme de décodage à entrée-sortie souple SISO.

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, nous parlons 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.

1.6.2 Types d'entrelaceurs

1.6.2.1 Entrelaceur par blocs

Ce type d'entrelaceurs est utilisé plus couramment dans les systèmes de communication [31]. En particulier, **Ramsey** [31] 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 1.5.

L'ordre des éléments ligne par ligne de la matrice de la figure 1.5 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,

$$i - 1 = (k - 1) \cdot n + (l - 1), \quad 1 \leq k \leq m, \quad 0 \leq l - 1 \leq n - 1 \quad (1.22)$$

ce qui donne

$$l - 1 = (i - 1) \bmod [n] \quad (1.22.a)$$

$$k - 1 = \lfloor (i - 1) / n \rfloor \quad (1.22.b)$$

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 (1.22.a) et (1.22.b) dans cette dernière équation nous trouvons

$$\pi(i) = ((i - 1) \bmod [n]) \cdot m + \left\lfloor \frac{i - 1}{n} \right\rfloor + 1, \quad 1 \leq i \leq m \cdot n \quad (1.23)$$

La fonction de désentrelacement est

$$\pi^{-1}(i) = ((i - 1) \bmod [m]) \cdot n + \left\lfloor \frac{i - 1}{m} \right\rfloor + 1, \quad 1 \leq i \leq m \cdot n \quad (1.24)$$

Dans le cas d'un entrelaceur à matrice carrée ($m = n$) nous obtenons $\pi(i) = \pi^{-1}(i)$. Afin de réordonner le bloc entrelacé, nous effectuons l'opération inverse. Ce type d'entrelacement est efficace quand le vecteur pénalisant se situe dans une seule colonne.

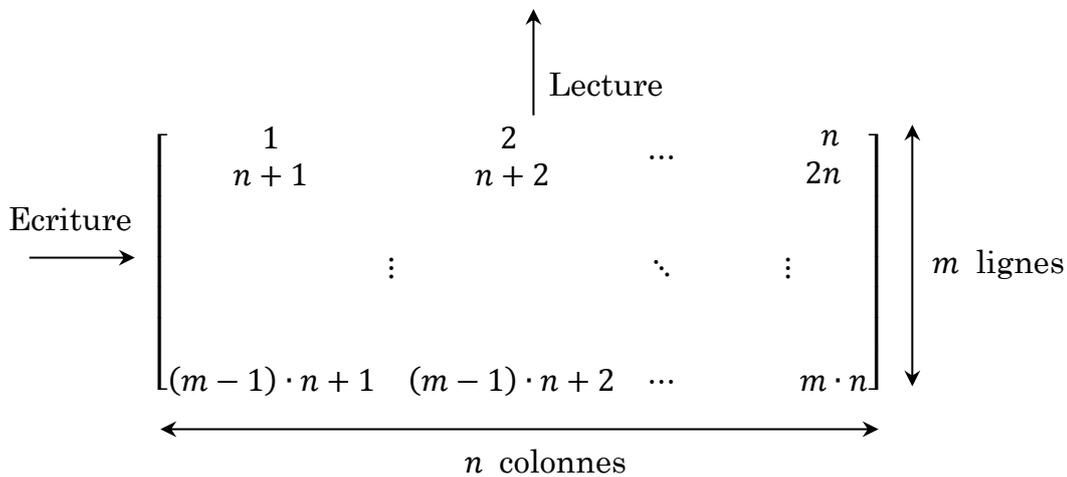


Fig. 1.5 Ecriture-Lecture dans un entrelaceur par blocs.

1.6.2.2 Entrelaceur hélicoïdal

Ce type d'entrelaceurs, qui est une version modifiée de l'entrelaceur par blocs, a été proposé par **Barbulescu** [32]. 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

$$\begin{cases} \pi(i) = i_r \cdot n + j_r, & 1 \leq i \leq m \cdot n \\ j_r = (i - 1) \bmod [n] + 1 \\ i_r = \left(m - \left\lfloor \frac{i-1}{n} \right\rfloor - j_r \right) \bmod [m] \end{cases} \quad (1.25)$$

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

$$\begin{cases} \pi(i) = (m \cdot n - i - (n - 2)j_r - 1) \bmod [m \cdot n] + 1, & 1 \leq i \leq m \cdot n \\ j_r = (i - 1) \bmod [n] + 1 \end{cases} \quad (1.26)$$

Nous pouvons vérifier facilement que la fonction de désentrelacement de cet entrelaceur est $\pi^{-1}(i) = \pi(i)$. La figure 1.6 montre la fonction d'entrelacement hélicoïdale pour $m = 5$ et $n = 4$.

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

Fig. 1.6 Entrelacement hélicoïdale pour $m = 5$ et $n = 4$.

1.6.2.3 Entrelaceur convolutif

L'entrelaceur convolutif a été enchâssé par Ramsey [31]. Il est constitué par des amas des registres à décalages et par deux commutateurs comme l'indique la figure 1.7 [33].

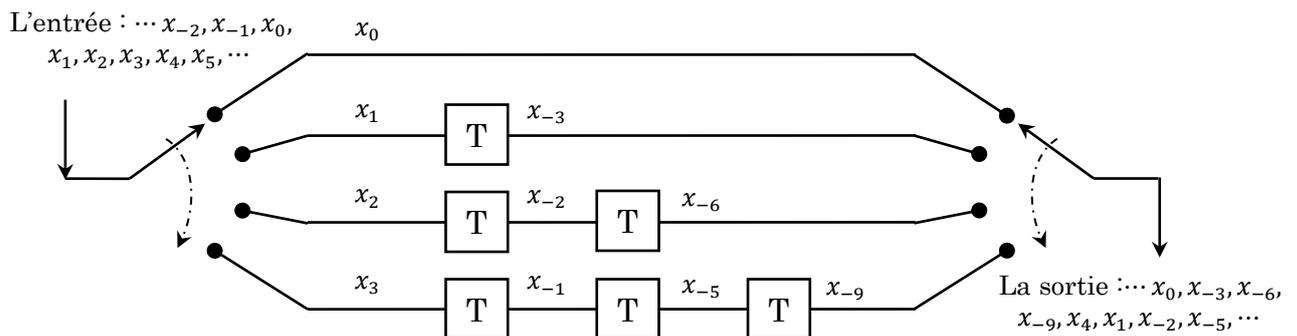


Fig. 1.7 Schéma de principe d'un entrelaceur convolutif avec $K = 4$ et $B = 1$.

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 trouve 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 [34]. La fonction d'entrelacement convolutif est donnée par [35]

$$\pi(i) = i - [(i - 1) \bmod [K]] \cdot K \cdot B \quad (1.27)$$

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

$$\pi^{-1}(i) = i + [(i - 1) \bmod [K]] \cdot K \cdot B \quad (1.28)$$

1.6.2.4 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

$$\pi(i) = (a \cdot i + s) \bmod [L] \quad (1.29)$$

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 [36]. a et s sont toujours inférieurs à L . a est premier avec L .

Dans le cas où $\sqrt{2L}$ est un entier, nous préconisons de prendre la valeur $a = \sqrt{2L} - 1$ [36]. 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 $a = \sqrt{2L} - 1$ nous pouvons prendre $\beta = L - (a + 2) = L - (\sqrt{2L} + 1)$.

La figure 1.8 montre un entrelacement à décalage circulaire avec $L = 8$, $a = 3$, et $s = 0$.

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

Fig. 1.8 Entrelacement à décalage circulaire pour $L = 8$, $a = 3$ et $s = 0$.

1.6.2.5 Entrelaceur aléatoire

Dans la pratique nous essayons 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) [37].

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

Tab. 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 $i \in \{1, 2, \dots, M\}$ nous avons $1 \leq j \leq N$.

L'entrelacement (ou la permutation) est obtenu en appliquant la formule (1.30).

$$\pi(k) = i_r \cdot N + j_r \quad (1.30)$$

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

$$\pi^{-1}(k) = i \cdot N + j \quad (1.31)$$

avec

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

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

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

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

où α et β sont deux entiers naturels tels que

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

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

$$v = \beta \cdot j_r \pmod{[8]}$$

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 (1.32) pour tout i tel que $0 \leq i \leq M \cdot N - 1$. Ici, le tableau 1.2 de nombres premiers avec 8 est utilisé [37].

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

Tab. 1.2 Tableau de nombres premiers avec 8.

$$\pi(i) = c(i) \cdot N + 2 \cdot r(i) - m(i) + 1 \quad (1.32)$$

avec

$$m(i) = i \pmod{[2]}$$

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

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

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

$$v = r(i) \pmod{[8]}$$

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

Cet entrelaceur a été proposé pour l'utiliser dans un standard CCSDS [37]. Nous pouvons démontrer que la fonction de désentrelacement est donnée par

$$\pi^{-1}(i) = 2 \cdot (r_0 \cdot M + c_0) + m \quad (1.33)$$

avec

$$m = (i - 1) \pmod{[2]}$$

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

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

$$v = r \pmod{[8]}$$

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

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

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

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

1.6.2.6 Entrelaceur 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 [38]. 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 [39], parce qu'il est classé parmi les entrelaceurs optimaux [40]. Pour cette raison, nous allons le présenter avec détaille.

Pratiquement, toutes les techniques de conception d'entrelacement des turbo codes dans la littérature sont basées sur l'algorithme de génération d'entrelacement S –aléatoire proposé par l'équipe du JPL [41]. 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 selon 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 [42][43]

$$\forall i, j: 1 \leq i, j \leq N, 0 < |i - j| \leq S \Rightarrow |\pi(i) - \pi(j)| \geq S \quad (1.34)$$

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 $S = 1$ nous retrouvons 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 [35][36].

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 1.9 représente l'entrelaceur S –aléatoire pour un bloc de longueur 12, c'est-à-dire $S = 2$.

$$M = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11]$$

$$\Downarrow$$

$$\pi(M) = [5 \ 2 \ 7 \ 9 \ 4 \ 11 \ 1 \ 6 \ 10 \ 8 \ 0 \ 4]$$

Fig. 1.9 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 nous choisissons $S < \sqrt{N/2}$. Avec la relation (1.34), 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. [44] a ajouté une condition supplémentaire dans la construction d'un entrelaceur S –aléatoire, qui est

$$\forall i : 1 \leq i \leq N, \quad |i - \pi(i)| \geq S' \quad (1.35)$$

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 (1.34) et (1.35) 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 .

L'entrelaceur S –aléatoire est le plus performant pour les turbo codes si sa taille est moyenne ou grande [38]. Pour ce motif, nous allons utiliser ce type d'entrelaceur dans la partie de simulation (voir le chapitre 4). Les turbo codes à utiliser dans cette partie sont les CTC parallèles.

1.7 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 translater) 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, nous pouvons distinguer dans la modulation analogique discrète [27]

- 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)
- 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 bruité, 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.

La BPSK est la forme la plus simple du PSK. Elle utilise deux phases qui sont séparées de 180° ; nous l'appelons 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, nous ne pouvons moduler qu'un seul bit par symbole (voir la figure 1.10), ce qui est un inconvénient pour les applications qui nécessitent un débit binaire élevé.

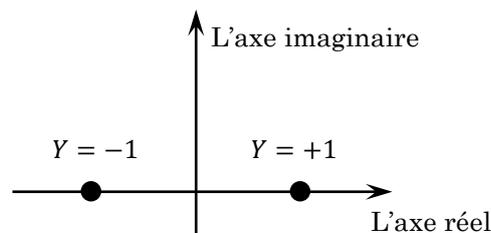


Fig. 1.10 Diagramme de constellation pour la BPSK.

Soit X la sortie du turbo code (qui est l'entrée du modulateur). La modulation BPSK d'un bit X , avec une énergie de bit E_b , est alors

$$Y = \begin{cases} \sqrt{E_b} & \text{pour } X = 0 \\ -\sqrt{E_b} & \text{pour } X = 1 \end{cases} \quad (1.36)$$

Sans restreindre la généralité, nous pouvons prendre $\sqrt{E_b} = 1$. La modulation devient tout simplement

$$Y = \begin{cases} +1 & \text{pour } X = 0 \\ -1 & \text{pour } X = 1 \end{cases} = 1 - 2X = (-1)^X \quad (1.37)$$

La probabilité d'erreur binaire P_e de la BPSK dans un canal bruité est la probabilité de décider qu'un 1 avait été transmis alors que c'était un 0 ou inversement. Si le signal reçu observé à l'instant k est $R_k = Y_k + W_k$, nous écrivons mathématiquement

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

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

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

Finalement nous obtenons

$$P_e = \frac{1}{2} - P(0 \leq W_k \leq 1) = \frac{1}{2} - \int_0^1 f_W(w) dw \quad (1.38)$$

Dans le cas où d'un bruit gaussien généralisé symétrique, de paramètre de forme α et de variance σ^2 (voir le paragraphe 2.2.1 du chapitre 2), nous trouvons

$$P_e = \frac{1}{2} - \int_0^1 \frac{\alpha \cdot (\Gamma(3/\alpha))^{1/2} \cdot \sqrt{SNR}}{\sqrt{2} \cdot (\Gamma(1/\alpha))^{3/2}} \exp\left\{-\left(\sqrt{2} \cdot \left(\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}\right)^{1/2} \cdot \sqrt{SNR} \cdot x\right)^\alpha\right\} dx \quad (1.39)$$

où $SNR = \frac{1}{2\sigma^2}$. La figure 1.11 illustre la variation de P_e de l'équation (1.39) en fonction de SNR en dB.

Pour un bruit $S\alpha S$ de **Cauchy** symétrique (voir le paragraphe 2.3.1 du chapitre 2) de paramètre de dispersion γ , nous trouvons

$$P_e = \frac{1}{2} - \int_0^1 \frac{\gamma}{\pi(x^2 + \gamma^2)} dx = \frac{1}{2} - \frac{1}{\pi} \arctan\left(\frac{1}{\gamma}\right) \quad (1.40)$$

La figure 1.12 montre la variation de P_e de l'équation (1.40) en fonction du rapport $1/\gamma$ en dB.

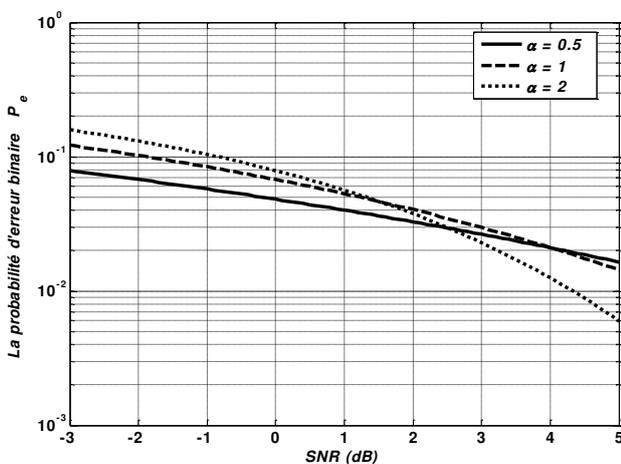


Fig. 1.11 La probabilité d'erreur P_e en fonction du SNR pour un bruit GG.

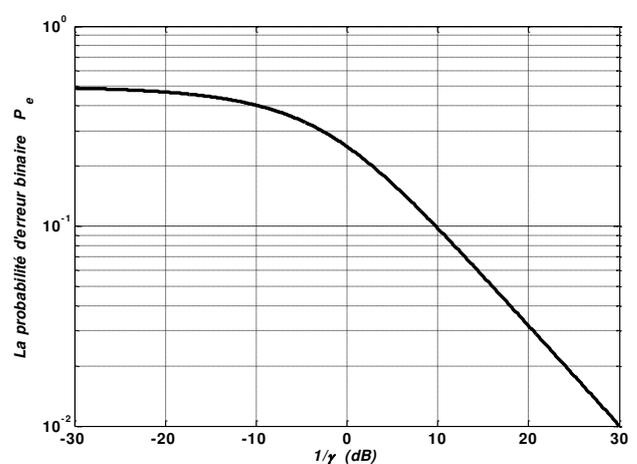


Fig. 1.12 La probabilité d'erreur P_e en fonction de $1/\gamma$ pour un bruit $S\alpha S$ de Cauchy.

1.8 Conclusion

Nous avons présenté dans ce chapitre quelques concepts de base sur les DCS et leurs modules constitutants, en particulier le canal et les fonctions du codage et de la modulation. Au fur et à mesure de la présentation de ces modules nous avons 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 -aléatoires, les bruits impulsifs à GGD symétrique et à distribution $S\alpha S$ de **Cauchy**, la modulation BPSK.

Toutefois dans cette présentation, nous avons essayé de donner l'essentielle de ce thème pour ne pas être exhaustif. Dans le paragraphe 1.4.2 nous avons calculé la capacité du canal pour trois canaux continus : canal gaussien, canal GGD et canal $S\alpha S$ de **Cauchy**. Nous avons vu que la capacité du canal est fortement dépend des paramètres du bruit envisagé. Nous avons constaté que cette capacité se dégrade si le bruit est fort ou très impulsif. Dans la section 1.6.2, nous avons présenté les différents types d'entrelacement généralement utilisés. Pour chaque fonction d'entrelacement nous avons 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 1.7, nous avons calculé et tracé la probabilité d'erreur de la modulation BPSK pour un signal numérique non codé, dans les cas des bruits gaussien, gaussien généralisé et $S\alpha S$ de **Cauchy**. Ceci est très utile si nous voulons faire des comparaisons 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 gaussiennes généralisées ou sur les distributions α -stables [6][45]. Les bruits qui suivent ces deux types de distribution font l'objet du chapitre suivant.

Chapitre 2

Distributions des bruits non-gaussiens : GG et α – stables

2.1 Introduction

En traitement du signal, le bruit est une composante inéluctable dont il faut tenir compte qu'il s'agisse de l'analyse, de la modélisation, de l'identification ou de la détection. Les méthodes d'analyse et les algorithmes associés supposent que le bruit est modélisé sous la forme simple d'un bruit blanc gaussien dont les caractéristiques statistiques et spectrales sont parfaitement connues. Or la réalité est fort éloignée de cette hypothèse de travail, et en particulier, nous rencontrons souvent du bruit non-gaussien de type impulsif. Ce bruit se présente généralement sous forme d'impulsions élémentaires de forte amplitude, superposées au signal utile.

L'étude de l'effet des bruits impulsifs sur les DCS fait un axe de recherche d'actualité, surtout lorsqu'il s'agit des systèmes à OFDM comme les systèmes filaires, y compris les lignes d'abonnés numériques DSL, les communications par courants porteurs en ligne PLC, et les systèmes sans fil standards par exemple, IEEE 802.11a/g/n/ac, WiMax (IEEE 802.16) et 3GPP LTE [46][47][48][49]. C'est pour cette raison que nous nous sommes intéressés aussi à l'étude de ce type de bruit qui apparaît dans ces DCS.

Ce chapitre s'est axé sur l'étude de deux modèles du bruit non-gaussiens qui permettent une représentation précise et fiable des phénomènes impulsifs. Ces deux modèles sont très étudiés dans la littérature. Le premier modèle non-gaussien est caractérisé par une variance finie à savoir le gaussien généralisé GG. Le deuxième est le modèle α – stable qui possédait toujours une variance infinie, sauf dans le cas où $\alpha = 2$.

Ce chapitre se divise en deux grandes parties. La première partie présente les distributions gaussiennes généralisées GGD et la deuxième partie introduit les distributions α – stables.

Il est bon de signaler qu'un problème de terminologie se pose : qu'est-ce que le « non-gaussien » ? Il est difficile d'échapper à une définition négative : le non-gaussien est tout ce qui n'est pas gaussien ! D'une manière générale, nous pouvons considérer qu'un modèle non-gaussien est un modèle où l'hypothèse de « gaussianité » de la loi du bruit n'est pas faite a priori, même si, en fin de compte, nous retrouvons la loi gaussienne comme dans les cas de GGD et α –stable.

2.2 Les distributions gaussiennes généralisées GGD

La famille paramétrique de distributions gaussiennes généralisées GGD est utilisée dans la science pour modéliser privilégiement certains phénomènes physiques et en ingénierie, par exemple, dans le domaine du traitement du signal et du codage audio et vidéo (voir, par exemple, [50][51]).

Certaines classes spéciales importantes des distributions appartiennent à cette famille telles que, entre autres, la distribution gaussienne et la distribution de **Laplace**. En outre, pour de valeurs du paramètre de forme dans une certaine fourchette, ce qui est d'intérêt dans de nombreuses applications pratiques, la distribution acquiert un caractère impulsif et présente des queues lourdes.

Dans la plupart des situations pratiques l'usage d'une simulation rapide et précise peut jouer un rôle important. Dans cette partie, nous analysons quelques techniques pour générer une distribution gaussienne généralisée.

En particulier, nous examinons les cas $\alpha = 1/3$ et $\alpha = 1/2$. Pour de telles valeurs du paramètre de forme, les différentes méthodes de simulation sont évaluées. Dans le cas particulier $\alpha = 1/2$, nous comparons l'efficacité de quatre techniques de simulation différentes. Les résultats numériques mettent en évidence que la technique basée sur la fonction de distribution cumulative inverse écrite en termes de la fonction de **Lambert W** est la plus efficace.

La fonction de **Lambert W**, également connu sous le nom la fonction « Omega », est une fonction à valeurs multiples complexe définie comme l'inverse de la fonction $f(w) = we^w$. Elle a de nombreuses applications en mathématiques pures et appliquées [52]. Nous examinons brièvement les principales propriétés de cette fonction spéciale.

2.2.1 La densité gaussienne généralisée

La fonction de densité de probabilité PDF d'un bruit gaussien généralisé X , avec moyenne μ et variance σ^2 , est définie comme [53]

$$f_X(x; \mu, \sigma, \alpha) = \frac{\alpha A(\alpha, \sigma)}{2 \Gamma(1/\alpha)} \exp(-(A(\alpha, \sigma)|x - \mu|)^\alpha) \quad x \in \mathbb{R}, \quad (2.1)$$

où

$$A(\alpha, \sigma) = \frac{1}{\sigma} \left[\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)} \right]^{1/2} \quad (2.2)$$

et

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt \quad z > 0 \quad (2.3)$$

est la fonction gamma complète.

La GGD est symétrique par rapport à $x = \mu$. $A(\alpha, \sigma)$ est un facteur d'échelle qui définit la dispersion de la distribution, d'où il est une mesure généralisée de la variance. $\alpha > 0$ est le paramètre de forme qui décrit le taux exponentiel de la décroissance : queues lourdes correspondent à des valeurs plus petites de α .

Il faut noter que α est appelé aussi l'exposant de la loi GG ou le coefficient de gaussianité. Si $\alpha < 2$, nous avons une loi sous-gaussienne et si $\alpha > 2$ une loi sur-gaussienne [8].

La famille du gaussien généralisé comprend une variété de distributions. Certaines classes bien connues de distributions sont générées par un paramétrage de la décroissance exponentielle de la GGD. Lorsque $\alpha = 1$, la GGD correspond à une distribution de **Laplace** (ou double exponentielle). Pour $\alpha = 2$, nous avons distribution gaussienne. Quand $\alpha \rightarrow +\infty$ la GGD converge vers une distribution uniforme dans $[\mu - \sqrt{3}\sigma, \mu + \sqrt{3}\sigma]$, tandis que lorsque $\alpha \rightarrow 0^+$ nous avons une fonction de probabilité impulsive à $x = \mu$.

Tous les moments centraux impairs de la GGD définie par l'équation (2.1) sont égaux à zéro, $E(|X - \mu|^r) = 0$ ($r = 1, 3, 5, \dots$), et les moments centraux pairs sont [53][8]

$$E(|X - \mu|^r) = \left[\frac{\sigma^2 \Gamma(1/\alpha)}{\Gamma(3/\alpha)} \right]^{r/2} \frac{\Gamma((r+1)/\alpha)}{\Gamma(1/\alpha)} \quad r = 2, 4, 6, \dots \quad (2.4)$$

La relation (2.4) reste valable pour tout nombre réel $r > -1$ différent des entiers impairs [8].

Avec une normalisation simple et certaines réductions de (2.1), nous obtenons un bruit GG de moyenne nulle et de variance unité ayant comme densité la fonction suivante

$$f_X(x; \alpha) = \frac{\alpha A(\alpha)}{2 \Gamma(1/\alpha)} \exp(-(A(\alpha)|x|)^\alpha) \quad x \in \mathbb{R}, \quad (2.5)$$

où $A(\alpha) = A(\alpha, 1)$.

Dans ce qui suit, nous limitons notre attention à des bruits gaussiens généralisés dont la densité est donnée dans (2.5). Pour $0 < \alpha < 2$ la densité (2.5) est convenable pour modéliser plusieurs phénomènes physiques et financiers à queues lourdes [53]. En outre, il est intéressant de noter que, tous les moments sont finis (ce qui n'est pas le cas pour d'autres queues lourdes, comme par exemple, les densités α – stables).

Le kurtosis \mathcal{K}_X d'un bruit X est le rapport entre le moment d'ordre 4 et le moment d'ordre 2 à la puissance 2 du bruit X [8]

$$\mathcal{K}_X = \frac{E(X^4)}{(E(X^2))^2}$$

Le kurtosis réalise une comparaison de la PDF du bruit X par rapport à la loi gaussienne de même variance qu'il possède un kurtosis égal à 3

- Si la densité de probabilité de X tend vers 0 à l'infini plus rapidement que la loi gaussienne, le bruit dit sous-gaussienne et le kurtosis est supérieur à 3 ;
- Si la densité de probabilité de X tend vers 0 à l'infini moins vite que la loi gaussienne, le bruit est dit sur-gaussienne et le kurtosis est à inférieur 3 ;
- Si la densité de probabilité de X est gaussienne, le kurtosis est égal à 3.

Le kurtosis caractérise l'aplatissement de la PDF par rapport à celui d'une distribution gaussienne.

Le kurtosis de la distribution (2.5) est

$$\mathcal{K}_X(\alpha) = \frac{\Gamma(1/\alpha)\Gamma(5/\alpha)}{[\Gamma(3/\alpha)]^2} \quad (2.6)$$

$\mathcal{K}_X(\alpha)$ diminue avec α , et nous pouvons constater que [54]

$$\lim_{\alpha \rightarrow 0^+} \mathcal{K}_X(\alpha) = +\infty \quad \lim_{\alpha \rightarrow +\infty} \mathcal{K}_X(\alpha) = 1.8 \quad (2.7)$$

La figure 2.1 montre la densité d'une GGD, à moyenne nulle et de variance unité, pour différentes valeurs du paramètre α . La figure 2.2 montre le comportement du kurtosis pour $\alpha \in [0, 2]$ et $\alpha \in [0, 20]$.

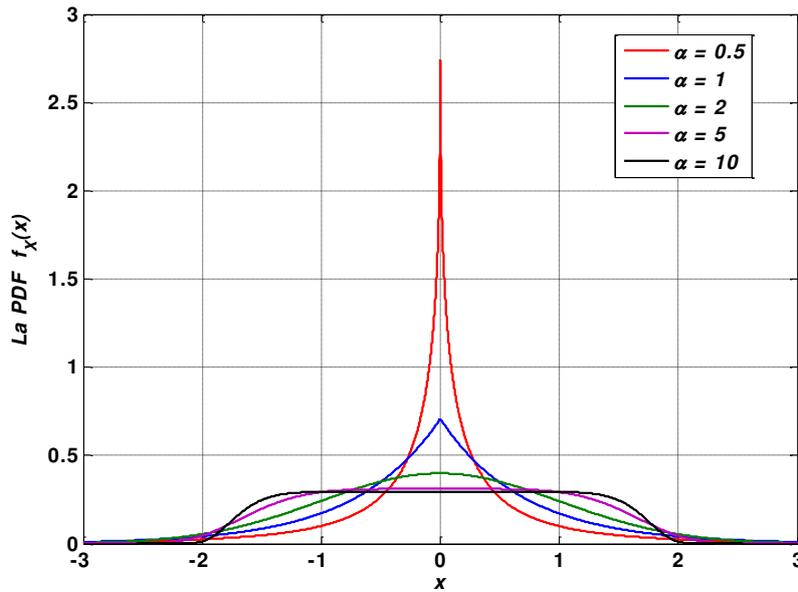


Fig. 2.1 La densité gaussienne généralisée pour différentes valeurs du paramètre α , avec $\mu = 0$ et $\sigma = 1$.

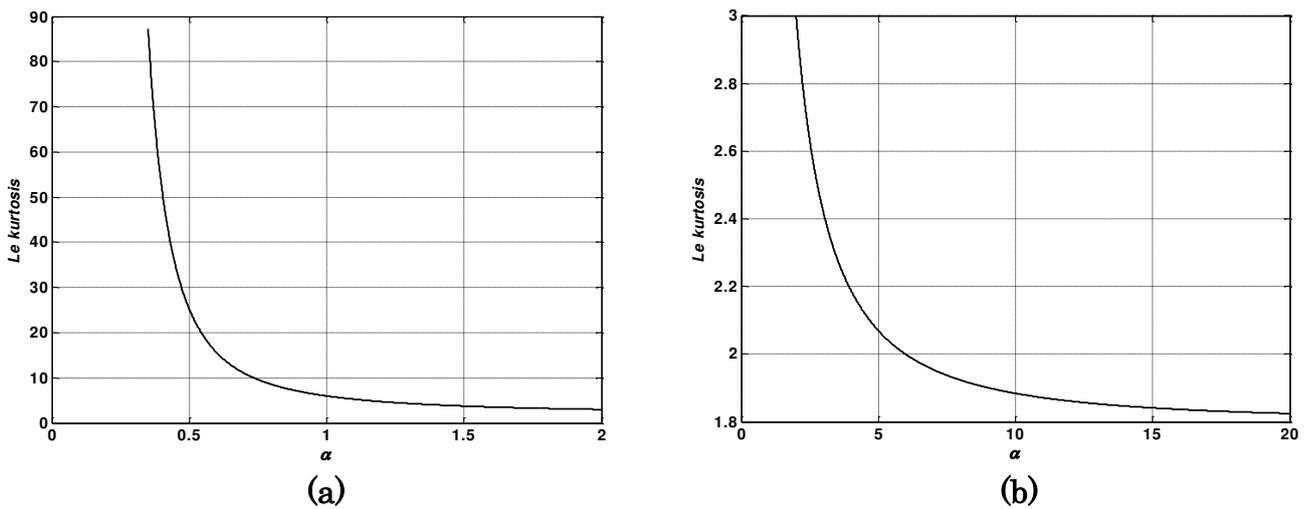


Fig. 2.2 Le Kurtosis de la GGD avec variance unité et α variable : (a) dans $[0, 2]$ (b) dans $[0, 20]$.

2.2.2 Simulation de la distribution gaussienne généralisée

Soient F_X la CDF, appelée aussi la fonction de répartition, d'un bruit X et F_X^{-1} son inverse. Il est bien connu que si F_X^{-1} peut être directement évaluée, un grand nombre de réalisations de X peuvent être obtenues par $x_i = F_X^{-1}(u_i)$, où u_i ($i = 1, 2, \dots, n$) sont de valeurs aléatoires distribuées uniformément dans $[0, 1]$. Si F_X^{-1} a une expression analytique, une telle méthode peut être appliquée de manière efficace, mais malheureusement, ce n'est pas toujours le cas. Néanmoins, si F_X^{-1} peut être évalué, il

peut toujours être possible de simuler le bruit X par inversion numérique de sa fonction de distribution cumulative. Lorsque F_X^{-1} n'a pas d'expression de forme ferme, l'intégration numérique ou d'autres méthodes d'approximation sont nécessaires, au détriment d'une quantité croissante de calcul. Une autre méthode de simulation basée sur la transformation du bruit X dont laquelle un générateur de nombres aléatoires est disponible. Une technique spécifique sera décrite ci-dessous pour le cas de GGD.

Soit X un bruit gaussien généralisé avec CDF

$$F_X(x) = \int_{-\infty}^x \frac{\alpha}{2} \frac{A(\alpha)}{\Gamma\left(\frac{1}{\alpha}\right)} \exp(-(A(\alpha)|t|)^\alpha) dt \quad (2.8.a)$$

Avec la fonction gamma incomplète $\Gamma_{\text{incomp}}(y, a) = \int_y^{+\infty} t^{a-1} e^{-t} dt$ nous pouvons écrire [55]

$$F_X(x) = \begin{cases} 1 - \frac{1}{2\Gamma\left(\frac{1}{\alpha}\right)} \Gamma_{\text{incomp}}\left((A(\alpha)x)^\alpha, \frac{1}{\alpha}\right) & x \geq 0 \\ \frac{1}{2\Gamma\left(\frac{1}{\alpha}\right)} \Gamma_{\text{incomp}}\left((-A(\alpha)x)^\alpha, \frac{1}{\alpha}\right) & x < 0 \end{cases} \quad (2.8.b)$$

où $A(\alpha)$ a été défini ci-dessus. Une telle fonction peut être écrite sous la forme fermée (ou explicite) seulement pour quelques cas spéciaux.

Comme nous allons le voir, les moments de ce type de bruits font une partie primordiale de notre travail. Pour cette raison il faut noter que ces moments sont finis, calculables analytiquement (par opposition à d'autres densités de probabilité à queue lourde [56], comme les densités α – stables).

Afin de générer de valeurs d'une distribution gaussienne généralisée avec paramètre de forme α , nous pouvons utiliser les trois étapes suivantes [6][53]

- i) simuler une variable aléatoire gamma $Z \sim \text{Gamma}(a, b)$, avec les paramètres $a = \alpha^{-1}$ et $b = (A(\alpha))^\alpha$;
- ii) appliquer d'abord la transformation $Y = Z^{1/\alpha}$;
- iii) finalement, appliquer une transformation de la forme

$$Y = |X| \quad (2.9)$$

La relation (2.9) présente deux racines. Le problème est de savoir comment déterminer la probabilité de choisir chaque racine. Nous pouvons démontrer, en utilisant

la symétrie de la distribution GG, que nous pouvons prendre les racines avec des probabilités égales (voir l'annexe A). Pour chaque observation y aléatoire, une racine est choisie ($x = -y$ ou $x = y$). A cette fin, un essai auxiliaire de **Bernoulli** avec une probabilité $p = 1/2$ peut être effectué. Autrement dit (2.9) donne $X = (-1)^B Y$ avec B une variable aléatoire de **Bernoulli** de paramètre $p = 1/2$.

Nous avons d'abord rencontré le problème de générer des valeurs aléatoires de la distribution gamma (voir l'annexe A). Le procédé repose sur l'hypothèse que si Z est une variable aléatoire à distribution $Gamma(a, b)$ (tels que a et b sont définis ci-dessus), puis en prenant $Y = Z^{1/\alpha}$ et en considérant la transformation (2.9), alors le bruit X a une GGD avec paramètre α .

Lors du test de la procédure ci-dessus, un grand nombre des échantillons ont été générés pour différentes valeurs du paramètre α . Le test de Kolmogorov-Smirnov a été appliqué à chaque expérience de simulation et n'a donné aucune indication que les observations simulées n'ont pas été générées à partir de la GGD [53].

2.2.2.1 Simulation d'un bruit gaussien généralisé avec $\alpha = 1/3$

Comme un cas particulier, nous avons étudié la densité gaussienne généralisée avec $\alpha = 1/3$, tandis que le cas $\alpha = 1/2$ est analysé dans le paragraphe suivant. Lorsque $\alpha = 1/3$, la densité donnée par l'équation (2.5) devient

$$f_X(x) = 2\sqrt{35} \exp\left\{-\left(24\sqrt{35}|x|\right)^{\frac{1}{3}}\right\} \quad (2.10)$$

et la CDF a la forme fermée suivante

$$F_X(x) = \begin{cases} \frac{1}{2} \exp\left\{-\left(24\sqrt{35}(-x)\right)^{\frac{1}{3}}\right\} \\ \cdot \left(1 + \left(24\sqrt{35}(-x)\right)^{\frac{1}{3}} + \frac{1}{2}\left(24\sqrt{35}(-x)\right)^{\frac{2}{3}}\right) & x \leq 0 \\ 1 - \frac{1}{2} \exp\left\{-\left(24\sqrt{35}x\right)^{\frac{1}{3}}\right\} \\ \cdot \left(1 + \left(24\sqrt{35}x\right)^{\frac{1}{3}} + \frac{1}{2}\left(24\sqrt{35}x\right)^{\frac{2}{3}}\right) & x > 0 \end{cases} \quad (2.11)$$

2.2.2.2 Les résultats numériques ($\alpha = 1/3$)

La figure 2.3 montre de réalisations d'un bruit à GGD avec $\alpha = 1/3$, générées avec l'algorithme décrit ci-dessus. Nous avons effectué une estimation de la fonction de densité de probabilité (2.10) en basant sur 10^8 valeurs simulées (échantillons), réunis en intervalles de largeur $\Delta x = 0.05$. La figure 2.4 montre la densité de probabilité simulée et la densité de probabilité théorique (2.10). Comme nous pouvons le remarquer, l'ajustement est assez précis.

Dans [53] ils ont également comparé la procédure de simulation basée sur la transformation gamma avec la méthode basée sur l'inverse de la CDF (2.11). Les résultats numériques montrent que l'ajustement avec la densité de probabilité théorique est encore bon pour la méthode de l'inversion de la CDF, mais en termes du temps de calcul, la première méthode qui est basée sur la transformation gamma est environ dix fois plus lente que la deuxième méthode.

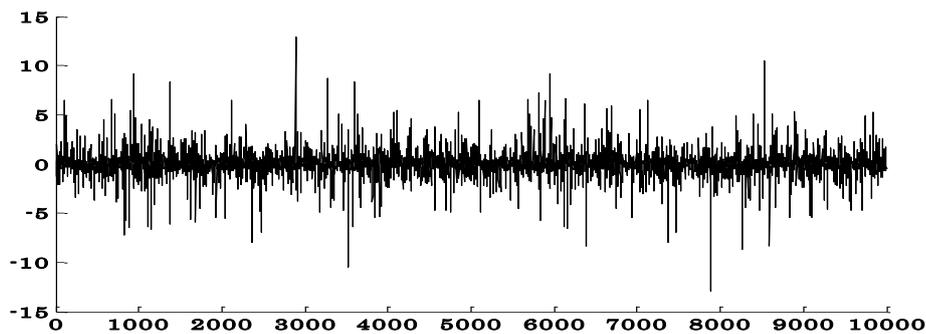


Fig. 2.3 Les réalisations d'un bruit à GGD avec les paramètres $\alpha = 1/3$, $\mu = 0$, et $\sigma = 1$ ($N = 10\,000$).

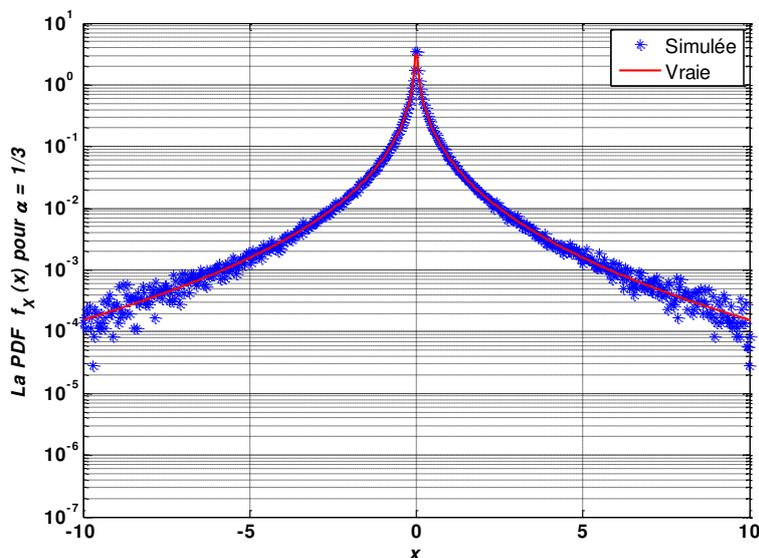


Fig. 2.4 Les fonctions de densité de probabilité estimée et théorique d'un bruit à GGD avec $\alpha = 1/3$ et kurtosis $\mathcal{K}_X(1/3) = 107.25$.

2.2.3 Simulation d'un bruit gaussien généralisé avec $\alpha = 1/2$

Dans ce paragraphe, nous allons proposer une approche numérique pour simuler le bruit X qui suit une loi dite gaussienne généralisée avec $\alpha = 1/2$. Nous allons mettre en évidence l'efficacité de la technique basée sur l'inverse de la CDF qui est exprimé au moyen d'une fonction spéciale W dite fonction de **Lambert**. La fonction W de **Lambert** est une fonction complexe à valeurs multiples [52]. Cette fonction n'a pas de forme explicite, ce qui nous conduit à fournir une méthode numérique pour l'évaluer, sur laquelle nous nous baserons pour construire un algorithme simple et rapide et générer plusieurs réalisations d'un bruit gaussien généralisé de paramètre de forme $\alpha = 1/2$.

2.2.3.1 La fonction W de Lambert

La fonction W de **Lambert**, est définie implicitement comme la racine de l'équation

$$W(z)e^{W(z)} = z \quad (2.12)$$

est une fonction à valeurs multiples définie en général pour z complexe et prenant des valeurs $W(z)$ complexe. Si z est réel et $z < -1/e$, alors $W(z)$ est à plusieurs valeurs complexes. Si $z \in \mathbb{R}$ et $-1/e \leq z < 0$, il existe deux valeurs réelles de $W(z)$: la branche satisfait $W(z) \geq -1$ est généralement désignée par $W_0(z)$ et est appelée la branche principale de la fonction W , et l'autre branche satisfait $W(z) \leq -1$ est désigné par $W_{-1}(z)$. Si $z \in \mathbb{R}$ et $z \geq 0$, il y a une valeur unique pour $W(z)$ qui appartient aussi à la branche principale $W_0(z)$.

Plusieurs problèmes qui se posent dans différentes applications mathématiques, physiques et sciences de l'ingénieur, font appel à cette fonction, ce qui donne à cette dernière un rôle central pour résoudre ces problèmes [52], à tel point que des scientifiques la considère comme une fonction mathématique spéciale. **Corless** et al. [52] ont présenté les principales propriétés et les applications d'une telle fonction.

Dans ce qui suit, nous allons donner un autre champ d'application de W , dans le domaine du traitement du signal, pour synthétiser le bruit gaussien généralisé de paramètre de forme (ou d'exposant) $\alpha = 1/2$.

La figure 2.5 montre à la fois la branche principale W_0 (ligne pointillée) et la branche notée W_{-1} (ligne continue) de la fonction W de **Lambert**.

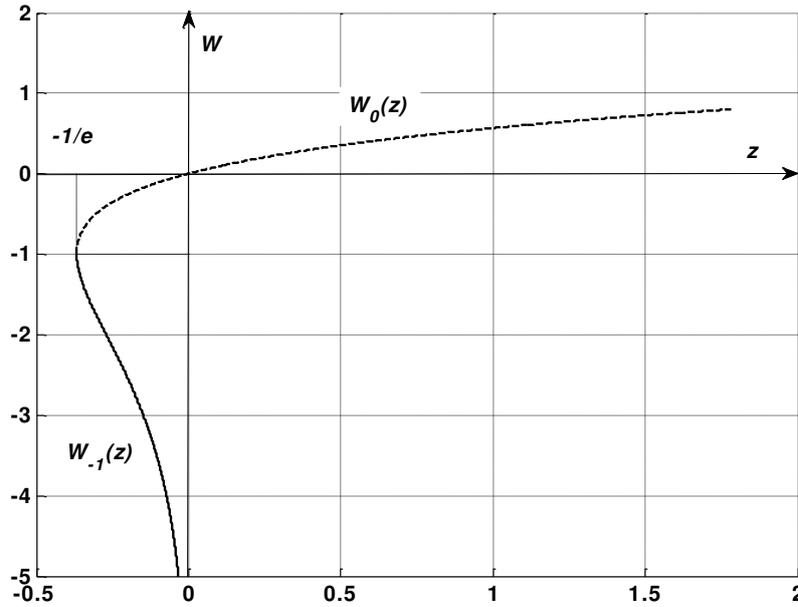


Fig. 2.5 Les deux branches réelles de la fonction W de Lambert. La branche $W_{-1}(z)$ (ligne continue) : définie pour $-1/e \leq z < 0$. La branche principale $W_0(z)$ (ligne pointillée) : définie pour $-1/e \leq z < +\infty$. Les deux branches ont un point commun $(-1/e, -1)$.

2.2.3.2 Bruit gaussien généralisé pour $\alpha = 1/2$

Dans l'équation (2.5) de la densité de probabilité, si nous prenons le cas $\alpha = 1/2$, nous obtenons la densité GG

$$f_X(x) = \frac{\sqrt{30}}{2} \exp \left\{ - (2\sqrt{30}|x|)^{\frac{1}{2}} \right\} \tag{2.13}$$

En s'appuyant sur la formule (2.8.b), la fonction de distribution cumulative CDF est donnée par

$$F_X(x) = \begin{cases} \frac{1}{2} \exp \left\{ - (2\sqrt{30}(-x))^{\frac{1}{2}} \right\} \left(1 + (2\sqrt{30}(-x))^{\frac{1}{2}} \right) & x \leq 0 \\ 1 - \frac{1}{2} \exp \left\{ - (2\sqrt{30}x)^{\frac{1}{2}} \right\} \left(1 + (2\sqrt{30}x)^{\frac{1}{2}} \right) & x > 0 \end{cases} \tag{2.14}$$

Afin de simuler ou générer de bruits à partir d'une distribution GG, nous pouvons appliquer la méthode basée sur la transformation gamma décrite précédemment dans le paragraphe 2.2.2 qui comporte trois étapes. Pour une simulation en un seul coup, procédons ci-dessous à l'inversion de $F_X(x)$ de l'équation (2.14); nous allons montrer que l'inverse peut être exprimé au moyen de la fonction W de Lambert.

Dans un premier temps nous allons considérer la branche $x \leq 0$ de $F_X(x)$.

Si nous posons $y = -(-2\sqrt{30}x)^{\frac{1}{2}}$, l'expression de $F_X(x)$ sur cette branche devient

$$F_X(x) = \frac{1}{2}(1 - y)e^y \quad (2.15)$$

Pour faire apparaître la fonction W , l'équation (2.15) peut être écrite comme suit

$$\frac{-2F_X(x)}{e} = (y - 1)e^{y-1} \quad (2.16)$$

inversible sous la forme

$$y - 1 = W_{-1}\left(\frac{-2F_X(x)}{e}\right) \quad (2.17)$$

En remplaçant x par sa valeur dans l'équation (2.17), nous obtenons

$$x = -\frac{1}{2\sqrt{30}}\left(1 + W_{-1}\left(\frac{-2F_X(x)}{e}\right)\right)^2 \quad (2.18)$$

La branche W_{-1} de la fonction de **Lambert** est identifiée puisqu'elle passe par les deux points $(x = 0, F_X = 1/2)$ et $(x = -\infty, F_X = 0)$.

Maintenant, nous se plaçons dans la branche $x > 0$ de $F_X(x)$ de l'équation (2.14); des arguments similaires peuvent être appliqués dans ce cas pour inverser $F_X(x)$.

Finalement nous obtenons l'expression de $F_X^{-1}(x)$ pour tout $x \in]0,1[$

$$F_X^{-1}(x) = \begin{cases} -\frac{1}{2\sqrt{30}}\left(1 + W_{-1}\left(-\frac{2x}{e}\right)\right)^2 & \text{pour } 0 < x \leq \frac{1}{2} \\ \frac{1}{2\sqrt{30}}\left(1 + W_{-1}\left(-\frac{2(1-x)}{e}\right)\right)^2 & \text{pour } \frac{1}{2} \leq x < 1 \end{cases} \quad (2.19)$$

Si nous pouvons effectuer l'évaluation numérique de la fonction W_{-1} , l'équation (2.19) résout le problème de synthèse du bruit gaussien généralisé de paramètre de forme $\alpha = 1/2$ à partir du bruit uniforme. Et par suite, l'étape suivante consiste à essayer d'évaluer numériquement cette fonction spéciale.

2.2.3.3 Evaluation numérique de W_{-1}

Nous en venons maintenant à l'évaluation numérique de $W_{-1}(z)$ pour tout $z \in [-1/e, 0[$. Comme nous l'avons mentionné, cette évaluation numérique est directement accessible dans le logiciel **Matlab** [57]. Bien que la fonction W soit très utile dans beaucoup de domaines, nous ne la trouvons pas dans les bibliothèques mathématiques standards de tous les logiciels. Nous décrirons ainsi ici les éléments explicites pour son évaluation numérique, qui permettent d'écrire des sous-programmes avec un langage de programmation.

2.2.3.3.1 Evaluation de précision arbitraire

Une spécificité de la fonction W de **Lambert** est qu'elle est définie comme une fonction inverse. Dès lors, des évaluations de précision arbitraires peuvent être obtenues au moyen de méthodes itératives de recherche de racine. Pour un z donné, nous pouvons trouver $W_{-1}(z)$ comme racine de l'équation $we^w = z$, de la branche $W_{-1}(z)$. De nombreuses méthodes itératives sont disponibles à cet égard. Un choix doit concilier à la fois la complexité de la mise en œuvre, les conditions et le nombre d'itérations jusqu'à convergence pour une précision donnée. Ces propriétés sont habituellement contrôlables par l'intermédiaire de l'ordre de la méthode (l'ordre le plus élevé des dérivés de la fonction à mettre à zéro par l'algorithme).

La méthode de **Newton** [58], est une méthode de second ordre simple qui est appropriée et avec une convergence rapide. Entre la simplicité et la rapidité, un meilleur compromis est réalisé par la méthode de **Helley**, une méthode de troisième ordre, qui constitue le choix mis en application par **Matlab** et qui mène à l'évaluation de précision élevée dans un temps raisonnable. Il est basé sur le procédé itératif suivant [6][59]

$$w_{j+1} = w_j - \frac{w_j e^{w_j} - z}{(w_j + 1)e^{w_j} - \frac{(w_j + 2)(w_j e^{w_j} - z)}{2w_j + 2}} \quad j \geq 0 \quad (2.20)$$

Des méthodes de quatrième ordre, plus rapides mais plus compliquées, sont proposées dans [60] et [61], mais elles sont limitées, comme elles sont exposées, à la seule branche principale $W_0(z)$.

En outre, le fait que la fonction de **Lambert** soit une fonction inverse fournit une voie simple pour contrôler l'erreur, ou la précision, d'une évaluation numérique donnée. Pour un $z \in]-1/e, 0[$, la valeur de $W_{-1}(z)$ est évaluée comme étant la racine approchée de $\varphi(w) = z$ où $\varphi(w) = we^w$, que nous noterons par w_{eval} . Si la racine exacte est notée w_{true} (c'est la valeur « exacte » de $W_{-1}(z)$), alors le résidu $z - w_{eval}e^{w_{eval}}$ permet d'accéder à l'erreur d'évaluation $w_{true} - w_{eval}$.

Pour n'importe quel algorithme de recherche de la racine, w_{eval} est très proche de w_{true} et nous avons, à une très bonne approximation

$$\frac{z - w_{eval}e^{w_{eval}}}{w_{true} - w_{eval}} \cong \varphi'(w_{eval}) = (1 + w_{eval})e^{w_{eval}} \quad (2.21)$$

d'où l'erreur d'évaluation

$$w_{true} - w_{eval} \cong \frac{z - w_{eval}e^{w_{eval}}}{(1 + w_{eval})e^{w_{eval}}} \quad (2.22)$$

Lorsque l'équation (2.19) est contrôlée par l'équation (2.22), une évaluation de précision arbitraire de la fonction W de **Lambert** peut être réalisée. Ceci est ce que **Matlab** le fait aussi.

2.2.3.3.2 Approximation rapide

Parfois, plutôt que d'utiliser un procédé itératif, des évaluations plus directes de la fonction W de **Lambert** peuvent être recherchées. Ceci peut être particulièrement vrai pour les trois raisons suivantes

D'abord, pour des applications comme les simulations de Monte Carlo, qui demandent un grand nombre de tirages, il peut parfois être préférable d'avoir une évaluation en « un seul coup » et rapide avec une précision limitée mais suffisante, plutôt qu'un procédé itératif plus précis mais plus lent.

En second lieu, l'approximation rapide en « un seul coup » peut être utile pour fournir une bonne valeur initiale à un procédé itératif afin de réduire le nombre d'itérations jusqu'à la convergence pour une précision donnée.

Troisièmement, la fonction $W_{-1}(z)$ dans son domaine d'existence a une branche infinie au voisinage de 0^- ($W_{-1}(z)$ tend vers $-\infty$). Ceci se traduit, pour un procédé itératif d'évaluation, par une erreur d'approximation de l'équation (2.22), qui décroît très lentement quand $W_{-1}(z)$ est approchée par w_{eval} , proche de -1 car z est choisi près de 0 . Par conséquent, pour z proche de 0 , les évaluations directes de $W_{-1}(z)$ au moyen d'un développement en série ou d'un développement asymptotique peuvent être préférables et sont en effet possibles. Dans ce volet, **Monir** et al. [59] ont décrit avec détails de différentes approximations de $W_{-1}(z)$ pour $z \in [-1/e, 0]$. Le résultat primordial de leurs approximations est présenté sous la forme d'un algorithme d'approximation comme suit [59][53]

Si $-\frac{1}{e} \leq z < -0.333$

$$p = -\sqrt{2(e \cdot z + 1)}$$

$$W_{-1}(z) = -1 + p - \frac{1}{3}p^2 + \frac{11}{72}p^3 - \frac{43}{540}p^4 + \frac{769}{17280}p^5 - \frac{221}{8505}p^6$$

Sinon Si $-0.333 \leq z \leq -0.033$

$$N = -8.096 + 391.0025 \cdot z - 47.4252 \cdot z^2 - 4877.633 \cdot z^3 - 5532.776 \cdot z^4$$

$$D = 1 - 82.9423 \cdot z + 433.8688 \cdot z^2 + 1515.306 \cdot z^3$$

$$W_{-1}(z) = N/D$$

Sinon $-0.033 < z < 0$

$$L_1 = \ln(-z)$$

$$L_2 = \ln(-\ln(-z))$$

$$W_{-1}(z) = L_1 - L_2 + \frac{L_2}{L_1} + \frac{L_2(-2+L_2)}{2L_1^2} + \frac{L_2(6-9L_2+2L_2^2)}{6L_1^3} + \frac{L_2(-12+36L_2-22L_2^2+3L_2^3)}{12L_1^4} + \frac{L_2(60-300L_2+350L_2^2-125L_2^3+12L_2^4)}{60L_1^5}$$

Fin Si

Cet algorithme nous permet d'avoir une fonction approximante de $W_{-1}(z)$ avec une erreur relative strictement inférieure à 10^{-4} pour tout $z \in [-1/e, 0[$ [59][53]. Il est important de signaler que dans la bibliothèque du **Matlab** il existe la fonction de **Lambert**, elle est représentée par la fonction **lambertw**. L'utilisation de telle fonction dans la génération des réalisations à GGD, à partir de la distribution uniforme sur $[0,1]$ en utilisant l'équation (2.19), requiert un temps d'exécution relativement considérable si nous augmentons le nombre des réalisations. Par contre, le maniement de l'algorithme d'approximation ci-dessus permet de générer les réalisations à GGD en un temps d'exécution très courte comme le montre la figure 2.6.

La figure 2.6 montre l'avantage d'utiliser l'algorithme d'approximation, que d'utiliser directement la fonction **lambertw** de **Matlab**, dans la génération du bruit à GGD pour $\alpha = 1/2$. Le gain en termes du temps d'exécution entre les deux méthodes, comme montre la figure 2.6, est de l'ordre de 10^3 pour $N = 10^5$ réalisations.

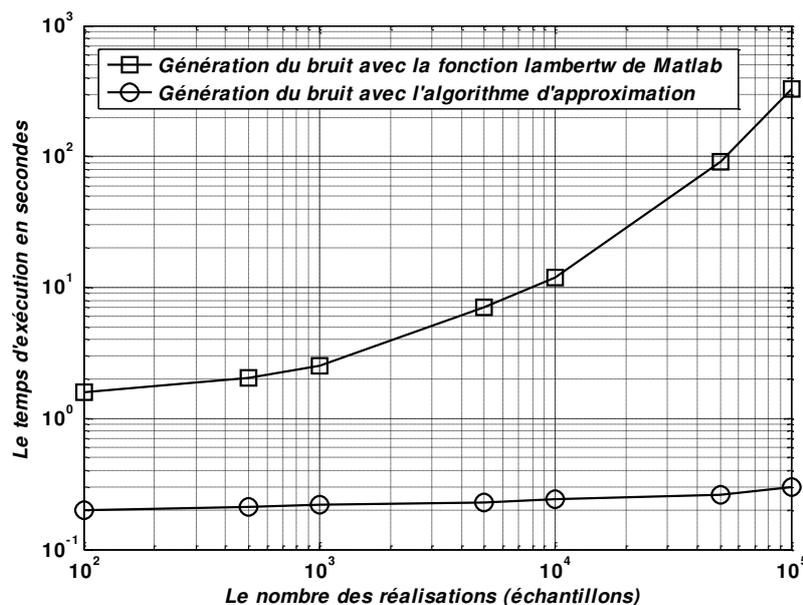


Fig. 2.6 Les temps d'exécution en secondes pour générer un bruit à GGD avec $\alpha = 1/2$ en utilisant directement la fonction de Lambert de Matlab puis l'algorithme d'approximation.

2.2.3.4 Génération du bruit gaussien généralisé de paramètre $\alpha = 1/2$

La figure 2.7 montre le tracé de 10^4 réalisations d'un bruit gaussien généralisé avec $\alpha = 1/2$, générées par l'algorithme d'approximation décrit ci-dessus.

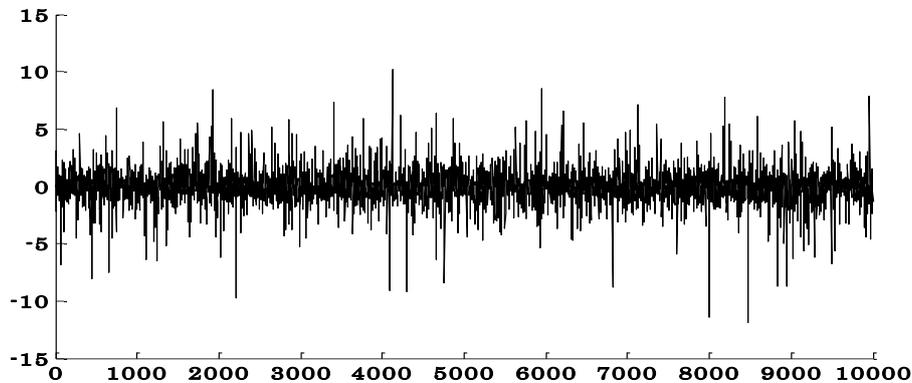


Fig. 2.7 Les réalisations d'un bruit à GGD avec les paramètres $\alpha = 1/2$, $\mu = 0$, et $\sigma = 1$ ($N = 10\ 000$).

Nous avons effectué une estimation de la fonction de densité de probabilité (2.13) en basant sur 10^6 valeurs simulées (échantillons), réunis en intervalles de largeur $\Delta x = 0.05$. Sur la figure 2.8 qui représente la densité de probabilité simulée superposée à la densité théorique (2.10), nous constatons qu'il y a un accord. Cette synthèse rapide du bruit gaussien généralisé basé sur l'algorithme d'approximation, est spécialement appropriée pour la simulation Monte Carlo, qui exige un grand nombre de réalisations de bruit. Le test de Kolmogorov-Smirnov appliqué sur les réalisations de simulation n'a donné aucune indication que ces réalisations n'ont pas été générées à partir de la distribution GG [53].

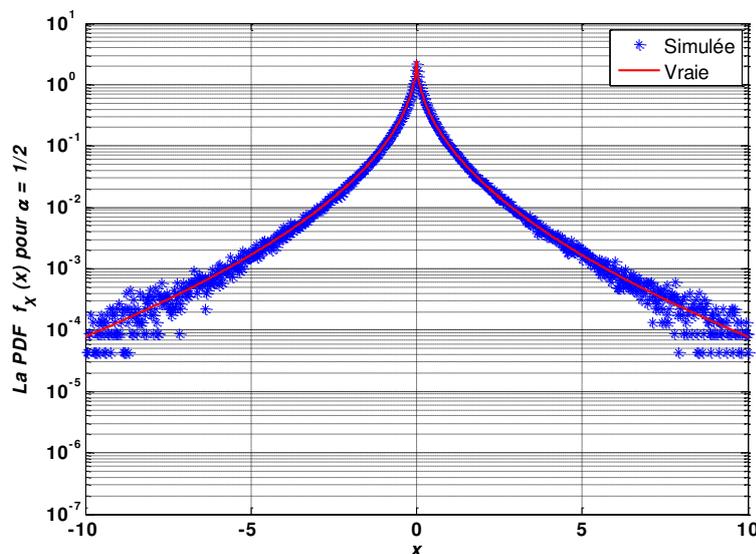


Fig. 2.8 Les fonctions de densité de probabilité estimée et théorique d'un bruit à GGD avec $\alpha = 1/2$.

2.3 Les lois α – stables

Les lois **Pareto**–stables, appelées aussi lois **Lévy**–stables ou tout simplement lois α –stables, forment une classe importante des distributions des probabilités, car elles permettent d’avoir l’asymétrie et les queues lourdes et possèdent de propriétés mathématiques intrigantes. Cette classe a été définie par **Paul Lévy** dans son étude des sommes de termes identiquement distribuées indépendants dans les années 1920. La non-existence des formules fermées pour les PDF de ces lois, sauf pour quelques cas particuliers (le cas gaussien et le cas **Cauchy**), présente un inconvénient majeur dans l’utilisation des distributions α –stables par les praticiens. Il y a maintenant des logiciels informatiques fiables pour calculer ces PDF. Avec ces logiciels, il est possible d’utiliser les modèles α –stables dans une variété de problèmes pratiques.

Les distributions α –stables ont été proposées comme un modèle pour de nombreux types de phénomènes physiques et économiques. Il existe plusieurs raisons d’utiliser une distribution α –stable pour décrire un phénomène [62]. La première est celle où il y a des raisons théoriques solides qui nous permettent de prévoir un modèle α –stable non-gaussienne, par exemple, la réflexion sur un miroir rotatif qui donne une distribution de **Cauchy**, les temps de passage d’un mouvement brownien qui donne une distribution de **Lévy**, le champ gravitationnel des étoiles qui donne une distribution **Holtmark** et d’autres exemples sont cités dans [63] et [64]. La deuxième raison est le théorème central limite généralisé qui stipule que la seule limite possible non négligeable des sommes normalisées de termes indépendants identiquement distribués est α –stable. La troisième raison est purement empirique : de nombreux grands ensembles de données présentent des queues lourdes et l’asymétrie.

Plusieurs livres sont consacrés à ces lois : **Zolotarev** [65] qui a étudié les lois α –stables dans le contexte univarié ; **Samorodnitsky** et al. [56] qui ont étudié de manière approfondie beaucoup de propriétés de ces lois dans le cas univarié comme dans le cas multivarié ; **Nikias** et al. [13] qui ont appliqué ces lois dans le domaine du traitement du signal.

Dans cette partie, nous traitons seulement le cas univarié. Les définitions et propriétés énoncées ici se trouvent en détail dans [56] et [66].

2.3.1 Rudiments des lois α – stables

2.3.1.1 Définitions

Définition 2.1 [67]

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

$$X_1 + X_2 + \dots + X_n \stackrel{d}{=} C_n X + D_n \quad (2.23)$$

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

Lorsque $D_n = 0$, nous parlons de loi strictement α – stable. Il est bon de noter que le mot « stable » signifie que la loi est stable par convolution. Nous pouvons 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** [63]. D'où il vient l'appellation « α – stable ».

Définition 2.2 [25]

Nous disons qu'une variable aléatoire réelle X qu'elle suit une loi α – stable de paramètres α, β, γ et δ , que le nous notons $X \sim S_\alpha(\beta, \gamma, \delta)$, si et seulement si sa fonction caractéristique est de la forme

$$\psi_X(t) = \exp\{-\gamma^\alpha |t|^\alpha [1 + i \cdot \beta \cdot \text{sign}(t) \cdot \omega(t, \alpha)] + i \cdot \delta \cdot t\} \quad (2.24)$$

où

$$\omega(t, \alpha) = \begin{cases} -\tan\left(\frac{\pi\alpha}{2}\right) & \text{si } \alpha \neq 1 \\ \frac{2}{\pi} \ln|t| & \text{si } \alpha = 1 \end{cases} \quad (2.25. a)$$

$$\text{sign}(t) = \begin{cases} 1 & \text{si } t > 0 \\ 0 & \text{si } t = 0 \\ -1 & \text{si } t < 0 \end{cases} \quad (2.25. b)$$

$i^2 = -1$, $\alpha \in]0, 2]$, $\beta \in [-1, 1]$, $\gamma > 0$ et $\delta \in \mathbb{R}$.

Par conséquent, pour identifier complètement une loi α – stable il faut connaître ces quatre paramètres α , β , γ et δ .

Dans la définition 2.2, nous avons utilisé la fonction caractéristique au lieu de la PDF pour définir une loi $S_\alpha(\beta, \gamma, \delta)$, parce que la PDF d'une telle loi ne possède pas une forme fermée ou explicite sauf pour les deux cas particuliers suivants [25]

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

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

$$f_X(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-itx} \psi_X(t) dt \quad (2.26)$$

Exemple 2.1

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

$$X_1 + X_2 \cong CX + D \quad (2.27)$$

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

$$\begin{cases} 2\lambda = C\lambda + D \\ 2\lambda = C^2\lambda \end{cases} \Rightarrow \begin{cases} D = (2 - \sqrt{2})\lambda \\ C = \sqrt{2} \end{cases} \quad (2.28)$$

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} .

Notations

Par convention, $S_\alpha(\beta, \gamma, \delta)$ représentera une loi α – stable de paramètres α, β, γ et δ et $S\alpha S(\gamma)$ ou tout simplement $S\alpha S$ une loi symétrique α – stable de paramètre γ où $\beta = \delta = 0$.

2.3.1.2 Définition et interprétation des paramètres d'une loi $S_\alpha(\beta, \gamma, \delta)$ [68]

Une loi α –stable alors est caractérisée par les quatre paramètres

- $0 < \alpha \leq 2$, ce paramètre est appelé le paramètre de forme, l'exposant caractéristique ou l'indice de stabilité. Il décrit la forme de la distribution ou le degré d'épaisseur de la queue de distribution (figures 2.9 jusqu'à 2.15). Plus ce paramètre est petit, plus les queues de la distribution sont épaisses, donc, plus d'impulsivité au niveau des échantillons du bruit. Une distribution gaussienne a la valeur maximum de α soit $\alpha = 2$.
- $-1 \leq \beta \leq 1$, est le paramètre d'asymétrie. Lorsque β vaut zéro alors la distribution est symétrique (figures 2.9 jusqu'à 2.11). La loi est symétrique par rapport au paramètre δ . Si en plus, $\delta = 0$, la loi est dite symétrique α –stable $S\alpha S$ de fonction caractéristique

$$\psi_X(t) = \exp(-\gamma|t|^\alpha) \quad (2.29)$$

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).

- $\gamma > 0$, est le paramètre de dispersion ou d'échelle. Plus γ est grand, plus les données sont volatiles. Le paramètre γ permet de cintrer plus ou moins le corps de la distribution (figures 2.9, 2.12 et 2.13).
- $-\infty < \delta < +\infty$, est le paramètre de localisation ou de position, 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 2.9, 2.14 et 2.15). Lorsque δ est positif (resp. négatif), la courbe décale vers la droite (resp. vers la gauche).

Remarque 2.1

Dans les tracés de la PDF $f_X(x)$ d'une loi α –stable (voir les figures 2.10 et 2.11), nous remarquons que $f_X(-x, \alpha, \beta) = f_X(x, \alpha, -\beta)$.

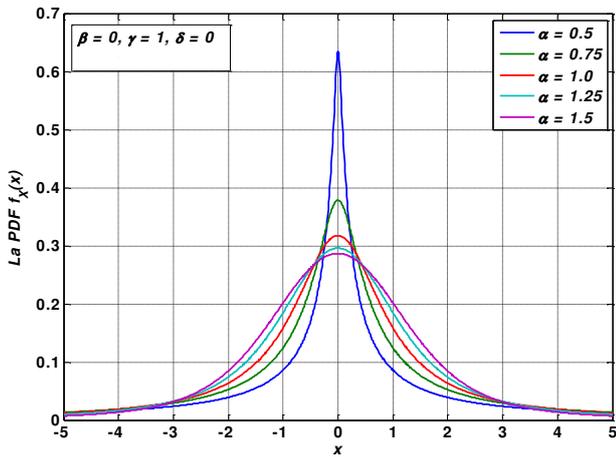


Fig. 2.9 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0, \gamma = 1, \delta = 0$.

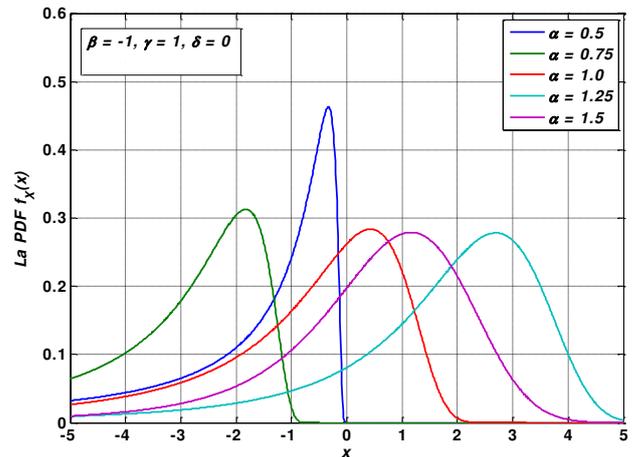


Fig. 2.10 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = -1, \gamma = 1, \delta = 0$.

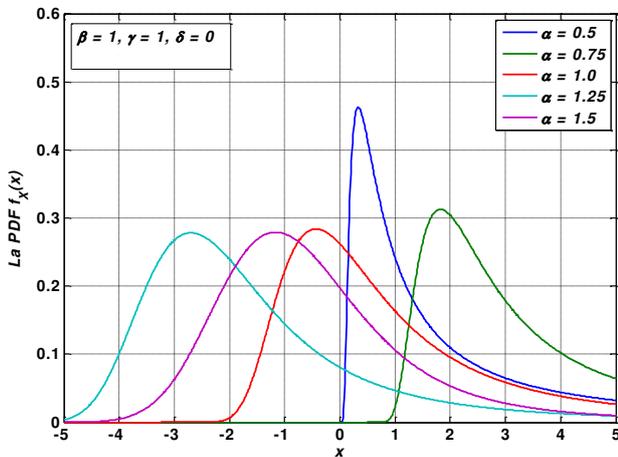


Fig. 2.11 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 1, \gamma = 1, \delta = 0$.

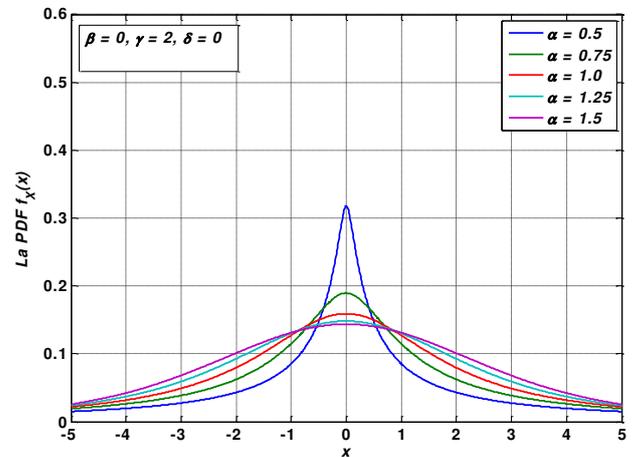


Fig. 2.12 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0, \gamma = 2, \delta = 0$.

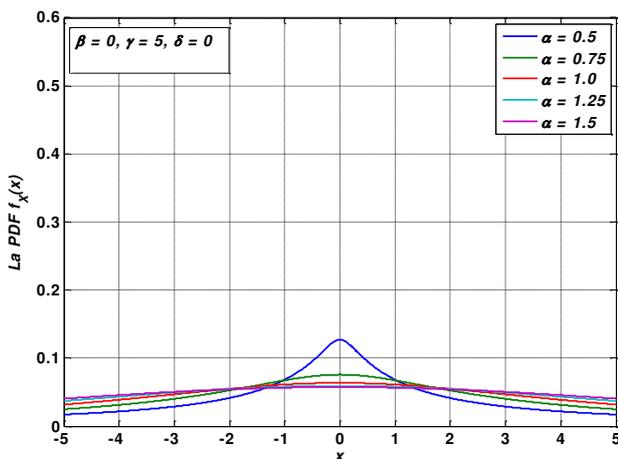


Fig. 2.13 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0, \gamma = 5, \delta = 0$.

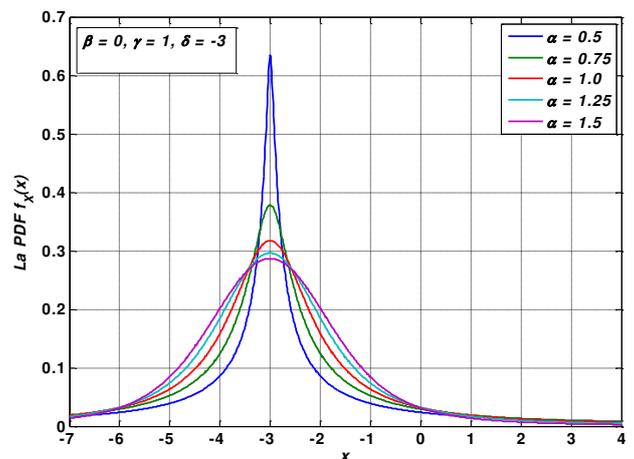


Fig. 2.14 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0, \gamma = 1, \delta = -3$.

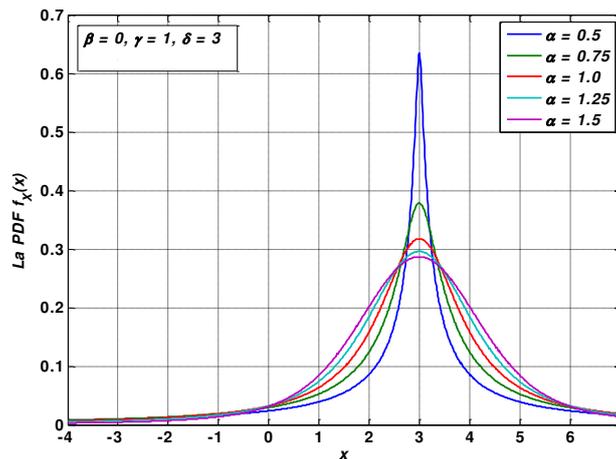


Fig. 2.15 La PDF pour $\alpha \in \{0.5, 0.75, 1, 1.25, 1.5\}$, $\beta = 0, \gamma = 1, \delta = 3$.

2.3.2 Diverses propriétés

Nous allons rappeler quelques propriétés importantes des bruits α –stables de loi $S_\alpha(\beta, \gamma, \delta)$ que nous utiliserons dans la suite de cette thèse.

Propriété 2.1 : La densité

Pour la plupart des lois connues, nous avons une forme explicite de la PDF (normale, **Cauchy**, gamma,...). Pour la loi α –stable, 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 (2.26) nous pouvons obtenir la PDF f_X d’une loi α –stable standardisée, i.e. $\gamma = 1$ et $\delta = 0$, sous la forme d’une intégrale comme suit

$$f_X(x) = \frac{1}{\pi} \int_0^{+\infty} e^{-t^\alpha} \cos[xt + \beta t^\alpha \omega(t, \alpha)] dt \tag{2.30}$$

où

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

Propriété 2.2 : La stabilité

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

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

La démonstration de cette propriété se trouve dans [25].

Remarque 2.2

Pour la simulation (qui sera présentée dans le paragraphe 2.3.4), il suffit de générer des lois $S_\alpha(\beta, 1, 0)$ et par changement de variables, nous pouvons obtenir des lois $S_\alpha(\beta, \gamma, \delta)$.

Propriété 2.3 : Les queues lourdes

Soit X un bruit $S_\alpha(\beta, \gamma, \delta)$ nous avons les deux résultats suivants [56]

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

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

L'égalité précédente nous fait penser à la caractérisation des lois de **Pareto**. En effet, un bruit 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** [69] a montré que pour les lois α – stables, nous avons

$$P(X \geq x) = x^{-\alpha} [f_1(\alpha) + f_2(\alpha)x^{-\alpha} + o(x^{-2\alpha})] \quad (2.31)$$

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

Propriété 2.4 : L'existence des moments

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

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

La preuve de cette propriété est bien détaillée dans [7] et [25].

Nous pouvons conclure à partir de cette propriété que si α est strictement inférieur à 2, la variance d'une loi α – stable devient infinie. Si α est plus grand que 1, la moyenne d'une loi α – stable est δ .

Propriété 2.5 : Calcul des moments

Si X suit une loi $S_\alpha(\beta, \gamma, 0)$, nous avons pour tout réel r tel que : $-1 < r < \alpha$

$$m_r = E(|X|^r) = \frac{\Gamma\left(1 - \frac{r}{\alpha}\right)}{\Gamma(1 - r)} \cdot \left| \frac{\gamma}{\cos \theta} \right|^{\frac{r}{\alpha}} \cdot \frac{\cos\left(\frac{r\theta}{\alpha}\right)}{\cos\left(\frac{r\pi}{2}\right)} \quad (2.32)$$

avec la fonction $\Gamma(z)$ est donnée par (2.3), et l'expression de θ est donnée par

$$\theta = \arctan\left(\beta \tan \frac{\alpha\pi}{2}\right) \quad (2.33)$$

Comme cas particulier, si la loi est $S\alpha S$, c'est-à-dire $\beta = 0$, nous trouvons

$$m_r = E(|X|^r) = \frac{\Gamma\left(1 - \frac{r}{\alpha}\right)}{\Gamma(1 - r)} \cdot \frac{\gamma^{\frac{r}{\alpha}}}{\cos\left(\frac{r\pi}{2}\right)} = \frac{2^{r+1} \cdot \Gamma\left(\frac{r+1}{2}\right) \cdot \Gamma\left(\frac{-r}{\alpha}\right)}{\alpha\sqrt{\pi} \cdot \Gamma\left(\frac{-r}{2}\right)} \cdot \gamma^{\frac{r}{\alpha}} \quad (2.34)$$

Ce résultat important a été démontré par **Nikias** dans la référence [13].

Propriété 2.6 : Les moments logarithmiques [70]

Pour éviter la présence de la fonction gamma, qui complique amplement les calculs, dans les moments d'une loi α – stable, nous utilisons ce que nous appelons les moments logarithmiques.

Soit X un bruit suit la loi $S_\alpha(\beta, \gamma, 0)$ et n un nombre entier strictement positif. D'après la relation

$$E((\ln|X|)^n) = \lim_{r \rightarrow 0} \frac{d^n}{dr^n} E(|X|^r), \quad n = 1, 2, 3, \dots$$

nous obtenons

$$L_1 = E(\ln|X|) = \psi_0 \cdot \left(1 - \frac{1}{\alpha}\right) + \frac{1}{\alpha} \cdot \ln \left| \frac{\gamma}{\cos \theta} \right| \quad (2.35)$$

$$L_2 = E\left((\ln|X| - E(\ln|X|))^2\right) = \psi_1 \cdot \left(\frac{1}{2} + \frac{1}{\alpha^2}\right) - \frac{\theta}{\alpha^2} \quad (2.36)$$

$$L_3 = E\left((\ln|X| - E(\ln|X|))^3\right) = \psi_2 \cdot \left(1 - \frac{1}{\alpha^3}\right) \quad (2.37)$$

sachant que les valeurs ψ_k de la fonction polygamma sont données par

$$\psi_k = \frac{d^{k+1}}{dx^{k+1}} \ln \Gamma(x) \Big|_{x=1}, \quad \psi_0 = -0.57721566 \dots, \quad \psi_1 = \frac{\pi^2}{6}, \quad \psi_2 = 1.2020569 \dots$$

et la quantité θ est donnée par (2.33).

Propriété 2.7 : L'additivité

Soient X_1 et X_2 deux bruits indépendants de lois α –stable $S_\alpha(\beta_1, \gamma_1, \delta_1)$ et $S_\alpha(\beta_2, \gamma_2, \delta_2)$ respectivement, alors $X_1 + X_2$ suit une loi α –stable $S_\alpha(\beta, \gamma, \delta)$ avec

$$\gamma = (\gamma_1^\alpha + \gamma_2^\alpha)^{\frac{1}{\alpha}}; \beta = \frac{\beta_1 \gamma_1^\alpha + \beta_2 \gamma_2^\alpha}{\gamma_1^\alpha + \gamma_2^\alpha}; \delta = \delta_1 + \delta_2 \quad (2.38)$$

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

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

$$AX_1 + BX_2 + C = S_\alpha\left(\beta, \gamma(A^\alpha + B^\alpha)^{\frac{1}{\alpha}}, \delta(A^\alpha + B^\alpha)^{\frac{1}{\alpha}} + C\right) \quad (2.39)$$

2.3.3 Puissance géométrique

Il est important pour les processus α –stables de pouvoir évaluer l'« intensité » du processus, autrement dit sa puissance. C'est un paramètre d'échelle qui permet de comparer l'importance de l'interférence par rapport au signal utile et, éventuellement, aux autres bruits. Classiquement, la puissance arithmétique est utilisée, qui représente le moment d'ordre 2 du bruit [7].

Il est clair que la puissance arithmétique n'existe plus pour un processus α –stable avec $\alpha < 2$. Nous pouvons nous demander alors s'il est possible de définir un opérateur puissance P vérifiant quelques propriétés élémentaires de la puissance arithmétique [7]. **Gonzalez** dans [14] et [15] a proposé une autre forme de puissance, nommée puissance géométrique, qui s'affranchit de la condition de linéarité.

2.3.3.1 Définitions [71][7]**Définition 2.3**

Nous disons qu'un bruit X est d'ordre logarithmique si et seulement si l'espérance $E(\ln|X|) < +\infty$.

Cette définition est établie par analogie avec l'expression « processus de second ordre » qui exprime que le processus considéré est à variance finie.

Définition 2.4

Soit X un bruit d'ordre logarithmique. La puissance géométrique de ce bruit s'écrit

$$S_0 = S_0(X) = e^{E(\ln(X^2))} e^{2E(\ln|X|)} \quad (2.40)$$

La puissance géométrique d'un bruit X de densité de probabilité $f_X(x)$ s'écrit alors comme suit

$$S_0 = e^{2 \int_{-\infty}^{+\infty} \ln|x| \cdot f_X(x) dx} \quad (2.41)$$

A partir de cette formule nous pouvons déterminer les puissances géométriques de quelques distributions usuelles.

- Pour la loi uniforme, $f_X(x) = 1/a$, $|x| \leq a/2$, nous avons $S_0(x) = (a/2e)^2$;
- Pour la loi gaussienne, $f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$, nous avons $S_0(x) = \sigma^2/2C_g$;
- Pour la loi de **Cauchy**, $f_X(x) = \frac{\gamma}{\pi(x^2+\gamma^2)}$, nous avons $S_0(x) = \gamma^2$.

2.3.3.2 Propriétés

Cette puissance géométrique présente d'importantes propriétés.

Propriété 2.8 : S_0 est un paramètre d'échelle

Pour tout processus d'ordre logarithmique X et pour toute constante c nous avons

- $S_0(X) \geq 0$;
- $S_0(cX) = c^2 S_0(X)$.

Propriété 2.9 : S_0 est un indicateur de puissance

Pour tout processus d'ordre logarithmique X et pour toutes constantes c et d nous avons

- $S_0(c) = c^2$;
- $c^2 < S_0(X) < d^2$ si pour toute réalisation x de X telle que $0 \leq c < |x| < d$;
- $S_0(X) = 0$ si et seulement si $Pr(|X| < \varepsilon) > 0 \forall \varepsilon > 0$, ce qui implique que la nullité de cette puissance quand la masse de probabilité discrète est localisée en zéro.

La démonstration se trouve dans la référence [15].

Propriété 2.10 : Multiplicativité de S_0

Pour tout couple de bruits (X, Y) et pour toute constante c nous avons

- $S_0(X \cdot Y) = S_0(X) \cdot S_0(Y)$;
- $S_0(X/Y) = S_0(X)/S_0(Y)$;
- $S_0(X^c) = S_0(X)^c$.

La démonstration est dans la référence [15].

Propriété 2.11 : L'inégalité de valeur absolue de S_0

Pour tout couple de bruits (X, Y) nous avons

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

La démonstration est dans la référence [15].

2.3.3.3 Puissance géométrique d'une distribution $S\alpha S$

La puissance géométrique d'un bruit α – stable symétrique s'exprime en fonction de α et γ par [65]

$$S_0 = e^{2E(\ln|X|)} = \gamma^2 C_g^{\frac{2}{\alpha}-2} \quad (2.42)$$

où $C_g = e^{C_e} \cong 1.7811$ est une constante qui exprime l'exponentiel de la constante d'**Euler**.

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

2.3.3.4 Le rapport signal sur bruit géométrique

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

$$GSNR = \frac{1}{2C_g} \cdot \frac{E_b}{S_0} \quad (2.43)$$

où E_b est la puissance du signal utile reçu.

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

2.3.4 La simulation d'un bruit α – stable

Pour simuler les lois α – stables, il existe un algorithme développé par **Chambers** et al. [72]. 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 $]-\frac{\pi}{2}, \frac{\pi}{2}[$ est 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

$$\begin{cases} \Phi = \pi U_1 - \frac{\pi}{2} \\ W = \log(1 - U_2) \end{cases} \quad (2.44)$$

nous obtenons bien le résultat désiré.

Deuxième étape

Elle consiste à générer une loi Y α – 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 $\alpha \neq 1$

$$Y = \left(\frac{\sin \alpha(\Phi - \Phi_0)}{(\cos \Phi)^{\frac{1}{\alpha}}} \right) \left(\frac{\cos(\Phi - \alpha(\Phi - \Phi_0))}{W} \right)^{\frac{1-\alpha}{\alpha}} \quad (2.45)$$

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

Pour $\alpha = 1$

$$Y = \frac{2}{\pi} \left(\left(\frac{1}{2}\pi + \beta\Phi \right) \tan \Phi - \beta \log \left(\frac{\frac{1}{2}\pi W \cos(\Phi)}{\frac{1}{2}\pi + \beta\Phi} \right) \right) \quad (2.46)$$

alors le bruit 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

$$\begin{cases} Y = \frac{\sin(\alpha\Phi)}{(\cos \Phi)^{\frac{1}{\alpha}}} \left(\frac{\cos((1-\alpha)\Phi)}{W} \right)^{\frac{1-\alpha}{\alpha}} & \text{si } \alpha \neq 1 \\ Y = \tan \Phi & \text{si } \alpha = 1 \end{cases} \quad (2.47)$$

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

$$Y = \frac{\sin 2\Phi}{\sqrt{\cos \Phi}} \left(\frac{\cos \Phi}{W} \right)^{-\frac{1}{2}} = 2\sqrt{W} \sin \Phi \quad (2.48)$$

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

$$Y = \tan \Phi \quad (2.49)$$

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

2.3.5 Statistiques sur les lois symétriques α – stables

Nous savons bien d'après la propriété 2.5 que pour $0 < \alpha < 2$, une loi α – stable admet des moments d'ordre $r < \alpha$ uniquement.

Nous supposons que nous avons N échantillons ou réalisations x_1, \dots, x_N d'un bruit X qui suit une loi symétrique α – stable $S\alpha S$. Si nous voulons prendre une idée sur la valeur de α , nous avons deux méthodes que nous pouvons les utiliser. La première méthode est basée sur le test du moment $E(|X|)$ et la variance du bruit X s'ils sont finis ou non. Il y a plusieurs façons pour faire ces tests [25]. La plus utilisée est la façon graphique [25]. La deuxième méthode est purement algébrique qui est basée sur l'estimation des paramètres de la distribution du bruit X .

La première méthode : la méthode de test

Soit à tester la variance par un test graphique. 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 nous sommes en présence d'une loi à variance infinie, le tracé diverge et α est strictement être inférieur à 2.

La figure 2.16 présente les tracés de $n = 5000$ observations d'un bruit $S\alpha S$ pour différentes valeurs de α et pour $\gamma = 1$. Nous remarquons que plus α est petit plus le tracé présente d'impulsivité.

La figure 2.17 montre le test graphique de la variance pour les mêmes échantillons du bruit de la figure 2.16. Dans la figure 2.17.a, nous remarquons que lorsque n augmente, la variance converge vers une valeur fixe égale à 2, ce qui prouve que $\alpha = 2$ et le bruit est gaussien avec variance $\sigma^2 = 2\gamma^2 = 2$, c'est-à-dire $\gamma = 1$. Dans les deux figures 2.17.b et 2.17.c, nous observons une divergence de la variance, ce qui signifie que $\alpha < 2$.

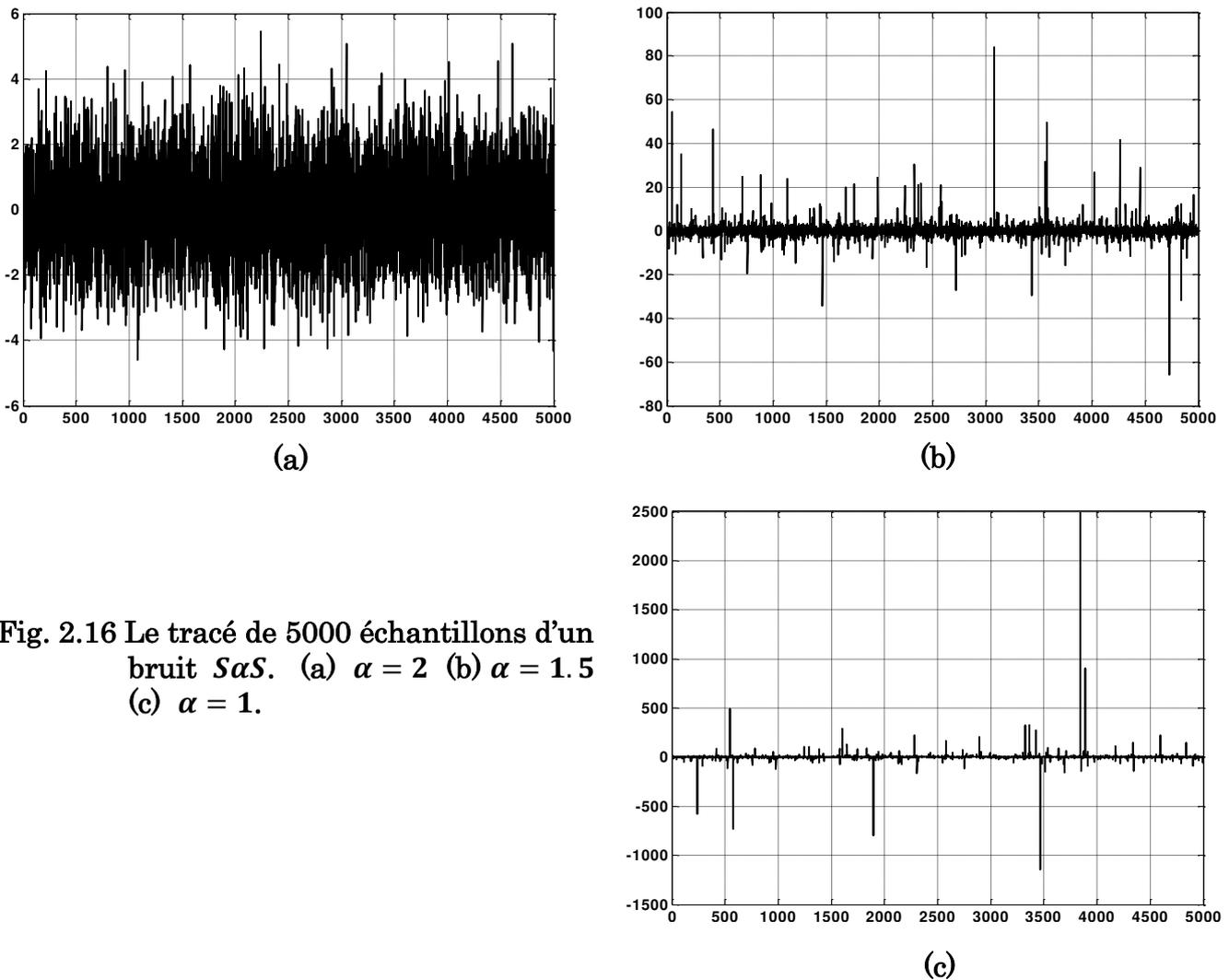


Fig. 2.16 Le tracé de 5000 échantillons d'un bruit $S\alpha S$. (a) $\alpha = 2$ (b) $\alpha = 1.5$ (c) $\alpha = 1$.

La deuxième méthode : la méthode d'estimation

La relation (2.36) nous permet d'écrire pour un bruit X qui suit une loi $S\alpha S$ de paramètres α et γ

$$\alpha = \left(\frac{L_2}{\psi_1} - \frac{1}{2} \right)^{-1/2}, \quad \psi_1 = \frac{\pi^2}{6} \tag{2.50}$$

Si nous approximos l'espérance mathématique par la moyenne arithmétique ou empirique dans les expressions de L_2 nous trouvons

$$L_2 = E \left((\ln|X| - E(\ln|X|))^2 \right) \cong \frac{1}{N} \sum_{n=1}^N \left(\ln|x_n| - \frac{1}{N} \sum_{i=1}^N \ln|x_i| \right)^2$$

et nous obtenons facilement la valeur estimée de α à partir de (2.50). Par exemple, les observations du bruit $S\alpha S$ de la figure 2.16.a donnent une valeur estimée de α égale 2.0221, celles de la figure 2.16.b donnent une valeur de 1.5304 et celles de la figure 2.16.c donnent une valeur de 0.9874.

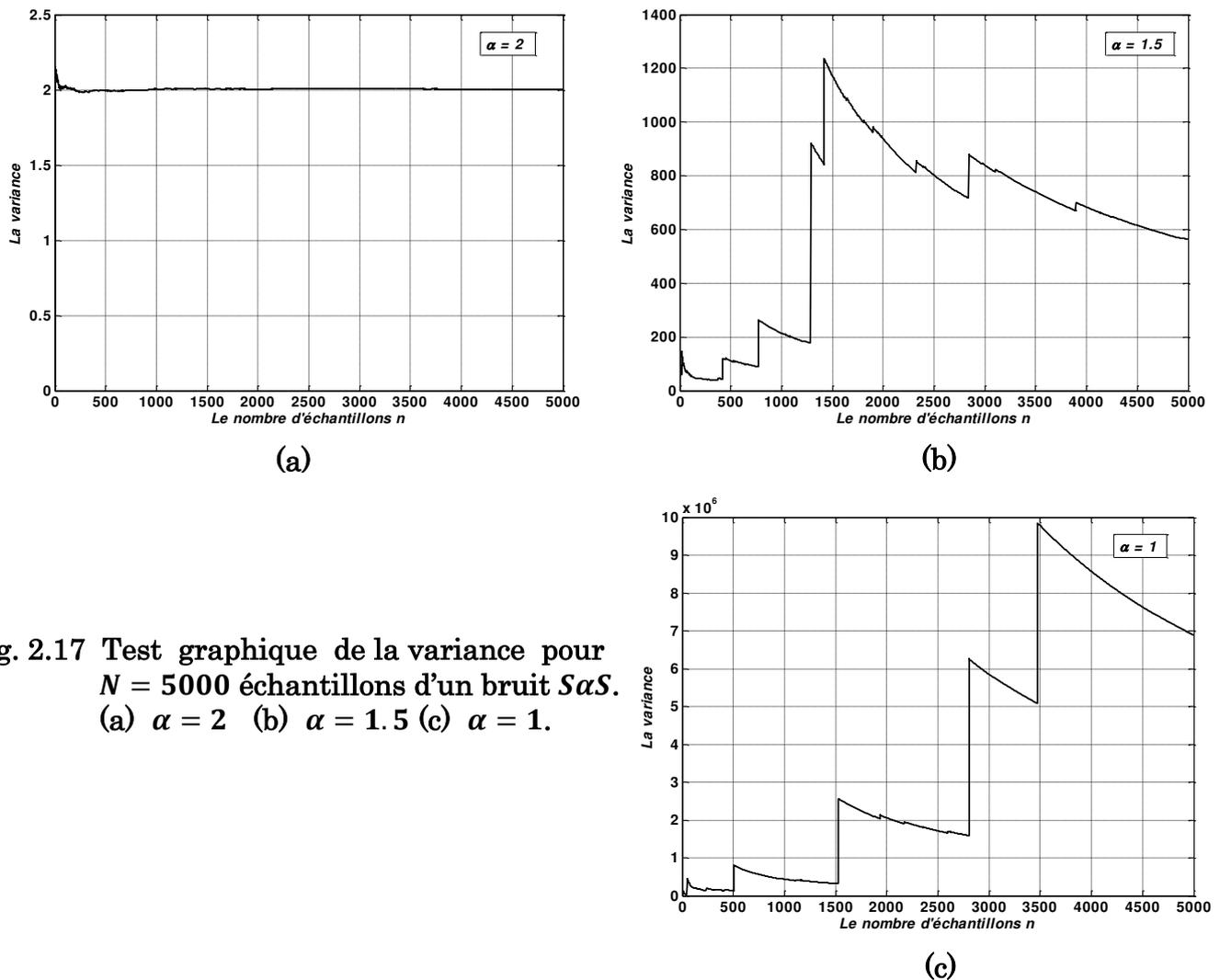


Fig. 2.17 Test graphique de la variance pour $N = 5000$ échantillons d'un bruit $S\alpha S$.
 (a) $\alpha = 2$ (b) $\alpha = 1.5$ (c) $\alpha = 1$.

2.4 Conclusion

Dans ce chapitre, deux grandes parties qui représentent les bruits impulsifs à queues lourdes ont été présentés, à savoir, les distributions gaussiennes généralisées abrégées par GGD ou les distributions GG, et les distributions α – stables.

Dans la première partie, nous avons étudié et comparé différentes techniques pour générer des bruits à partir d'une distribution gaussienne généralisée. Dans ce contexte, nous avons présenté une procédure en trois étapes basée sur une transformation gamma qui donne de bons résultats en temps d'exécution raisonnable. Cette procédure est testé pour $\alpha = 1/3$. Aussi, nous avons montré que la réalisation du bruit gaussien généralisé avec un paramètre de forme $\alpha = 1/2$ peut être obtenue grâce à une transformation non linéaire formulée en fonction de $W_{-1}(z)$ de **Lambert**, avec une précision maîtrisable. Autrement dit, nous avons présenté clairement une méthode qui s'applique sans équivoque à l'appréciation de la fonction $W_{-1}(z)$, qui est très avantageux pour la problématique de synthèse du bruit.

Nous avons en plus exhibé un algorithme d'approximation simple et rapide pour évaluer numériquement la fonction $W_{-1}(z)$ dans l'intervalle $[-1/e, 0[$ avec une précision imposée. Cet algorithme peut être sans ambiguïté programmé dans n'importe quel langage de programmation. Il offre une tactique unique notamment propice pour la simulation de Monte Carlo qui requiert un grand nombre de réalisations du bruit à GGD, qui peut concourir à une meilleure cognition et une meilleure maîtrise des signaux non-gaussiens.

L'algorithme d'approximation que nous venons de décrire n'est plus valable que pour un paramètre de forme $\alpha = 1/2$. Dès lors, sauf quelques cas particuliers, la méthode de l'inversion de la fonction de répartition CDF devient non pratique dans la synthèse des bruits à GGD. Dès lors, pour une valeur du paramètre de forme $\alpha \neq 1/2$ nous utilisons toujours la technique à trois étapes basée sur une transformation gamma. Le temps d'exécution nécessaire pour la génération du bruit à GGD à l'aide de cette technique est raisonnable, mais il reste relativement grand par rapport à celui de la technique de l'inversion de la CDF.

Dans la deuxième partie, nous avons présenté une classe particulière des lois statistiques, à savoir, 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 un bruit α – stable ($0 < \alpha \leq 2$), 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 du bruit. En plus, nous avons expliqué le sens et l'influence des autres paramètres d'un tel bruit, ainsi que l'algorithme de simulation qui permet de le générer.

Nous avons constaté que si $0 < \alpha < 2$, la variance d'un bruit α – stable est infinie. Ce fait entraîne la nullité du rapport signal sur bruit SNR , et pour caractériser l'intensité d'un tel bruit, une des solutions proposées est d'utiliser les moments d'ordre logarithmique [71], où nous avons défini la puissance géométrique d'un bruit et le rapport signal sur bruit géométrique $GSNR$.

L'un des inconvénients d'une loi α – stable réside dans la non-existence d'une expression explicite de sa PDF sauf pour $\alpha = 2$ et $\alpha = 1$. En effet, elle est calculée à partir d'un calcul d'intégrale aux bornes infinies d'une fonction complexe comme le montre l'équation (2.26).

Les algorithmes à entrée-sortie douces SISO utilisés dans le décodage itératif, à savoir l'algorithme de **Viterbi** à sortie douce SOVA et l'algorithme de maximum a posteriori MAP, exigent l'expression de la PDF du bruit qui affecte le signal utile pour calculer une information extrinsèque fiable, dans le but d'avoir des meilleurs performances du décodeur itératif. L'usage de ces algorithmes dans le cas d'un bruit α – stable où $\alpha \neq 1$ et $\alpha \neq 2$ nous oblige alors de passer aux traitements numériques à l'aide de la FFT pour surmonter le problème de non-existence de la forme explicite de la PDF du bruit [73].

Le décodage itératif avec les deux algorithmes SOVA et MAP, fait l'objet du chapitre suivant.

Chapitre 3

Le décodage itératif avec les algorithmes SISO : SOVA et MAP

3.1 Introduction

Un des plus gros problèmes en théorie du codage consiste à élaborer des techniques de décodage efficaces réalisant ou approximant le décodage au maximum de vraisemblance ML pour des codes de grande longueur tels que l'algorithme de **Viterbi**. Ces algorithmes prennent des décisions en tenant compte d'un nombre important de symboles reçus. Clairement, la décision est d'autant plus fiable que le nombre de symboles pris en compte est élevé. En revanche, plus ce nombre est élevé, plus le décodeur est complexe. La place mémoire nécessaire devient rapidement très importante, de même que les temps de calcul correspondants. Heureusement, un processus de décodage itératif d'une complexité moindre peut être appliqué tout en offrant des performances quasi-optimales. C'est pourquoi l'avènement du processus de décodage itératif, proprement dit turbo décodage, pour les codes concaténés parallèles, il y a deux décennies [74], se classe parmi les plus importants développements dans les communications numériques modernes dans le dernier demi-siècle.

Récemment, il a été démontré que les systèmes de décodage itératif peuvent fonctionner à des rendements très proches de la limite de **Shannon**, imposée par le théorème du codage sur canaux bruités [75], avec toutefois une complexité raisonnable. C'est pour cette raison une grande part de la recherche est focalisée sur l'évaluation des performances du décodage itératif pour de futurs DCS filaires et non-filaires, en particulier, les systèmes avec codes LDPC irréguliers et les turbo codes qui sont des candidats prometteurs pour de futures applications. De plus, le principe turbo a été

étendu à l'ensemble des fonctionnalités constituant le récepteur d'un DCS telles que la turbo détection dans les domaines des canaux sélectifs en fréquence ou des canaux MIMO, la turbo égalisation ou la turbo modulation [76].

De manière générale, le décodage itératif consiste à décoder un code par étapes successives à l'aide de plusieurs décodeurs à faible coût au lieu de décoder le code avec un seul décodeur complexe. Dans chaque itération il y a un échange mutuel d'information dite extrinsèque entre les décodeurs élémentaires constitutifs du décodeur itératif. Le module de décodage associé à chaque décodeur constitutif est constitué par un algorithme à entrée-sortie douces SISO.

Puisque nous sommes intéressé au décodage itératif des CTC basés sur les codes convolutifs RSC, comme il est indiqué à la fin du paragraphe 1.6 dans le chapitre 1, alors, nous présentons dans ce chapitre les deux algorithmes SISO généralement utilisés pour ce type de décodage, à savoir : l'algorithme de **Viterbi** à sortie douce SOVA et l'algorithme de maximum a posteriori MAP appelé aussi BCJR [77][78][79][80][81].

3.2 La métrique de vraisemblance logarithmique (de corrélation)

Nous définissons dans ce paragraphe une grandeur importante dite la métrique de vraisemblance logarithmique ou la métrique de corrélation [82]. Cette métrique sera utilisée par la suite dans l'algorithme SOVA.

Une fois codés, les bits d'information sont transmis sur le canal. A la réception, en partant des symboles reçus qui comportent, 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 [83].

Dans un DCS 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 le cas d'un canal à bruit de type AWGN à moyenne nulle et de variance σ^2 , et le signal à transmettre est modulé avec la modulation BPSK avec une énergie égale à 1 (voir chapitre 1), le signal reçu s'écrit

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

où $Y_k = 1 - 2X_k$ est le symbole transmis, $X_k \in \{0, 1\}$ est la sortie du turbo code et W_k est le bruit AWGN de PDF donnée par

$$f_W(w) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-w^2}{2\sigma^2}\right) \quad (3.2)$$

Les densités de probabilité de transition du canal s'écrivent alors

$$P(R_k/Y_k \text{ émis}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(R_k - Y_k)^2}{2\sigma^2}\right) \quad (3.3)$$

Pour tirer profit de l'information portée par le symbole reçu R_k dans la sortie du canal, nous devons 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. Nous parlons donc d'un canal à décision douce, souple ou soft.

Au niveau du récepteur nous devons 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 du 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** [27], ça veut dire il maximise le logarithme népérien de cette probabilité. Nous éliminons toutes les constantes dans l'expression de $\ln(P(R_k/Y_k \text{ émis}))$ nous trouvons l'expression à maximiser qui est $-(R_k - Y_k)^2$. 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. Nous pouvons 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 nous utilisons la règle suivante « **Sachant que $Y_k = 1 - 2X_k$, 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)** » [82].

Nous pouvons montrer facilement que cette métrique de corrélation reste valable si le bruit envisagé est non-gaussien comme le cas des bruits GG et SaS (voir le paragraphe 4.3.1.3 du chapitre 4).

3.3 Les algorithmes de décodage SISO

Historiquement de nombreux algorithmes existent pour décoder un code convolutif. Les premiers algorithmes de **Fano** [84] puis **Viterbi** [85] étaient à entrées et sorties binaires. L'algorithme de **Viterbi**, le plus efficace des deux, a d'abord été modifié

pour accepter des entrées douces ou souples [86] avec à la clé une amélioration du décodage. La concaténation des codes a ensuite poussé l'apparition d'algorithmes de décodage à entrées et sorties pondérées SISO tels que l'algorithme SOVA [87] et l'algorithme MAP aussi appelé BCJR [88]. Ces deux algorithmes sont couramment utilisés pour le décodage de codes représentés en treillis, particulièrement les codes convolutifs. La différence entre ces deux algorithmes est que le SOVA calcule le chemin dont la vraisemblance est maximale et en plus à chacun des symboles associés aux transitions du chemin choisi, il fait correspondre une information de fiabilité en approximant sa probabilité a posteriori. Il s'agit donc d'un algorithme de décodage à ML par séquence ou par chemin sous optimal. Par contre l'algorithme MAP ou BCJR, il calcule de manière exacte la probabilité a posteriori des symboles présents sur les branches. Il s'agit donc aussi d'un algorithme à décision MAP qui calcule la probabilité de chaque symbole à partir des probabilités de tous les chemins possibles dans le treillis entre l'état initial et l'état final. Il est un algorithme optimal d'une maximisation symbole par symbole.

Puisque notre travail est basé sur les CTC présentés dans le chapitre 1, nous présentons dans ce paragraphe les deux algorithmes SISO déjà cités SOVA et MAP qui seront utilisés pour le décodage itératif des codes CTC.

3.3.1 Algorithme de Viterbi à sortie douce SOVA

3.3.1.1 Algorithme de Viterbi classique avec la métrique de corrélation

Parmi les algorithmes les plus utilisés dans le décodage des codeurs convolutifs est celui de **Viterbi** classique qui est surtout basé sur le diagramme en treillis du codeur. Cet algorithme suit la règle de décision du ML. Le décodage optimal au sens du ML d'un code convolutif est obtenu par la recherche dans le treillis du code de la séquence la plus probable au moyen de cet algorithme [89][86][85]. Selon la nature de l'estimation fournie par le démodulateur, dure ou douce, les métriques utilisées seront respectivement une métrique de **Hamming** ou une métrique euclidienne (métrique de corrélation). Une métrique cumulative dite métrique de branches est évaluée le long du treillis du code. Nous appelons survivants les chemins conservés à une certaine étape du décodage. Il y a un unique survivant par état, et autant de survivants que d'états. Chaque survivant est associé à une valeur de la métrique. A un instant donné, le choix entre deux chemins possibles se fait en conservant celui de métrique maximale. Si deux ou plusieurs chemins

ont la même métrique, nous en choisissons aléatoirement un survivant parmi les candidats possibles.

Pour présenter clairement l'algorithme de **Viterbi** classique avec la métrique euclidienne de corrélation définie dans le paragraphe précédent, nous supposons que le canal de transmission soit de type stationnaire gaussien et sans mémoire avec une modulation BPSK. Le codeur convolutif a une longueur de contrainte L . L'algorithme peut être partagé en deux parties

- La première s'étend du début du treillis, $k = 1$ jusqu'au niveau $k = L = \nu + 1$, où L est la longueur de contrainte et ν est le nombre des registres (mémoire du codeur). Le décodage consiste à attribuer une métrique nulle à l'état initial, et à chaque autre état $S^{(m_2)}$, atteint par la transition d'un état $S^{(m_1)}$, une métrique, qui est la somme des métriques à l'état $S^{(m_1)}$ et la métrique de transition ou de branche. La métrique de branche à un niveau k du treillis entre deux états $S^{(m_1)}$ et $S^{(m_2)}$ notée $BM_{(m_1)}^{(m_2)}$, est la somme des métriques de corrélation des symboles reçus à ce niveau.
- La deuxième partie s'étend du niveau $k = L + 1$ jusqu'au niveau $k = \left\lfloor \frac{N}{m_e} \right\rfloor + L - 2$ où N est la longueur du message, et m_e est le nombre des entrées du codeur.

Le décodage consiste à choisir la branche pour laquelle la métrique au nœud $S^{(m)}$ sera maximale et éliminer toutes les autres. La branche choisie est appelée *survivant* et les autres branches sont appelées *concurrents*. Si le choix de plusieurs branches donne la même métrique maximale au nœud $S^{(m)}$, alors peu importe, nous choisirons une de ces branches. Cette opération est répétée avec tous les 2^ν états, et cela pour tous les niveaux de cette partie. A la fin de cette partie nous obtiendrons 2^ν chemins, desquels le décodeur choisira celui qui a la métrique de corrélation la plus grande.

En notant, qu'il est nécessaire de stoker 2^ν chemins jusqu'à la fin, où la décision sera prise, nous se rendons compte que le décodeur devient très complexe et nécessite une capacité mémoire importante pour une longueur de contrainte importante.

Toutefois, lorsque nous examinons les différents survivants à une étape k , nous remarquons qu'avec une grande probabilité, ils ont tous le même passé entre les étapes 1 et $k - dp + 1$ où dp est la profondeur de décision [90].

Nous pouvons par conséquent décider définitivement à l'étape k toutes les données émises jusqu'à l'étape $k - dp$. Comme règle empirique, nous pouvons fixer le paramètre dp égal à 5 ou 6 fois le nombre d'états du décodeur [90]. Avec ces valeurs les performances du décodeur sont quasiment les mêmes que lorsque la décision se fait en fin de la séquence.

Cette stratégie permet de réduire la complexité du décodeur et le retard de décision. La complexité d'un décodeur de **Viterbi** est proportionnelle au nombre d'états de treillis. Celui-ci croît exponentiellement avec la longueur de contrainte. Pour cette raison les codes utilisés en pratique ont une longueur de contrainte inférieure à 10 [90].

Pour un CTC la profondeur de décision dp peut être prise égale à la taille de l'entrelaceur si cette dernière n'est pas importante.

Exemple 3.1

Considérons le codeur RSC de la figure 1.4.a dont la représentation en treillis est donnée par la figure 1.4.b, le décodage à ML consiste à choisir le chemin correspondant à la séquence des symboles qui possède une métrique de corrélation maximale.

Soit $d = (11011)$ la séquence à envoyer. La sortie du codeur RSC (5,7) est $X = (1110001011)$. La sortie du modulateur est $Y = (-1 - 1 - 1 + 1 + 1 + 1 - 1 + 1 - 1 - 1)$. Cette dernière séquence est envoyée dans le canal bruité. Nous supposons que nous avons trouvé à la sortie du canal les observations $R = (-4 - 3 - 3 - 2 + 2 + 4 - 3 + 3 - 3 + 1)$.

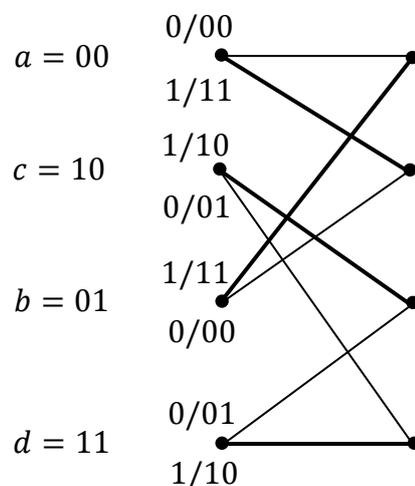
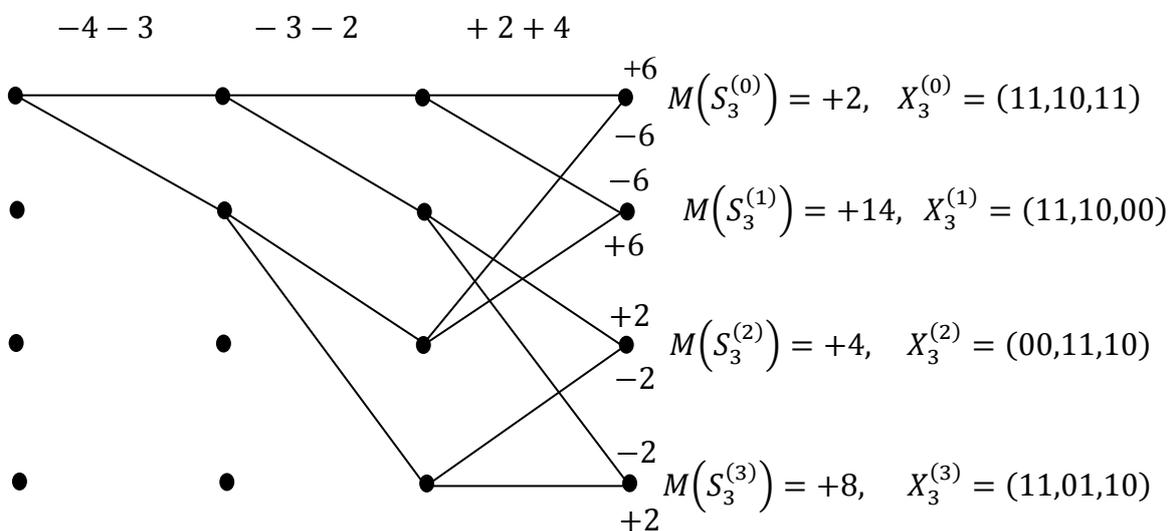
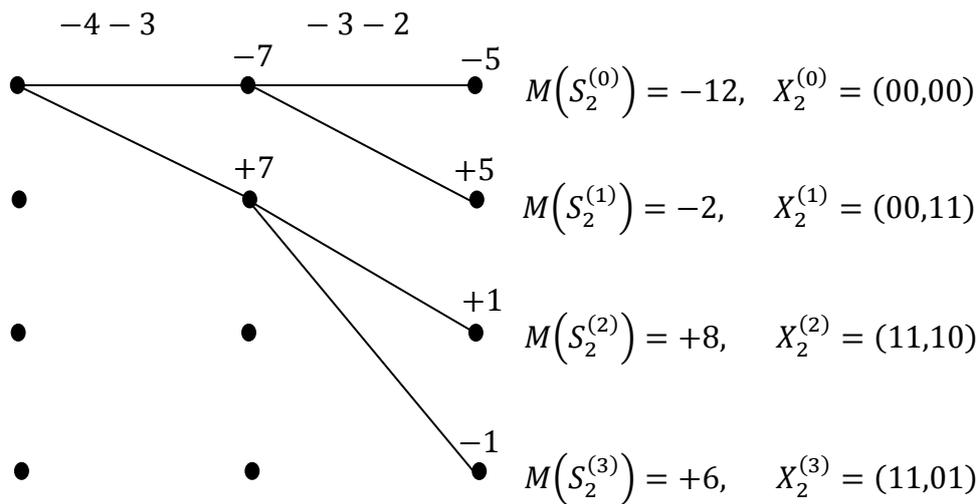
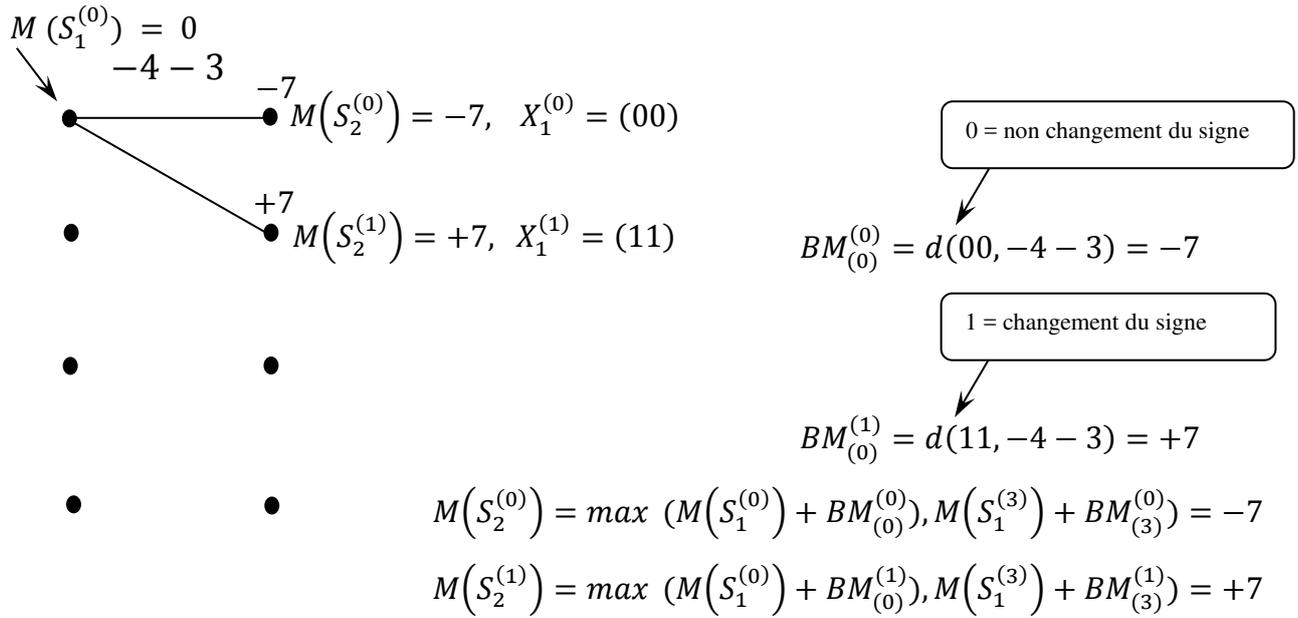


Fig. 3.1 Le treillis simplifié du RSC (5,7).



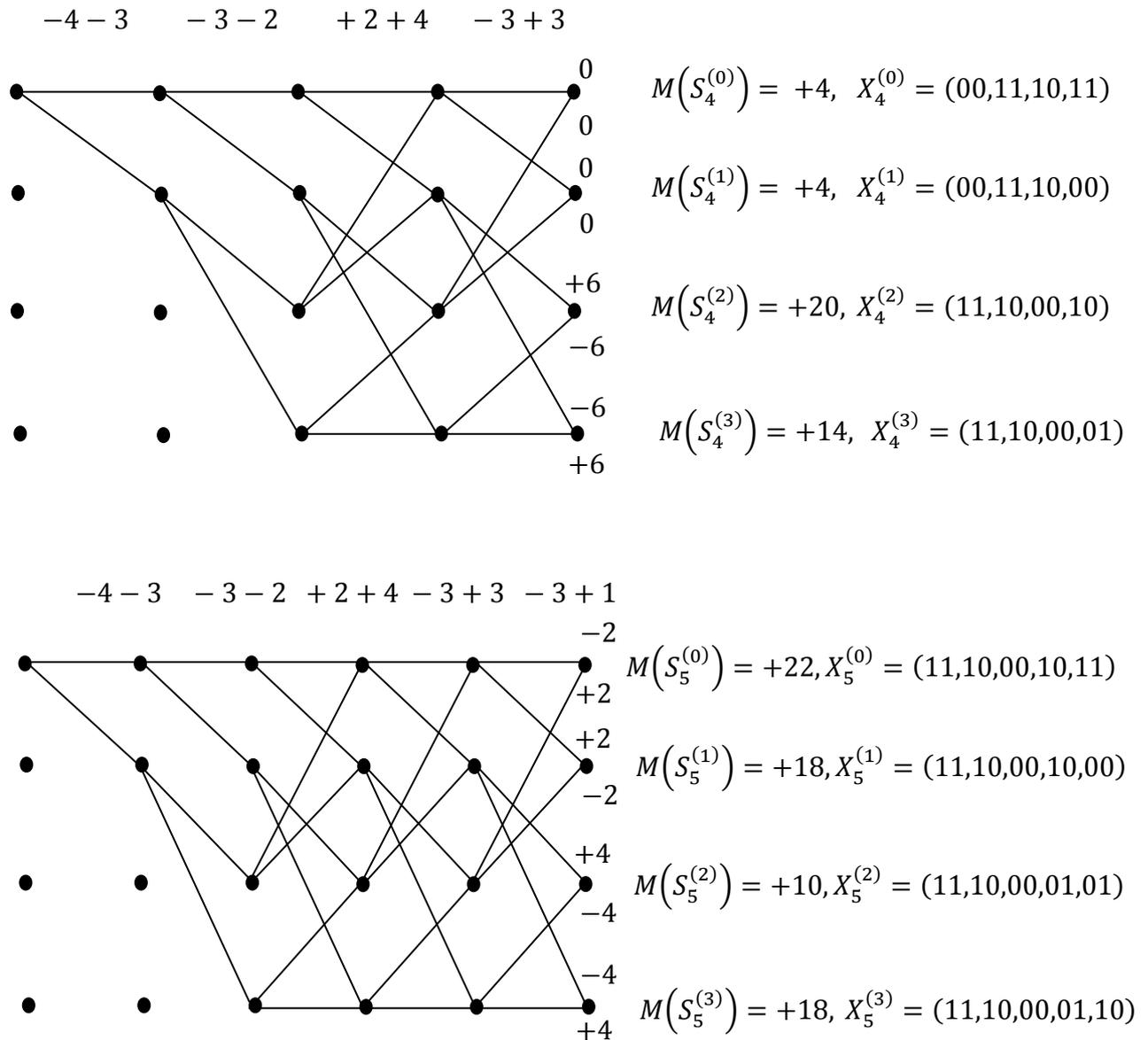


Fig. 3.2 L'algorithme de Viterbi classique avec métrique de corrélation pour le RSC (5,7).

Etat/Etape	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
$S_k^{(0)}$	0	-7	-12	2	4	22
$S_k^{(1)}$	∅	7	-2	14	4	18
$S_k^{(2)}$	∅	∅	8	4	20	10
$S_k^{(3)}$	∅	∅	6	8	14	18

Tab. 3.1 Tableau des survivants du treillis de la figure 3.2.

D'après le treillis illustré ci-dessus, le survivant est le chemin qui passe par les états : $S_1^{(0)} S_2^{(1)} S_3^{(2)} S_4^{(1)} S_5^{(2)} S_6^{(0)}$, qui correspond à une séquence de sortie du codeur $X_5^{(0)} = (11,10,00,10,11)$ et une séquence d'entrée estimée $\hat{d} = (11011)$, qui est exactement la séquence envoyée.

Il est important de signaler qu'un décodeur à décision ferme donne à partir de la séquence reçue R la séquence estimée $\hat{Y} = (-1 - 1 - 1 \underline{-1} + 1 + 1 - 1 + 1 - 1 \underline{+1})$ qui contient deux erreurs mentionnées en gras.

3.3.1.2 Algorithme de Viterbi à sortie douce SOVA

L'algorithme SOVA est une modification de l'algorithme de **Viterbi** classique pour lui permettre d'être utilisé comme élément de décodage dans le turbo décodeur [91][81]. Cette modification réside en deux points

- Le calcul des métriques de branches est modifié pour permettre de prendre en compte les informations a priori.
- L'algorithme est modifié de façon qu'il produise une information de sortie douce sous la forme d'information a posteriori d'un rapport logarithmique de vraisemblance LLR pour chaque bit décodé.

Autrement dit, l'algorithme de **Viterbi** est modifié afin de fournir à sa sortie une valeur de confiance ou de fiabilité (approximant une probabilité a posteriori) associée à chaque bit décodé. La valeur des bits décodés est toujours donnée par le chemin ξ ayant la métrique de branches (ou la métrique cumulée) maximale $M(\xi)$ fournie par l'algorithme classique.

La probabilité que le chemin ξ soit émis sachant la séquence R soit reçue est proportionnelle à la probabilité conditionnelle de la séquence R soit reçue sachant le chemin ξ soit émis. Notons $M(1)$ la métrique de branches maximale jusqu'à l'étape k dans le treillis avec le bit estimé de d_k est $\hat{d}_k = 1$. De même, notons $M(0)$ la métrique de branches maximale jusqu'à l'étape k dans le treillis avec le bit estimé de d_k est $\hat{d}_k = 0$. Alors le rapport logarithmique de vraisemblance du bit d_k dans le chemin ξ est approximé par [77]

$$LLR(d_k) \triangleq \ln \left(\frac{P(d_k = 1|R)}{P(d_k = 0|R)} \right) \cong M(1) - M(0) \quad (3.4)$$

Cette valeur s'appelle la décision douce ou souple correspondant au bit d_k . La relation (3.4) contient trois membres. Le troisième membre est calculé à partir du treillis du codeur en appliquant l'algorithme de **Viterbi** classique avec la métrique de corrélation

euclidienne. Quant au deuxième membre, nous pouvons le développer de plus dans le but de calculer ce que nous appelons l'information extrinsèque. Pour ce but, nous devons passer par les fondations mathématiques du décodage à décisions douces qui reposent sur le théorème de **Bayes**.

Soient A_i , $i = 1, 2, \dots, K$, les signaux à transmettre d'un ensemble donné de signaux, et soient $P(A_i)$ les probabilités a priori de ces signaux. Soit B le signal reçu qui n'est qu'un des signaux A_i corrompu par du bruit. B est une variable aléatoire continue de probabilité $P(B)$. L'expression $P(A_i|B)$ est la probabilité a posteriori APP du signal 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 [89]

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} \quad (3.5)$$

où $P(B|A_i)$ est la probabilité conditionnelle du signal observé B , et $P(A_i)$ est la probabilité a priori du $i^{\text{ème}}$ signal transmis. Le dénominateur de l'équation (3.5) est donné par

$$P(B) = \sum_{i=1}^K P(B|A_i)P(A_i) \quad (3.6)$$

Notons que $P(B|A_i)$ est appelée aussi la fonction de vraisemblance. Le critère de décision qui consiste à prendre le maximum de $P(B|A_i)$ parmi les K signaux est le principe du ML si les signaux sont équiprobables.

Si la distribution des signaux A_i est uniforme, c'est-à-dire 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 du ML.

Dans notre étude les signaux A_i sont les signaux binaires d_k , donc $K = 2$, et le signal reçu B représente la séquence R observée à la sortie du canal. Alors l'équation (3.5) devient

$$P(d_k = i|R) = \frac{P(R|d_k = i)P(d_k = i)}{P(R)}, \quad i = 0, 1 \quad (3.7)$$

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

$$P(d_k = 1|R) \underset{H_1}{\overset{H_0}{\leq}} P(d_k = 0|R) \quad (3.8)$$

Dans l'équation (3.8) 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 qui donne \hat{d}_k au sens du MAP est alors la suivante

$$\hat{d}_k = \begin{cases} 1 & \text{si } P(d_k = 1|R) > P(d_k = 0|R) \\ 0 & \text{sinon.} \end{cases} \quad (3.9)$$

La décision douce est définie par le rapport logarithmique de vraisemblance LLR comme suit (voir l'équation (3.4))

$$LLR(d_k) \triangleq \ln \left[\frac{P(d_k = 1|R)}{P(d_k = 0|R)} \right] \quad (3.10)$$

Il s'ensuit, d'après (3.9) et (3.10) que le signe de la décision douce détermine la décision ferme et que la valeur absolue de la décision douce détermine la fiabilité de cette décision. Il est important à signaler que le LLR obtenu est valable pour un seul bit ou symbole.

Si nous notons par \overline{R}_k toute la séquence reçue R sauf R_k qui est le symbole reçu à l'instant k , nous pouvons écrire grâce à la loi de **Bayes**

$$P(R|d_k = i) = P(\overline{R}_k | R_k, d_k = i) \cdot P(R_k | d_k = i), \quad i = 0, 1 \quad (3.11)$$

et avec la relation (3.7), le rapport $LLR(d_k)$ peut être décomposé sous la forme [82][92]

$$LLR(d_k) = \underbrace{\ln \left(\frac{P(R_k | d_k = 1)}{P(R_k | d_k = 0)} \right)}_{LLR_c(R_k)} + \underbrace{\ln \left(\frac{P(\overline{R}_k | R_k, d_k = 1)}{P(\overline{R}_k | R_k, d_k = 0)} \right)}_{LLR_e(\overline{R}_k)} + \underbrace{\ln \left(\frac{P(d_k = 1)}{P(d_k = 0)} \right)}_{LLR_a(d_k)} \quad (3.12)$$

où $LLR_c(R_k) = \ln \left(\frac{P(R_k | d_k = 1)}{P(R_k | d_k = 0)} \right)$ est le rapport logarithmique de vraisemblance du canal,

$LLR_e(\overline{R}_k) = \ln \left(\frac{P(\overline{R}_k | R_k, d_k = 1)}{P(\overline{R}_k | R_k, d_k = 0)} \right)$ est le rapport logarithmique de vraisemblance représentant

la connaissance acquise grâce au processus de décodage et $LLR_a(d_k) = \ln \left(\frac{P(d_k = 1)}{P(d_k = 0)} \right)$ est le rapport logarithmique de vraisemblance a priori.

Finalement, nous pouvons écrire la sortie douce $LLR(d_k)$ d'un décodeur SOVA comme suit [82][92]

$$LLR(d_k) = LLR_c(R_k) + LLR_e(\overline{R}_k) + LLR_a(d_k) = LLR_c(R_k) + \Psi(d_k) \quad (3.13)$$

où la nouvelle quantité $\Psi(d_k) = LLR_e(\overline{R}_k) + LLR_a(d_k)$ est appelée **information extrinsèque**, qui est indépendante de R_k .

Le critère de décision au sens du MAP peut alors s'écrire

$$\hat{d}_k = \begin{cases} 1 & \text{si } LLR(d_k) > 0 \\ 0 & \text{sinon.} \end{cases} \quad (3.14)$$

Il s'ensuit que le signe de la décision douce détermine la décision ferme et que la valeur absolue de la décision douce détermine la fiabilité de cette décision.

En particulier, si nous considérons un bruit de type AWGN de moyenne nulle et de variance σ^2 qui sera utilisé par la suite dans le chapitre 4, le rapport logarithmique de vraisemblance du canal a pour expression

$$LLR_c(R_k) = \ln \left(\frac{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(R_k + 1)^2}{2\sigma^2}\right)}{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(R_k - 1)^2}{2\sigma^2}\right)} \right) = -\frac{2}{\sigma^2} R_k \quad (3.15)$$

et l'information extrinsèque $\Psi(d_k)$ devient alors

$$\Psi(d_k) = \frac{2}{\sigma^2} R_k + LLR(d_k) \quad (3.16)$$

Remarque 3.1

Généralement $\Psi(d_k)$ et $LLR(d_k)$ sont de même signe [2], alors

$$\hat{d}_k = \begin{cases} 1 & \text{si } \Psi(d_k) > 0 \\ 0 & \text{sinon.} \end{cases} \quad (3.17)$$

Calcul pratique des $LLR(d_k)$

L'approximation du $LLR(d_k)$ indiquée dans l'équation (3.4) est utilisable lorsque nous appliquons l'algorithme de **Viterbi** classique avec la métrique de corrélation dans un seul sens du treillis (sens d'aller seulement). Pour améliorer amplement cette approximation, nous devons utiliser en plus le sens de retour du treillis, c'est-à-dire nous devons appliquer l'algorithme de **Viterbi** classique dans les deux sens du treillis, aller et retour après avoir fermé ce treillis. La fermeture du treillis pour l'algorithme SOVA est bien illustré dans le paragraphe 4.2.5.1 dans le chapitre 4. Avec les mêmes données du paragraphe 3.3.1.1, l'algorithme SOVA à utiliser consiste à appliquer l'algorithme de **Viterbi** classique dans les deux sens du treillis, pour calculer le $LLR(d_k)$ de chaque bit d'information d_k , en respectant les étapes suivantes

- **Etape 1 : Le sens d'aller ou direct** : Dans ce sens nous appliquons l'algorithme de **Viterbi** classique sur le treillis du codeur de l'étape $k = 1$ jusqu'à l'étape finale

$k = N + 1$, où N est la longueur de l'information envoyée. Cette étape a deux buts. Le premier but est de calculer et sauvegarder toutes les métriques de branche qui lient les états entre eux. Le deuxième but est de trouver le survivant qui correspond à la métrique maximale M_{max} . Les métriques sauvegardées sont utilisées par la suite dans le calcul des LLR.

- **Etape 2 : Le sens de retour ou inverse** : Le but de cette étape est le calcul des contributions minimales dues à la prolongation des chemins (à chaque niveau de profondeur et pour chaque nœud). Nous appliquons toujours l'algorithme de **Viterbi** classique sur le treillis du codeur mais dans le sens inverse, donc de l'étape finale $k = N + 1$ jusqu'à l'étape initiale $k = 1$. Dans cette deuxième phase, nous ne calculons et ne sauvegardons que les métriques des états dans toutes les étapes du treillis, parce que les métriques de branche sont déjà calculées et sauvegardées dans l'étape 1.
- **Etape 3 : Calcul des $LLR(d_k)$** : Le survivant trouvé dans l'étape 1 qui possède une métrique notée M_{max} , donne la séquence estimée de la vraie séquence d'information. Le $LLR(d_k)$ d'un bit d_k est pratiquement donné par [82][93]

$$LLR(d_k) = M(1) - M(0), \quad M(d_k) = M_{max} \quad (3.18)$$

où $M(i)$ est la métrique de branches maximale jusqu'à l'étape k avec la valeur estimée de d_k qui est $\hat{d}_k = i$ où $i = 0, 1$. La branche du survivant dans l'étape k , qui correspond au bit d_k , possède plusieurs concurrents qui correspondent au complément de d_k noté $d_k \oplus 1$. La métrique de branches du meilleur concurrent est donnée par la relation [82][93][94]

$$M(d_k \oplus 1) = \max_{m1, m2} \left\{ M_f(S_{k-1}^{(m1)}) + BM(d_k \oplus 1) + M_b(S_k^{(m2)}) \right\} \quad (3.19)$$

où $m1, m2 \in \{0, 1, 2, \dots, 2^v - 1\}$, $M_f(S_{k-1}^{(m1)})$ est la métrique de survivant dans l'état $S_{k-1}^{(m1)}$ à l'étape $k - 1$ dans le sens d'aller (direct), $BM(d_k \oplus 1)$ est la métrique de branche correspond au complément de d_k qui lie les deux états $S_{k-1}^{(m1)}$ et $S_k^{(m2)}$ et $M_b(S_k^{(m2)})$ est la métrique de survivant dans l'état $S_k^{(m2)}$ à l'étape k dans le sens de retour (inverse).

L'algorithme SOVA à utiliser nécessite alors d'appliquer l'algorithme de **Viterbi** classique avec la métrique de corrélation deux fois : la première est dans le sens d'aller

(forward) et la seconde est dans le sens inverse (backward). C'est pourquoi cet algorithme est parfois appelée SOVA bidirectionnel.

Pour tirer au clair le calcul pratique du $LLR(d_k)$ nous donnons l'exemple suivant.

Exemple 3.2

Nous considérons les mêmes données de l'exemple 3.1. Le treillis fermé illustré ci-dessous dans la figure 3.3, contient toutes les métriques de branches dans les deux sens. Ce qui sont indiquées en orange représentent les métriques des branches, celles indiquées en noir sont les métriques cumulatives sur les états en aller et celles indiquées en bleu sont les métriques cumulatives sur les états en retour. Le chemin dessiné en rouge représente le survivant. Une branche fine ou simple correspond à un bit d'information égal à 0 et une branche en gras correspond à un bit d'information égal à 1.

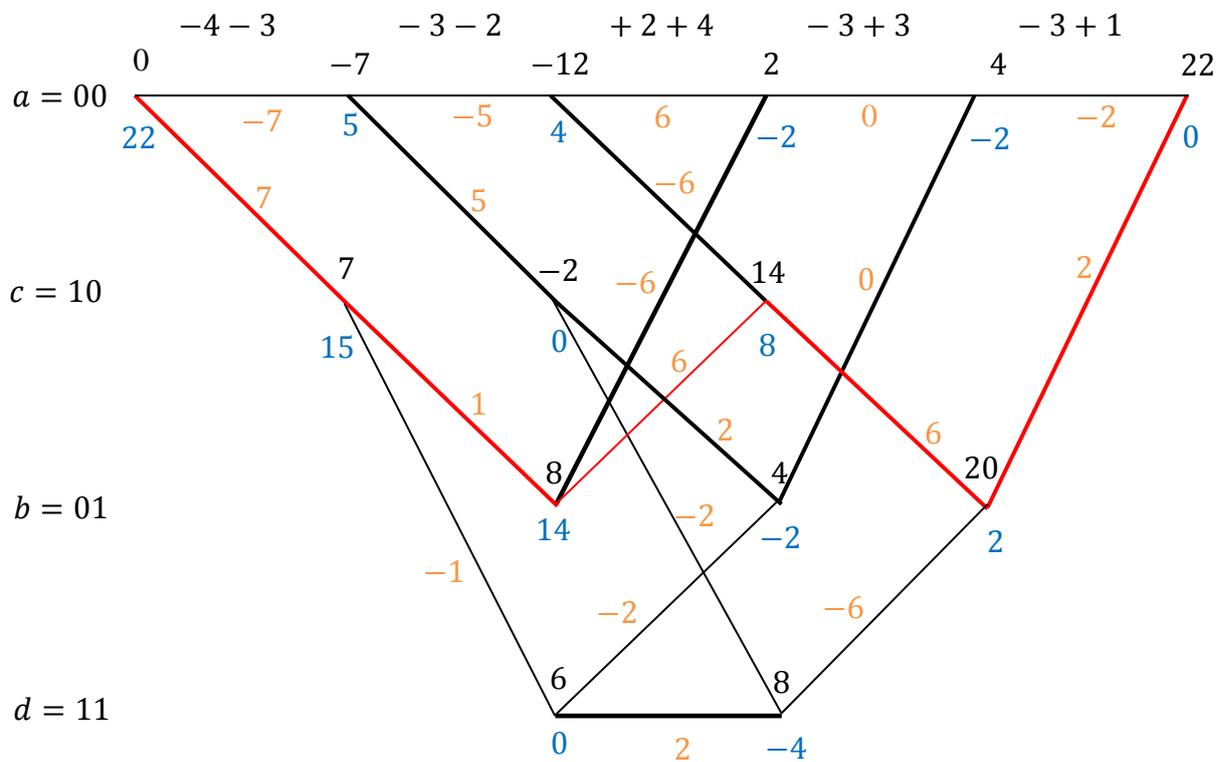


Fig. 3.3 Calcul des métriques de branches dans les deux sens du treillis.

Le survivant est le chemin qui passe par les états : $S_1^{(0)}S_2^{(1)}S_3^{(2)}S_4^{(1)}S_5^{(2)}S_6^{(0)}$, qui correspond à une séquence d'entrée estimée $\hat{d} = (11011)$, et sa métrique $M_{max} = 22$.

Pour le premier bit $d_1 = 1$, $M(1) = M(d_1) = 22$ et $M(0) = M(d_1 \oplus 1) = 0 - 7 + 5 = -2$, donc $LLR(d_1) = 22 - (-2) = 24$.

Pour le deuxième bit $d_2 = 1$, $M(1) = M(d_1) = 22$ et $M(0) = M(d_1 \oplus 1) = \max(7 - 1 + 0, -7 - 5 + 4) = 6$, donc $LLR(d_2) = 22 - 6 = 16$.

Pour le troisième bit $d_3 = 0$, $M(1) = M(d_1 \oplus 1) = \max(0, 2, -10, 8) = 8$ et $M(0) = M(d_1) = 22$, donc $LLR(d_3) = 8 - 22 = -14$.

Pour le quatrième bit $d_4 = 1$, $M(1) = M(d_1) = 22$ et $M(0) = M(d_1 \oplus 1) = \max(0, 2) = 2$, donc $LLR(d_4) = 22 - 2 = 20$.

Pour le cinquième bit $d_5 = 1$, $M(1) = M(d_1) = 22$ et $M(0) = M(d_1 \oplus 1) = 2$, donc $LLR(d_5) = 22 - 2 = 20$.

Par conséquent, $LLR(d) = (24, 16, -14, 20, 20)$ et nous pouvons vérifier facilement que $d_k = (1 + \text{sign}(LLR(d_k)))/2$.

3.3.2 Algorithme de maximum a posteriori MAP (ou BCJR)

L'algorithme BCJR, nommé d'après ses auteurs [88], a été présenté en 1974 comme une solution alternative pour le décodage de codes convolutifs. Le critère de décodage de cet algorithme est la minimisation de la probabilité d'erreur de chaque symbole ou chaque bit du message. Il s'agit d'un algorithme optimal pour le décodage des codes convolutifs. Tandis que l'algorithme SOVA qui minimise la probabilité d'erreur par mot ou par séquence, est un algorithme est sous-optimal.

L'algorithme BCJR est ainsi capable de fournir, en plus des décisions fermes, une estimation de la fiabilité du décodage, qui se base sur la probabilité a posteriori du bit concerné. Par conséquent, dans certaines publications, la méthode est citée aussi sous le terme d'algorithme Maximum a Posteriori MAP.

La règle du maximum a posteriori appliquée sur un symbole d'une séquence reçue, signifie de prendre le maximum des probabilités a posteriori APP de ce symbole 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 du ML. En effet, il suffit d'appliquer la règle de **Bayes** afin de prouver l'identité entre les deux règles (voir le paragraphe 3.3.1.2). Avec l'algorithme

MAP (ou BCJR) à la sortie du décodeur, les bits décodés ne sont pas comparés 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 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.

Comme dans l'algorithme SOVA, la décision ferme optimale qui donne \hat{d}_k au sens du MAP est alors la suivante

$$\hat{d}_k = \begin{cases} 1 & \text{si } P(d_k = 1|R) > P(d_k = 0|R) \\ 0 & \text{sinon.} \end{cases} \quad (3.20)$$

et la décision douce ou souple est définie par

$$LLR(d_k) \triangleq \ln \left[\frac{P(d_k = 1|R)}{P(d_k = 0|R)} \right] \quad (3.21)$$

Ce rapport logarithmique est valable pour un seul bit ou symbole. Il faut noter que le signe du $LLR(d_k)$ détermine la décision ferme optimale au sens du MAP, et que sa valeur absolue détermine la fiabilité de cette décision.

Exactement comme dans le cas de l'algorithme SOVA, le $LLR(d_k)$ peut s'écrire sous la forme de la somme du rapport logarithmique de vraisemblance du canal $LLR_c(R_k)$ et l'information extrinsèque $\Psi(d_k)$ du bit d_k qui est indépendante de R_k , où R_k est le symbole observé à la sortie du canal à l'instant k . Nous écrivons alors

$$\Psi(d_k) = LLR(d_k) - LLR_c(R_k) \quad (3.22)$$

sachant que

$$LLR_c(R_k) = \ln \left(\frac{P(R_k|d_k = 1)}{P(R_k|d_k = 0)} \right) \quad (3.23)$$

Nous pouvons calculer le $LLR_c(R_k)$ pour les trois types de bruit qui seront utilisés par la suite avec l'algorithme MAP dans notre simulation numérique (voir chapitre 4).

- Pour un bruit GG de moyenne nulle, de variance σ^2 et de PDF donnée par (2.1), nous trouvons

$$LLR_c(R_k) = \left(\sigma^2 \frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)} \right)^{-\alpha/2} \cdot (|R_k - 1|^\alpha - |R_k + 1|^\alpha) \quad (3.24)$$

- Pour un bruit gaussien de moyenne nulle et de variance σ^2 , il suffit de mettre $\alpha = 2$ dans l'équation (3.24), nous trouvons l'équation (3.15).

- Pour un bruit impulsif $S\alpha S$ de **Cauchy** de paramètre de dispersion γ et de PDF $f_X(x) = \frac{\gamma}{\pi(x^2 + \gamma^2)}$, nous obtenons l'expression

$$LLR_c(R_k) = \ln \left[\frac{(R_k - 1)^2 + \gamma^2}{(R_k + 1)^2 + \gamma^2} \right] \quad (3.25)$$

3.3.2.1 Notations et règle de décision

Afin d'appliquer le principe MAP aux turbo codes CTC, considérons un RSC de rendement 1/2 (voir chapitre 1, figure 1.4.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 ;
- $\nu = L - 1$ est la mémoire du codeur et L est la longueur de contrainte ;
- $S_k = S_k^{(m)} = m$, $m = 0, 1, \dots, 2^M - 1$ 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 bruitée (sortie du canal).

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

$$LLR(d_k) = \ln \left[\frac{P(d_k = 1 | R_1^N)}{P(d_k = 0 | R_1^N)} \right] \quad (3.26)$$

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

$$\lambda_k^i(m) = P(d_k = i, S_k = m | R_1^N) \quad (3.27)$$

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

$$P(d_k = i | R_1^N) = \sum_{m=0}^{2^\nu - 1} \lambda_k^i(m) \quad (3.28)$$

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

En remplaçant l'expression de la probabilité conjointe dans l'équation (3.26), nous obtenons

$$LLR(d_k) = \ln \left[\frac{\sum_{m=0}^{2^\nu - 1} \lambda_k^1(m)}{\sum_{m=0}^{2^\nu - 1} \lambda_k^0(m)} \right] \quad (3.29)$$

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

$$\text{Si } LLR(d_k) > 0 \text{ le bit décodé est } 1 \quad (3.30)$$

$$\text{Si } LLR(d_k) < 0 \text{ le bit décodé est } 0 \quad (3.31)$$

Notons que l'expression de la JD diffère d'un auteur à un autre. Dans l'algorithme d'origine BJRC [88], JD a été définie sous forme de deux équations

$$\delta_k(m) = P(S_k = m | R_1^N) \quad (3.32)$$

$$\sigma_k(m', m) = P(S_{k-1} = m', S_k = m, R_1^N) \quad (3.33)$$

En analysant, la deuxième équation (3.33), 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

$$\begin{aligned} \sigma_k(m', m) &= P(S_{k-1} = m', S_k = m, R_1^N) \\ &= P(S_{k-1} = m', S_k = m | R_1^N) \cdot P(R_1^N) \\ &= P(d_{k-1} = i, S_k = m | R_1^N) \cdot P(R_1^N) \\ &= \lambda_{k-1}^i(m) \cdot P(R_1^N) \end{aligned} \quad (3.34)$$

La majorité des auteurs ont adopté la définition de la probabilité conjointe que nous avons présenté en premier (équation (3.27)). 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 [95][38]. Évidemment, la façon dont l'expression est développée affectera le nombre d'opérations ou plus c'est-à-dire la complexité de l'algorithme MAP. A l'origine du MAP est l'article phare paru en 1993 [96]. Les auteurs de ce dernier article ont adopté une version modifiée de l'algorithme original BCJR. Cette version a été appelée BCJR modifiée du fait qu'elle a été adoptée en particulier pour le codes RSC.

Dans ce paragraphe, nous allons décrire la dernière version de l'algorithme MAP utilisée dans les références [97][98][99], qui sera utilisé dans la suite de cette thèse.

Il est bon de noter qu'il existe plusieurs variantes de l'algorithme MAP visant à réduire sa complexité. Les plus connues sont les algorithmes Log-MAP et Max-Log-MAP [100], mais ils restent des algorithmes sous optimaux contrairement à l'algorithme primitif MAP.

3.3.2.2 L'Algorithme MAP utilisé

3.3.2.2.1 Expression de la probabilité conjointe

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

$$\begin{aligned}
 \lambda_k^i(m) &= P(d_k = i, S_k = m | R_1^N) \\
 &= \frac{P(d_k = i, S_k = m, R_1^{k-1}, R_k^N)}{P(R_1^N)} \\
 &= \frac{P(R_1^{k-1} | d_k = i, S_k = m, R_k^N) \cdot P(d_k = i, S_k = m, R_k^N)}{P(R_1^N)} \\
 &= \frac{P(R_1^{k-1} | d_k = i, S_k = m, R_k^N) \cdot P(d_k = i, S_k = m, R_k, R_{k+1}^N)}{P(R_1^N)} \\
 &= \frac{P(R_1^{k-1} | d_k = i, S_k = m, R_k^N) \cdot P(R_{k+1}^N | d_k = i, S_k = m, R_k) \cdot P(d_k = i, S_k = m, R_k)}{P(R_1^N)} \quad (3.35)
 \end{aligned}$$

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

$$\alpha_k(m) = P(R_1^{k-1} | d_k = i, S_k = m, R_k^N) \quad (3.36)$$

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

$$\alpha_k(m) = P(R_1^{k-1} | S_k = m) \quad (3.37)$$

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

$$\beta_k(m) = P(R_k^N | S_k = m) \quad (3.38)$$

Nous définissons la métrique de branche BM par

$$\delta_k^i(m) = P(d_k = i, S_k = m, R_k) \quad (3.39)$$

En remplaçant les expressions des FSM, BSM et BM dans l'équation (3.35)

$$\begin{aligned}
 \lambda_k^i(m) &= \frac{\alpha_k(m) \cdot P(R_{k+1}^N | d_k = i, S_k = m, R_k) \cdot \delta_k^i(m)}{P(R_1^N)} \\
 &= \frac{\alpha_k(m) \cdot P(R_{k+1}^N | S_{k+1} = f(i, m)) \cdot \delta_k^i(m)}{P(R_1^N)} \\
 &= \frac{\alpha_k(m) \cdot \beta_{k+1}(f(i, m)) \cdot \delta_k^i(m)}{P(R_1^N)} \quad (3.40)
 \end{aligned}$$

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 (3.29), nous obtenons

$$LLR(d_k) = \ln \left[\frac{\sum_{m=0}^{2^v-1} \alpha_k(m) \cdot \beta_{k+1}(f(1, m)) \cdot \delta_k^1(m)}{\sum_{m=0}^{2^v-1} \alpha_k(m) \cdot \beta_{k+1}(f(0, m)) \cdot \delta_k^0(m)} \right] \quad (3.41)$$

D'après l'équation (3.41), 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.

3.3.2.2 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

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

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 égal j .

$$\begin{aligned} \alpha_k(m) &= \sum_{m'=0}^{2^v-1} \sum_{j=0}^1 P(R_1^{k-2} | S_{k-1} = b(j, m)) \cdot P(d_{k-1} = j, S_{k-1} = b(j, m), R_{k-1}) \\ &= \sum_{j=0}^1 \alpha_{k-1}(b(j, m)) \cdot \delta_{k-1}^j(b(j, m)) \end{aligned} \quad (3.42)$$

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 3.4.

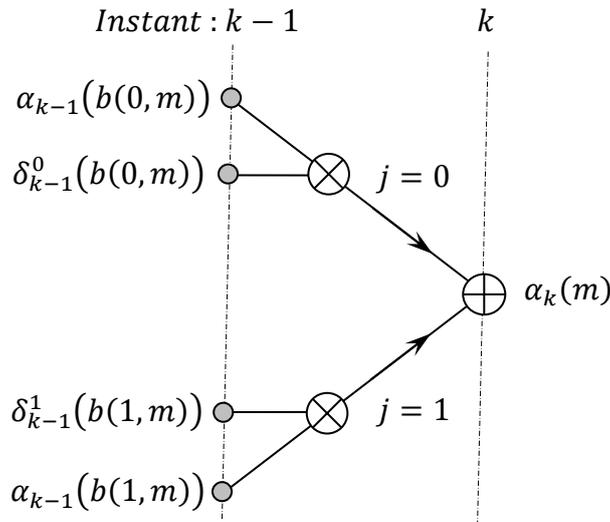


Fig. 3.4 Représentation graphique de la métrique d'état en avant.

3.3.2.2.3 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

$$\begin{aligned}
 \beta_k(m) &= P(R_k^N | S_k = m) \\
 &= \sum_{m'=0}^{2^v-1} \sum_{j=0}^1 P(d_k = j, S_{k+1} = m', R_k^N | S_k = m) \\
 &= \sum_{m'=0}^{2^v-1} \sum_{j=0}^1 P(R_{k+1}^N | S_k = m, d_k = j, S_{k+1} = m', R_k) \cdot P(d_k = j, S_{k+1} = m', R_k | S_k = m) \\
 \beta_k(m) &= \sum_{j=0}^1 P(R_{k+1}^N | S_{k+1} = f(j, m)) \cdot P(d_k = j, S_k = m, R_k) = \sum_{j=0}^1 \beta_{k+1}(f(j, m)) \cdot \delta_k^j(m) \quad (3.43)
 \end{aligned}$$

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 être aussi schématisée sous forme graphique à la figure 3.5.

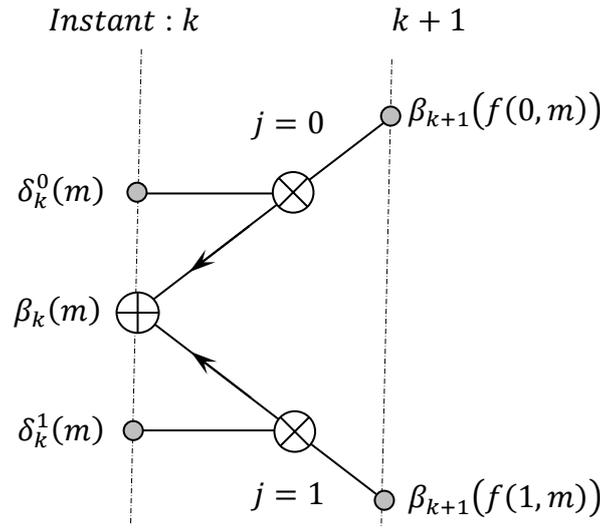


Fig.3.5 Représentation graphique de la métrique d'état en arrière.

3.3.2.2.4 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 (3.39)), 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.

$$\begin{aligned} \delta_k^i(m) &= P(d_k = i, S_k = m, R_k) = P(R_k | d_k = i, S_k = m) \cdot P(d_k = i, S_k = m) \\ &= P(R_k | d_k = i, S_k = m) \cdot P(S_k = m | d_k = i) \cdot P(d_k = i) \end{aligned} \quad (3.44)$$

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

$$\delta_k^i(m) = P(r_k^s | d_k = i, S_k = m) \cdot P(r_k^p | d_k = i, S_k = m) \cdot P(S_k = m | d_k = i) \cdot P(d_k = i) \quad (3.45)$$

Les deux premiers termes de la métrique de branche dépendent des symboles recueillis à la sortie du canal. Donc le type du canal utilisé affecte effectivement l'expression de la BM.

Nous pouvons exprimer la BM pour les trois canaux suivants (qui seront utilisés par la suite dans le chapitre 4)

a. Canal à bruit de type GG de moyenne nulle et de variance σ^2

Nous pouvons exprimer respectivement la densité de probabilité de l'information reçue et celle de l'information de parité reçue de l'équation (3.45) comme suit

$$P(r_k^s | d_k = i, S_k = m) = \frac{\alpha}{2 \cdot A \cdot \Gamma(1/\alpha)} \exp(-|r_k^s - (1 - 2i)|/A)^\alpha \quad (3.46)$$

$$P(r_k^p | d_k = i, S_k = m) = \frac{\alpha}{2 \cdot A \cdot \Gamma(1/\alpha)} \exp(-|(r_k^p - (1 - 2c_m^i))/A|^\alpha) \quad (3.47)$$

où $A = \left(\sigma^2 \cdot \frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)}\right)^{1/2}$, $\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt$, $z > 0$ et c_m^i est le bit codé sachant $d_k = i$ et $S_k = m$.

En remplaçant les deux précédentes expressions dans l'équation (3.45), la métrique de branche devient

$$\begin{aligned} \delta_k^i(m) &= \frac{\alpha}{2 \cdot A \cdot \Gamma(1/\alpha)} \exp(-|(r_k^s - (1 - 2i))/A|^\alpha) \cdot \frac{\alpha}{2 \cdot A \cdot \Gamma(1/\alpha)} \exp(-|(r_k^p - (1 - 2c_m^i))/A|^\alpha) \\ &\quad \cdot \frac{1}{2^\nu} \cdot P(d_k = i) \\ &= \frac{1}{2^\nu} \cdot P(d_k = i) \cdot \left(\frac{\alpha}{2 \cdot A \cdot \Gamma(1/\alpha)}\right)^2 \\ &\quad \cdot \exp\left(-A^{-\alpha}(|(r_k^s - (1 - 2i))|^\alpha + |(r_k^p - (1 - 2c_m^i))|^\alpha)\right) \\ &= C(\nu, \alpha, \sigma) \cdot P(d_k = i) \cdot \exp\left(-A^{-\alpha}(|(r_k^s - (1 - 2i))|^\alpha + |(r_k^p - (1 - 2c_m^i))|^\alpha)\right) \end{aligned} \quad (3.48)$$

où la grandeur $C(\nu, \alpha, \sigma)$ ne dépend pas de la variable i et de l'état m , et va disparaître dans l'expression du $LLR(d_k)$.

b. Canal à bruit de type AWGN de moyenne nulle et de variance σ^2

Nous posons $\alpha = 2$ dans l'équation (3.48), la métrique de branche $\delta_k^i(m)$ pour un canal AWGN s'écrit

$$\begin{aligned} \delta_k^i(m) &= C(\nu, \sigma) \cdot P(d_k = i) \cdot \exp\left(-\frac{1}{2\sigma^2} \left((r_k^s - (1 - 2i))^2 + (r_k^p - (1 - 2c_m^i))^2 \right)\right) \\ &= C(\nu, \sigma) \cdot P(d_k = i) \cdot \exp\left(-\frac{1}{2\sigma^2} \left((r_k^s)^2 - 2r_k^s(1 - 2i) + \underbrace{(1 - 2i)^2}_{=1} + (r_k^p)^2 - \right. \right. \\ &\quad \left. \left. 2r_k^p(1 - 2c_m^i) + \underbrace{(1 - 2c_m^i)^2}_{=1} \right)\right) \\ &= C(\nu, \sigma) \cdot \underbrace{\exp\left(-\frac{1}{2\sigma^2} \left((r_k^s)^2 - 2r_k^s + (r_k^p)^2 - 2r_k^p + 2 \right)\right)}_{=C(\nu, \sigma, r_k^s, r_k^p)} \\ &\quad \cdot P(d_k = i) \cdot \exp\left(-\frac{2}{\sigma^2} (r_k^s \cdot i + r_k^p \cdot c_m^i)\right) \end{aligned} \quad (3.49)$$

où $C(\nu, \sigma, r_k^s, r_k^p)$ va disparaître lorsque nous faisons le LLR.

c. Canal à bruit de type *SaS* de Cauchy de paramètre de dispersion γ

Pour ce canal, nous avons

$$P(r_k^s | d_k = i, S_k = m) = \frac{\gamma}{\pi((r_k^s - (1 - 2i))^2 + \gamma^2)} \quad (3.50)$$

$$P(r_k^p | d_k = i, S_k = m) = \frac{\gamma}{\pi(r_k^p - (1 - 2c_m^i))^2 + \gamma^2} \quad (3.51)$$

et la métrique de branche devient

$$\begin{aligned} \delta_k^i(m) &= \frac{1}{2^v} \cdot P(d_k = i) \cdot \frac{\gamma}{\pi((r_k^s - (1 - 2i))^2 + \gamma^2)} \cdot \frac{\gamma}{\pi(r_k^p - (1 - 2c_m^i))^2 + \gamma^2} \\ &= \frac{1}{2^v} \cdot P(d_k = i) \cdot \frac{\gamma^2}{\pi^2((r_k^s - (1 - 2i))^2 + \gamma^2) \cdot (r_k^p - (1 - 2c_m^i))^2 + \gamma^2)} \\ &= C(v, \gamma) \cdot P(d_k = i) \cdot \frac{1}{((2i - 1)r_k^s + h(r_k^s)) \cdot ((2c_m^i - 1)r_k^p + h(r_k^p))} \end{aligned} \quad (3.52)$$

où $h(x) = (x^2 + \gamma^2 + 1)/2$, $(1 - 2i)^2 = (1 - 2c_m^i)^2 = 1$ et le coefficient $C(v, \gamma)$ se simplifie en faisant le LLR.

Exemple 3.3

Nous considérons les données de l'exemple 3.1. La séquence à envoyer est $d = (11011)$. La sortie du codeur RSC (5,7) est $X = (1110001011)$. La sortie du modulateur est $Y = (-1 - 1 - 1 + 1 + 1 + 1 - 1 + 1 - 1 - 1)$. Nous supposons que la séquence Y est envoyée dans un canal AWGN. Nous supposons que nous avons trouvé à la sortie du canal les observations $R = (r^s, r^p) = (-4 - 3 - 3 - 2 + 2 + 4 - 3 + 3 - 3 + 1)$. Nous pouvons alors calculer la variance du bruit W à partir de la relation (3.1).

Nous considérons qu'à la sortie du codeur RSC la probabilité d'avoir un 0 est égale à celle d'avoir un 1, c'est-à-dire $P(X_k = 0) = P(X_k = 1) = 1/2$. Dans ce cas nous avons $E(Y) = 1/2 \cdot (-1) + 1/2 \cdot (+1) = 0$ et $var(Y) = 1/2 \cdot (-1)^2 + 1/2 \cdot (+1)^2 = 1$. Ipso facto, $var(R) = var(Y) + var(W) = 1 + \sigma^2$, c'est-à-dire

$$\sigma^2 = var(R) - 1 \quad (3.53)$$

Une application numérique donne $\sigma^2 \cong 7.84$.

Pour un bruit AWGN toujours, la BM est donnée par l'équation (3.49). Si nous supposons que les bits d'information d_k sont équiprobables, c'est-à-dire $P(d_k = 0) = P(d_k = 1) = 1/2$, cette BM s'écrit sous la forme $\delta_k^i(m) = C e^{-\frac{2}{\sigma^2}(r_k^s \cdot i + r_k^p \cdot c_m^i)}$. Puisque la constante C va disparaître en faisant le rapport LLR, nous pouvons mettre $C = 1$, et la BM devient tout simplement comme suit

$$\delta_k^i(m) = e^{-\frac{2}{\sigma^2}(r_k^s \cdot i + r_k^p \cdot c_m^i)} \quad (3.54)$$

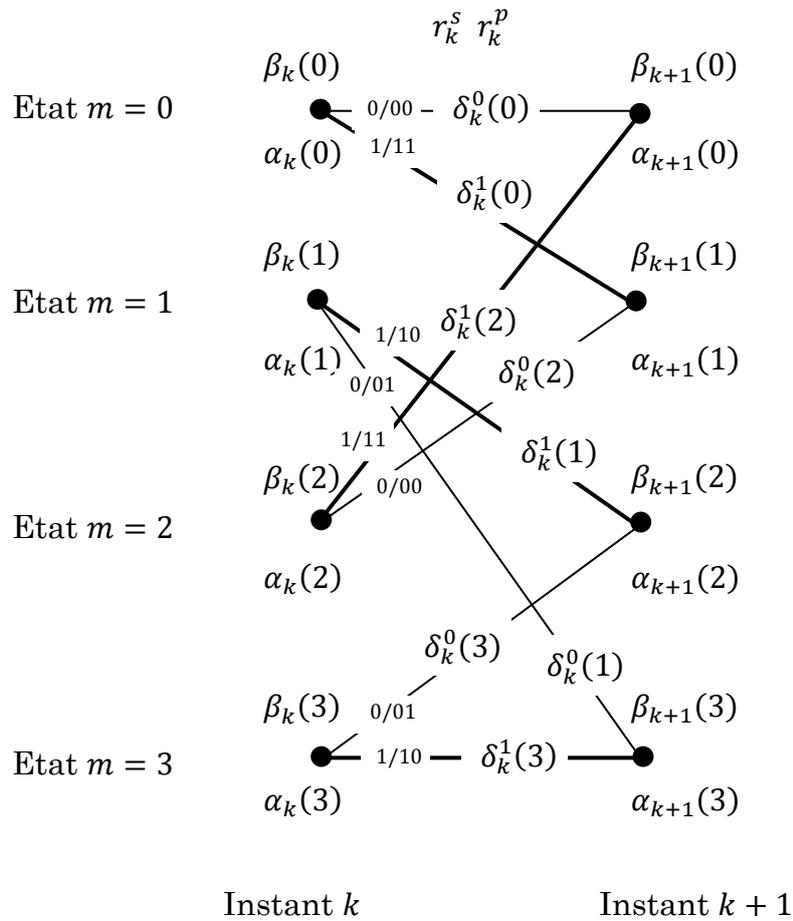


Fig. 3.6 Les FSM, BSM et BM sur le treillis simplifié du RSC (5,7) pour calculer le LLR(d_k).

A un instant k , dans chaque nœud (état) m du treillis de la figure 3.6, il existe deux métriques d'état, une en avant notée $\alpha_k(m)$ et une autre en arrière notée $\beta_k(m)$, et à partir du même nœud m elles partent deux branches, une correspond à un bit d'entrée $d_k = 0$ (ou $i = 0$) et qui possède une métrique de branche notée $\delta_k^0(m)$ et une autre correspond à un bit $d_k = 1$ (ou $i = 1$) et qui possède une métrique de branche notée $\delta_k^1(m)$. Ces deux métriques de branche dépendent des deux symboles r_k^s et r_k^p reçus à l'instant k . L'écriture binaire qui contient un slash sur les branches du treillis est formée de trois bits. Le premier bit représente le bit d'entrée du codeur noté d_k ou i et les deux autres bits représentent respectivement la sortie systématique $X_k^1 = i$ et la sortie codée $X_k^2 = c_m^i$ du codeur (voir figure 1.4.a). Par exemple, la forme « 0/01 » implique que l'entrée du codeur est $i = 0$, la sortie systématique est $i = 0$ et la sortie codée est $c_m^i = 1$. En méditant sur le treillis de la figure 3.6, nous trouvons facilement que

$$c_m^i = \begin{cases} i & \text{si } m \text{ est pair} \\ 1 - i & \text{si } m \text{ est impair} \end{cases} \quad (3.55)$$

Le tableau 3.2 résume toutes les valeurs de la métrique de branche $\delta_k^i(m)$ pour le message reçu $R = (-4 - 3 - 3 - 2 + 2 + 4 - 3 + 3 - 3 + 1)$.

		k				
		$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$\delta_k^i(m)$						
Etat $m = 0$	$\delta_k^0(0)$	1.0000	1.0000	1.0000	1.0000	1.0000
	$\delta_k^1(0)$	5.9578	3.5779	0.2166	1.0000	1.6651
Etat $m = 1$	$\delta_k^0(1)$	2.1487	1.6651	0.3607	0.4654	0.7749
	$\delta_k^1(1)$	2.7727	2.1487	0.6005	2.1487	2.1487
Etat $m = 2$	$\delta_k^0(2)$	1.0000	1.0000	1.0000	1.0000	1.0000
	$\delta_k^1(2)$	5.9578	3.5779	0.2166	1.0000	1.6651
Etat $m = 3$	$\delta_k^0(3)$	2.1487	1.6651	0.3607	0.4654	0.7749
	$\delta_k^1(3)$	2.7727	2.1487	0.6005	2.1487	2.1487

Tab. 3.2 Les valeurs des BM du treillis du RSC (5,7) pour le message reçu $R = (-4 - 3 - 3 - 2 + 2 + 4 - 3 + 3 - 3 + 1)$ dans le but de calculer le $LLR(d_k)$.

En basant sur les figures 3.4 et 3.5, nous pouvons calculer respectivement les FSM et BSM à partir du treillis de la figure 3.6 des tous les nœuds pour $k = 1, 2, \dots, 6$.

Sans perte de généralité, nous pouvons prendre pour les FSM les conditions initiales $\alpha_1(0) = 1, \alpha_1(m) = 0, \forall m \neq 0$. Nous avons à partir du treillis

- Pour $k = 2$

$$\alpha_2(0) = \alpha_1(0) \cdot \delta_1^0(0) + \alpha_1(2) \cdot \delta_1^1(2) = 1.0000$$

$$\alpha_2(1) = \alpha_1(2) \cdot \delta_1^0(2) + \alpha_1(0) \cdot \delta_1^1(0) = 5.9578$$

$$\alpha_2(2) = \alpha_1(3) \cdot \delta_1^0(3) + \alpha_1(1) \cdot \delta_1^1(1) = 0.0000$$

$$\alpha_2(3) = \alpha_1(1) \cdot \delta_1^0(1) + \alpha_1(3) \cdot \delta_1^1(3) = 0.0000$$

- Pour $k = 3$

$$\alpha_3(0) = \alpha_2(0) \cdot \delta_2^0(0) + \alpha_2(2) \cdot \delta_2^1(2) = 1.0000$$

$$\alpha_3(1) = \alpha_2(2) \cdot \delta_2^0(2) + \alpha_2(0) \cdot \delta_2^1(0) = 3.5779$$

$$\alpha_3(2) = \alpha_2(3) \cdot \delta_2^0(3) + \alpha_2(1) \cdot \delta_2^1(1) = 12.8017$$

$$\alpha_3(3) = \alpha_2(1) \cdot \delta_2^0(1) + \alpha_2(3) \cdot \delta_2^1(3) = 9.9206$$

- Pour $k = 4$

$$\alpha_4(0) = \alpha_3(0) \cdot \delta_3^0(0) + \alpha_3(2) \cdot \delta_3^1(2) = 3.7727$$

$$\alpha_4(1) = \alpha_3(2) \cdot \delta_3^0(2) + \alpha_3(0) \cdot \delta_3^1(0) = 13.0183$$

$$\alpha_4(2) = \alpha_3(3) \cdot \delta_3^0(3) + \alpha_3(1) \cdot \delta_3^1(1) = 5.7267$$

$$\alpha_4(3) = \alpha_3(1) \cdot \delta_3^0(1) + \alpha_3(3) \cdot \delta_3^1(3) = 7.2482$$

- Pour $k = 5$

$$\alpha_5(0) = \alpha_4(0) \cdot \delta_4^0(0) + \alpha_4(2) \cdot \delta_4^1(2) = 9.4994$$

$$\alpha_5(1) = \alpha_4(2) \cdot \delta_4^0(2) + \alpha_4(0) \cdot \delta_4^1(0) = 9.4994$$

$$\alpha_5(2) = \alpha_4(3) \cdot \delta_4^0(3) + \alpha_4(1) \cdot \delta_4^1(1) = 31.3459$$

$$\alpha_5(3) = \alpha_4(1) \cdot \delta_4^0(1) + \alpha_4(3) \cdot \delta_4^1(3) = 21.6330$$

- Pour $k = 6$

$$\alpha_6(0) = \alpha_5(0) \cdot \delta_5^0(0) + \alpha_5(2) \cdot \delta_5^1(2) = 61.6949$$

$$\alpha_6(1) = \alpha_5(2) \cdot \delta_5^0(2) + \alpha_5(0) \cdot \delta_5^1(0) = 47.1638$$

$$\alpha_6(2) = \alpha_5(3) \cdot \delta_5^0(3) + \alpha_5(1) \cdot \delta_5^1(1) = 37.1760$$

$$\alpha_6(3) = \alpha_5(1) \cdot \delta_5^0(1) + \alpha_5(3) \cdot \delta_5^1(3) = 53.8448$$

Pour les BSM, nous pouvons prendre les conditions finales $\beta_6(0) = 1, \beta_6(m) = 0, \forall m \neq 0$. Nous avons à partir du treillis

- Pour $k = 5$

$$\beta_5(0) = \beta_6(0) \cdot \delta_5^0(0) + \beta_6(1) \cdot \delta_5^1(0) = 1.0000$$

$$\beta_5(1) = \beta_6(3) \cdot \delta_5^0(1) + \beta_6(2) \cdot \delta_5^1(1) = 0.0000$$

$$\beta_5(2) = \beta_6(1) \cdot \delta_5^0(2) + \beta_6(0) \cdot \delta_5^1(2) = 1.6651$$

$$\beta_5(3) = \beta_6(2) \cdot \delta_5^0(3) + \beta_6(3) \cdot \delta_5^1(3) = 0.0000$$

- Pour $k = 4$

$$\beta_4(0) = \beta_5(0) \cdot \delta_4^0(0) + \beta_5(1) \cdot \delta_4^1(0) = 1.0000$$

$$\beta_4(1) = \beta_5(3) \cdot \delta_4^0(1) + \beta_5(2) \cdot \delta_4^1(1) = 3.5779$$

$$\beta_4(2) = \beta_5(1) \cdot \delta_4^0(2) + \beta_5(0) \cdot \delta_4^1(2) = 1.0000$$

$$\beta_4(3) = \beta_5(2) \cdot \delta_4^0(3) + \beta_5(3) \cdot \delta_4^1(3) = 0.7749$$

- Pour $k = 3$

$$\beta_3(0) = \beta_4(0) \cdot \delta_3^0(0) + \beta_4(1) \cdot \delta_3^1(0) = 1.7749$$

$$\beta_3(1) = \beta_4(3) \cdot \delta_3^0(1) + \beta_4(2) \cdot \delta_3^1(1) = 0.8800$$

$$\beta_3(2) = \beta_4(1) \cdot \delta_3^0(2) + \beta_4(0) \cdot \delta_3^1(2) = 3.7945$$

$$\beta_3(3) = \beta_4(2) \cdot \delta_3^0(3) + \beta_4(3) \cdot \delta_3^1(3) = 0.8260$$

- Pour $k = 2$

$$\beta_2(0) = \beta_3(0) \cdot \delta_2^0(0) + \beta_3(1) \cdot \delta_2^1(0) = 4.9237$$

$$\beta_2(1) = \beta_3(3) \cdot \delta_2^0(1) + \beta_3(2) \cdot \delta_2^1(1) = 9.5289$$

$$\beta_2(2) = \beta_3(1) \cdot \delta_2^0(2) + \beta_3(0) \cdot \delta_2^1(2) = 7.2307$$

$$\beta_2(3) = \beta_3(2) \cdot \delta_2^0(3) + \beta_3(3) \cdot \delta_2^1(3) = 8.0934$$

- Pour $k = 1$

$$\beta_1(0) = \beta_2(0) \cdot \delta_1^0(0) + \beta_2(1) \cdot \delta_1^1(0) = 61.6949$$

$$\beta_1(1) = \beta_2(3) \cdot \delta_1^0(1) + \beta_2(2) \cdot \delta_1^1(1) = 37.4392$$

$$\beta_1(2) = \beta_2(1) \cdot \delta_1^0(2) + \beta_2(0) \cdot \delta_1^1(2) = 38.8632$$

$$\beta_1(3) = \beta_2(2) \cdot \delta_1^0(3) + \beta_2(3) \cdot \delta_1^1(3) = 37.9776$$

Finalement, il nous reste que de remplacer les valeurs trouvées des FSM, BSM et BM dans l'expression suivante du $LLR(d_k)$ (voir la relation (3.41))

$$LLR(d_k) = \ln \left[\frac{\alpha_k(0) \cdot \beta_{k+1}(1) \cdot \delta_k^1(0) + \alpha_k(1) \cdot \beta_{k+1}(2) \cdot \delta_k^1(1) + \alpha_k(2) \cdot \beta_{k+1}(0) \cdot \delta_k^1(2) + \alpha_k(3) \cdot \beta_{k+1}(3) \cdot \delta_k^1(3)}{\alpha_k(0) \cdot \beta_{k+1}(0) \cdot \delta_k^0(0) + \alpha_k(1) \cdot \beta_{k+1}(3) \cdot \delta_k^0(1) + \alpha_k(2) \cdot \beta_{k+1}(1) \cdot \delta_k^0(2) + \alpha_k(3) \cdot \beta_{k+1}(2) \cdot \delta_k^0(3)} \right] \quad (3.56)$$

Après tout calcul fait, nous obtenons pour $k = 1, \dots, 5$

$$LLR(d) = (+2.4450 \quad + 1.6464 \quad - 1.6058 \quad + 1.7175 \quad + 1.7038)$$

et facilement nous déduisons la séquence estimée

$$\hat{d} = \frac{1}{2} \cdot \left(1 + \text{sign}(LLR(d)) \right) = (11011)$$

qui est exactement la séquence envoyée.

3.4 Schéma de principe d'un décodeur itératif

Nous considérons les turbo codes CTC formés de deux codes convolutifs RSC identiques concaténés en parallèle comme les montre la figure 1.3.b. Au niveau du récepteur le signal reçu est $R_k = Y_k + W_k$, où Y_k sont les symboles émis et W_k sont des variables aléatoires i.i.d qui représentent les observations du bruit à moyenne nulle. A l'aide d'un démultiplexeur 1 vers 3, la séquence d'observations R_k se dégroupé en trois sous séquences R_k^1 , R_k^2 et R_k^3 .

Le décodage itératif de ces turbo codes peut être réalisé à partir de deux décodeurs à entrées-sorties douces DEC1 et DEC2 associés selon le principe représenté sur la figure 3.7. Si l'algorithme SISO utilisé est optimal au sens du ML, ces deux décodeurs élémentaires sont ainsi, mais malgré ceci le décodage itératif ne l'est pas.

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 , se 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 douces, le décodeur DEC1 associe à chaque symbole décodé d_k une mesure de fiabilité sous forme du logarithme de son rapport de vraisemblance $LLR_1(d_k)$, et DEC2 associe au même symbole d_k une autre mesure de fiabilité $LLR_2(d_k)$. 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), et 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 valeur estimée \hat{d}_k du bit originale d_k en basant sur le signe du rapport $LLR_2(d_k)$ (ou $LLR_1(d_k)$) : $\hat{d}_k = \frac{1}{2} \cdot (1 + \text{sign}(LLR_2(d_k)))$.

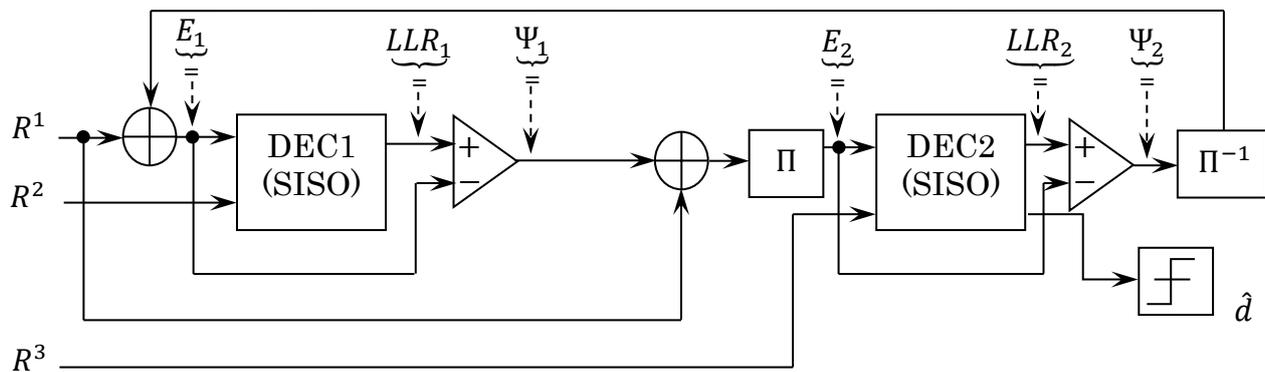


Fig. 3.7 Schéma bloc d'un décodeur itératif associé à un CTC.

3.5 Conclusion

Ce chapitre est consacré au turbo décodage, proprement dit décodage itératif avec les algorithmes à entrées-sorties douces SISO, à savoir l'algorithme de **Viterbi** à sortie douce SOVA et l'algorithme de maximum a posteriori MAP ou BCJR.

Nous avons défini une métrique euclidienne dite métrique de corrélation en basant

sur la maximisation de la probabilité de transition du canal au sens du ML. L'algorithme de **Viterbi** classique qui utilise cette métrique euclidienne a été exposé avec subtilité. Le survivant qui donne la séquence estimée correspond ici à la métrique maximale, contrairement à ce qui existe dans la grande majorité des littératures.

Nous avons vu que la technique d'appliquer l'algorithme de **Viterbi** classique avec la métrique de corrélation dans les deux sens du treillis, permet de prendre une décision ferme sur le message émis en utilisant le survivant, et de calculer en plus le LLR qui représente une mesure très opportune de la fiabilité de cette décision. La technique en question est appelée l'algorithme SOVA bidirectionnel ou SOVA tout court.

Pour n'est pas être exhaustif, nous avons essayé de donner l'essentiel dans le développement des formules qui apportent le LLR. A propos de ceci, nous avons exposé avec un grand développement un exemple qui montre les calculs numériques des fiabilités de décision LLR.

Nous avons vu que le SOVA est un algorithme de décodage à ML par séquence ou par chemin sous optimal, contrairement au MAP qui est un algorithme à ML symbole par symbole optimal.

L'algorithme MAP a été exposé avec subtilité, où nous avons défini la métrique d'état en avant FSM, la métrique d'état en arrière BSM et la métrique de branche BM pour calculer le LLR qui représente une mesure de la fiabilité de décision. L'expression de la BM a été donnée pour les trois types du bruit à savoir : AWGN, GG et $S\alpha S$ de **Cauchy**. Pour tirer au clair le calcul des BM, FSM, BSM et LLR, nous avons donné un exemple numérique bien détaillé.

A la fin, nous avons présenté le principe du décodage itératif d'un CTC qui permet aux deux décodeurs constitutifs SISO de travailler conjointement, c'est à dire que le premier décodeur SISO profite de l'entrée non systématique R_k^3 du deuxième décodeur SISO, et que ce dernier avec son rôle profite de l'entrée non systématique R_k^2 du premier décodeur SISO en utilisant ce que nous appelons l'information extrinsèque.

Nous avons vu comme montre l'équation (3.22) que le calcul de l'information extrinsèque d'un bit donné doit passer par le calcul des LLR et LLR du canal. Ces deux LLR dépendent étroitement des paramètres du bruit qui contamine le canal. Par conséquent, au niveau du récepteur, il faut trouver un moyen pour estimer ces paramètres à partir du signal reçu. Pour cette raison, nous allons présenter dans le chapitre suivant deux méthodes originales d'estimation, la première est pour un bruit

GG et la deuxième est pour un bruit $S\alpha S$ de **Cauchy**. Ces deux nouvelles méthodes permettent alors d'avoir une information extrinsèque très fiable, ce qui améliore amplement les performances du décodeur itératif. Dans ce même contexte, et pour un bruit AWGN, nous allons présenter aussi dans ce même chapitre une nouvelle technique pour optimiser un décodeur itératif à algorithme SISO.

Ces trois techniques originales et leurs simulations numériques font l'objet du chapitre suivant.

Chapitre 4

Nouvelles contributions et simulation numérique

4.1 Introduction

Dans le cas d'un bruit de type AWGN, un turbo code possède des performances proches des limites de Shannon. Pour un rapport signal sur bruit $SNR \geq 0.65 \text{ dB}$ le turbocode permet d'atteindre des meilleures performances si nous augmentons le nombre d'itérations, et ceci grâce à son décodeur itératif à maximum de vraisemblance et à sortie souple. Cependant, pour un faible SNR ($SNR < 0.65 \text{ dB}$), le turbocode ne donne pas de performances acceptables même si nous augmentons le nombre d'itérations. Nous allons proposer une nouvelle approche qui consiste à ajouter une fonction porte ou binaire dans l'entrée systématique du décodeur, afin de rendre les informations extrinsèques délivrées par les décodeurs constitutifs plus fiables, ce qui permet ainsi d'améliorer les performances en termes de BER et de convergence. Nous allons montrer à l'aide de différents exemples que les performances du décodeur itératif proposé sont meilleures que celles du décodeur itératif classique.

Dans cette nouvelle approche nous avons supposé que le bruit est modélisé sous la forme simple d'un bruit additif qui suit la loi gaussienne, or les bruits réels ne suivent que rarement cette loi, de bonnes performances ne peuvent être obtenues qu'en s'affranchissant de l'hypothèse gaussienne. C'est pour cette raison que nous nous sommes intéressés à l'étude d'une classe importante des bruits à savoir les bruits impulsifs.

Nous allons étudier les performances du décodage itératif en présence d'un bruit impulsif personnifié par le modèle gaussien généralisé défini par sa PDF. Dans cette étude, nous allons proposer une technique semi-aveugle d'estimation du paramètre de forme du bruit GG. Cette technique originale s'appuie sur deux points à savoir la probabilité que le signal reçu soit de même signe que celui transmis, et sur des

statistiques d'ordre inférieur fractionnel FLOS. Nous allons voir que les résultats obtenus montrent que l'approche proposée offre une amélioration considérable des performances du décodeur itératif.

Avec le même principe de l'approche semi-aveugle précédente, nous allons présenter une autre nouvelle approche d'estimation dans le turbo décodage mais pour un bruit impulsif $S\alpha S$ de **Cauchy** où $\alpha = 1$. Le paramètre à estimer par cette méthode est le paramètre de dispersion du bruit γ . Nous allons voir que la précision d'estimation est très bonne et que les performances du décodeur itératif pour les deux valeurs exacte et estimée de γ sont très proches.

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 code convolutif CTC parallèle ;
- Un modulateur BPSK ;
- Un canal supposé idéal (sans distorsions) à bruit additif ;
- Une source de bruit additif (les bruits envisagés ici sont : le bruit gaussien, le bruit impulsif gaussien généralisé et le bruit impulsif α –stable de **Cauchy**) ;
- Un décodeur itératif à ML et à sortie souple (se servir de l'algorithme SOVA ou MAP) ;
- Un comparateur pour calculer le nombre de bits erronés après le décodage afin d'estimer la probabilité d'erreurs binaire *BER*.

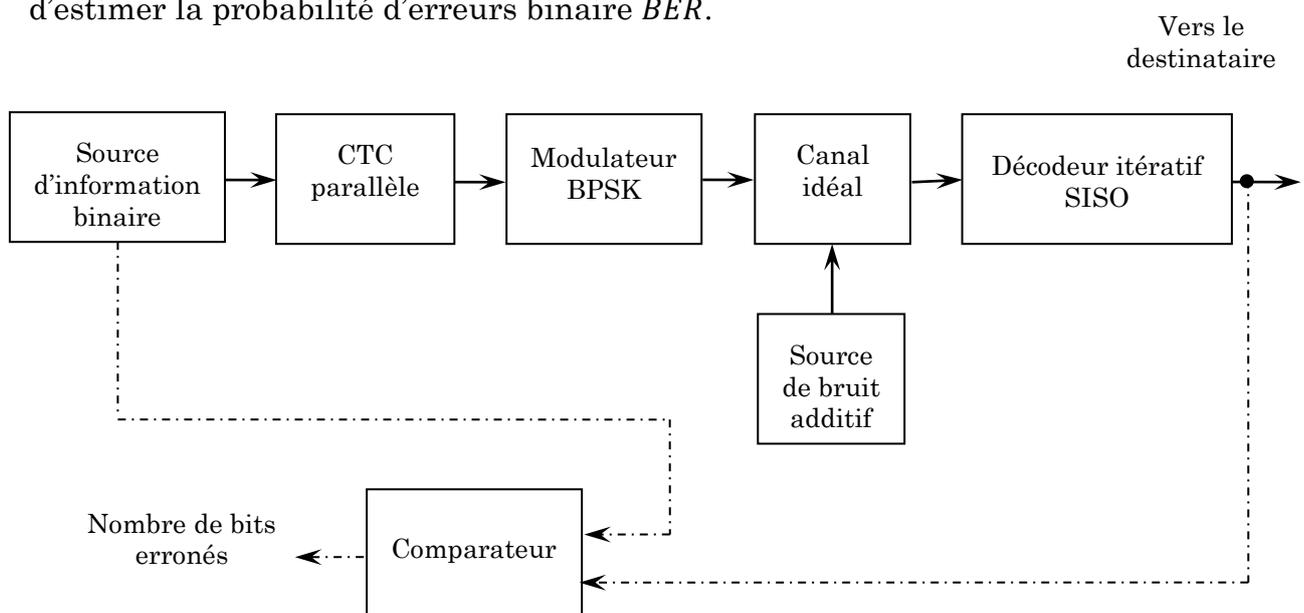


Fig. 4.1 Modèle de simulation.

4.2.1 Source d'information

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

La méthode universellement adoptée par la CCITT (Comité Consultatif International Téléphonique et Télégraphique), consiste à utiliser une séquence dite pseudo-aléatoire, séquence périodique dont les propriétés statistiques sont « voisines » de celle 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 (voir la figure 4.2). 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

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

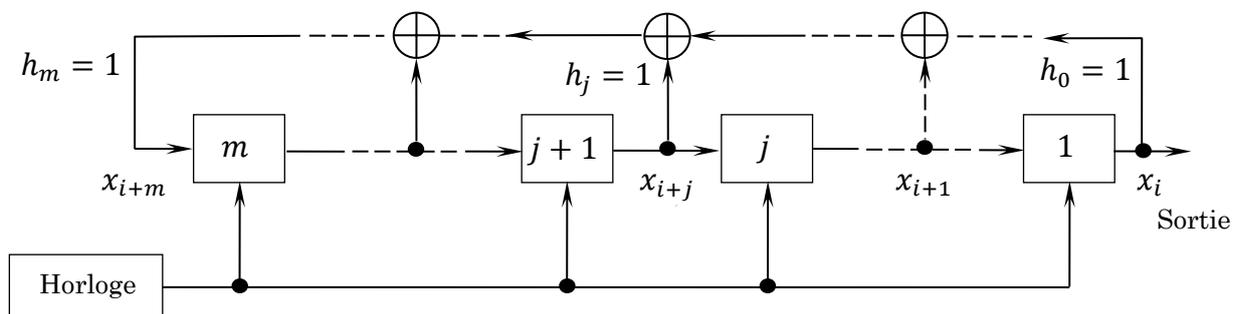


Fig. 4.2 Générateur pseudo-aléatoire.

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 rappelées à l'annexe B [22][101].

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) nous risquons 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 [102]. A cause du facteur de temps et de la mémoire limitée de l'ordinateur utilisé, nous avons choisi une séquence pseudo-aléatoire de longueur égale à $2^{20} - 1 = 1\,048\,575$ bits engendrée par le polynôme primitif $h(x) = x^{20} + x^3 + 1$. Alors, la séquence binaire qui représente l'information à envoyer d est de longueur égale à $2^{20} = 1\,048\,576$ bits.

4.2.2 Les CTC parallèles utilisés

Les codeurs de canal utilisés lors de cette simulation sont trois turbo codes parallèles de rendement 1/3 illustrés dans les figures 4.5.a, 4.5.b et 4.5.c. Ils sont obtenus par concaténation parallèle de deux codes RSC de rendement 1/2, très employés par **C. Berrou** [2], de polynômes générateurs : (5, 7) à 4 états (voir la figure 4.4.a), (13, 15) à 8 états (voir la figure 4.4.b) et (35, 23) à 16 états (voir la figure 4.4.b).

Concernant le choix de l'entrelaceur utilisé dans les turbo codes sus-cités il y a plusieurs types d'entrelaceurs comme les entrelaceurs uniformes, pseudo-aléatoire, aléatoire, S -aléatoire. Dans [36] ils ont noté que les entrelaceurs S -aléatoires, ayant des bons paramètres d'irrégularité et de capacité de dispersion [103], sont très performants pour les turbo codes dans toutes les tailles. Leurs inconvénients par rapport aux autres entrelaceurs (blocs classiques, pseudo-aléatoire) résident dans leur complexité. Ces derniers sont des entrelaceurs aléatoires avec une restriction de ses capacités de dispersion [40][104]. Ces entrelaceurs assurent que tous les bits, qui sont séparés dans le temps par une longueur inférieure à S dans la séquence originale, seront séparés par S bits au moins après l'entrelacement.

Dans notre travail nous avons utilisé l'entrelaceur S -aléatoire pour deux tailles $K = 64 \times 64$ et $K = 128 \times 128$. Dans [36] ils ont proposé une méthode pour générer des nombres i compris entre 1 et K . Chaque nombre généré est comparé aux S nombres précédents. Le nombre courant n'est pris que s'il est distant de $\pm S$ de n'importe lequel des S précédents nombres. Le temps de calcul de cet algorithme augmente en fonction de S et la convergence vers une solution n'est pas garantie. Cependant il a été observé dans [36] qu'en choisissant le paramètre S inférieur à $\sqrt{K/2}$, l'algorithme produit une solution en un temps raisonnable. Dans notre simulation nous avons délibérément choisi $S = 15$ pour les deux entrelaceurs afin de montrer l'effet de la taille de l'entrelaceur sur notre proposition.

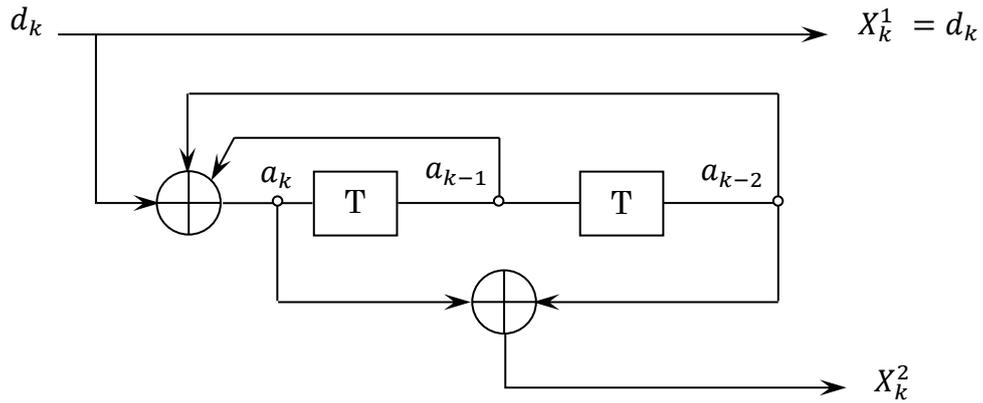


Fig. 4.3.a Code convolutif récursif systématique (5,7).

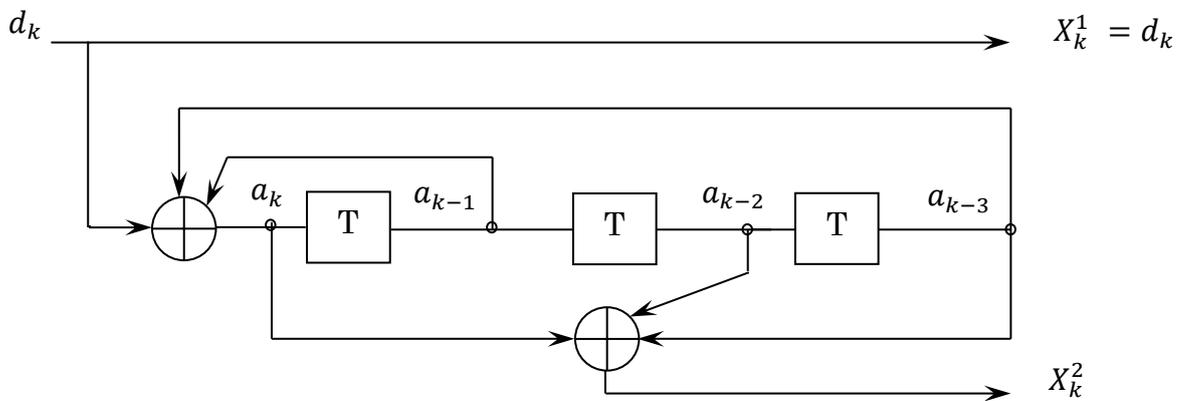


Fig. 4.3.b Code convolutif récursif systématique (13,15).

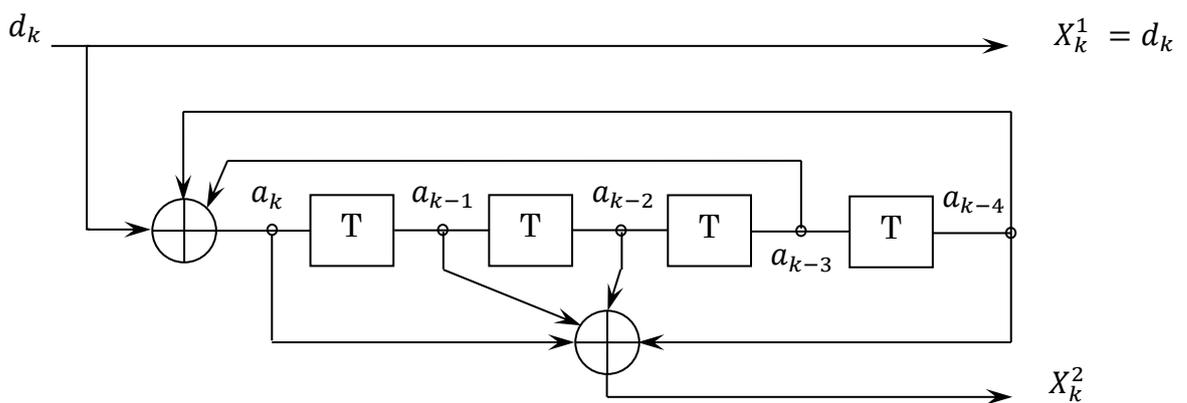


Fig. 4.3.c Code convolutif récursif systématique (35,23).

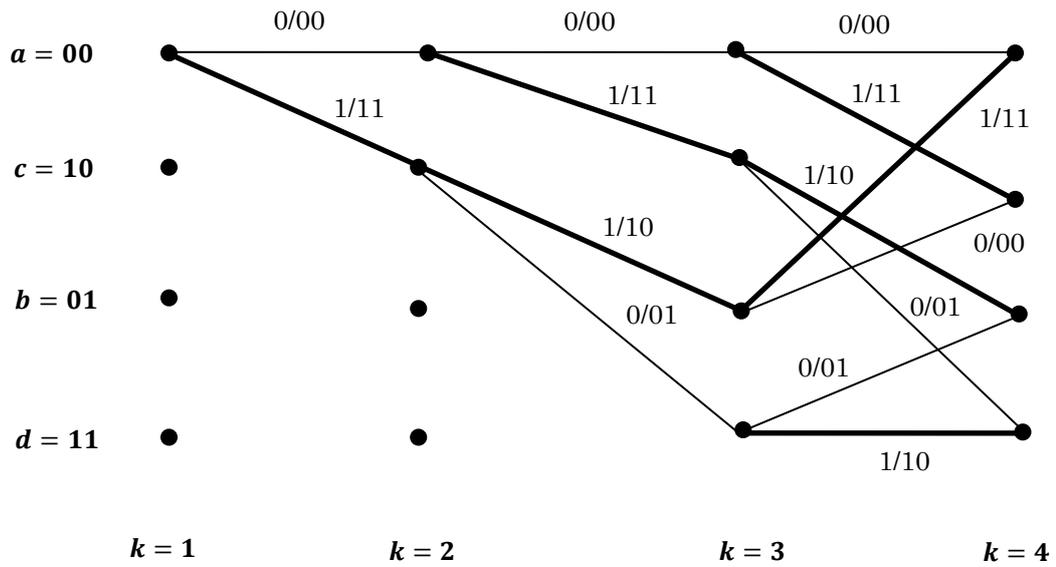


Fig. 4.4.a Représentation en treillis du codeur de la figure 4.3.a.

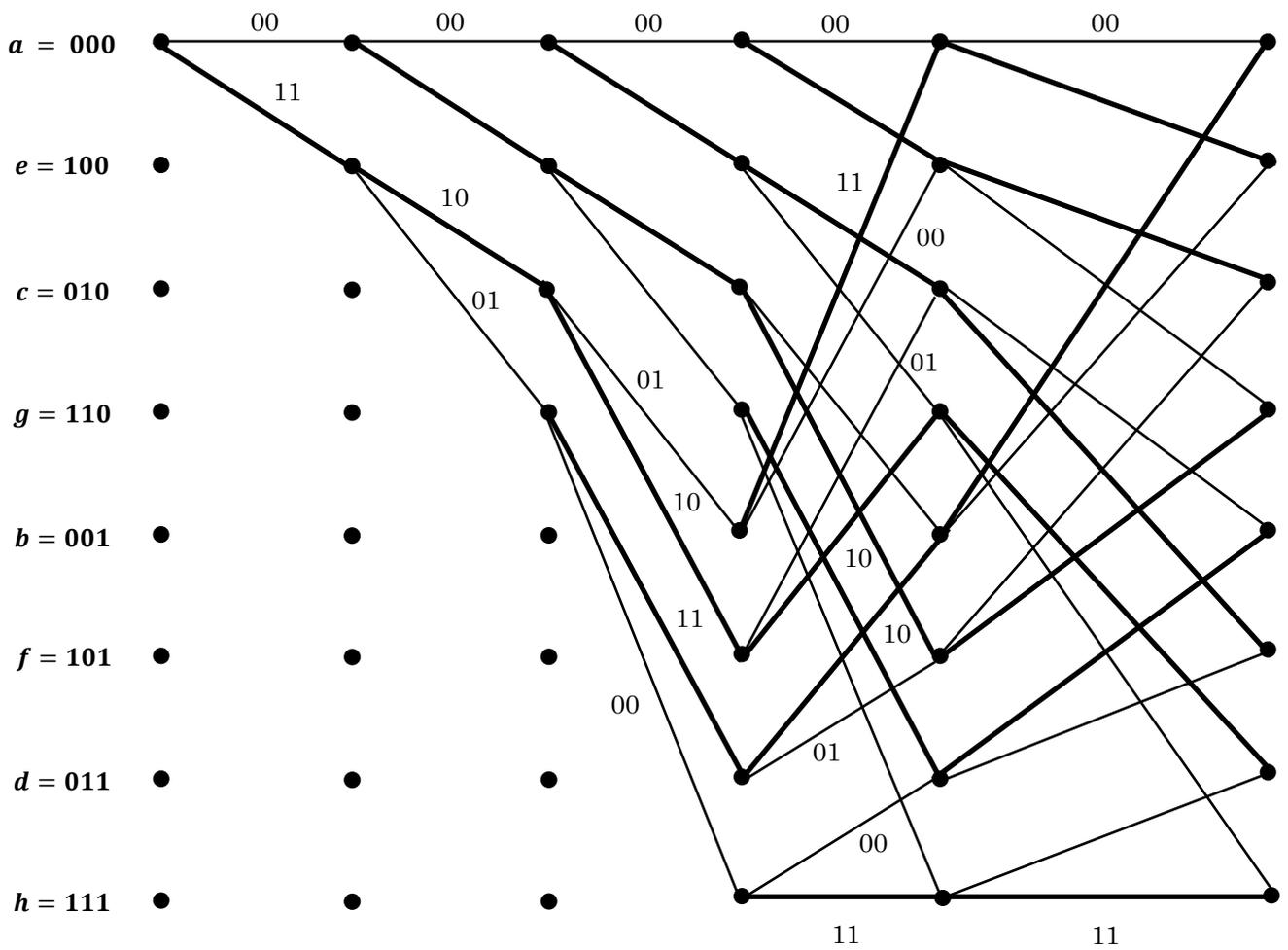


Fig. 4.4.b Représentation en treillis du codeur de la figure 4.3.b.

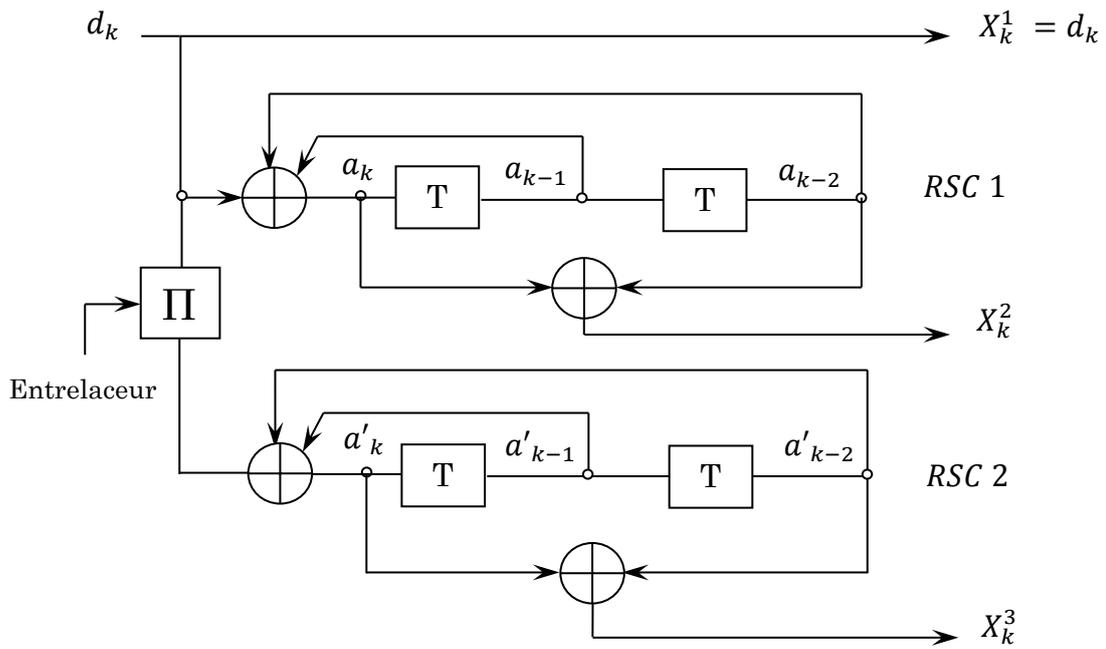


Fig. 4.5.a Turbo code convolutif parallèle (5,7).

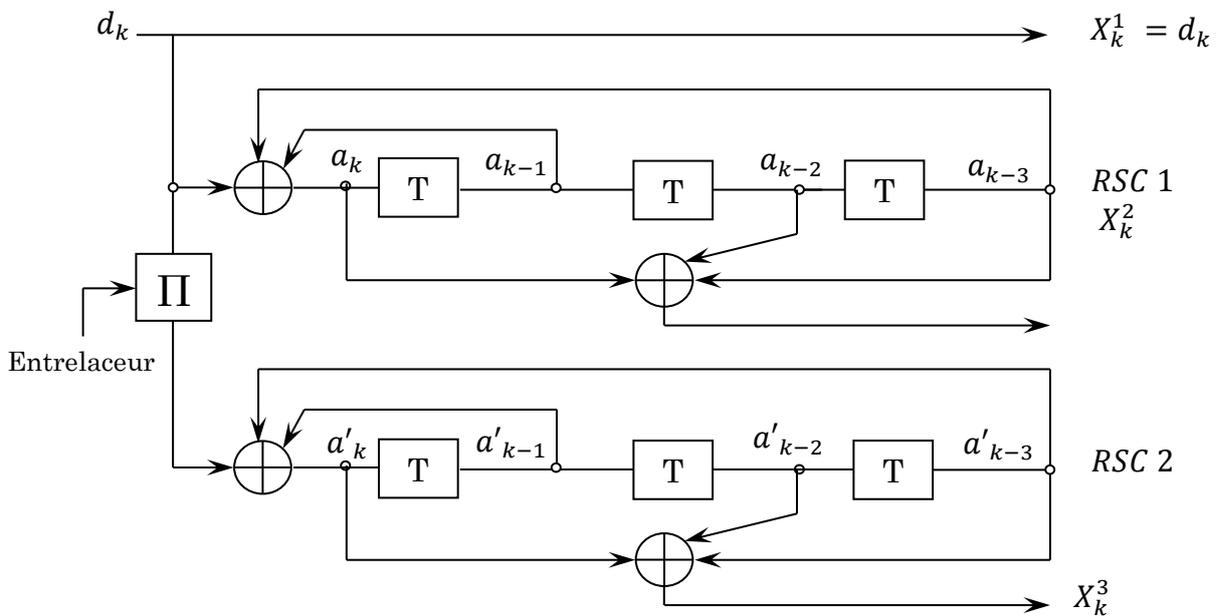


Fig. 4.5.b Turbo code convolutif parallèle (13,15).

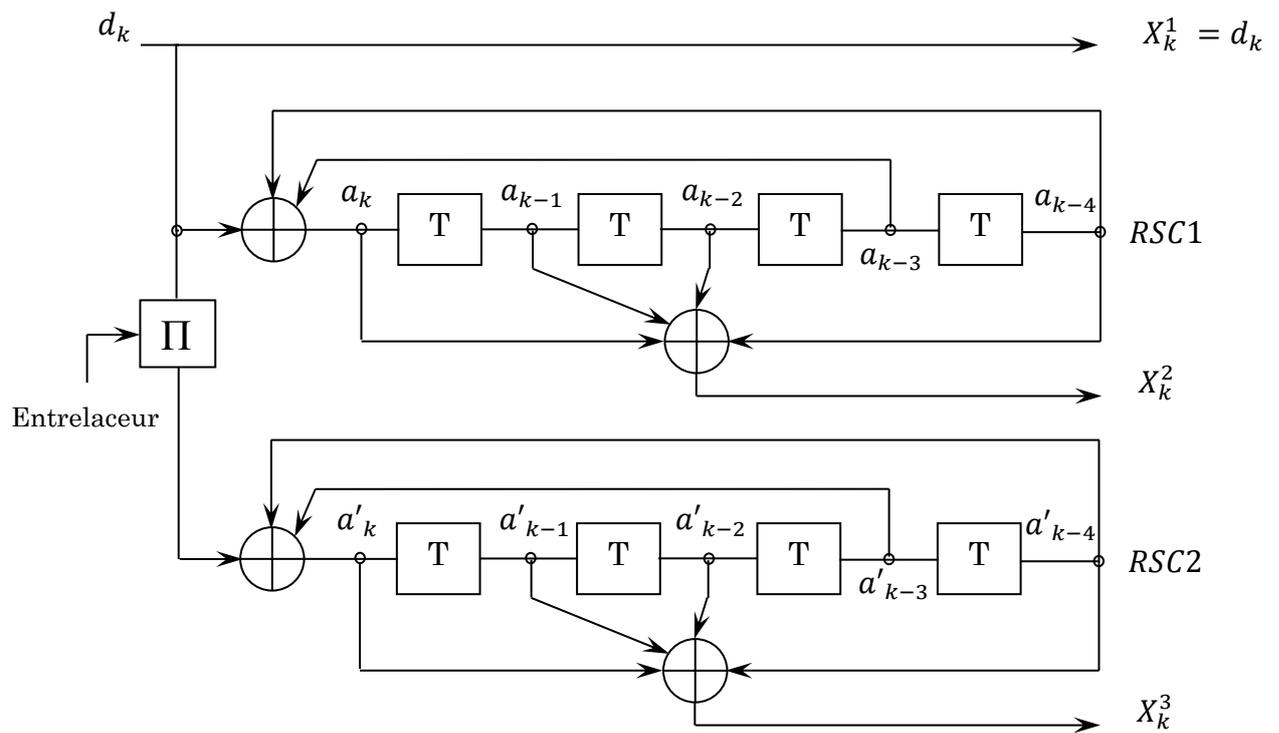


Fig. 4.5.c Turbo code convolutif parallèle (35,23).

4.2.3 Modulation utilisée

Puisque notre travail s'intéresse à l'étude de l'effet du bruit sur les turbo codes, pour cette raison, dans cette simulation notre choix reste limité à la modulation BPSK pour tous les turbo codes employés. La modulation d'un bit X délivré par le turbo code est donnée par (voir chapitre 1)

$$Y = \begin{cases} +\sqrt{E_b} & \text{pour } X = 0 \\ -\sqrt{E_b} & \text{pour } X = 1 \end{cases} \quad (4.1.a)$$

Sans restreindre la généralité, nous pouvons prendre $\sqrt{E_b} = 1$. La modulation devient tout simplement

$$Y = \begin{cases} +1 & \text{pour } X = 0 \\ -1 & \text{pour } X = 1 \end{cases} \quad \text{ou} \quad Y = 1 - 2X \quad (4.1.b)$$

Par conséquent, pour un bruit blanc de variance finie σ^2 , le rapport signal sur bruit qui est défini par $SNR = E_b/N_0$, où $N_0 = 2\sigma^2$ représente la densité de puissance du bruit, devient tout simplement

$$SNR = \frac{1}{N_0} = \frac{1}{2\sigma^2} \quad (4.2.a)$$

Pour un bruit à variance infinie comme le bruit $S\alpha S$ de **Cauchy** de puissance géométrique $S_0 = \gamma^2$, le rapport signal sur bruit géométrique défini par $G\text{SNR} = (2C_g)^{-1} E_b/S_0$ (voir le paragraphe 2.2.3.4 du chapitre 2) devient

$$G\text{SNR} = \frac{1}{2C_g\gamma^2} \quad (4.2. b)$$

4.2.4 Source de bruit

Dans ce travail nous avons trois nouvelles approches comme indique le paragraphe 4.3. La première est basée sur un bruit gaussien AWGN à moyenne nulle, la deuxième utilise un bruit impulsif GG centré et la troisième est appuyée sur un bruit impulsif à variance infinie et moyenne nulle à savoir le bruit α -stable à exposant caractéristique $\alpha = 1$. Par conséquent, nous considérons dans notre simulation trois types de bruit à moyenne nulle

- Bruit gaussien (variance finie);
- Bruit impulsif GG avec un paramètre de forme $\alpha \leq 1$ (variance finie);
- Bruit impulsif $S\alpha S$ de **Cauchy** où $\alpha = 1$ (variance infinie). Le choix du bruit de **Cauchy** est justifié dans le paragraphe 4.3.1.2.

La simulation d'un bruit, représenté mathématiquement par une variable aléatoire, peut être effectuée à l'aide de plusieurs méthodes comme la méthode de l'inversion de la CDF, la méthode du rejet, la méthode de composition et la méthode de **Box-Muller** [105], mais la première méthode (l'inversion de la CDF) est la plus majeure.

4.2.4.1 Simulation d'un bruit gaussien AWGN

Il est possible de simuler une variable aléatoire gaussienne à l'aide de la méthode de **Box-Muller** [105]. Soient U et V deux variables aléatoires uniformément distribuées sur $[0, 1]$. Nous posons $R = \sqrt{-2 \ln V}$ et $\theta = 2\pi U$. Les deux variables $X_0^1 = R \cdot \cos(\theta)$ et $Y_0^1 = R \cdot \sin(\theta)$ sont deux variables gaussiennes centrées et réduites.

Pour obtenir une variable aléatoire gaussienne X de moyenne μ et de variance σ^2 , il reste à multiplier X_0^1 par l'écart type σ et ajouter la moyenne μ .

La figure 4.6 montre le tracé de 5000 échantillons d'un bruit gaussien AWGN générés par la procédure précédente.

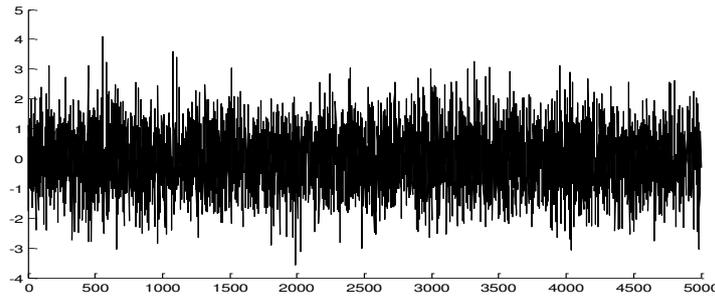


Fig. 4.6 Le tracé de 5000 échantillons d'un bruit gaussien avec $\mu = 0$ et $\sigma = 1$.

4.2.4.2 Simulation d'un bruit GG

Afin de simuler ou de générer de valeurs d'une distribution gaussienne généralisée avec paramètre α , nous pouvons utiliser les trois étapes suivantes (voir l'annexe A)

- simuler une variable aléatoire gamma $Z \sim \text{Gamma}(a, b)$, avec les paramètres $a = \alpha^{-1}$ et $b = A^{-\alpha}$ avec $A = \left[\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)} \right]^{1/2}$;
- appliquer d'abord la transformation $Y = Z^{1/\alpha}$;
- finalement, appliquer une transformation de la forme

$$X = (-1)^B Y \quad (4.3)$$

où B est une variable aléatoire de **Bernoulli** avec paramètre $p = 1/2$.

Avec cette procédure nous pouvons générer des échantillons i.i.d d'un bruit GG de variance égale à l'unité et de moyenne nulle avec un paramètre de forme α .

Pour avoir des échantillons de variance σ^2 et de moyenne μ , nous multiplions les échantillons déjà générés du bruit par σ et nous leurs ajoutons μ .

Il est à noter que si $\alpha = 2$ le bruit GG devient un bruit gaussien AWGN, d'où une deuxième méthode pour simuler un tel bruit. Si $0 < \alpha \leq 1$ le bruit GG simulé est un bruit impulsif. Dans le cas où $\alpha = 1/2$ il est préférable de simuler le bruit GG à l'aide de la méthode de l'inversion de la CDF en basant sur l'évaluation numérique de l'inverse de fonction de **Lambert** W_{-1} (voir chapitre 3).

La figure 4.7 montre le tracé de 5000 échantillons d'un bruit impulsif GG générés par la procédure précédente pour deux valeurs de α . Cette figure montre que plus α est proche de 0 plus le bruit est impulsif.

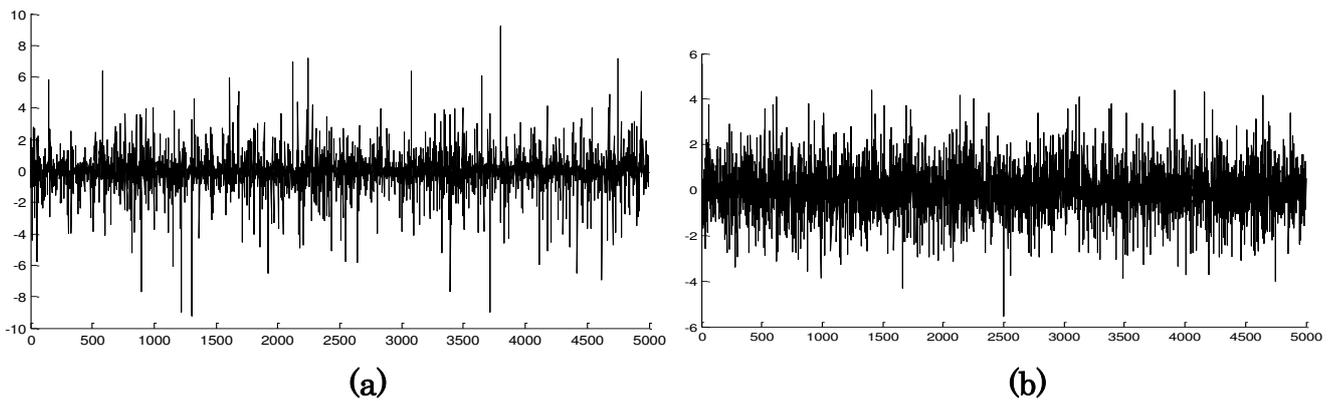


Fig. 4.7 Le tracé de 5000 échantillons d'un bruit impulsif GG avec $\mu = 0$ et $\sigma = 1$ pour deux valeurs de α : (a) $\alpha = 0.5$ (b) $\alpha = 1$.

4.2.4.3 Simulation d'un bruit SaS de Cauchy

Pour ce type de bruit nous utilisons la méthode de l'inversion de la CDF (voir le paragraphe 2.2.2 du chapitre 2). Sachant que la PDF de la loi de **Cauchy** centrée est $f_X(x) = \frac{\gamma}{\pi(x^2 + \gamma^2)}$, nous pouvons évaluer directement la CDF et nous trouvons

$$F(x) = \int_{-\infty}^x \frac{\gamma}{\pi(t^2 + \gamma^2)} dt = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{\gamma}\right) \tag{4.4}$$

et facilement nous ôtons F^{-1} comme suit

$$x = F^{-1}(u) = \gamma \cdot \tan(\pi(u - 1/2)) \tag{4.5}$$

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

La figure 4.8 montre le tracé de 5000 échantillons d'un bruit impulsif SaS de **Cauchy** générés à l'aide de l'équation (4.5) pour deux valeurs de γ . Un simple coup d'œil sur cette figure montre que plus nous augmentons le paramètre γ , plus la dispersion des échantillons augmente aussi, et inversement.

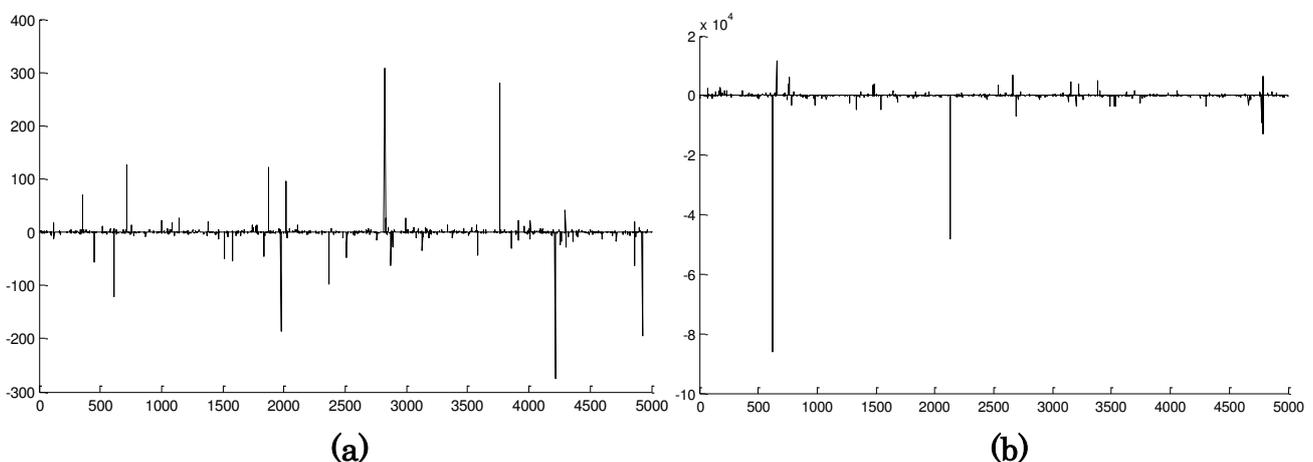


Fig. 4.8 Le tracé de 5000 échantillons d'un bruit impulsif SaS de Cauchy (a) $\gamma = 0.2$ (b) $\gamma = 20$.

4.2.5 Décodeur itératif avec algorithme SISO

Le décodeur itératif avec algorithme SISO de la figure 3.7 dans le chapitre 3 converge lentement pour les faibles rapports signal sur bruit. Pour remédier à cela, nous utilisons une idée simple, inspirée des méthodes numériques itératives, qui consiste à multiplier l'information extrinsèque calculée de chaque décodeur constitutif par un facteur ω , dit facteur de réglage de gain, avant de la réinjectée dans l'autre décodeur constitutif (voir la figure 4.9) [1][106][107][108]. L'utilisation de ce facteur diminue le nombre des itérations presque de la moitié, pour les deux algorithmes SISO, SOVA et MAP [109]. Ce facteur ω est choisi de façon à obtenir un *BER* le plus faible possible pour un *SNR* et un nombre d'itérations donnés [110]. Par exemple, pour le CTC de la figure 4.5.a, la simulation montre que le meilleur facteur de réglage est $\omega \cong 0.1$ si nous utilisons l'algorithme de décodage SOVA, et $\omega \cong 0.2$ si nous utilisons l'algorithme MAP. Bien entendu, nous pouvons encore améliorer les performances de notre décodeur en faisant varier la valeur de ω à chaque itération [1][110].

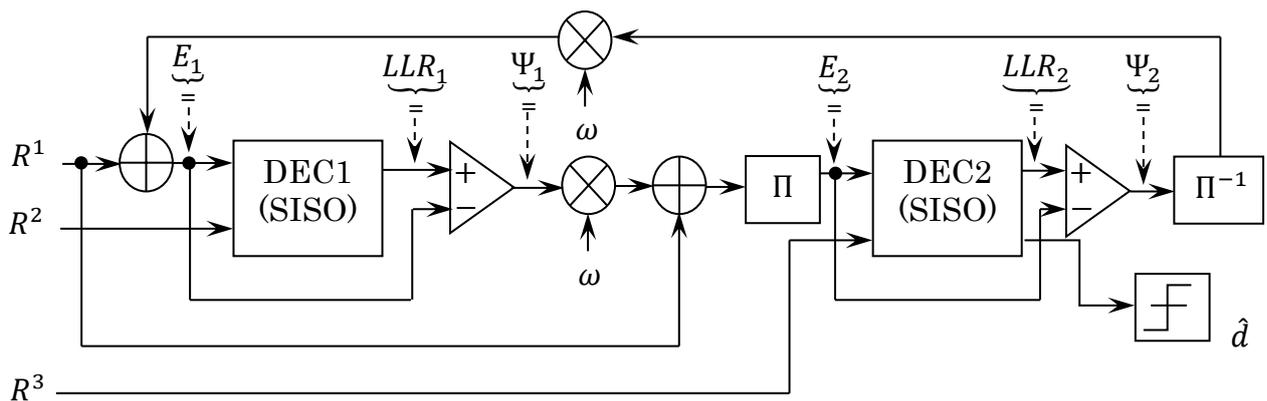


Fig. 4.9 Schéma bloc d'un décodeur itératif avec facteur de réglage de gain.

Dans le schéma de la figure 4.9 nous avons

- $R = R^1 R^2 R^3$ est le signal reçu (noté aussi $R = (R^s, R^p)$ où $R^s = R^1$ et $R^p = R^2 R^3$);
- LLR_1, LLR_2 sont les rapports logarithmiques de vraisemblance;
- Ψ_1, Ψ_2 sont les informations extrinsèques ;
- E_1, E_2 sont les entrées systématiques ;
- ω est le facteur de réglage de gain ;
- Π est l'entrelaceur ;
- Π^{-1} est le désentrelaceur ;
- DEC1, DEC2 sont les décodeurs élémentaires qui travaillent avec un algorithme SISO (SOVA ou MAP).

Au niveau du récepteur le signal reçu est

$$R_k = Y_k + W_k, \quad k = 1, 2, \dots, 3N \quad (4.6)$$

où $3N$ est le nombre des symboles reçu (le coefficient 3 vient du fait que les CTC utilisés ont 3 bits de sortie), $Y_k \in \{-1, +1\}$ sont les symboles émis et W_k sont des variables aléatoires i.i.d qui représentent le bruit avec une moyenne nulle.

D'après le principe de fonctionnement du décodeur itératif, la longueur de la séquence des symboles reçus doit être un multiple de 3 fois la taille de l'entrelaceur utilisé, car le turbo décodeur traite les symboles reçus par blocs de tailles égales à celle de l'entrelaceur. Alors, le traitement de la séquence reçue est périodique avec une période égale à la taille de l'entrelaceur. Nous pouvons prendre alors sans perte de généralité $3N$ égale à 3 fois la taille de l'entrelaceur, c'est-à-dire N égal à la taille de l'entrelaceur.

Dans les deux paragraphes suivants nous présentons l'implémentation des deux algorithmes SISO utilisés par le décodeur itératif à savoir le SOVA et le MAP.

4.2.5.1 L'implémentation de l'algorithme SOVA

Avant d'implémenter l'algorithme SOVA, nous devons expliquer quelques notions de base. Nous commençons tout d'abord par la notion de la fermeture d'un treillis, puis nous définissons les terminologies suivantes : treillis fermé direct et treillis fermé indirect ou inverse.

Inversement à un code convolutif simple qui fait le codage d'une manière continue, un CTC fait le codage de la séquence d'entrée par paquets de taille égale à celle de l'entrelaceur.

Pour le premier paquet de la séquence d'information les codeurs convolutifs du CTC sont à l'état initial, généralement, 00 (dit aussi l'état 0). A seule fin que tous les autres paquets de la séquence en question confrontent le même état (donc les mêmes conditions de codage), nous devons forcer le CTC à l'état 0 par l'ajout des bits non informatifs appelés la queue de paquet ou de séquence, nous obtenons alors pour chaque paquet de la séquence un treillis fermé.

Par exemple, le CTC de la figure 4.5.a, la fermeture de son treillis est illustré dans la figure 4.10.a, et sa queue de séquence correspondante (ou les bits de forçage) qui assure la fermeture du treillis est donnée par le tableau 4.1.

L'état $a_{k-1}a_{k-2}$	La queue q_1q_2	
0	0	0
1	0	1
0	1	0
1	1	1

Tab. 4.1 La queue de séquence du CTC de la figure 4.5.a.

Ce tableau nous donne $q_1 = a_{k-1}(1 - a_{k-2}) + (1 - a_{k-1})a_{k-2}$ et $q_2 = a_{k-1}$.

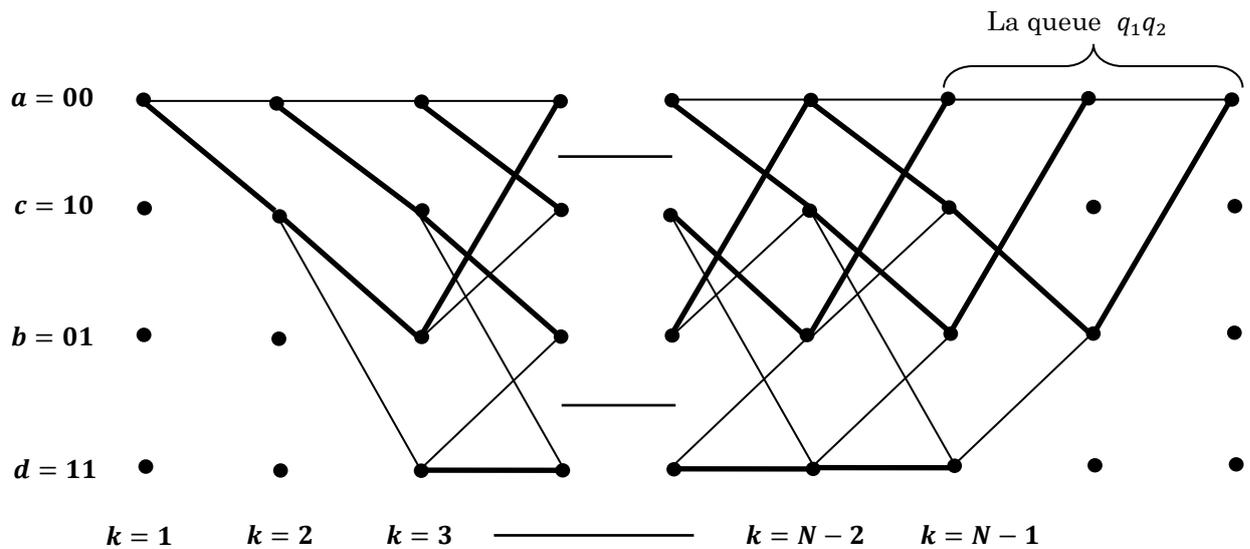


Fig. 4.10.a Fermeture du treillis.

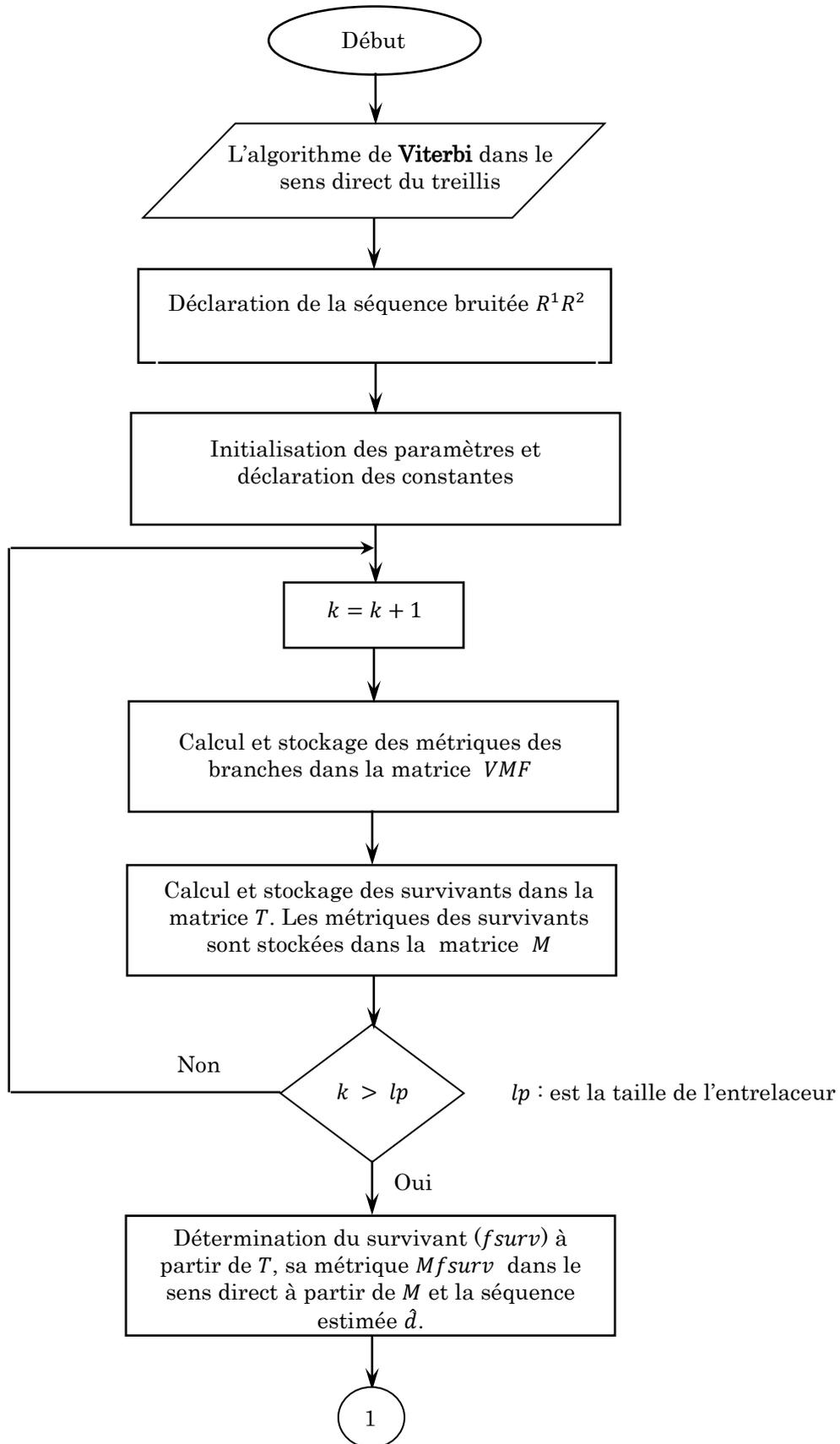
La fermeture du treillis est nécessaire dans le décodage itératif avec l’algorithme SOVA, car ce dernier s’est fondé sur le calcul du chemin le plus probable au sens du ML dans le deux sens du treillis, le sens d’aller et le sens de retour.

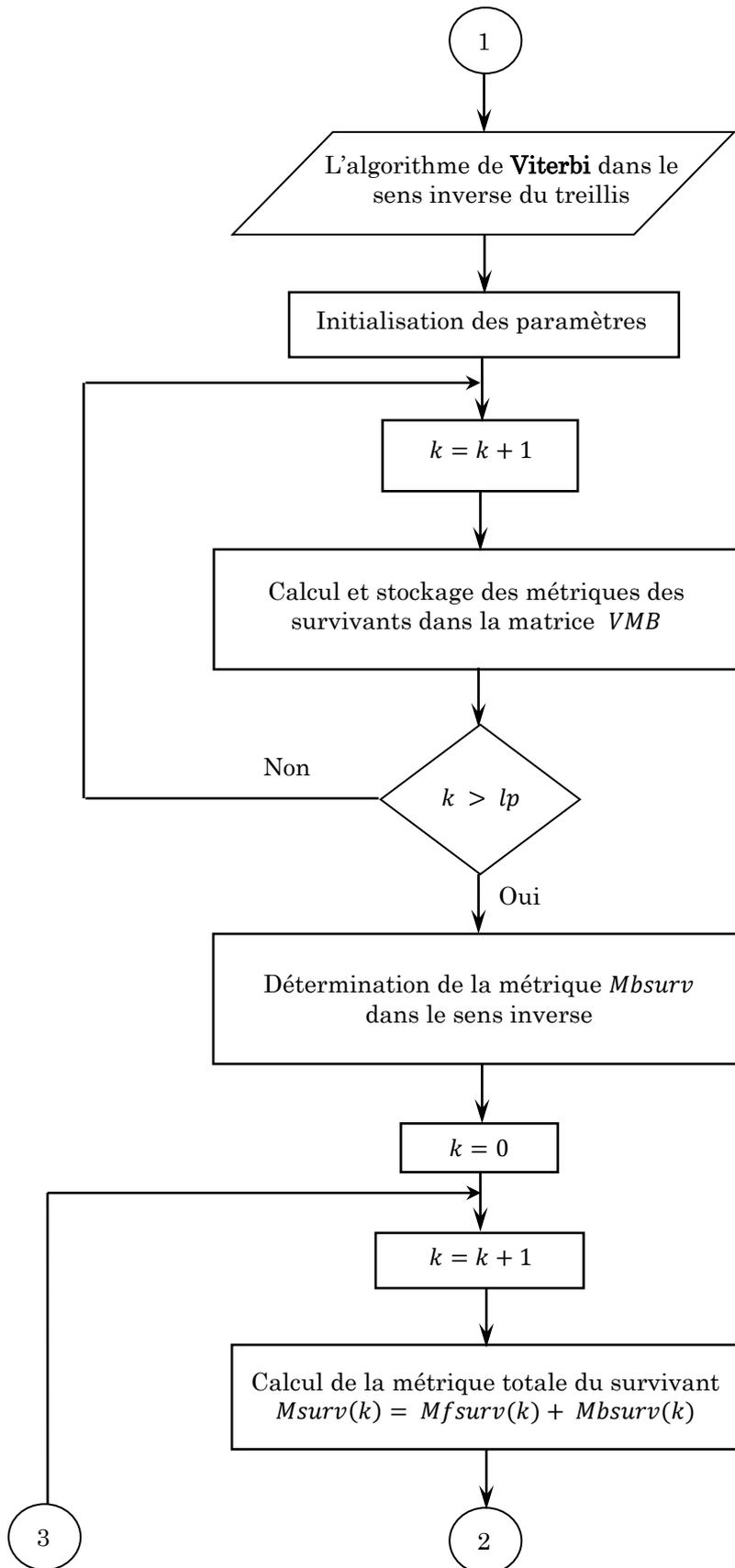
Définition 4.1 : Nous disons qu’un treillis fermé est direct si dans son ouverture les branches qui lient les états entre eux ne chevauchent pas. Par exemple le treillis de la figure 4.4.a.

Définition 4.2 : Le treillis fermé indirect ou inverse est un treillis direct inversé en considérant l’étape finale comme était l’étape initiale.

Comme il est déjà expliqué dans le chapitre 3, le SOVA nécessite les calculs des métriques de corrélation dans les deux sens du treillis. A ce niveau une idée prodigieuse est utilisée dans la programmation du SOVA qui facilite amplement les calculs des métriques. Cette idée est basée sur la remarque qu’un treillis inverse, dans les faits, est un treillis direct en arrangeant les états comme suit : « nous prenons les états du treillis inverse, qui appartiennent aux branches, écrits en binaire, puis nous faisons l’inversion des bits, nous obtenons les états du treillis direct ». Avec cette idée, le SOVA devient comme étant l’algorithme de **Viterbi** classique appliqué deux fois sur le treillis fermé direct. Par exemple, si nous lisons le treillis de la figure 4.4.b dans le sens inverse, le chemin inverse fermé $S_1^{(0)} S_2^{(4)} S_3^{(6)} S_4^{(3)} S_5^{(1)} S_6^{(0)}$, n’est autre que le chemin direct fermé $S_1^{(inv(0))} S_2^{(inv(4))} S_3^{(inv(6))} S_4^{(inv(3))} S_5^{(inv(1))} S_6^{(inv(0))}$ où $inv(n)$ représente l’inversion des bits du nombre n écrit en binaire. Donc, $inv(0) = 0$, $inv(4) = 1$, $inv(6) = 3$, $inv(3) = 6$ et $inv(1) = 4$ et le chemin direct s’écrit tout simplement $S_1^{(0)} S_2^{(1)} S_3^{(3)} S_4^{(6)} S_5^{(4)} S_6^{(0)}$.

L'organigramme de l'implémentation de l'algorithme SOVA est représenté dans la figure 4.10.b.





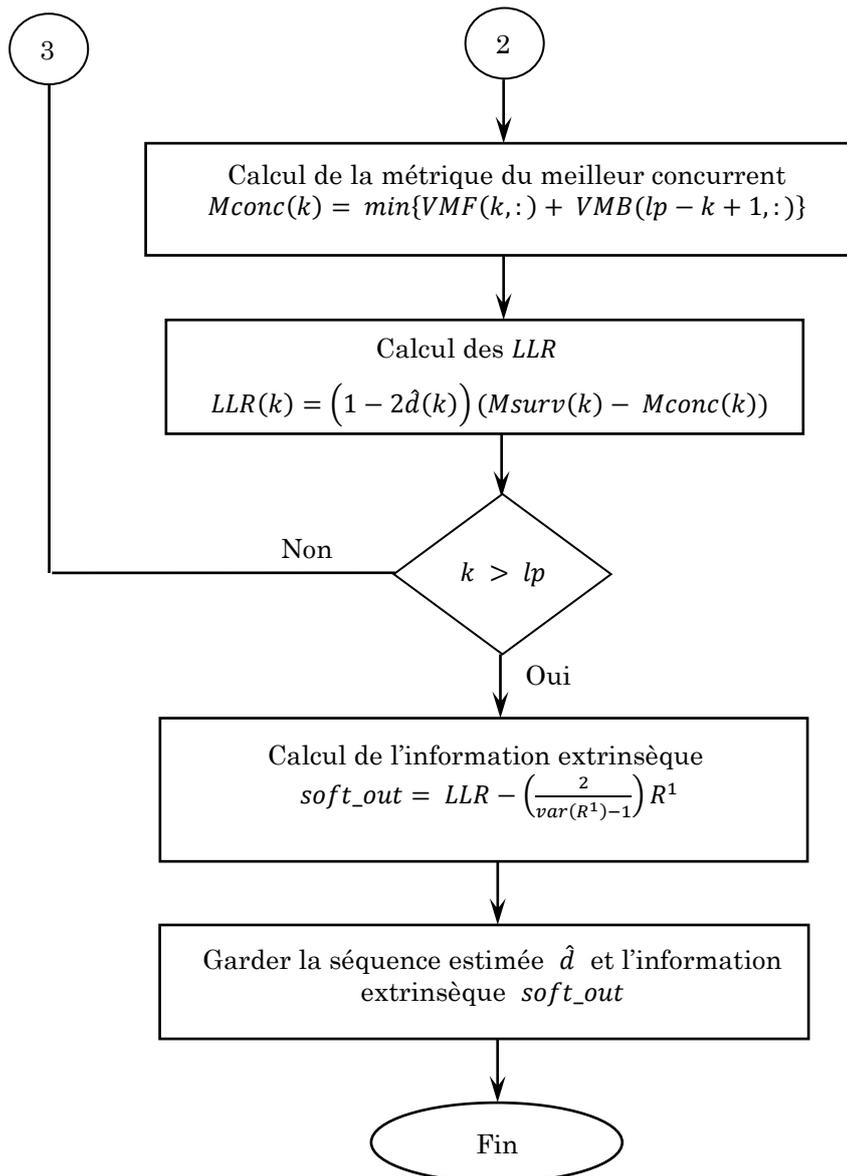


Fig. 4.10.b Organigramme de l'algorithme SOVA.

4.2.5.2 L'implémentation de l'algorithme MAP

Contrairement à l'algorithme SOVA, nous avons vu dans le chapitre 3 que l'algorithme MAP nécessite de connaître les paramètres de la PDF du bruit qui affecte le signal émis. Néanmoins, au niveau de la réception nous n'avons que le signal reçu, qui est une superposition du signal émis et le bruit. Par conséquent, pour chaque type de bruit, nous avons besoin d'une méthode d'estimation méticuleuse afin d'estimer correctement ces paramètres, ce qui pose une grande aporie pour un algorithme qualifié d'optimal.

Pour les deux types du bruit impulsif, gaussien généralisé et α -stable symétrique, nous allons voir dans les paragraphes 4.3.2 et 4.3.3 respectivement, deux méthodes

originales d'estimation utilisées dans l'algorithme MAP. Ces deux méthodes s'appuient principalement sur deux points essentiels, à savoir

- L'utilisation de statistiques d'ordre inférieur fractionnel FLOS ;
- Le calcul de la nouvelle quantité notée ρ , qui représente la probabilité pour que l'observation R_k du signal reçu donnée par l'équation (4.6), soit de même avec l'observation Y_k du signal envoyé ;

La notion de FLOS pour les bruits impulsif GG et SaS, ainsi que la définition et le calcul de la probabilité ρ , sont bien exhibés dans le paragraphe 4.3.1.

Les principales étapes de l'algorithme de décodage MAP sont

Étape 1 : Déclarer le signal reçu $R_k = (R_k^s, R_k^p)$ et estimer judicieusement les paramètres du bruit à partir de la partie systématique $R = (R_k^s)$ et $k = 1, 2, \dots, N$ où N est pris égal à la taille de l'entrelaceur.

Étape 2 : A partir de l'instant $k = 1$ pour $i = 0, 1$ et $m = 0, \dots, 2^\nu - 1$, où ν est la mémoire du CTC, calculer la métrique de branche $\delta_k^i(m)$ suivant l'expression (3.45) pour tous les symboles reçus et les sauvegarder dans un tableau.

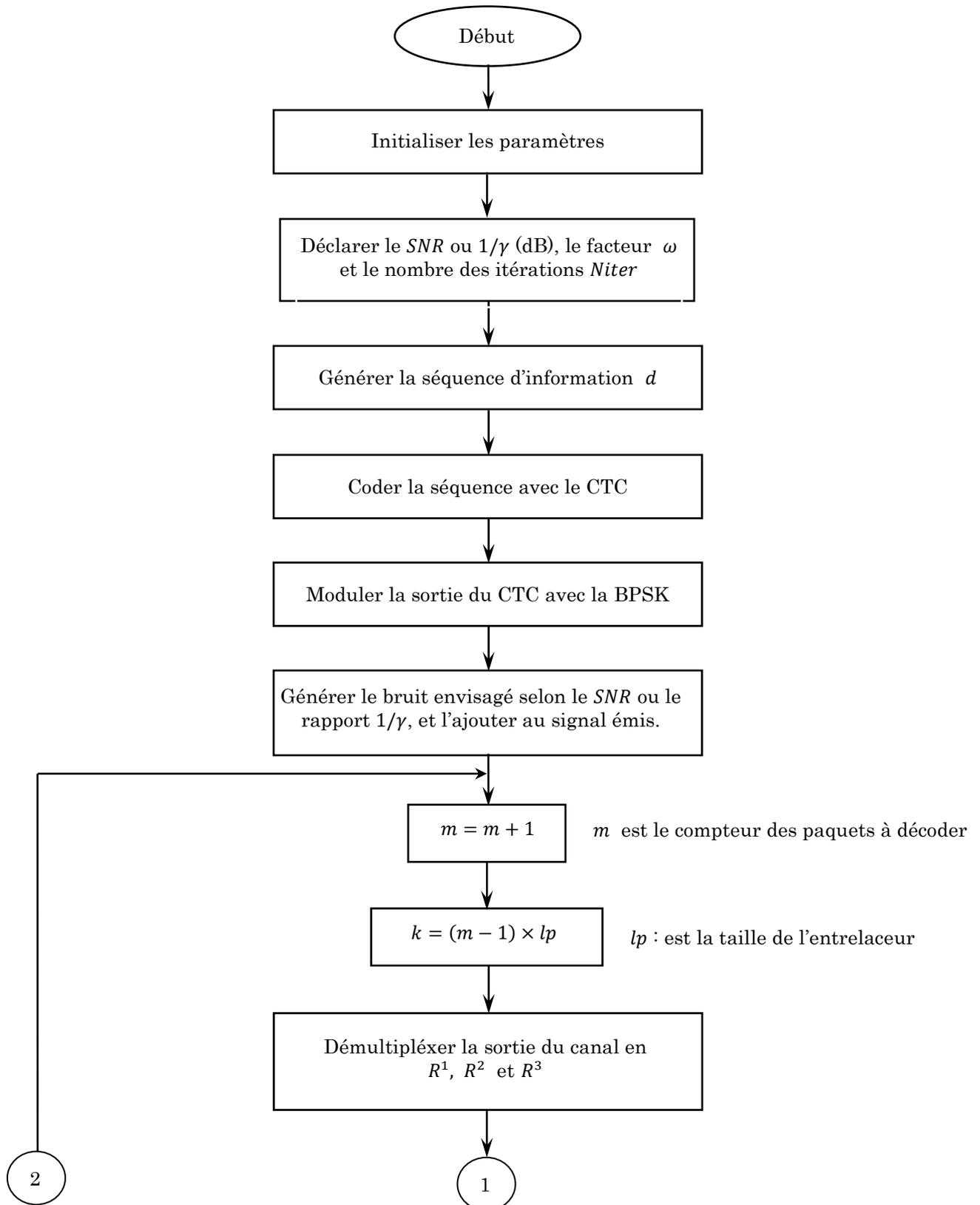
Étape 3 : Initialiser la variable $\alpha_k(m)$ telle que $\alpha_1(0) = 1$ et $\alpha_1(m) = 0$, pour tout $m \neq 0$, et calculer la métrique d'état en avant $\alpha_k(m) = 0$ à partir de l'instant $k = 2$ et pour tous les états ($m = 0, \dots, 2^\nu - 1$) en utilisant l'équation (3.42).

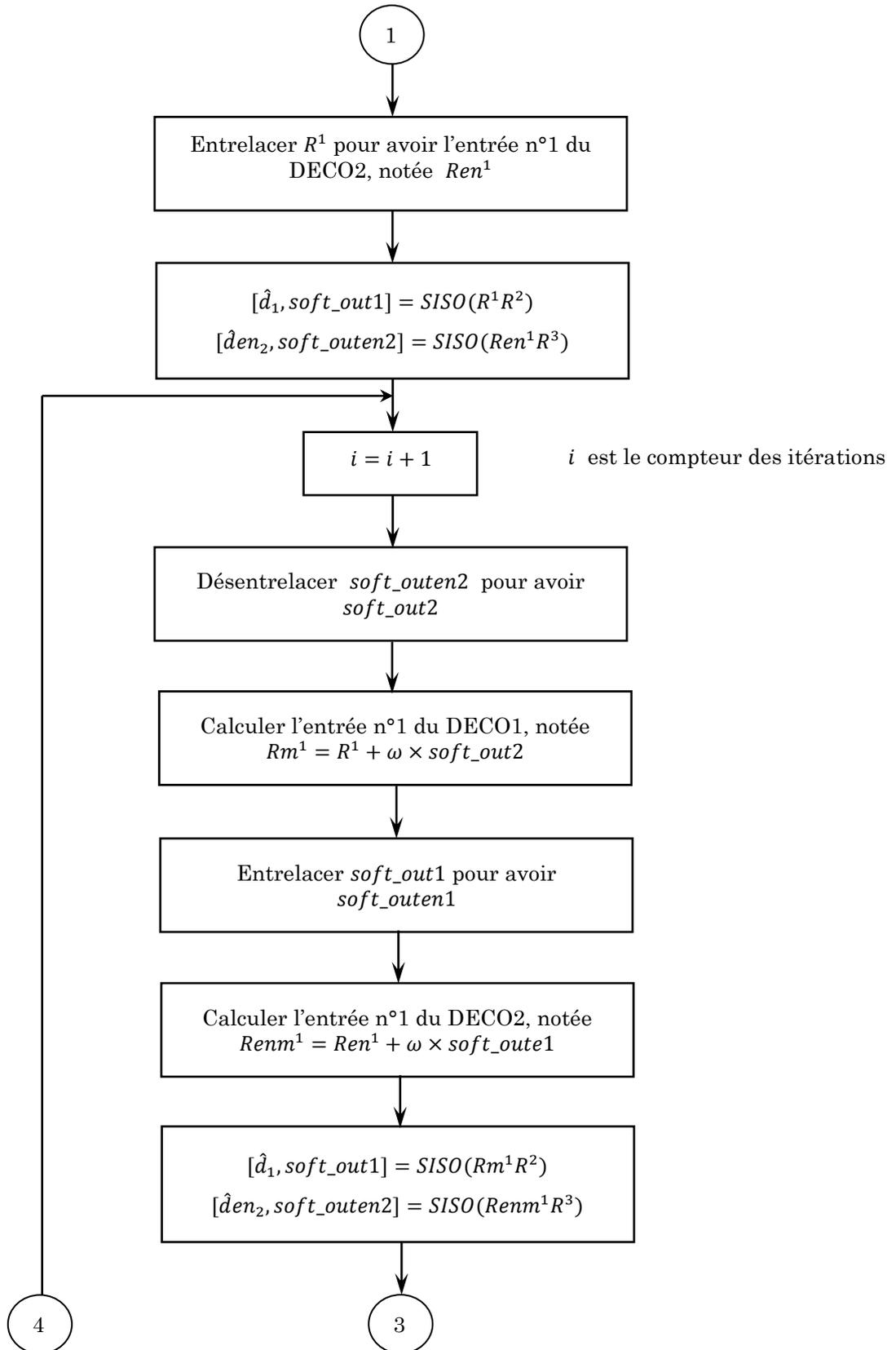
Étape 4 : Initialiser la variable $\beta_k(m)$ telle que $\beta_N(0) = 1$ et $\beta_N(m) = 0$, pour tout $m \neq 0$, et calculer la métrique d'état en arrière $\beta_k(m) = 0$ à partir de l'instant $k = N - 1$ et pour tous les états ($m = 0, \dots, 2^\nu - 1$) en utilisant l'équation (3.43).

4.2.6 Comparateur (Calcul du taux d'erreurs binaire *BER*)

Nous considérons les deux équations (4.2.a) et (4.2.b) : $SNR = \frac{1}{N_0} = \frac{1}{2\sigma^2}$ et $GSNR = \frac{1}{2C_g} \cdot \frac{E_b}{\gamma^2}$. Dans le modèle de la figure 4.1, si la variance du bruit qui affecte le canal est finie, les performances du système numérique sont étudiées et comparées en analysant le *BER* en fonction du *SNR* donné par l'équation (4.2.a). Mais si la variance du bruit est infinie comme le cas du bruit SaS de **Cauchy**, ces performances sont présentées en fonction du *GSNR* donné par l'équation (4.2.b). Ceci représente un grand inconvénient si nous voulons comparer ces performances utilisant ce *GSNR* avec les performances d'un système numérique contaminé par autre bruit SaS différent de celui de **Cauchy**, car le *GSNR* est devient différent de celui de **Cauchy** (voir l'équation (2.43)). Afin de résoudre ce

problème nous avons proposé de présenter les comportements de ces systèmes numériques à étudier en fonction de $1/\gamma$ [7]. Ce choix sera utilisé le long de cette thèse.





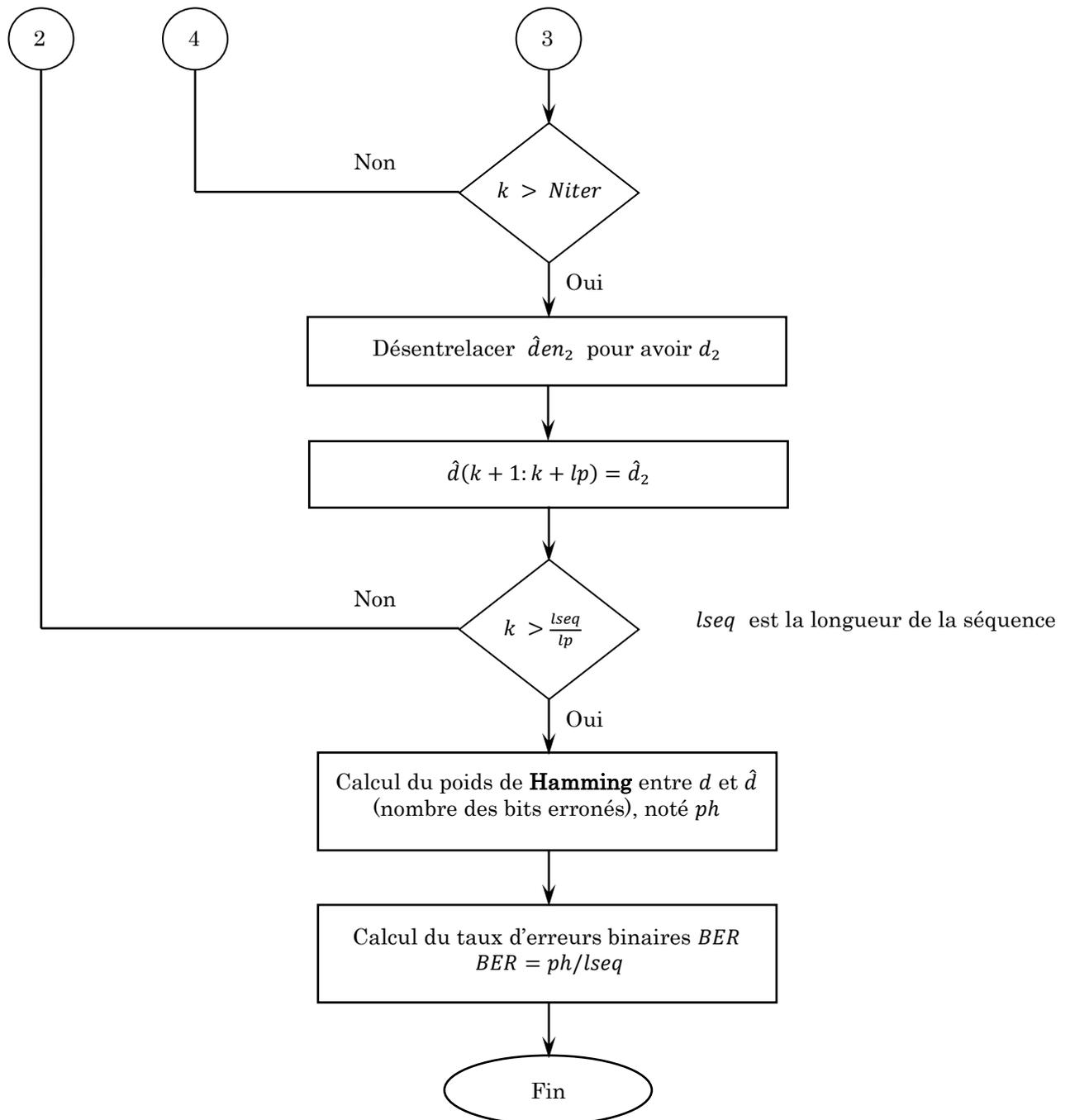


Fig. 4.11 Organigramme de l'implémentation du modèle de la figure 4.1.

Pour un SNR ou un rapport $1/\gamma$ et un nombre d'itérations donnés, nous comptons le nombre des bits erronés par le biais 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 émise.

D'après la méthode de Monte Carlo, pour avoir une meilleure estimation du BER , il faut répéter un grand nombre de fois l'expérience de transmission modélisée par le

modèle de la figure 4.1. La moyenne arithmétique des BER trouvés est une estimation de la valeur réelle du BER , qui est d'autant meilleure que le nombre d'essais est grand.

$$\overline{BER} \cong \frac{1}{N_e} \sum_{n=1}^{N_e} BER_n \quad (4.7)$$

où N_e est le nombre des essais et BER_n est le taux d'erreur binaire calculé au $n^{\text{ème}}$ essai. Nous avons pris le long de notre simulation $N_e = 100$.

Après avoir étalé les éléments spécifiques de chaque bloc dans le modèle de notre simulation de la figure 4.1, nous présentons dans la figure 4.11 l'organigramme d'implémentation du tel modèle.

4.3 Nouvelles contributions à l'amélioration des décodeurs itératifs

Dans ce paragraphe nous présentons trois nouvelles approches pour optimiser un décodeur itératif où chaque approche est établie pour un seul type de bruit parmi les types envisagés dans le paragraphe 4.2.4.

Egalement, nous étudions avec détail l'effet de l'ordre de moment sur l'approximation pratique de l'espérance mathématique par la moyenne arithmétique dans le calcul des moments. En outre, nous définissons une nouvelle quantité appelée la probabilité ρ , la façon de la calculer et son rôle dans l'approximation des statistiques d'ordre inférieur fractionnel FLOS.

En statistique, le terme « statistiques d'ordre inférieur » LOS se réfère à des fonctions qui utilisent au maximum la puissance 2 d'un échantillon. Si la puissance considérée est strictement inférieure à 1, nous parlons de « statistiques d'ordre inférieur fractionnel » FLOS.

Par contre, le terme « statistiques d'ordre supérieur » HOS se réfère à des fonctions qui utilisent la troisième puissance ou plus d'un échantillon [111].

4.3.1 Les FLOS d'un bruit impulsif et la probabilité ρ

Soient W_1, W_2, \dots, W_N , N observations d'un bruit W . La meilleure approximation du moment

$$m_r = E(|W|^r) \quad (4.8)$$

en utilisant les W_k ($k = 1, \dots, N$), est la moyenne arithmétique donnée par

$$\hat{m}_r = \frac{1}{N} \sum_{k=1}^N |W_k|^r \quad (4.9)$$

D'après la méthode de Monte Carlo, pour obtenir une estimation pertinente de m_r , il faut répéter un grand nombre de fois la réalisation des N observations W_k . L'estimation du moment m_r au $n^{\text{ème}}$ essai est alors

$$\hat{m}_{rn} = \frac{1}{N} \sum_{k=1}^N |W_{kn}|^r \quad (4.10)$$

Si nous notons par N_e le nombre d'essais, dès lors, la moyenne arithmétique des moments \hat{m}_{rn} trouvés est une estimation de la valeur réelle du m_r , qui est d'autant meilleure que le nombre d'essais est grand.

$$\hat{m}_r = \frac{1}{N_e} \sum_{n=1}^{N_e} \left(\frac{1}{N} \sum_{k=1}^N |W_{kn}|^r \right) \quad (4.11)$$

Pour toiser la précision donc la qualité de l'estimation du moment m_r nous utilisons la racine carrée de l'erreur quadratique moyenne $RMSE$ donnée par [112]

$$RMSE = \sqrt{E((m_r - \hat{m}_{rn})^2)} \cong \sqrt{\frac{1}{N_e} \sum_{n=1}^{N_e} (m_r - \hat{m}_{rn})^2} = \sqrt{\frac{1}{N_e} \sum_{n=1}^{N_e} \left(E(|W|^r) - \frac{1}{N} \sum_{k=1}^N |W_{kn}|^r \right)^2} \quad (4.12)$$

ou aussi la $RMSE$ relative

$$RMSE_{rel} = \frac{RMSE}{m_r} \cong \sqrt{\frac{1}{N_e} \sum_{n=1}^{N_e} \left(1 - \frac{\hat{m}_{rn}}{m_r} \right)^2} \quad (4.13)$$

La relation (4.12) nous montre théoriquement que

$$\lim_{N \rightarrow +\infty} RMSE = 0, \quad \forall r. \quad (4.14. a)$$

$$\lim_{r \rightarrow 0} RMSE = 0, \quad \forall N. \quad (4.14. b)$$

Pour valider numériquement et mettre à profit ces deux résultats nous avons choisi deux bruits impulsifs GG et SaS, comme montrent les deux paragraphes suivants.

4.3.1.1 Les FLOS d'un bruit impulsif GG

Soient W un bruit GG à moyenne nulle et r un nombre réel différent des entiers impairs avec $r > -1$. Nous avons vu dans le chapitre 2 (voir l'équation (2.4)) que le moment d'ordre r est donné par

$$m_r = E(|W|^r) = \left[\frac{\sigma^2 \Gamma(1/\alpha)}{\Gamma(3/\alpha)} \right]^{r/2} \frac{\Gamma((r+1)/\alpha)}{\Gamma(1/\alpha)} \quad (4.15)$$

Dans ce paragraphe, nous s'intéressons à l'étude de la *RMSE* de l'estimation du moment m_r en fonction de l'ordre r pour un bruit impulsif GG où le paramètre de forme vérifie $0 < \alpha \leq 1$. Dans cette étude, sans perte de généralité, nous posons l'écart type du bruit $\sigma = 1$. Les courbes des figures 4.12 sont considérées pour $N_e = 10\,000$ essais. L'analyse de ces courbes permet de conclure que

- La *RMSE* est une fonction croissante en fonction de r et elle augmente rapidement d'une façon presque exponentielle dans l'intervalle $r \geq 1$;
- La *RMSE* reste faible et inférieur ou égale à 10^{-1} dans l'intervalle $0 < r < 1$ même pour un nombre d'échantillons relativement petit $N = 100$;
- La *RMSE* se diminue avec l'augmentation de N , ce qui est conforme avec la théorie d'après (4.14.a);
- La *RMSE* s'augmente avec la diminution de α si $r \geq 1$. Donc, plus que le bruit est impulsif plus la *RMSE* est grande lorsque r est plus grand que 1 ;
- Dans l'intervalle $0 < r < 1$, la *RMSE* est pratiquement la même pour toutes les valeurs de α comprises entre 0 et 1.
- Plus l'ordre r est petit plus la *RMSE* est faible, ce qui est corroboré par l'égalité (4.14.b).

Par conséquent, pour que l'estimation du moment m_r soit pertinente dans le cas d'un bruit impulsif GG, il faut

- augmenter considérablement le nombre d'échantillons N surtout lorsque l'ordre de moment $r \geq 1$, ce qui est mathématiquement exprimé par l'égalité (4.14.a) ;
- choisir un ordre de moment r proche de 0, et ceci quelque soit le nombre d'échantillons N considéré, même relativement petit. Ce choix est traduit théoriquement par la relation (4.14.b).

Dans le cas d'un système qui travaille en temps réel comme les turbo codes, le premier choix est à éviter. Par contre, tangiblement il est très recommandé de choisir la deuxième proposition. D'où la raison de travailler avec de statistiques d'ordre inférieur fractionnel FLOS dans l'estimation du paramètre de forme d'un bruit impulsif GG.

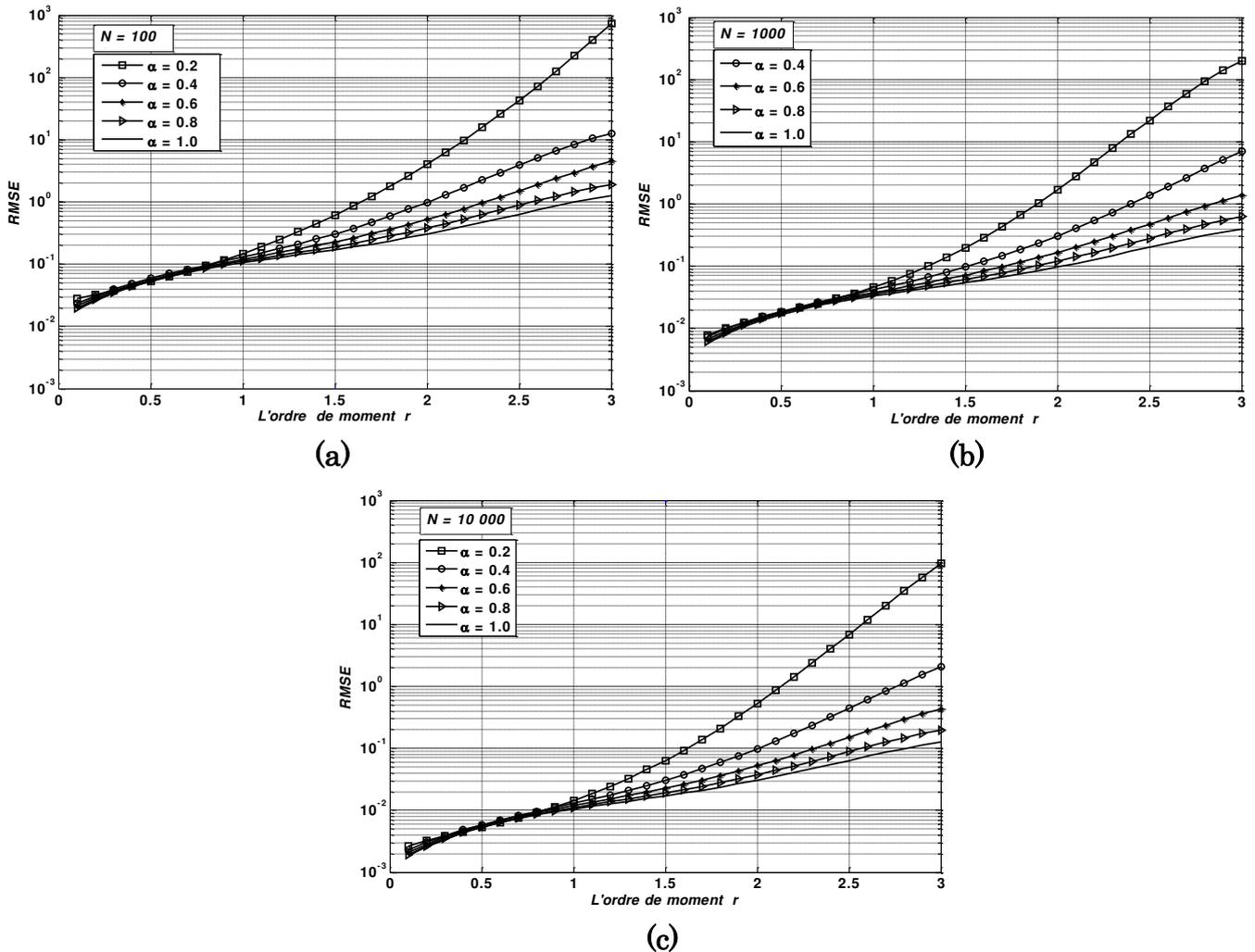


Fig. 4.12 La *RMSE* de l'estimation du moment m_r en fonction de r pour un bruit impulsif GG et pour différentes valeurs de N . (a) $N = 100$ (b) $N = 1000$ (c) $N = 10\,000$.

4.3.1.2 Les FLOS d'un bruit impulsif $S\alpha S$

Soient W un bruit $S\alpha S$ et r un nombre réel tel que $-1 < r < \alpha$. Nous avons vu dans le chapitre 2 (voir l'équation (2.34)) que le moment d'ordre r d'un tel bruit est donné par

$$m_r = E(|W|^r) = \frac{2^{r+1} \cdot \Gamma\left(\frac{r+1}{2}\right) \cdot \Gamma\left(\frac{-r}{\alpha}\right)}{\alpha\sqrt{\pi} \cdot \Gamma\left(\frac{-r}{2}\right)} \cdot \gamma^{r/\alpha} \quad (4.16)$$

La *RMSE* de l'estimation du moment m_r à partir de N observations W_1, W_2, \dots, W_N d'un bruit impulsif $S\alpha S$, est donnée toujours par la formule (4.12).

Dans ce paragraphe, nous s'intéressons à l'étude de cette *RMSE* en fonction de l'ordre r tel que $0 < r < \alpha$. Le paramètre de forme doit suffire à la condition $0 < \alpha < 2$ pour assurer l'impulsivité du bruit $S\alpha S$. Donc, tous les moments m_r d'un tel bruit sont

des statistiques d'ordre inférieur LOS car $r < \alpha < 2$. Dans cette étude, sans restreindre la généralité, nous prenons le paramètre de dispersion $\gamma = 1$. Les courbes des figures 4.13 sont tracées pour $N_e = 10\ 000$ essais.

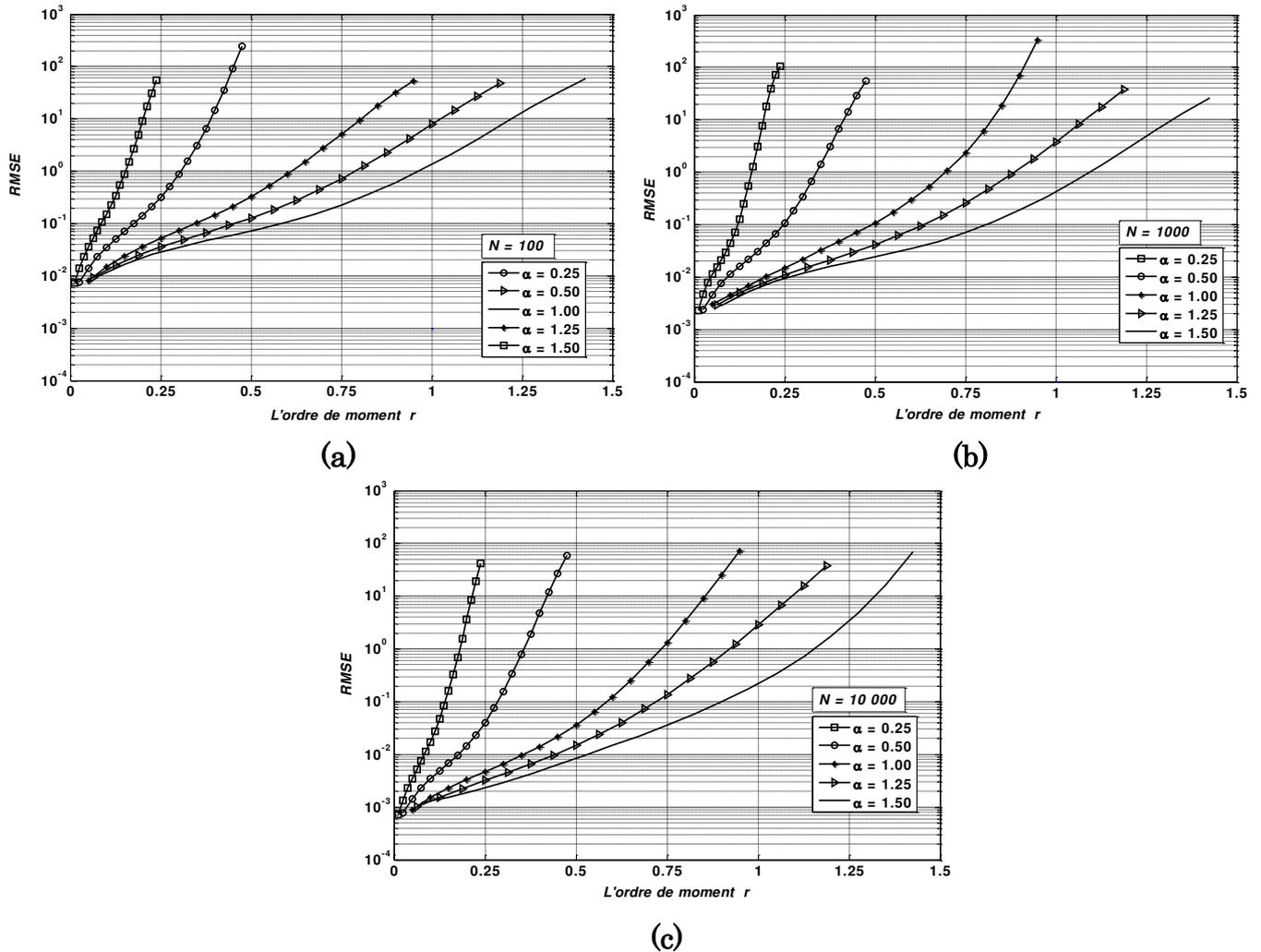


Fig. 4.13 La *RMSE* de l'estimation du moment m_r en fonction de r pour un bruit impulsif *SaS* et pour différentes valeurs de N .
 (a) $N = 100$ (b) $N = 1000$ (c) $N = 10\ 000$.

Ces courbes nous permettent de constater que

- La *RMSE* se diminue si nous augmentons le nombre d'échantillons N , ce qui est devinable selon l'égalité (4.14.a) ;
- La *RMSE* s'accroît en fonction de r et son taux d'accroissement suit un decrescendo en fonction de α ;
- Lorsque r est proche de α , la *RMSE* devient très grande car le moment m_r donné par (4.16) tend vers plus l'infini si r tend vers α ;

- La *RMSE* devient trop petite si l'ordre r est proche de 0, ce qui est confirmé par la relation (4.14.b).

Corollairement, quant à un bruit impulsif *SaS*, nous avons deux choix pour obtenir une estimation rigoureuse du moment m_r , c'est comme dans le cas d'un bruit impulsif GG. Ces deux choix sont

- agrandir gigantesquement le nombre d'échantillons N surtout lorsque l'ordre de moment r est proche de α , ce qui est rigoureusement exprimé par l'égalité (4.14.a) ;
- prendre une valeur de r proche de 0, et ceci quelque soit le nombre d'échantillons N considéré, même comparativement petit. Ce choix est glosé analytiquement par l'équation (4.14.b).

La première proposition est pratiquement à éviter. En revanche, il est vital de pencher à la deuxième proposition en ceci qu'il est très avantageux d'utiliser les FLOS pour estimer les paramètres d'un bruit impulsif *SaS*.

Le problème majeur rencontré dans l'utilisation des FLOS dans le cas d'un bruit impulsif *SaS* est le choix du l'ordre de moment r qui doit vérifier $0 < r < \alpha$. Or, au niveau de la réception le décodeur itératif doit estimer les paramètres du bruit envisagé, à savoir α et γ , en utilisant le moment m_r qui est dépend de r . Autrement dit, comment choisir r qui est majorée par α qui est à son tour dépend de r ?

Pour dépasser ce problème nous choisissons un type bien précis des bruits impulsifs *SaS* à savoir le bruit de **Cauchy** où le paramètre $\alpha = 1$.

Dans tout ce qui suit nous adoptons le bruit de **Cauchy** pour représenter la classe des bruits *SaS*.

4.3.1.3 La notion et le calcul de la probabilité ρ

Dans le chapitre 2 nous avons vu que les deux lois GG et *SaS* envisagées sont symétriques à moyennes nulles, alors leurs PDF $f_X(x)$ sont des fonctions paires. De plus, elles sont des fonctions décroissantes sur \mathbb{R}^+ . Par conséquent, ces deux PDF s'écrivent sous la forme $f_X(x) = g(|x|)$ où g est une fonction continue et décroissante, définie de \mathbb{R}^+ vers \mathbb{R}^+ . Dans ces conditions, comme dans le paragraphe 3.3 du chapitre 3, les densités de probabilité de transition des canaux considérés peuvent s'écrivent donc sous la forme

$$P(R_k/Y_k \text{ émis}) = f_X(R_k - Y_k) = g(|R_k - Y_k|) \quad (4.17)$$

Nous savons bien qu'au niveau du récepteur nous devons 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 (4.17) soit maximisée. Maximiser la dernière expression revient à minimiser la quantité $|R_k - Y_k|$ car la fonction g est décroissante. Pour que cette dernière quantité soit minimale il faut que R_k soit de même signe que Y_k en tenant compte que $Y_k \in \{-1, +1\}$, autrement dit, il faut que le produit $R_k \times Y_k$ soit positif (voir le paragraphe 3.2 du chapitre 3). D'où vient l'idée de calculer la probabilité $\rho = P(R_k \cdot Y_k > 0)$. Nous avons

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

$$\rho = P(1 + Y_k \cdot W_k > 0) = P(Y_k \cdot W_k > -1) \text{ car } R_k = Y_k + W_k \text{ et } Y_k^2 = 1, \text{ alors}$$

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

et avec $P(Y_k = -1) = P(Y_k = +1) = 1/2$, nous trouvons

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

enfin

$$\rho = \frac{1}{2} \cdot \left(1 + \int_{-1}^1 f(w) dw \right) = \frac{1}{2} + \int_0^1 f(w) dw \quad (4.18)$$

Nous remarquons que cette probabilité est toujours supérieure à 1/2.

Tant que les bruits considérés dans ce travail sont impulsifs, alors la densité $f_W(w)$ présente de grandes valeurs au voisinage de 0, donc l'intégrale sur le segment $[0, 1]$ siégée dans l'égalité (4.18) est souvent comparable avec 1/2 et ceci selon l'impulsivité du bruit. Pour cette raison ρ est généralement n'est pas loin de 1. Par conséquent, il est fortement indispensable d'utiliser cette probabilité dans l'estimation des paramètres des bruits impulsifs envisagés.

Dans le cas d'un bruit impulsif GG nous trouvons

$$\rho = \frac{1}{2} + \frac{\alpha}{2 \cdot A \cdot \Gamma(1/\alpha)} \int_0^1 e^{-(w/A)^\alpha} dw, \quad A = \sigma \left(\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)} \right)^{1/2} \quad (4.19)$$

Dans le cas d'un bruit impulsif S α S de **Cauchy** nous obtenons

$$\rho = \frac{1}{2} + \int_0^1 \frac{\gamma}{\pi(w^2 + \gamma^2)} dw = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{1}{\gamma}\right) \quad (4.20)$$

La figure 4.14 montre la variation de la probabilité ρ donnée par l'équation (4.19) en fonction de α pour différentes valeurs de σ . D'après les courbes de cette figure, nous remarquons que pour un bruit impulsif GG faible ($\sigma = 0.5$) la probabilité ρ reste toujours

supérieure à 0.95, et pour un fort bruit ($\sigma = 2$) cette probabilité reste toujours supérieure à 0.75.

La figure 4.15 illustre la variation de ρ de la relation (4.20) en fonction du paramètre de dispersion γ . Cette figure nous permet de conclure que la probabilité ρ varie inversement avec la dispersion du bruit. Néanmoins, cette probabilité reste strictement supérieur à 1/2 quelque soit la dispersion du bruit.

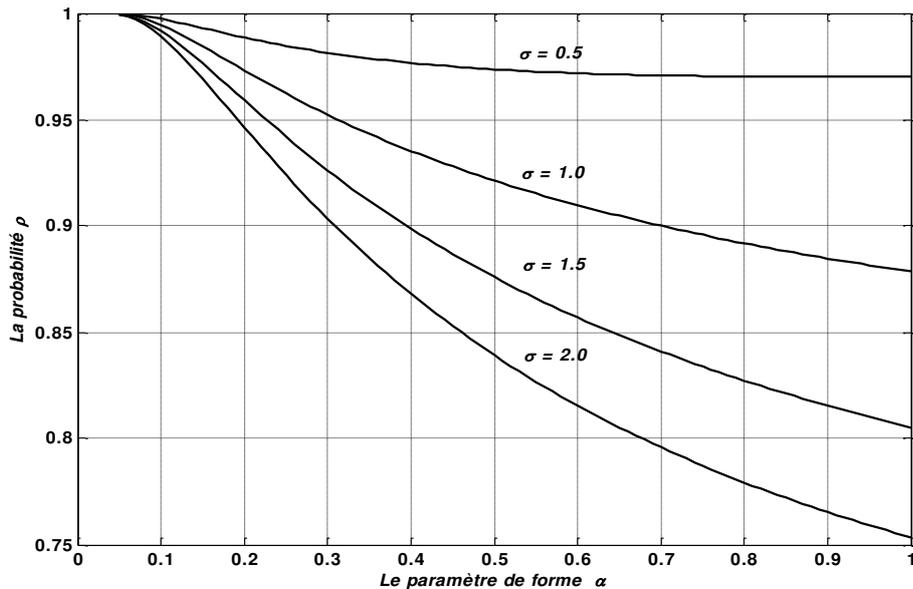


Fig. 4.14 La probabilité ρ en fonction du paramètre de forme α pour un bruit impulsif GG.

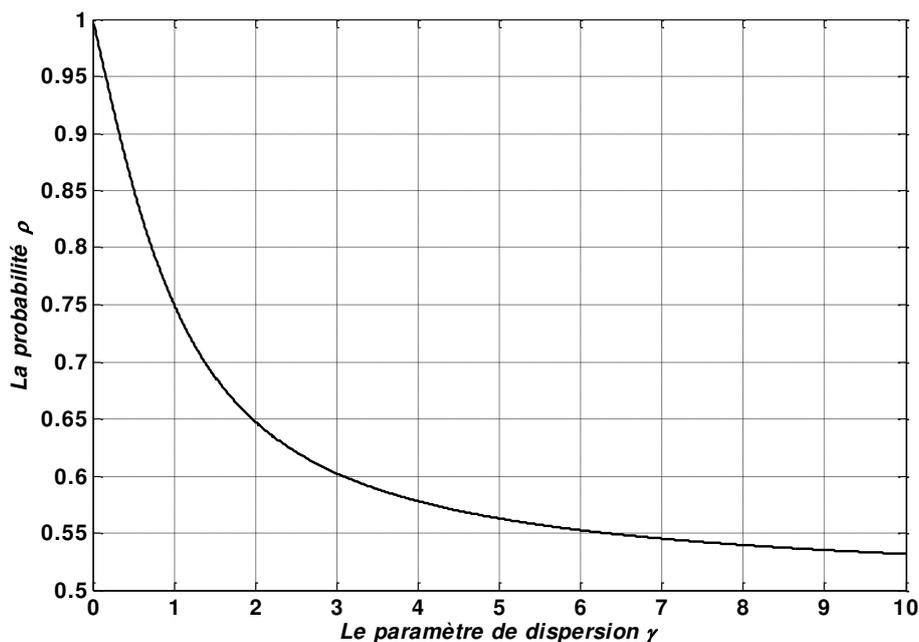


Fig. 4.15 La probabilité ρ en fonction du paramètre de dispersion γ pour un bruit impulsif $S\alpha S$ de Cauchy.

Nous cherchons maintenant à exprimer le moment FLOS m_r en fonction de ρ . Tout d'abord, nous avons

$$Y_k = \begin{cases} \text{sign}(R_k) & \text{avec probabilité } \rho \\ -\text{sign}(R_k) & \text{avec probabilité } 1 - \rho \end{cases} \quad (4.21)$$

alors, pour tout $0 < r < 1$ nous avons

$$|W_k|^r = |R_k - Y_k|^r = \begin{cases} |R_k - \text{sign}(R_k)|^r & \text{avec probabilité } \rho \\ |R_k + \text{sign}(R_k)|^r & \text{avec probabilité } 1 - \rho \end{cases} \quad (4.22)$$

et nous pouvons approximer dans ce cas $|W_k|^r$ par sa valeur moyenne comme suit

$$|W_k|^r \cong \rho \cdot |R_k - \text{sign}(R_k)|^r + (1 - \rho) \cdot |R_k + \text{sign}(R_k)|^r \quad (4.23)$$

Donc, le moment m_r d'ordre $0 < r < 1$ est approximé par

$$m_r = E(|W_k|^r) \cong \rho \cdot E(|R_k - \text{sign}(R_k)|^r) + (1 - \rho) \cdot E(|R_k + \text{sign}(R_k)|^r) \quad (4.24)$$

ou sous la forme simplifiée suivante

$$m_r \cong \rho \cdot b_r + (1 - \rho) \cdot c_r = \rho \cdot (b_r - c_r) + c_r \quad (4.25.a)$$

$$\text{où} \quad b_r = E(|R_k - \text{sign}(R_k)|^r) \quad \text{et} \quad c_r = E(|R_k + \text{sign}(R_k)|^r) \quad (4.25.b)$$

Nous signalons que la formule (4.25.a) reste valable pour tout $r > 0$, bien entendu à condition que le moment m_r est fini.

4.3.2 Approche 1 : Nouvelle approche pour optimiser un décodeur itératif pour un canal AWGN [113]

Pour un faible SNR , les turbo codes ne donnent pas une grande chose en termes du BER même si nous augmentons le nombre d'itérations du décodeur itératif. Les travaux [74][114][115] dans le cas d'une modulation BPSK, ont permis de mettre en évidence cet inconvénient. Un tel inconvénient rend les turbo codes non optimaux. Par conséquent, une question qui se pose ici : « y a-t-il un moyen pour perfectionner des tels codes ? »

Il y a beaucoup des travaux qui traitent le problème de perfectionnement des décodeurs itératifs. Par exemple, en augmentant la distance libre minimale du codeur, soit par poinçonnage de la partie systématique du turbo code [116], ou soit par un choix adéquat de l'entrelaceur [117].

Dans [118], les auteurs utilisent, pour optimiser un décodeur itératif, la théorie du chaos et la théorie de la bifurcation pour analyser l'influence réciproque des processus itératifs et réduire la sous-optimalité d'un processus itératif. Il ressort de cette étude que corriger l'information extrinsèque avec des fonctions f correctement choisies (par exemple, $f(x) = a \cdot x \cdot e^{-b|x|}$ avec $a \in [0.8, 1]$ et $b \in [10^{-3}, 10^{-2}]$) permet une amélioration des performances de décodage.

Dans la pratique, des corrections linéaires sont préférables car moins complexe à mettre en œuvre. Pour cette raison les auteurs dans [107] et [108] ont multiplié l'information extrinsèque par un facteur bien choisi. Il est également possible de faire varier ce facteur au cours des itérations [1].

Dans cette section, pour optimiser un décodeur itératif et le rendre plus efficace surtout pour les faibles SNR , nous proposons une approche originale qui consiste à incorporer une fonction porte ou binaire $p(i)$, qui ne peut prendre que deux valeurs 0 ou 1 en fonction de l'itération i , dans l'entrée systématique du décodeur itératif. La détermination de la fonction $p(i)$ est faite en minimisant le taux d'erreurs binaire BER lors de chaque itération.

Le modèle d'un DCS utilisé pour cette approche est représenté dans la figure 4.1. Les turbo codes employés sont ceux des figures 4.5.a et 4.5.b avec un entrelaceur S –aléatoire pour deux tailles $K = 64 \times 64$ et $K = 128 \times 128$. Le bruit envisagé est un bruit AWGN à moyenne nulle. A partir des observations R_k du signal reçu données par (4.6), le seul paramètre à estimer pour ce bruit est sa variance

$$\sigma^2 = E(W_k^2) = E(R_k^2) - 1 \cong \frac{1}{N} \sum_{k=1}^N R_k^2 - 1 \quad (4.26)$$

Dans le schéma du décodeur itératif de la figure 4.2, que nous allons l'appeler décodeur itératif classique CID, les entrées systématiques, à chaque itération i sont

$$\begin{cases} E_1(i) = R^1 + \omega \cdot \Psi_2(i) \\ E_2(i) = R^1 + \omega \cdot \Psi_1(i) \end{cases} \quad \text{avec } E_1(0) = E_2(0) = R^1 \quad (4.27)$$

Les performances d'un décodeur itératif sont données par les taux d'erreurs binaires BER_1 et BER_2 des décodeurs élémentaires

$$\begin{cases} BER_1 = \text{fonction de } E_1(i) \text{ et } R^2 = \text{fonction de } i \\ BER_2 = \text{fonction de } E_2(i) \text{ et } R^3 = \text{fonction de } i \end{cases} \quad (4.28)$$

Ces taux d'erreurs binaires sont en fait, fonctions du nombre d'itérations pour un SNR donné. Le caractère itératif du décodeur rend BER_1 et BER_2 décroissant avec l'augmentation de i . Il faut noter que ces deux taux d'erreurs convergent vers la même valeur.

Nous remémorons que le problème remarquable dans un décodeur itératif classique est que pour les faibles SNR ne donne pas de meilleures performances en termes de BER si nous augmentons le nombre d'itérations surtout lorsqu'on utilise

l'algorithme SOVA [74][114][115]. Dans le paragraphe suivant nous allons exposer une nouvelle technique qui permet d'améliorer les performances d'un décodeur itératif pour les faibles SNR et nous allons voir que cette technique reste très avantageuse dans la zone des faibles BER .

4.3.2.1 Approche proposée et décodeur itératif modifié MID

Pour minimiser l'effet de l'inconvénient du problème cité ci-dessus dans le paragraphe précédent, nous devons jouer astucieusement sur le paramètre le plus primordial dans le décodeur itératif, à savoir l'information extrinsèque Ψ , parce que le pouvoir de correction d'un tel décodeur dépend étroitement de ce paramètre, et plus ce paramètre est fiable pour chaque itération i , plus le décodeur itératif est performant.

Dans ce contexte, nous avons proposé une méthode originale qui permet d'améliorer colossalement les performances du décodeur itératif surtout dans la zone des faibles SNR [113]. Notre nouvelle méthode consiste à mettre les entrées $E_1(i)$ et $E_2(i)$ du CID sous la forme suivante

$$\begin{cases} E_1(i) = p(i) \cdot R^1 + \omega \cdot \Psi_2(i) \\ E_2(i) = p(i) \cdot R^1 + \omega \cdot \Psi_1(i) \end{cases} \quad (4.29)$$

donc,

$$\begin{cases} BER_1 = \text{fonction de } E_1(i) \text{ et } R^2 = \text{fonction de } i \text{ et } p(i) \\ BER_2 = \text{fonction de } E_2(i) \text{ et } R^3 = \text{fonction de } i \text{ et } p(i) \end{cases} \quad (4.30)$$

où la nouvelle fonction $p(i)$ est une fonction binaire de i ($p(i) \in \{0,1\}$) à déterminer de façon à minimiser le $BER(i, p(i))$ à chaque itération.

Il faut noter que l'idée de cette approche est inspirée de la méthode mathématique dite « variation des constantes » utilisée dans les équations différentielles. Ipso facto, au lieu de multiplier l'entrée R^1 par un « 1 » pour chaque itération i , dans les expressions de E_1 et E_2 , nous la multiplie par une fonction variable $p(i)$.

La détermination de la fonction binaire $p(i)$ est faite par simulation numérique en travaillant avec le modèle de la figure 4.1. Nous rappelons que $p(i) = 0$ (resp. $p(i) = 1$) signifie que le décodeur itératif travaille à l'itération i sans (resp. avec) l'entrée systématique. L'algorithme de calcul de la fonction $p(i)$ est le suivant

Début

Lire le nombre d'itérations Nr

Pour $i = 1 \rightarrow Nr$

Calculer $BER(i, p(i) = 0)$ et $BER(i, p(i) = 1)$

Si $BER(i, p(i) = 0) \leq BER(i, p(i) = 1)$

Alors $p(i) = 0,$

sinon $p(i) = 1.$

Garder la valeur de $p(i)$ dont le BER est minimal.

Fin pour

Fin.

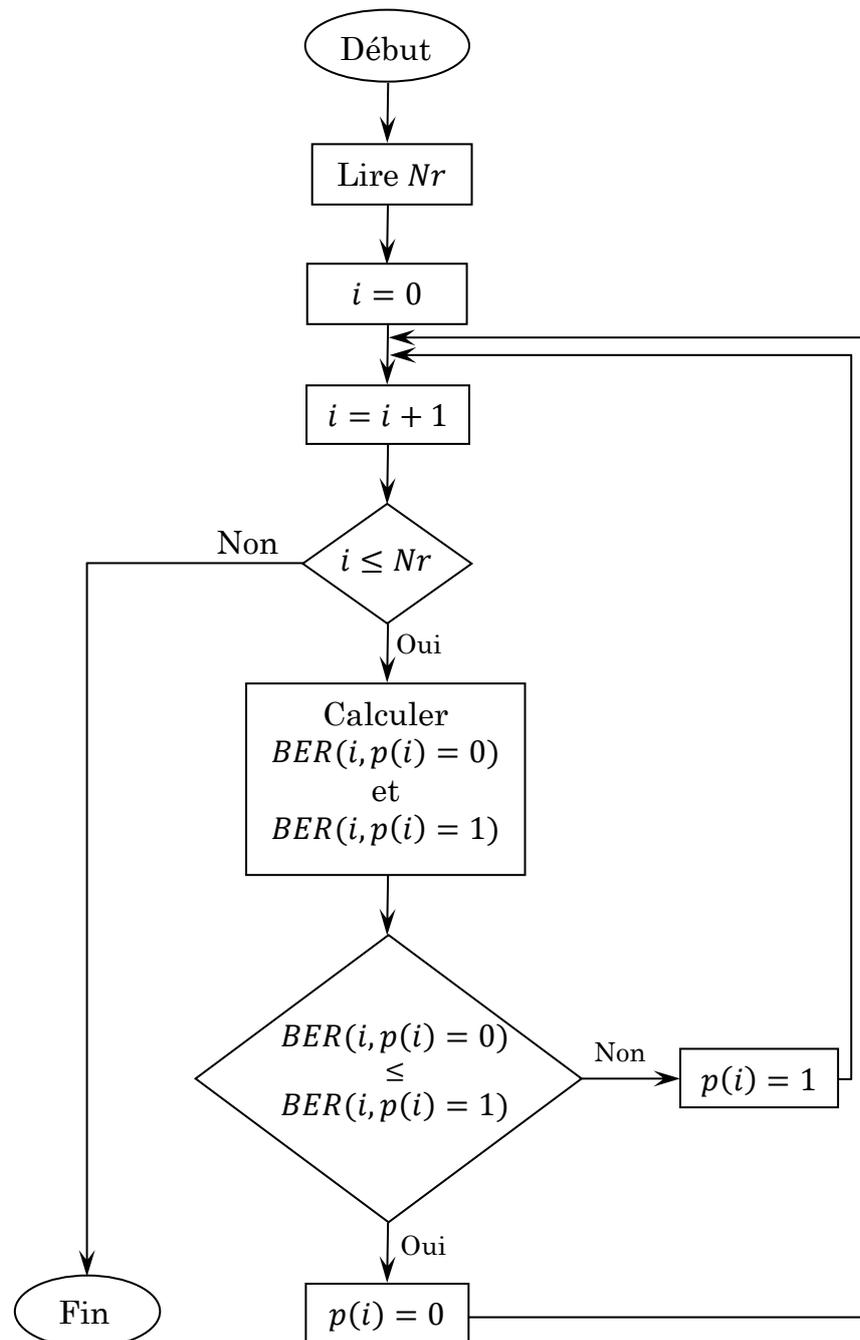


Fig. 4.16 L'organigramme de l'algorithme de calcul de $p(i)$.

- Figure 4.18.a : illustre le *BER* en fonction du *SNR* pour 6 itérations (#6) des décodeurs itératifs à 4 états et entrelaceur 15 –aléatoire de taille 64×64 ;
- Figure 4.18.b : illustre le *BER* en fonction du *SNR* pour 12 itérations (#12) des décodeurs itératifs à 4 états et entrelaceur 15 –aléatoire de taille 64×64 ;
- Figure 4.19.a : illustre le *BER* en fonction du *SNR* pour 6 itérations (#6) des décodeurs itératifs à 8 états et entrelaceur 15 –aléatoire de taille 64×64 ;
- Figure 4.19.b : illustre le *BER* en fonction du *SNR* pour 12 itérations (#12) des décodeurs itératifs à 8 états et entrelaceur 15 –aléatoire de taille 64×64 ;
- Figure 4.20.a : illustre le *BER* en fonction du *SNR* pour 6 itérations (#6) des décodeurs itératifs à 4 états et entrelaceur 15 –aléatoire de taille 128×128 ;
- Figure 4.20.b : illustre le *BER* en fonction du *SNR* pour 12 itérations (#12) des décodeurs itératifs à 4 états et entrelaceur 15 –aléatoire de taille 128×128 ;
- Figure 4.21.a : illustre le *BER* en fonction du *SNR* pour 6 itérations (#6) des décodeurs itératifs à 8 états et entrelaceur 15 –aléatoire de taille 128×128 ;
- Figure 4.21.b : illustre le *BER* en fonction du *SNR* pour 12 itérations (#12) des décodeurs itératifs à 8 états et entrelaceur 15 –aléatoire de taille 128×128 ;
- En plus, les figures 4.20.b et 4.21.b illustrent le *BER* en fonction du *SNR* pour 12 itérations (#12) des décodeurs itératifs modifiés à 4 états et à 8 états, respectivement, avec entrelaceur 15 –aléatoire de taille 128×128 , algorithme de décodage MAP et fonction porte périodique $pp(i) = 100100100100$.

A. Performance d'un décodeur itératif

L'analyse des figures 4.18, 4.19, 4.20 et 4.21 permet de constater que

- L'algorithme MAP est plus performant que l'algorithme SOVA en termes de *BER* et en termes de rapidité de convergence car l'algorithme MAP est considéré comme étant l'algorithme optimal pour le décodage des CTC basés sur les codes RSC, contrairement à l'algorithme SOVA qui est sous optimal [1][79][119].
- Le *BER* d'un décodeur itératif est inversement proportionnelle au *SNR*, ce qui confirme davantage la théorie, puisque pour un grand *SNR*, la séquence émise n'est affectée que peu par le bruit. Donc, le décodeur de vraisemblance maximale aura à son entrée une séquence avec un nombre faible de bits erronés, d'où une probabilité faible, et vice versa ;
- La rapidité de convergence du décodeur itératif augmente avec le nombre d'états et avec le *SNR*.

La comparaison des figures 4.18 et 4.19 avec les figures 4.20 et 4.21 respectivement, permet de conclure que l'augmentation du nombre d'états améliore les performances du décodeur itératif, mais en contrepartie la complexité du décodeur augmente aussi.

La comparaison des figures 4.18.a, 4.19.a, 4.20.a et 4.21.a avec les figures 4.18.b, 4.19.b, 4.20.b et 4.21.b respectivement, permet de conclure que l'augmentation du nombre d'itérations améliore amplement les performances du décodeur itératif, ce qui prouve le pouvoir correctif d'un tel décodeur, mais au détriment du temps de décodage qui devient plus grand.

La comparaison des figures 4.18 et 4.20 avec les figures 4.19 et 4.21 respectivement, permet de conclure que l'augmentation de la taille de l'entrelaceur perfectionne les performances du décodeur itératif, mais il nécessite plus de ressources et de temps de calcul et induit un retard plus important. Il est important de signaler que les performances du décodeur itératif sont étroitement liées à la taille et à la nature de l'entrelaceur [114].

B. Comparaison entre un MID et un CID

Pour les décodeurs itératifs associés aux figures 4.3.a et 4.3.b avec deux entrelaceurs S -aléatoires ($S = 15$ pour les deux entrelaceurs 64×64 et 128×128) et pour les deux algorithmes de décodage SOVA et MAP, la simulation avec notre approche nous permet de trouver la fonction porte $p(i)$ dans chaque cas. Les résultats trouvés sont illustrés dans le tableau 4.2.

		$p(i = 1, \dots, 12)$														
		$i =$	1	2	3	4	5	6	7	8	9	10	11	12		
Décodeur itératif modifié	4 états	64×64	SOVA	1	0	0	1	1	0	1	1	0	0	1	0	
			MAP	1	0	0	1	0	0	1	0	0	1	1	1	
		128×128	SOVA	1	0	0	1	1	0	0	1	1	1	1	0	0
			MAP	1	0	0	1	0	0	1	0	1	1	0	1	
	8 états	64×64	SOVA	1	0	0	1	1	0	0	1	0	1	1	0	
			MAP	1	0	0	1	0	0	1	0	1	1	1	0	
		128×128	SOVA	1	0	0	1	1	0	0	0	1	1	0	1	
			MAP	1	0	0	1	0	0	1	1	0	0	1	0	

Tab. 4.2 La fonction porte $p(i)$ pour différents décodeurs itératifs.

D'après le tableau 4.2 nous remarquons que

1. Pour les 4 premières itérations, la fonction $p(i)$ est la même pour tous les décodeurs itératifs étudiés. Donc, dans notre travail, $p(i \leq 4)$ ne dépend ni du nombre d'états, ni de la taille de l'entrelaceur S –aléatoire et ni de l'algorithme de décodage ;
2. Pour les 6 premières itérations, la fonction $p(i)$ est la même pour tous les décodeurs itératifs étudiés qui travaille avec l'algorithme SOVA. Donc, dans notre travail, $p(i \leq 6)$ ne dépend ni du nombre d'états et ni de la taille de l'entrelaceur S –aléatoire ;
3. Pour les 7 premières itérations, la fonction $p(i)$ est la même pour tous les décodeurs itératifs étudiés qui travaille avec l'algorithme MAP. Donc, dans notre travail, $p(i \leq 7)$ ne dépend ni du nombre d'états et ni de la taille de l'entrelaceur S –aléatoire ;
4. La fonction $p(i)$ ($i > 6$ pour le SOVA et $i > 7$ pour le MAP) dépend évidemment du nombre d'états, de la taille de l'entrelaceur S –aléatoire et de l'algorithme de décodage utilisé ;
5. Tous les MID étudiés travaillent comme suit
 - La présence de la partie systématique du signal reçu est obligatoire dans la 1^{ère} et la 4^{ème} itération ;
 - Dans la 2^{ème}, la 3^{ème} et la 6^{ème} itération les MID relâchent la partie systématique du signal reçu ;
 - Dans la 5^{ème} itération les MID avec l'algorithme SOVA nécessite la présence de la partie systématique du signal reçu, contrairement aux MID avec l'algorithme MAP ;
6. Dans le cas de l'algorithme MAP la fonction $p(i)$ ne diffère que par quelques bits par rapport à la fonction périodique $pp(i) = 100100100100$ pour $Nr = 12$. Pour cette raison nous avons fait une comparaison entre un MID avec une $p(i)$ trouvée dans notre approche et un MID avec la fonction périodique $pp(i)$ en utilisant bien entendu l'algorithme MAP (voir les figures 4.20.b et 4.21.b) ;
7. Dans la fonction $p(i)$ et pour $Nr = 12$ itérations nous avons la propriété suivante « il y a autant des « zéros » que des « un », sauf la fonction $p(i)$ dans la dernière ligne du tableau 4.2 où il y a 5 « un » et 7 « zéros » ». Néanmoins, nous avons constaté que pour une grande valeur de Nr nous retrouvons que cette propriété est vérifiée.

La comparaison des figures 4.18.a, 4.19.a, 4.20.a et 4.21.a avec les figures 4.18.b, 4.19.b, 4.20.b et 4.21.b respectivement, permet de conclure que dans la zone des faibles SNR , l'augmentation du nombre d'itérations n'améliore pas les performances du CID surtout avec l'algorithme SOVA, contrairement au cas du MID où l'augmentation du nombre d'itérations améliore avec une façon remarquable les performances du décodeur même avec l'algorithme SOVA. Le tableau 4.3 montre un exemple numérique tiré des figures 4.19.a et 4.19.b qui confirme ce résultat.

		$SOVA(SNR = 0 \text{ dB})$	$MAP(SNR = -0.5 \text{ dB})$
CID	# 6	$BER \cong 2 \times 10^{-2}$	$BER \cong 3 \times 10^{-2}$
	# 12	$BER \cong 2 \times 10^{-2}$	$BER \cong 10^{-2}$
MID	# 6	$BER \cong 10^{-2}$	$BER \cong 2 \times 10^{-2}$
	# 12	$BER \cong 6 \times 10^{-3}$	$BER \cong 3 \times 10^{-3}$

Tab. 4.3 Comparaison MID / CID dans la zone des faibles SNR en fonction du nombre d'itérations.

Les figures 4.18, 4.19, 4.20 et 4.21 montrent que

- Le MID est plus performant que le CID quelque soit le SNR , et ceci pour les deux algorithmes de décodage SOVA et MAP, et que son pouvoir correctif des erreurs augmente par rapport à celui du CID si nous faisons augmenter le nombre d'itérations, le SNR ou le nombre d'états du décodeur itératif ;
- Le CID avec l'algorithme de décodage MAP est toujours plus performant que le MID avec l'algorithme de décodage SOVA ;
- Malgré que notre approche est faite au début pour améliorer le décodeur itératif dans la zone des forts BER (faibles SNR) mais la simulation montre que cette approche est très avantageux dans la zone de faibles BER (fortes SNR), car la figure 4.20.b, par exemple, montre que pour un $BER = 10^{-5}$ nous avons gagné en termes de SNR pour le MID par rapport au CID avec le SOVA 0.36 dB et avec le MAP 0.37 dB et la figure 4.21.b montre aussi que pour un $BER = 10^{-5}$ nous avons gagné en termes de SNR pour le MID par rapport au CID avec le SOVA 0.42 dB et avec le MAP 0.32 dB . Donc, nous pouvons conclure que notre approche améliore le CID presque de 0.35 dB en termes de SNR .

La figure 4.20.b illustre une comparaison entre le MID à 4 états et entrelaceur de taille 128×128 avec l'algorithme MAP (donc $p(i) = 100100101101$) et le même MID mais avec $pp(i) = 100100100100$. Nous remarquons que la distance de **Hamming** entre $pp(i)$ et $p(i)$ est $d_H(pp(i), p(i)) = 2$.

La figure 4.21.b illustre une comparaison entre le MID à 8 états et entrelaceur de taille 128×128 avec l'algorithme MAP (donc $p(i) = 100100110010$) et le même MID mais avec $pp(i) = 100100100100$. Nous remarquons que $d_H(pp(i), p(i)) = 3$, qui est la plus grande distance entre $pp(i)$ et toutes les fonctions $p(i)$ trouvées avec l'algorithme MAP dans le tableau 4.2.

D'après les figures 4.20.b et 4.21.b nous remarquons que la fonction périodique $pp(i)$ pour $Nr = 12$ itérations introduit une légère dégradation des performances par rapport à la fonction $p(i)$ sur toute la gamme du SNR et ceci pour un MID à 4 états ou à 8 états. Si nous prenons $BER = 10^{-5}$ comme référence, la dégradation dans la figure 4.20.b est presque de 0.075 dB et elle est presque de 0.095 dB dans la figure 4.21.b. donc, plus la distance de **Hamming** entre $pp(i)$ et $p(i)$ augmente, plus la dégradation du MID augmente. Mais malgré cette dégradation le MID avec $pp(i)$ reste toujours performant que le CID d'au moins 0.25 dB . Alors, nous pouvons conclure que la fonction porte périodique $pp(i) = 100100100100$ pour $Nr = 12$ fait une bonne solution pratique, pour améliorer un CID qui utilisent l'algorithme MAP comme algorithme de décodage, avec un gain estimé de 0.25 dB en termes de SNR .

Par conséquent, cette nouvelle technique qui est testée pour les deux algorithmes de décodage SOVA et MAP pour un bruit AWGN, offre au décodeur itératif modifié avec l'algorithme MAP de très meilleures performances en termes de pouvoir de correction des erreurs. En plus, la fonction porte périodique $pp(i) = 100100 \dots$ donne une solution pratique très acceptable pour rendre ce décodeur plus performant.

En outre, cette nouvelle technique réduit le nombre des additions dans le processus de décodage itératif, par conséquent elle réduit le temps de traitement du signal reçu.

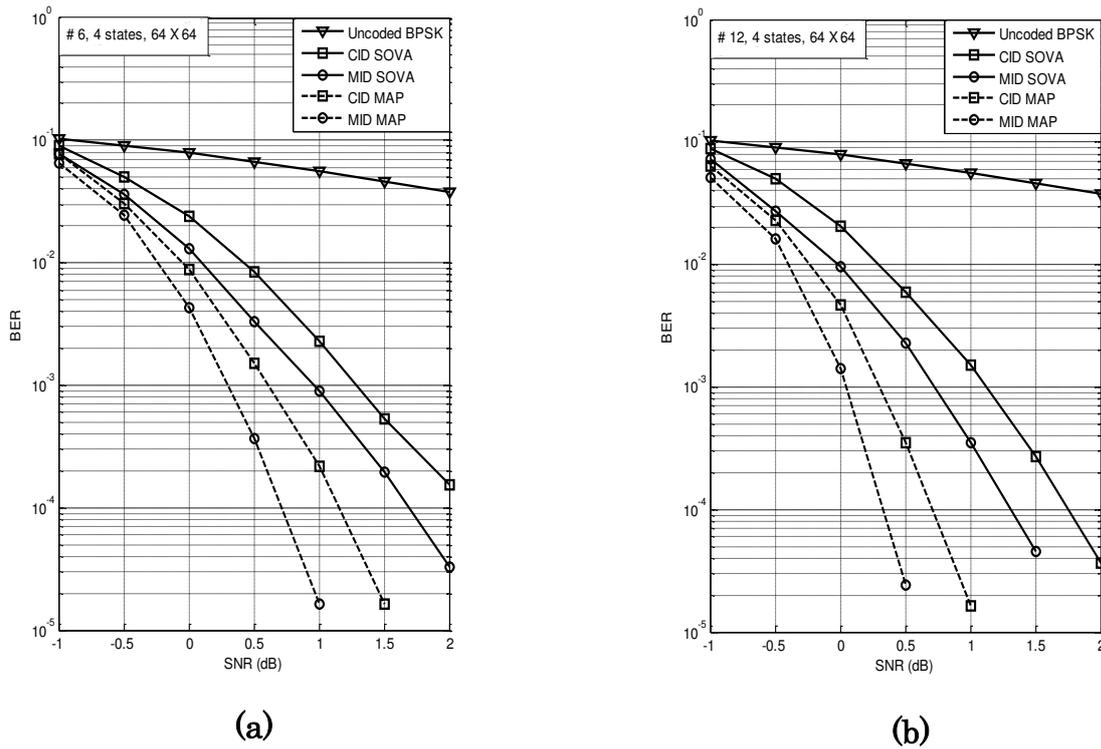


Fig. 4.18 Le *BER* en fonction du *SNR* des MID et CID à 4 états et entrelaceur 15 –aléatoire de taille 64×64 pour les deux algorithmes SOVA et MAP : (a) 6 itérations (b) 12 itérations.

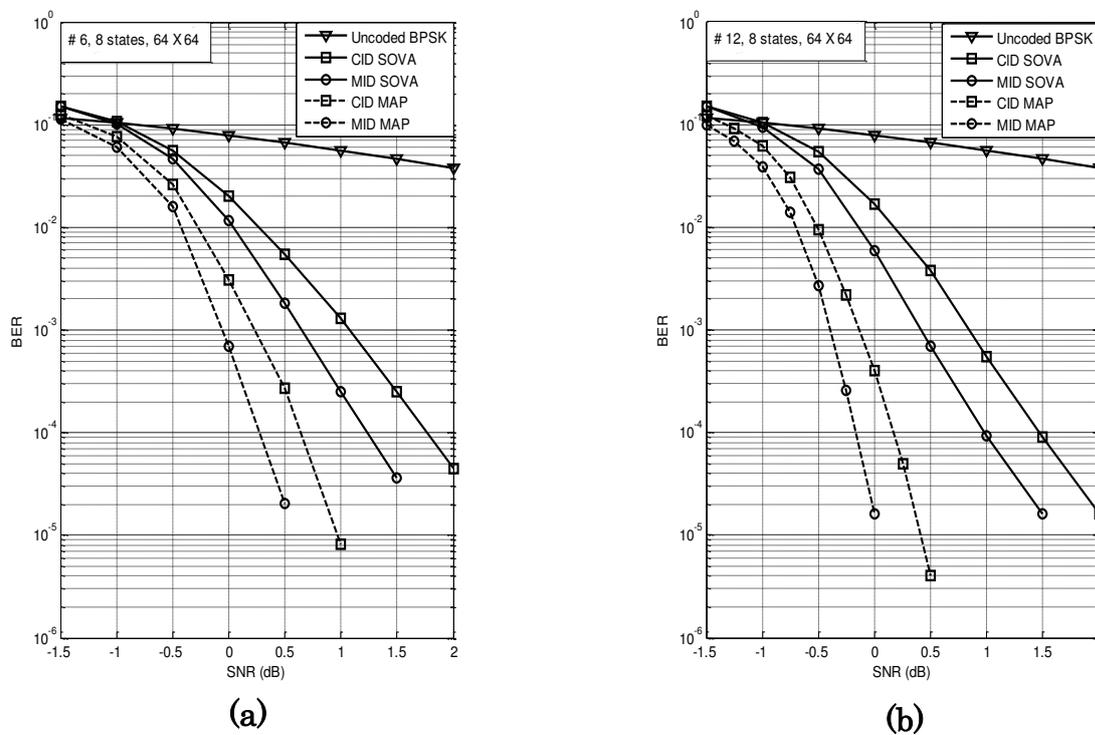


Fig. 4.19 Le *BER* en fonction du *SNR* des MID et CID à 8 états et entrelaceur 15 –aléatoire de taille 64×64 pour les deux algorithmes SOVA et MAP : (a) 6 itérations (b) 12 itérations.

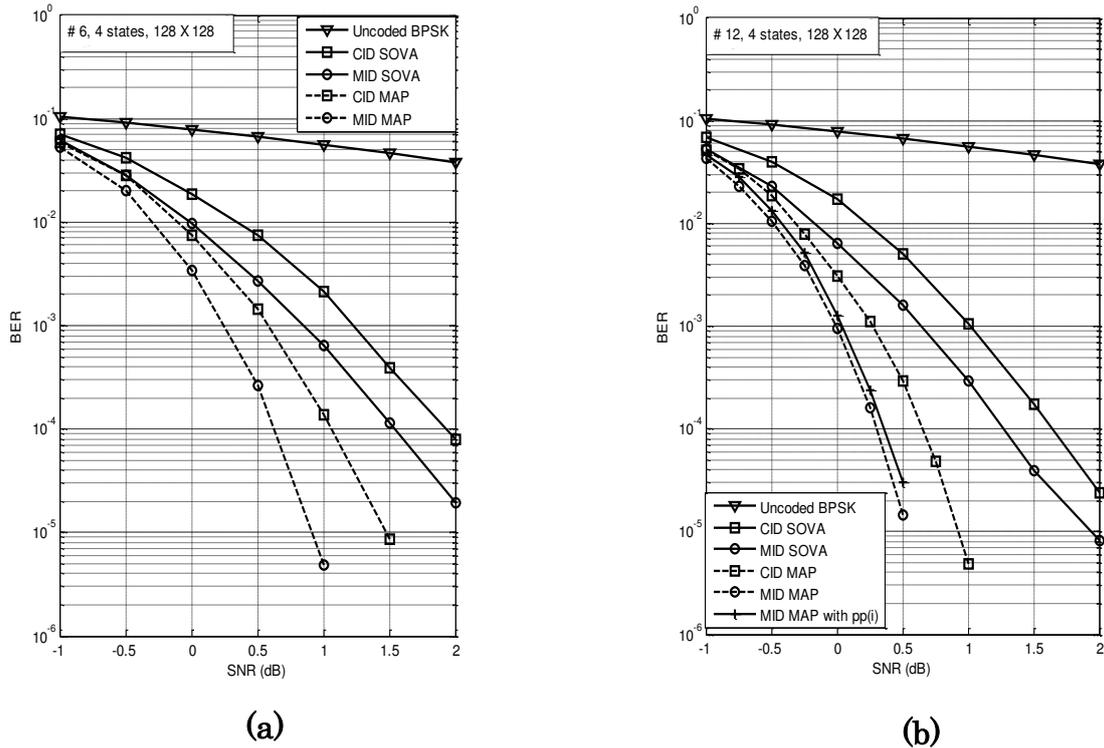


Fig. 4.20 Le *BER* en fonction du *SNR* des MID et CID à 4 états et entrelaceur 15 –aléatoire de taille 128×128 pour les deux algorithmes SOVA et MAP : (a) 6 itérations (b) 12 itérations.

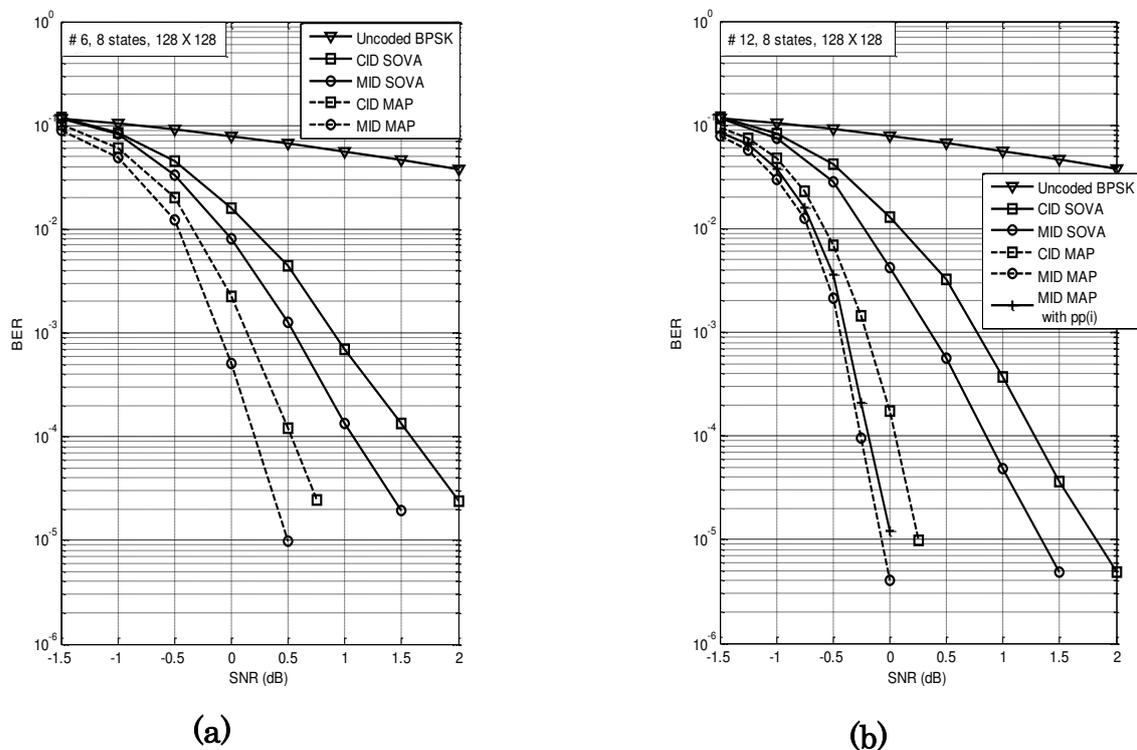


Fig. 4.21 Le *BER* en fonction du *SNR* des MID et CID à 8 états et entrelaceur 15 –aléatoire de taille 128×128 pour les deux algorithmes SOVA et MAP : (a) 6 itérations (b) 12 itérations.

En conclusion, nous avons vu lors de la présentation de cette approche que l'algorithme MAP est toujours meilleur que l'algorithme SOVA pour un bruit AWGN. D'ailleurs, l'algorithme MAP est considéré comme étant un algorithme optimal contrairement à l'algorithme SOVA qui est sous optimal [1][79][119].

Cette comparaison reste vraie même pour un bruit impulsif [120]. Pour ce motif, dans le reste de notre étude, le SOVA n'est pas l'algorithme adéquat à utiliser pour montrer l'efficacité de nos deux nouvelles techniques qui seront présentées dans la suite. En revanche, l'algorithme MAP fait parfaitement l'affaire.

Afin d'éviter la dégradation éventuelle des performances du décodeur itératif avec MAP, même si le bruit s'éloigne du modèle de bruit gaussien, il est nécessaire de concevoir un algorithme de décodage qui permet à ce décodeur de prendre en compte les valeurs des paramètres du bruit, parce que des telles valeurs permettent au décodeur MAP de calculer les métriques de branche afin d'estimer le message binaire envoyé. Dans le cas d'un bruit AWGN l'estimation des paramètres est simple, il suffit d'utiliser l'équation (4.8). Par contre, dans le cas de bruits impulsifs GG et $S\alpha S$, l'estimation des paramètres est fortement problématique [121][73].

Dans le cas d'un bruit impulsif GG, nous proposons dans le paragraphe 4.3.3 une nouvelle méthode d'estimation du paramètre de forme, et pour un bruit impulsif $S\alpha S$ de **Cauchy** une technique originale est présentée dans le paragraphe 4.3.4.

4.3.3 Approche 2 : Nouvelle approche semi-aveugle pour le turbo décodage dans les canaux à bruit impulsif GG [122]

L'estimation du paramètre de forme d'un bruit GG fait l'objet de plusieurs travaux de recherches. Par exemple, **X. Li** et al. [123] ils ont utilisé la méthode proposée par **Mallat** [124], pour estimer ce paramètre, qui est une méthode des moments. En sus de la méthode de **Mallat** utilisée par [123], d'autres méthodes d'estimation du paramètre de forme d'une GGD et leurs principales caractéristiques sont résumées et comparées entre elles théoriquement et pratiquement dans [125] et [126]. Aussi une nouvelle procédure pratique est bien présentée dans [127]. Tous ces travaux prennent en compte que les estimations proposées sont faites à partir des échantillons du bruit GG pré-connus, ce qui n'est pas le cas au niveau d'un turbo décodeur où l'estimation des paramètres du bruit impulsif GG doit être effectuée à partir des échantillons du signal reçu qui est une combinaison du bruit avec le signal envoyé.

Généralement, les statistiques du bruit impulsif qui affecte le canal ne sont pas constantes avec le temps, donc le récepteur qui travaille avec l'estimation de ses paramètres devient compliquée [128]. Pour cette raison jusqu'à présent, peu de recherches sur les effets des bruits impulsifs sur les décodeurs itératifs ont été faites. L'effet de la disparité du SNR sur les performances des tels décodeurs a été étudié par **Summers** et **Wilson** [129], qui ont proposé un nouvel schéma en ligne simple qui estime le SNR inconnu de chaque bloc de code, avant le décodage. Ils ont montré que ce schéma est suffisamment adéquat en termes de précision car il ne dégrade pas notablement les performances du décodage. La procédure d'estimation de la distribution du bruit basée sur la quantification du signal reçu démontrée par **Xiaoling** et **Nam** [130] a donnée de bons résultats pour un bruit GG.

Majoul et al. [121] ont étudié la sensibilité des décodeurs itératifs avec MAP à la disparité des paramètres du bruit impulsif GG. Ils ont montré que plus la valeur estimée du paramètre de forme s'éloigne de la valeur exacte plus que les performances du décodeur itératif se dégradent. Pour cette raison, ils ont présenté une nouvelle approche semi-aveugle pour les canaux à bruit impulsif GG. Cette approche, qui est basée sur des statistiques d'ordre supérieur HOS, permet de définir une meilleure estimation des paramètres du bruit qui sont directement utilisés dans le processus du décodage MAP, à savoir dans le calcul des métriques de branche.

La principale contribution dans ce paragraphe est de définir et d'expliquer une méthode originale pour estimer correctement le paramètre de forme du bruit impulsif GG, afin d'améliorer les performances du décodeur itératif avec MAP. Cette nouvelle méthode est qualifiée de semi-aveugle car elle fait l'estimation du paramètre de forme non à partir d'échantillons du bruit mais à partir du d'échantillons du signal reçu, qui est la superposition du bruit et le signal envoyé.

Contrairement à l'approche proposée dans [121] qui est basée sur de HOS, notre approche est basée sur de FLOS, et sur la probabilité ρ donnée par (4.19). Les résultats obtenus montrent que l'approche FLOS proposée est meilleure que l'approche HOS utilisée dans [121], surtout lorsque le bruit est très impulsif, donc elle offre une amélioration considérable des performances du turbo décodeur avec MAP.

4.3.3.1 L'effet de la disparité des paramètres du bruit sur les performances du décodeur

Il est connu que les performances d'un décodeur itératif avec l'algorithme MAP dépendent étroitement des LLR calculés par cet algorithme. Comme nous avons vu dans

le chapitre 3, que le calcul des LLR dans l'algorithme MAP est basé principalement sur le calcul des valeurs de la métrique de branche, ipso facto, les performances du turbo décodeur dépendent de l'exactitude de ces valeurs. Nous avons montré dans le chapitre 3 que le calcul de la métrique de branche introduit les paramètres du modèle de bruit envisagé. Par conséquent, si les caractéristiques du canal s'écartent du modèle considéré, alors l'erreur commise dans le calcul de la métrique de branche affecte les performances du turbo décodeur avec MAP.

Pour montrer ce que nous déjà dit, nous prenons le modèle du bruit GG à moyenne nulle où les paramètres influents sont celui de forme et la variance. Nous étudions le comportement du turbo décodeur à la disparité de ces deux paramètres selon le *BER*.

Cette étude est faite avec le turbo code de la figure 4.5.c, de rendement 1/3, de polynôme générateur (35,23) et avec un entrelaceur 15 –aléatoire de taille 64×64 . L'algorithme de décodage utilisé bien entendu est le MAP avec trois itérations (#3) seulement.

La figure 4.22 illustre la variation du *BER* avec des valeurs exactes pour l'écart type et des valeurs disparates du paramètre de forme autour de la valeur exacte $\alpha = 0.9$.

Nous montrons que, par exemple, à un $BER = 10^{-2}$, lorsque le décodeur fonctionne avec $\alpha = 1.5$, $\alpha = 2$ et $\alpha = 0.5$, nous avons respectivement des pertes de disparité en termes du *SNR* de 0.37 dB, 0.66 dB et 1.33 dB.

Dans la figure 4.23, nous traçons le *BER* pour une valeur exacte du paramètre de forme $\alpha = 0.9$ et pour des mauvaises valeurs de l'écart type σ .

Au $BER = 10^{-2}$, lorsque le décodeur remplace la valeur exacte σ par les valeurs erronées 2σ , 4σ et 6σ , nous observons respectivement des pertes en termes du *SNR* de 0.41 dB, 0.61 dB et 1.37 dB.

Les figures 4.22 et 4.23 montrent encore que si le *SNR* augmente, la sensibilité du décodeur itératif à la disparité des paramètres augmente aussi, et vice versa.

Cette étude montre effectivement que les performances du turbo décodeur dépendent de la précision des valeurs des paramètres du bruit considéré. Généralement, le bruit a de caractéristiques inconnues. Ainsi, une estimation de ces caractéristiques dans chaque bloc de signal reçu est nécessaire.

Pour un bruit GG il est facile d'en déduire la variance du bruit, il suffit d'utiliser la relation (4.8) comme dans le cas d'un bruit AWGN. Le problème est de trouver une manière efficace pour estimer le paramètre de forme.

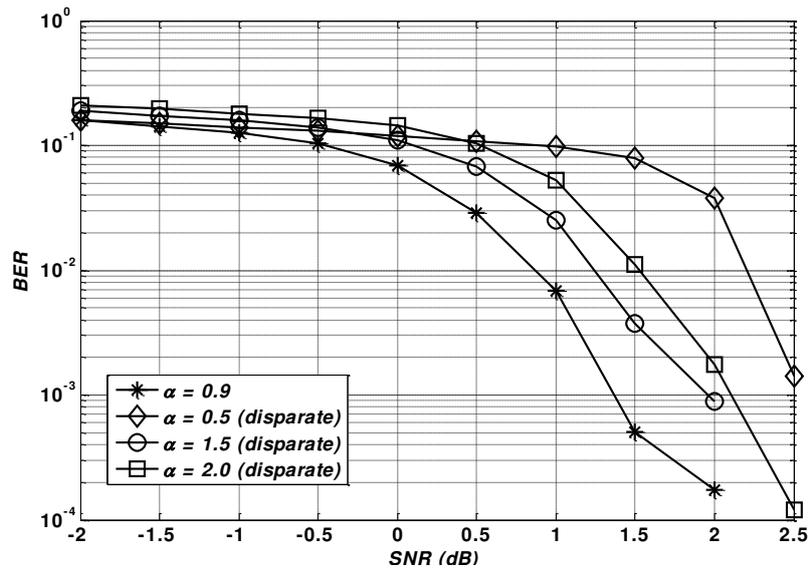


Fig. 4.22 L'effet de la disparité du paramètre de forme sur les performances du décodeur itératif. Les valeurs disparates de la valeur exacte $\alpha = 0.9$ sont $\alpha = 0.5$, $\alpha = 1.5$ et $\alpha = 2$. La valeur de σ est exacte.

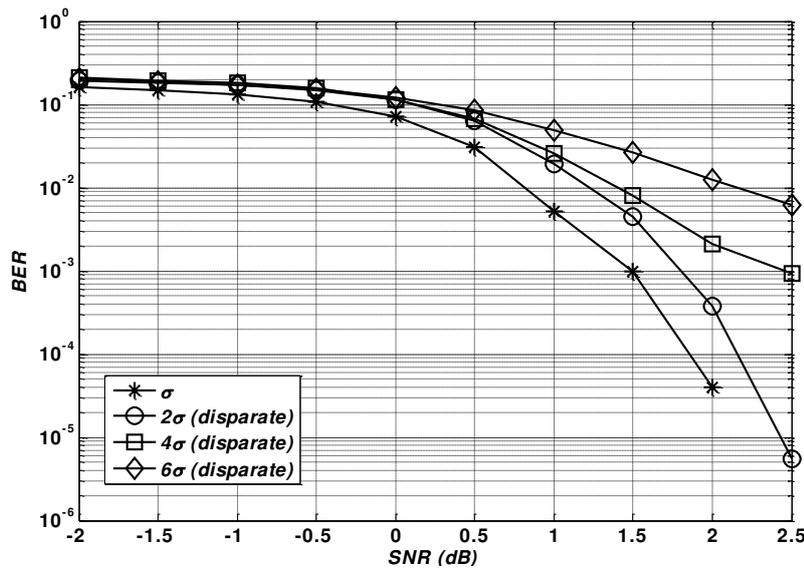


Fig. 4.23 L'effet de la disparité de l'écart type sur les performances du décodeur itératif. Les valeurs disparates de la valeur exacte σ sont 2σ , 4σ et 6σ . La valeur exacte du paramètre de forme est $\alpha = 0.9$.

4.3.3.2 Estimation semi-aveugle du paramètre de forme utilisant HOS [121]

L'estimation du paramètre de forme en utilisant les HOS est proposée dans [121]. Nous calculons à partir du signal reçu R le kurtosis \mathcal{K}_W du bruit impulsif W , qui est nécessaire pour estimer le paramètre de forme α .

Nous avons

$$E(R^2) = E((Y + W)^2) = E(Y^2) + E(W^2) + 2 \cdot E(Y) \cdot E(W) = 1 + E(W^2)$$

car $E(Y^2) = 1$ et $E(Y) = E(W) = 0$ ($Y \in \{-1, +1\}$),
 et $E(R^4) = E((Y + W)^4)$
 $= E(Y^4) + 4 \cdot E(Y^3) \cdot E(W) + 6 \cdot E(Y^2) \cdot E(W^2) + 4 \cdot E(Y) \cdot E(W^3) + E(W^4)$

puisque $E(Y^4) = 1$ et $E(W^2) = E(R^2) - 1$. Alors

$$\begin{cases} E(W^2) = E(R^2) - 1 \\ E(W^4) = E(R^4) - 1 - 6 \cdot (E(R^2) - 1) \end{cases} \quad (4.32)$$

Donc le kurtosis \mathcal{K}_W du bruit impulsif W s'écrit

$$K_W = \frac{E(W^4)}{(E(W^2))^2} = \frac{E(R^4) - 1 - 6 \cdot (E(R^2) - 1)}{(E(R^2) - 1)^2} \quad (4.33)$$

L'identification de la dernière équation avec l'équation (2.6) qui donne l'expression du kurtosis, donne

$$\frac{\Gamma(1/\alpha)\Gamma(5/\alpha)}{[\Gamma(3/\alpha)]^2} = \frac{E(R^4) - 1 - 6 \cdot (E(R^2) - 1)}{(E(R^2) - 1)^2} \quad (4.34)$$

Avec les approximations suivantes (voir [131] et [132])

$$\begin{cases} E(R^n) \cong \frac{1}{N} \sum_{k=1}^N R^n \quad n = 1, 2, \dots \\ \Gamma(x) \cong \sqrt{2\pi} \cdot x^{x-1/2} \cdot e^{-x} \text{ si } x \text{ est grande} \end{cases} \quad (4.35)$$

nous trouvons la valeur estimée $\hat{\alpha}$ du paramètre de forme α [121]

$$\alpha \cong \hat{\alpha} = \frac{\ln(5^5/3^6)}{\ln \left[\frac{\sqrt{5}}{3} \left(\frac{\frac{1}{N} \sum_{k=1}^N R^4 - 1 - 6 \cdot \left(\frac{1}{N} \sum_{k=1}^N R^2 - 1 \right)}{\left(\frac{1}{N} \sum_{k=1}^N R^2 - 1 \right)^2} - \frac{2}{9} \right) \right]} \quad (4.36)$$

4.3.3.3 Estimation semi-aveugle du paramètre de forme utilisant FLOS

Dans la section B de ce paragraphe nous proposons le principe de notre nouvelle technique pour estimer directement le paramètre de forme d'un bruit impulsif GG à partir du signal reçu. Nous allons montrer que cette nouvelle méthode est très efficace en termes de précision.

4.3.3.3.1 Méthodes d'estimation du paramètre de forme d'une GGD

De nombreuses études traitent du problème de l'estimation du paramètre de forme d'une GGD. Par exemple, la procédure présentée par **Sharifi** et al. [133] basé sur le calcul des moments utilisant l'approximation de l'inverse de la fonction $M(\alpha)$ donnée par $M(\alpha) = (E(|W|))^2 / E(|W|^2)$ selon un tableau de consultation. Cette procédure nécessite de grandes tailles des échantillons pour assurer la convergence de l'estimateur. Une

méthode basée sur le principe du ML pour estimer α a été proposée par **Varanasi** et **Aazhang** [134]. Cette méthode a été cependant trouvée fastidieuse en termes de coût de calcul.

Regazzoni et al. [131] a présenté une simple et fiable approche à l'aide du HOS pour évaluer le paramètre de forme d'une GGD. Ils ont montré que cette méthode donne des résultats tout à fait acceptables dans une gamme du kurtosis \mathcal{K}_W dans le cas du bruit impulsif. Pour cette dernière raison, **Majoul** et al. [121] ont bien exploité l'approche de **Regazzoni** et al. [131] pour estimer le paramètre de forme à partir du signal reçu comme il est déjà démontré le paragraphe précédent 4.3.3.2.

Taiyue et al. [135] ont proposé une estimation approximative, du paramètre de forme α d'une GGD, sous la forme d'une fonction du kurtosis \mathcal{K}_W . Cette estimation est bonne si $\alpha > 1$, mais n'est plus le cas le bruit est impulsif ($\alpha \leq 1$).

Yunfei et al. [112] ont exposé, pour estimer le paramètre de forme α d'une GGD, une équation, obtenue à l'aide du principe du maximum de vraisemblance, qui contient une expression implicite de α . Pour trouver la valeur estimée de α il faut résoudre cette équation en utilisant une méthode numérique adéquate. L'inconvénient remarquable dans cette méthode est l'existence de la fonction digamma dans l'équation en question. Pour cette raison, **Yunfei** et al. [112] a considéré que α soit un nombre rationnel, ce qui facilite amplement les calculs au niveau de la fonction digamma.

Graciela et al. [127][136] ont proposé, pour estimer le paramètre de forme d'une GGD, un estimateur qui est une solution explicite approximative de l'estimateur transcendantal obtenue par la méthode des moments.

Maintenant, si nous voudrions estimer le paramètre de forme d'une GGD à partir du signal reçu, en utilisant directement une des méthodes déjà citées, il faut trouver un moyen qui permet de tirer complètement le bruit impulsif à partir du signal reçu, ce qui est impossible ! Pour cette raison **Majoul** et al. [121] ont exprimé le kurtosis du bruit en fonction des HOS du signal reçu, puis ils sont appliqué l'approximation proposé par [131].

Nous avons montré dans le paragraphe 4.3.1.1 que l'utilisation des HOS ne donne pas de résultats pertinents si le bruit GG est impulsif, c'est-à-dire $0 < \alpha < 1$, car les moments d'ordre 2 et 4 dans l'expression du kurtosis donné par (4.33), sont exprimés avec des erreurs relativement non négligeables en utilisant la formule (4.9).

Pour cette raison, et contrairement à ce qui présenté dans [121] où l'approche utilisée est basée sur des HOS, nous avons construit une nouvelle approche basée sur des FLOS et sur la probabilité ρ donnée par l'équation (4.19).

4.3.3.3.2 Approche proposée

Soit m_r le moment d'ordre r d'un bruit impulsif GG qui est donné par l'équation (4.15). Nous posons

$$k_W(r) = \frac{m_r}{\sigma^r} = \left[\frac{\Gamma(1/\alpha)}{\Gamma(3/\alpha)} \right]^{r/2} \frac{\Gamma((r+1)/\alpha)}{\Gamma(1/\alpha)} \quad (4.37)$$

Dans le cas d'un bruit impulsif (α est très petit), les quantités $1/\alpha$, $3/\alpha$ et $(r+1)/\alpha$ pour $r > 0$ devient grandes. Or, la fonction gamma $\Gamma(x)$ peut-être approximée, dans le cas où x est grande, par l'expression donnée par les approximations (4.35). Par conséquent, nous pouvons écrire

$$\begin{aligned} \Gamma(1/\alpha) &\cong \sqrt{2\pi\alpha} \cdot \alpha^{-1/\alpha} \cdot e^{-1/\alpha} \\ \Gamma(3/\alpha) &\cong 3^{3/\alpha} \sqrt{2\pi\alpha/3} \cdot \alpha^{-3/\alpha} \cdot e^{-3/\alpha} \\ \Gamma((r+1)/\alpha) &\cong (r+1)^{(r+1)/\alpha} \sqrt{2\pi\alpha/(r+1)} \cdot \alpha^{-(r+1)/\alpha} \cdot e^{-(r+1)/\alpha} \end{aligned}$$

Après la substitution des approximations suscitées dans l'équation (4.37) nous obtenons

$$k_W(r) \cong \left(\frac{(r+1)^{(r+1)}}{3^{(3/2)r}} \right)^{\frac{1}{\alpha}} \cdot \frac{3^{\frac{r}{4}}}{\sqrt{(r+1)}} \quad (4.38)$$

A partir de l'équation (4.38) nous tirons la valeur de α en fonction de r et $k_W(r)$

$$\alpha \cong \ln \left(\frac{(r+1)^{(r+1)}}{3^{(3/2)r}} \right) / \ln \left(k_W(r) \cdot \frac{\sqrt{(r+1)}}{3^{\frac{r}{4}}} \right) \quad (4.39)$$

Nous rappelons des formules (4.25)

$$\begin{aligned} m_r &\cong \rho \cdot b_r + (1 - \rho) \cdot c_r = \rho \cdot (b_r - c_r) + c_r \\ b_r &= E(|R_k - \text{sign}(R_k)|^r) \quad \text{et} \quad c_r = E(|R_k + \text{sign}(R_k)|^r) \end{aligned}$$

D'un côté, si nous posons $r = 2$ dans les expressions précédentes nous trouvons

$$\begin{aligned} m_2 &\cong \rho \cdot (b_2 - c_2) + c_2 \\ b_2 &= E(|R_k - \text{sign}(R_k)|^2) = E(R_k^2) - 2E(|R_k|) + 1 \\ c_2 &= E(|R_k + \text{sign}(R_k)|^2) = E(R_k^2) + 2E(|R_k|) + 1 \end{aligned}$$

c'est-à-dire

$$m_2 \cong -4E(|R_k|) \cdot \rho + E(R_k^2) + 2E(|R_k|) + 1$$

et d'un autre côté nous avons d'après (4.32)

$$m_2 = E(W_k^2) = \sigma^2 = E(R_k^2) - 1 \quad (4.40)$$

et par conséquent

$$\rho \cong \frac{1}{2} \cdot \left(1 + \frac{1}{E(|R_k|)} \right) \quad (4.41)$$

Les relations (4.25), (4.37), (4.40) et (4.41) nous donnent alors

$$k_W(r) = \frac{m_r}{\sigma^r} \cong \frac{\frac{1}{2} \cdot \left(1 + \frac{1}{E(|R_k|)} \right) \cdot E(|R_k - \text{sign}(R_k)|^r - |R_k + \text{sign}(R_k)|^r) + E(|R_k + \text{sign}(R_k)|^r)}{(E(R_k^2) - 1)^{r/2}}$$

Nous introduisons cette dernière relation dans (4.39), nous obtenons la nouvelle estimation semi-aveugle du paramètre de forme, en fonction des observations du signal reçu seulement, comme suit

$$\alpha \cong \frac{\ln \left(\frac{(r+1)^{r+1}}{3 \left(\frac{3}{2} \right)^r} \right)}{\ln \left(\frac{\frac{1}{2} \left(1 + \frac{1}{E(|R_k|)} \right) E(|R_k - \text{sign}(R_k)|^r - |R_k + \text{sign}(R_k)|^r) + E(|R_k + \text{sign}(R_k)|^r)}{(r+1)^{-1/2} \cdot \left(\sqrt{3} (E(R_k^2) - 1) \right)^{\frac{r}{2}}} \right)} \quad (4.42)$$

avec $E(\cdot)$ est donné par la formule (4.9).

Remarque 4.1 : A partir des observations R_k du signal reçu, nous pouvons estimer géométriquement le paramètre de forme α en calculant l'écart type σ à l'aide de (4.40) et la probabilité ρ à l'aide de (4.41) puis en utilisant l'abaque de la figure 4.14.

La relation (4.41) montre que $\rho > 1/2$, ce qui est conforme avec (4.19). Comme $\rho \leq 1$, alors d'après (4.21) nous trouvons

$$E(|R_k|) \geq 1 \quad (4.43)$$

Alors

$$E(R_k^2) \geq (E(|R_k|))^2 \geq 1 \quad (4.44)$$

Les relations (4.21) et (4.44) permettent de conclure que

- Si $E(R_k^2) \rightarrow 1$ alors $E(|R_k|)$ et ρ sont ainsi et le bruit est très impulsif et faible dans ce cas ;
- Si $E(|R_k|) \rightarrow 1$ et $E(R_k^2)$ est loin de 1 alors $\rho \rightarrow 1$ et le bruit est très impulsif et fort;
- Si $E(|R_k|)$ est loin de 1 alors $E(R_k^2)$ et ρ sont ainsi et le bruit est moins impulsif et fort.

4.3.3.3 Résultats et discussions

Dans ce paragraphe nous montrons l'efficacité de notre nouvelle approche FLOS sur l'estimation du paramètre de forme. Cette efficacité est bien extériorisée dans ce paragraphe à l'aide de deux comparaisons.

La première comparaison est entre cette approche et celle proposée dans [121], proprement dite approche HOS. Les performances des deux approches sont étudiées en termes de $RMSE$, qui est donnée par la relation (4.12), en fonction de la valeur exacte de α . Pour cette comparaison nous avons pris l'ordre de moment $r = 1/10$ et le nombre d'essais $N_e = 10\,000$. Le nombre des échantillons est pris égal à la taille d'un entrelaceur c'est-à-dire égal à une puissance de 2. Dans notre cas $N = 32 \times 32 = 1024$, $N = 64 \times 64 = 4096$ et $N = 128 \times 128 = 16384$. Pour les valeurs de l'écart type σ qui sont exprimées en fonction du SNR , nous avons pris trois valeurs : $SNR = -3\,dB$, $SNR = 0\,dB$ et $SNR = 3\,dB$. Pour calculer et tracer la $RMSE$ en fonction de α , nous avons adopté le modèle simulation illustré dans la figure 4.24. Les figures 4.25, 4.26 et 4.27 montrent les résultats de cette comparaison.

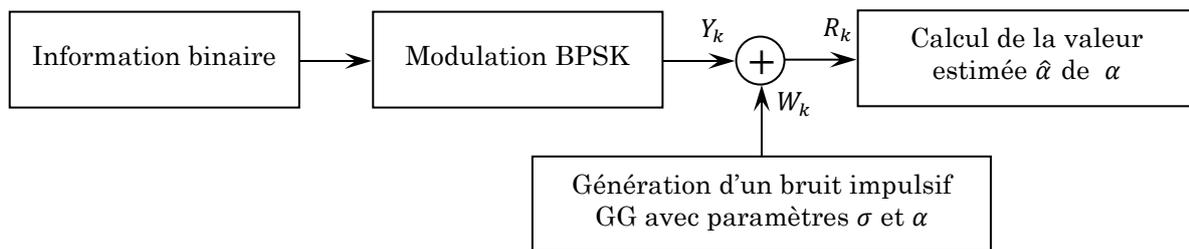


Fig. 4.24 Le modèle de simulation adopté pour calculer la $RMSE$ en fonction de α .

La deuxième comparaison est entre les trois comportements du décodeur itératif

- Il fonctionne avec les valeurs exactes des paramètres du bruit ;
- Il tient compte des valeurs estimées par notre approche ;
- Il considère le bruit comme étant AWGN.

Le comportement du décodeur itératif dans les trois cas est envisagé au prorata du BER . Pour illustrer tangiblement cette comparaison nous avons considéré le turbo code de la figure 4.5.c, de rendement $1/3$, de polynôme générateur $(35,23)$ et avec un entrelaceur 15-aléatoire de taille 64×64 . Pour le décodage nous se servir de l'algorithme MAP avec uniquement trois itérations (#3). Les valeurs traitées du paramètre de forme sont $\alpha = 0.7$ et $\alpha = 0.3$. La figure 4.28 extériorise la variation des BER du turbo décodeur en fonction du SNR pour les trois cas sus-cités et ceci avec $\alpha = 0.7$ et $\alpha = 0.3$.

L'analyse des figures 4.25, 4.26 et 4.27 permet de conclure que

- La *RMSE* diminue avec l'augmentation du nombre d'échantillons N pour les deux approches, ce qui confirme la théorie ;
- La *RMSE* de notre approche augmente d'une façon légère en fonction du paramètre de forme, et ceci revient principalement aux approximations utilisées dans cette approche qui résident dans les formules approximatives (4.25) et (4.41). Plus le bruit est impulsif ($\alpha \rightarrow 0$) plus ces approximations sont meilleures ;
- Dans le cas où le nombre d'échantillons est relativement petit ($N = 1024$) l'approche HOS donne de mauvais résultats si le bruit est faible ($SNR = 3 \text{ dB}$) ou il est très impulsif ($\alpha \leq 0.5$) (voir la figure 4.25) et ceci revient à l'erreur commise dans le calcul des moments HOS dans l'expression du kurtosis à l'aide de la formule (4.9). Par contre, dans les mêmes conditions, notre approche reste fiable ;
- Nous remarquons que la *RMSE* de notre approche FLOS est considérablement inférieure à celle de l'approche HOS et ceci quel que soit le nombre des échantillons et le *SNR*, ce qui prouve l'avantage de l'approche FLOS. L'efficacité de notre approche par rapport à l'approche HOS devient très remarquable dans le cas où le bruit est très impulsif ($\alpha \leq 0.5$).

Pour montrer numériquement l'avantage de notre approche FLOS par rapport à celle présentée dans [121], nous prenons par exemple le cas où le $SNR = 0 \text{ dB}$, le nombre d'échantillons $N = 1024, 4096$ et 16384 , et ceci pour les trois valeurs exactes du paramètre de forme 0.2, 0.5 et 0.8 (voir les figures 4.25.b, 4.26.b et 4.27.b). Nous calculons pour chaque cas la *RMSE* de l'approche HOS notée $RMSE_{HOS}$ et celle de notre approche FLOS notée $RMSE_{FLOS}$, puis nous faisons le rapport $RMSE_{HOS}/RMSE_{FLOS}$. Ce rapport permet de faire la comparaison entre les deux approches. Les résultats obtenus sont arrangés dans le tableau 4.4. Par exemple, pour $N = 1024$ et $\alpha = 0.5$ (voir la figure 4.25.b) nous trouvons $RMSE_{HOS} = 1.62 \times 10^{-1}$ et $RMSE_{FLOS} = 4.0 \times 10^{-2}$, donc le rapport entre les deux est $RMSE_{HOS}/RMSE_{FLOS} = 4.05$, c'est-à-dire la *RMSE* de l'approche HOS est quatre fois supérieure à celle de l'approche FLOS. Le tableau 4.4 illustre l'efficacité de notre approche FLOS par rapport à l'approche HOS surtout lorsque le bruit est très impulsif ($\alpha \rightarrow 0$). D'après ce tableau, la *RMSE* de l'approche FLOS est en moyenne presque 5 fois inférieure à celle de l'approche HOS.

	$\alpha = 0.2$	$\alpha = 0.5$	$\alpha = 0.8$
$N = 1024$	7.66	4.05	2.98
$N = 4096$	7.91	3.59	2.53
$N = 16384$	8.55	3.57	2.22

Tab. 4.4 Le rapport $RMSE_{HOS}/RMSE_{FLOS}$.

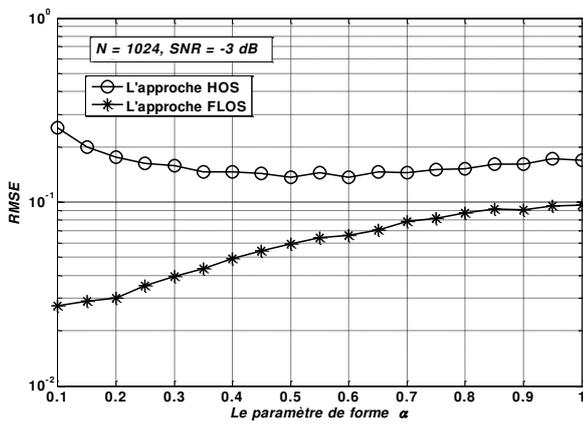
A partir de la figure 4.28 nous constatons que lorsque le canal est soumis à un bruit impulsif GG, et que le décodeur itératif utilise l'algorithme MAP classique, c'est-à-dire il suppose que le bruit est AWGN, il y a une dégradation importante des performances. A un BER de 10^{-3} et en comparant avec le cas où le turbo décodeur utilise la valeur exacte du paramètre de forme du bruit impulsif GG, nous observons à partir de la figure 4.28.a où $\alpha = 0.7$, une perte de gain en termes du SNR de 1.43 dB , et sur la figure 4.28.b où $\alpha = 0.3$, une perte de 3.16 dB . Comme prévu, cette dégradation est plus importante au fur et à mesure que le bruit envisagé écarte du modèle AWGN. Pour cette raison, dans le but d'avoir de meilleures performances, l'algorithme MAP exige les valeurs des paramètres du modèle de canal.

La nouvelle approche d'estimation semi-aveugle proposée offre des performances très proches des celles où les paramètres du bruit sont connus exactement. Pour un bruit impulsif GG de paramètre de forme $\alpha = 0.7$, nous remarquons quant au SNR une perte de gain d'environ 0.2 dB pour un $BER = 10^{-3}$. Si le bruit est très impulsif où $\alpha = 0.3$, nous observons que le turbo décodeur avec l'approche proposée présente des performances trop proches aux celles du même décodeur avec les paramètres exactes du bruit.

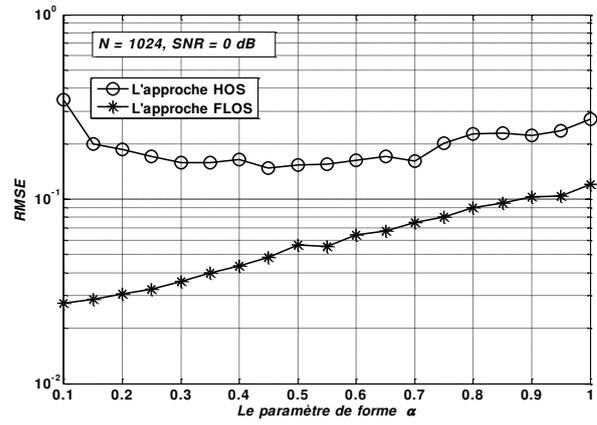
En conclusion, la nouvelle approche permet d'estimer le paramètre de forme directement en un seul coup à partir du signal reçu uniquement. Pour ce prétexte, cette approche est qualifiée de semi-aveugle. Nous avons montré que l'approche FLOS est plus précise que l'approche HOS proposée dans [121], et surtout lorsque le bruit est très impulsif. Numériquement l'approche proposée donne une $RMSE$ en moyenne presque cinq fois inférieure à celle de l'approche HOS.

En termes de BER , si le turbo décodeur qui utilise l'approche proposée donne des performances trop proches de celles obtenues s'il utilise les valeurs exactes des paramètres du bruit impulsif GG.

Dans le paragraphe suivant nous étendons le principe de cette approche au bruit $S\alpha S$ de **Cauchy** où $\alpha = 1$.

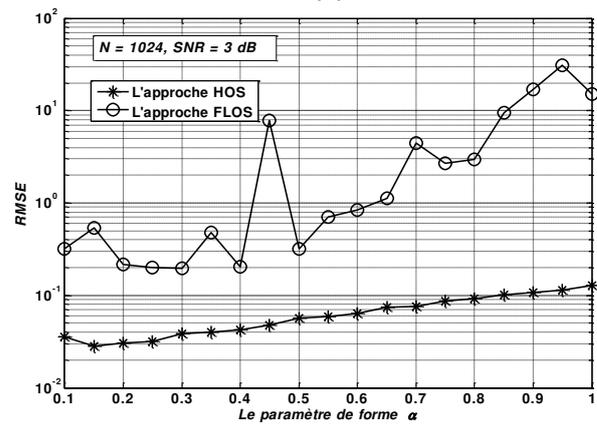


(a)

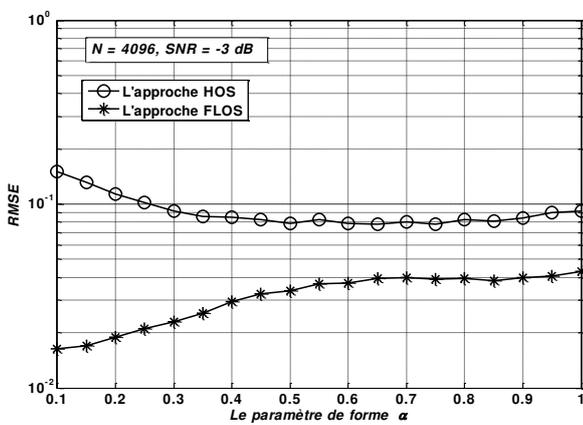


(b)

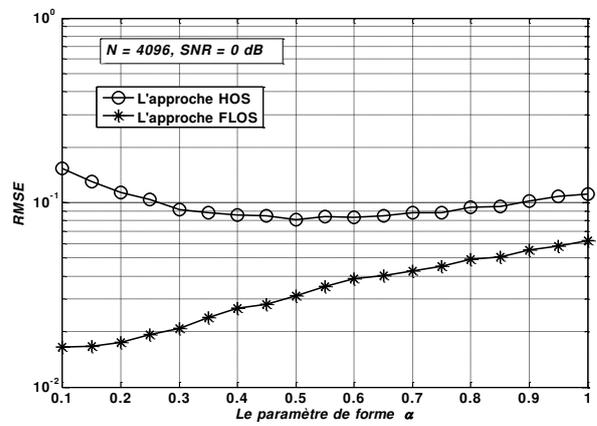
Fig. 4.25 La RMSE des approches HOS et FLOS en fonction du paramètre de forme pour $N = 1024$. (a) $SNR = -3 \text{ dB}$ (b) $SNR = 0 \text{ dB}$ (c) $SNR = 3 \text{ dB}$.



(c)

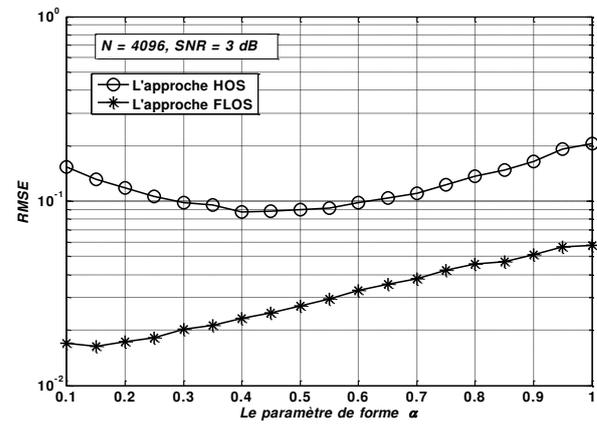


(a)

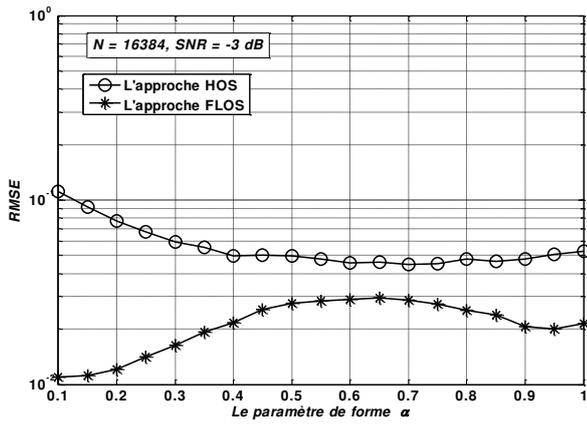


(b)

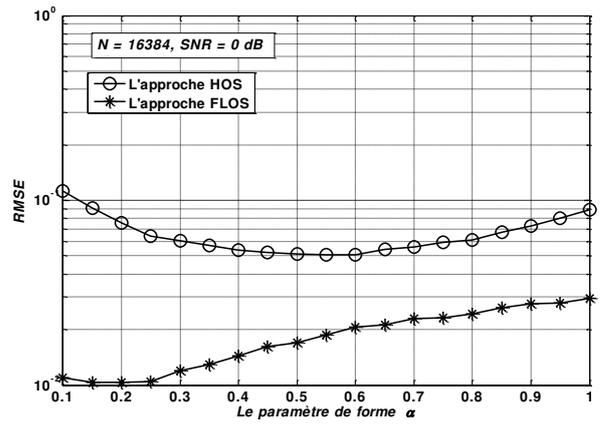
Fig. 4.26 La RMSE des approches HOS et FLOS en fonction du paramètre de forme pour $N = 4096$. (a) $SNR = -3 \text{ dB}$ (b) $SNR = 0 \text{ dB}$ (c) $SNR = 3 \text{ dB}$.



(c)

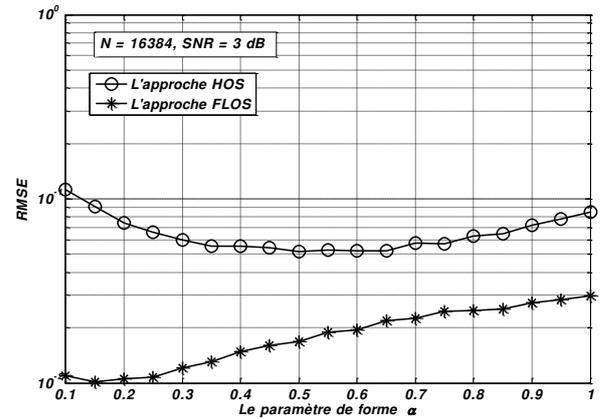


(a)

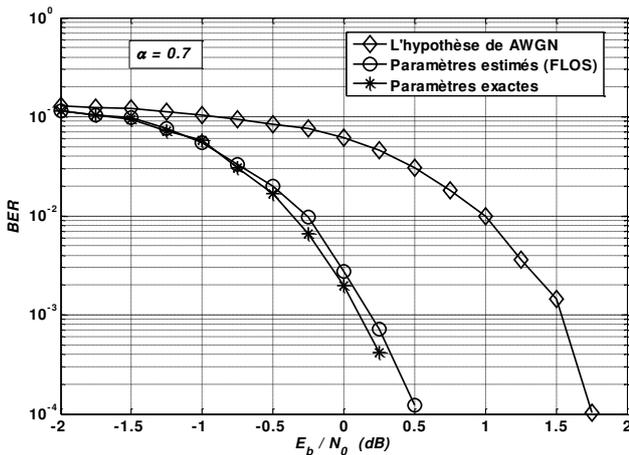


(b)

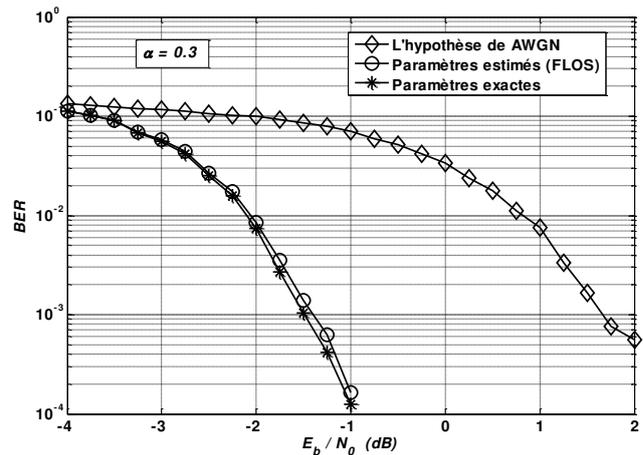
Fig. 4.27 La RMSE des approches HOS et FLOS en fonction du paramètre de forme pour $N = 16384$. (a) $SNR = -3 \text{ dB}$ (b) $SNR = 0 \text{ dB}$ (c) $SNR = 3 \text{ dB}$.



(c)



(a)



(b)

Fig. 4.28 Les performances du décodeur itératif pour un bruit impulsif GG dans les trois contextes : (\diamond) L'hypothèse de AWGN (\circ) Paramètres estimés avec l'approche FLOS ($*$) Paramètres exactes. Deux bruits impulsifs GG sont considérés : (a) $\alpha = 0.7$ (b) $\alpha = 0.3$.

4.3.4 Approche 3 : Nouvelle approche semi-aveugle pour le turbo décodage dans les canaux à bruit impulsif $S\alpha S$ de Cauchy

Différentes méthodes ont été proposées dans la littérature pour estimer les paramètres α , β , γ et δ d'une variable aléatoire α -stable. **Fama** et **Roll** [137] ont proposé une méthode basée sur les quantiles pour estimer le paramètre de forme α et le paramètre de dispersion γ d'une variable aléatoire $S\alpha S$. Cependant, cette méthode est applicable uniquement pour $\alpha > 1$. Cette méthode a été modifiée par **McCulloch** [138] pour inclure même les variables aléatoires à distribution non symétrique avec α dans l'intervalle $[0.6, 2.0]$. **Koutrouvelis** [139] a proposé une méthode basée sur la fonction caractéristique qui implique une procédure de régression itérative pour estimer les paramètres d'une variable aléatoire α -stable. **Kogon** et **Williams** [140] ont amélioré cette méthode en éliminant la procédure itérative et en simplifiant la régression. **Ma** et **Nikias** [141] et **Tsihrintzis** et **Nikias** [142] ont proposé l'utilisation de statistiques d'ordre inférieur fractionnel FLOS pour estimer les paramètres d'une variable aléatoire α -stable symétrique. **Bates** et **McLaughlin** [143] ont étudié les performances des méthodes proposées par **McCulloch** [138], **Kogon** et **Williams** [140], **Ma** et **Nikias** [141], et **Tsihrintzis** et **Nikias** [142] en utilisant deux ensembles de données réelles. Ils ont constaté qu'il existe de différences marquées entre les résultats obtenus à l'aide de différentes méthodes.

Cependant, la plupart de ces travaux considèrent que le cas particulier des variables aléatoires α -stables symétriques. Or la plupart des signaux de la vie réelle sont biaisés. En plus, toutes les techniques existantes soit elles exigent trop de calcul, soit leurs estimations possèdent des variances élevées. **Dance** et **Kuruoğlu** [70] ont résolu analytiquement le problème général de l'estimation des paramètres d'une variable aléatoire α -stable. Ils ont présenté trois nouvelles classes d'estimateurs des quatre paramètres α , β , γ et δ . Ces nouvelles classes d'estimateurs sont basées sur les FLOS positives et négatives d'une distribution α -stable asymétrique. Ce sont des généralisations des méthodes précédemment citées pour une variable aléatoire α -stable symétrique.

Dans [144] les paramètres d'une variable aléatoire α -stable sont estimés en utilisant une procédure d'optimisation en minimisant la somme des carrés de la différence entre la CDF observée ou empirique et la CDF théorique de la distribution α -stable.

Une étude comparative récente en termes d'erreur entre quelques méthodes d'estimation déjà citées, a été présentée par **Wang** et al. [145].

Pour un DCS avec décodeur itératif et qui est contaminé par un bruit impulsif de type *SaS* de **Cauchy** par exemple, si nous essayons d'appliquer une méthode d'estimation parmi les méthodes sus-citées pour estimer le paramètre de dispersion γ du bruit, il faut à partir du signal reçu, séparer le bruit du signal transmis, parce que toutes ces méthodes d'estimation sont applicables pour un nombre d'échantillons du bruit qui représente la variable aléatoire *SaS*. Néanmoins, c'est irréalisable de séparer deux signaux aléatoires superposés.

Malheureusement, jusqu'à présent, aucune méthode n'est présentée dans la littérature qui permet d'estimer les paramètres d'un bruit *SaS* à partir d'un signal reçu qui représente la superposition de ce bruit et un signal transmis, comme dans le cas d'un DCS avec turbo code. Pour ce motif, nous présentons dans ce paragraphe une approche originale qui permet de surmonter ce problème pour un bruit *SaS* de **Cauchy**.

4.3.4.1 L'effet de la disparité des paramètres du bruit sur les performances du décodeur

Nous avons vu dans le paragraphe 4.3.3.1 l'effet de la disparité des paramètres du bruit GG sur les performances du turbo décodeur.

Dans le cas d'un bruit *SaS* de **Cauchy**, nous devons trouver les mêmes conclusions, car les performances d'un décodeur itératif dépendent des LLR calculés par l'algorithme MAP, puisque le calcul de ces LLR inclut les paramètres du modèle de bruit envisagé. Donc, une mauvaise estimation de ces paramètres affecte les performances du turbo décodeur.

Dans ce paragraphe, pour étudier la sensibilité du décodeur itératif à la disparité du paramètre de dispersion du bruit de **Cauchy**, nous pouvons utiliser un autre paramètre que le *BER*, à savoir, le rapport logarithmique de vraisemblance LLR conditionnel [73][7], appelé aussi le LLR du canal, défini par (voir l'équation (3.25))

$$LLR_c(R_k) = \ln \left[\frac{p(R_k | d_k = 1)}{p(R_k | d_k = 0)} \right] = \ln \left[\frac{(R_k - 1)^2 + \gamma^2}{(R_k + 1)^2 + \gamma^2} \right] \quad (4.45)$$

Le signe de $LLR_c(R_k)$ indique si le bit transmis d_k est plus vraisemblablement d'être 0 ou 1, et sa valeur donne une indication sur la qualité de la décision prise en basant sur son signe.

Nous avons vu dans le paragraphe 1.4.2 du chapitre 1 que la capacité du canal dépend étroitement du rapport $LLR_c(R_k)$. En plus, si nous méditons sur l'expression (3.36) du chapitre 3, nous constatons que le rapport $LLR_c(R_k)$ fait la partie primordiale

dans le calcul de la métrique de branche, donc sur le calcul des LLR de l'algorithme MAP. Ceci signifie que les performances du décodeur itératif dépendent étroitement du rapport $LLR_c(R_k)$. Pour cette raison, dans [73] et [7] les auteurs basent sur ce rapport pour étudier quelques performances du turbo décodeur.

Nous étudions maintenant la sensibilité de $LLR_c(R_k)$ à la disparité du paramètre de dispersion γ . Pour ce faire, il suffit d'étudier le taux de variation de $LLR_c(R_k)$ par rapport à γ . Autrement dit, nous étudions la valeur absolue de la dérivée partielle $\frac{\partial LLR_c(R_k)}{\partial \gamma}$ en fonction de γ ou $1/\gamma$. Nous avons à partir de (4.45)

$$\left| \frac{\partial LLR_c(R_k)}{\partial \gamma} \right| = \left| \frac{8R_k\gamma}{((R_k - 1)^2 + \gamma^2) \cdot ((R_k + 1)^2 + \gamma^2)} \right| \quad (4.46)$$

Le signal reçu $R_k = Y_k + W_k$ où $Y_k \in \{-1, +1\}$ le signal envoyé, $W_k = \gamma \cdot W_k^1$ et W_k^1 est un bruit de **Cauchy** de paramètre de dispersion égal 1, donc W_k est de paramètre de dispersion égal à γ . Dans ces conditions, la valeur absolue de la dérivée partielle $\left| \frac{\partial LLR_c(R_k)}{\partial \gamma} \right|$ est une variable aléatoire qui s'écrit sous la forme suivante

$$\left| \frac{\partial LLR_c(R_k)}{\partial \gamma} \right| = \left| \frac{8(Y_k + \gamma \cdot W_k^1)\gamma}{((Y_k - 1 + \gamma \cdot W_k^1)^2 + \gamma^2) \cdot ((Y_k + 1 + \gamma \cdot W_k^1)^2 + \gamma^2)} \right| \quad (4.47)$$

et pour étudier la sensibilité de $LLR_c(R_k)$ à la disparité de γ , nous devons prendre l'espérance mathématique de l'équation (4.47), qui est remplacée pratiquement par la moyenne arithmétique. Pour cela, nous utilisons la méthode de Monte Carlo avec un nombre d'essais $N_e = 10\,000$. La figure 4.29 montre le tracé de la sensibilité de $LLR_c(R_k)$ par rapport à γ en fonction de $1/\gamma$ en dB.

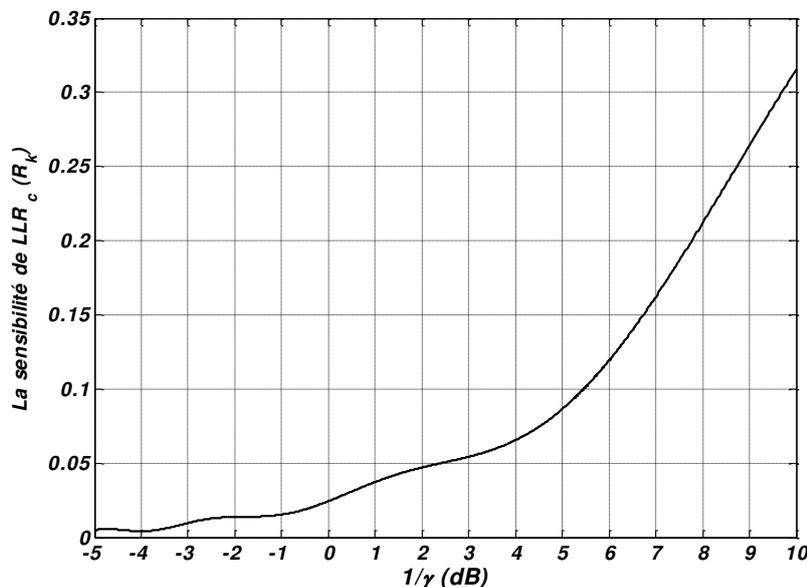


Fig. 4.29 La sensibilité de $LLR_c(R_k)$ à la disparité de paramètre γ .

La figure 4.29 nous confirme que dans la zone des faibles rapports $1/\gamma$, c'est-à-dire si γ est grand, la quantité $LLR_c(R_k)$ n'est pratiquement pas sensible à la disparité de γ . Par contre, dans la zone des forts rapports $1/\gamma$, ce qui correspond aux petites valeurs de γ , la grandeur $LLR_c(R_k)$ est sensible à la disparité de γ .

Par conséquent, comme dans le paragraphe 4.3.3.1, pour des faibles rapports $1/\gamma$, les performances du décodeur itératif ne subissent presque pas à des variations remarquables si le paramètre γ est mal estimé. En revanche, pour des forts rapports $1/\gamma$, ces performances souffrent une dégradation relativement grande pour une mauvaise estimation de γ .

Pour cette raison, dans le turbo décodage avec l'algorithme MAP et dans un environnement à bruit additif *SaS* de **Cauchy**, une méthode précise d'estimation du paramètre de dispersion γ est nécessaire.

4.3.4.2 Estimation semi-aveugle du paramètre de dispersion utilisant FLOS

Dans ce paragraphe nous présentons le principe de notre nouvelle approche pour estimer directement le paramètre de dispersion d'un bruit *SaS* de **Cauchy** à partir du signal reçu. Cette approche est basée sur les FLOS et sur la probabilité ρ donnée par l'équation (4.20). Nous allons montrer que cette méthode est très efficace en termes de précision.

Nous savons que les moments m_r d'un bruit *SaS* sont existes que si l'ordre r vérifie la condition $-1 < r < \alpha$. Par conséquent, tous les moments d'un bruit *SaS* de **Cauchy** où $\alpha = 1$ sont de type FLOS.

4.3.4.2.1 Approche proposée

Soit m_r le moment d'ordre r d'un bruit *SaS* de **Cauchy** qui est donné à partir de l'équation (4.16) par

$$m_r = E(|W|^r) = \frac{2^{r+1} \cdot \Gamma\left(\frac{r+1}{2}\right) \cdot \Gamma(-r)}{\sqrt{\pi} \cdot \Gamma\left(\frac{-r}{2}\right)} \cdot \gamma^r \quad (4.48)$$

En utilisant les deux propriétés suivantes de la fonction Γ

$$\Gamma(x) \cdot \Gamma(1-x) = \frac{\pi}{\sin(\pi x)} \text{ et } \Gamma(2x) = \frac{2^{2x-1}}{\sqrt{\pi}} \Gamma(x) \cdot \Gamma\left(x + \frac{1}{2}\right)$$

nous trouvons que le moment m_r s'écrit sous la forme simplifiée suivante

$$m_r = \frac{1}{\sin\left(\frac{\pi}{2}(r+1)\right)} \cdot \gamma^r \quad (4.49)$$

Nous posons $r = 1/n$ avec $n \geq 2$ ($n \in \mathbb{N}$). L'équation (4.49) devient

$$m_{1/n} = a_n \cdot \gamma^{1/n} \quad (4.50)$$

où
$$a_n = \frac{1}{\sin\left(\frac{\pi(n+1)}{2n}\right)} \quad (4.51)$$

et aussi les formules (4.25) deviennent

$$m_{1/n} \cong \rho \cdot b_n + (1 - \rho) \cdot c_n = \rho \cdot (b_n - c_n) + c_n \quad (4.52)$$

où
$$b_n = E\left(|R_k - \text{sign}(R_k)|^{\frac{1}{n}}\right) \quad \text{et} \quad c_n = E\left(|R_k + \text{sign}(R_k)|^{\frac{1}{n}}\right) \quad (4.53)$$

En remplaçant ρ par son expression (4.20) dans (4.52) nous trouvons

$$m_{1/n} \cong \frac{b_n - c_n}{\pi} \cdot \arctan\left(\frac{1}{\gamma}\right) + \frac{b_n + c_n}{2} \quad (4.54)$$

L'indentification de (4.54) et (4.50) nous permet d'écrire l'approximation

$$\varphi(\gamma) = \frac{b_n - c_n}{\pi} \cdot \arctan\left(\frac{1}{\gamma}\right) + \frac{b_n + c_n}{2} - a_n \gamma^{\frac{1}{n}} \cong 0 \quad (4.55)$$

Il faut noter que l'espérance mathématique $E(\cdot)$ dans l'équation (4.53), est considérée comme étant la moyenne arithmétique.

Mathématiquement nous avons $\lim_{n \rightarrow +\infty} (b_n - c_n) = 0^-$, et nous pouvons montrer par simulation que si $n \geq 10$ la quantité $(b_n - c_n)$ est toujours négative et relativement petite. Or la fonction $\frac{1}{\pi} \cdot \arctan\left(\frac{1}{\gamma}\right)$ est bornée entre $-1/2$ et $1/2$, alors la quantité $\frac{b_n - c_n}{\pi} \times \arctan\left(\frac{1}{\gamma}\right)$ devient très petite et $\varphi(\gamma) \cong \frac{b_n + c_n}{2} - a_n \gamma^{\frac{1}{n}}$. Donc, la dérivée de $\varphi(\gamma)$ est $\varphi'(\gamma) \cong \frac{-a_n}{n} \gamma^{\frac{1}{n}-1}$, ce qui signifie que $\varphi(\gamma)$ est 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 (4.55), qui représente la valeur estimée du paramètre de dispersion γ .

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

- Le calcul des moments de type FLOS b_n et c_n de l'équation (4.53) à partir des observations du signal reçu R_k ;
- La résolution numérique de l'équation (4.55) sachant qu'elle possède une solution unique $\hat{\gamma}$.

4.3.4.2 Résultats et discussions

Dans cette partie nous allons montrer l'avantage de notre nouvelle approche pour estimer le paramètre de dispersion γ d'un bruit W de type $S\alpha S$ de **Cauchy**. Pour ce faire, nous avons besoin d'une comparaison en termes de $RMSE$, c'est-à-dire de précision, et en termes de simplicité des calculs, avec une autre méthode qui fait l'estimation de ce paramètre à partir du signal reçu $R = Y + W$, comme dans le cas d'un bruit impulsif GG. Malheureusement, jusqu'à ce moment, nulle méthode n'est présentée dans la littérature sur ce contexte.

Pour surmonter ce problème de comparaison, nous avons un autre moyen pour toiser l'efficacité de notre approche, c'est le maniement de la racine carrée de l'erreur quadratique moyenne relative $RMSE_{rel}$ donnée par (4.13).

Pour étudier la $RMSE_{rel}$ nous avons pris l'ordre de moment $r = 1/10$ et le nombre d'essais $N_e = 10\,000$. Le nombre des échantillons N prend trois valeurs : $N = 32 \times 32 = 1024$, $N = 64 \times 64 = 4096$ et $N = 128 \times 128 = 16384$. Le modèle simulation adopté pour cette étude est illustré dans la figure 4.30. La figure 4.31 montre les tracés de la $RMSE_{rel}$ en fonction de la valeur exacte du paramètre γ .

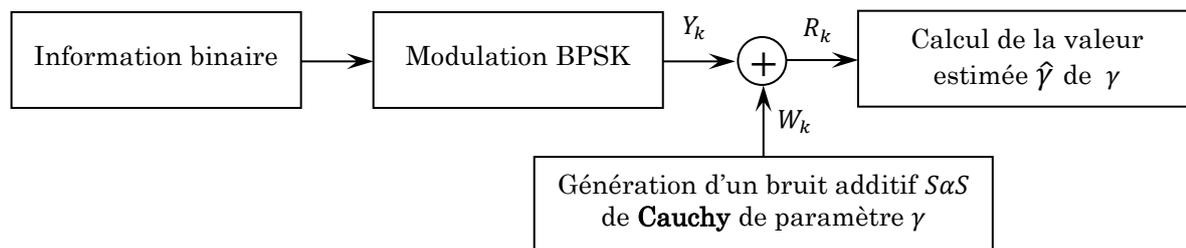


Fig. 4.30 Le modèle de simulation adopté pour calculer l'erreur $RMSE_{rel}$ en fonction de γ .

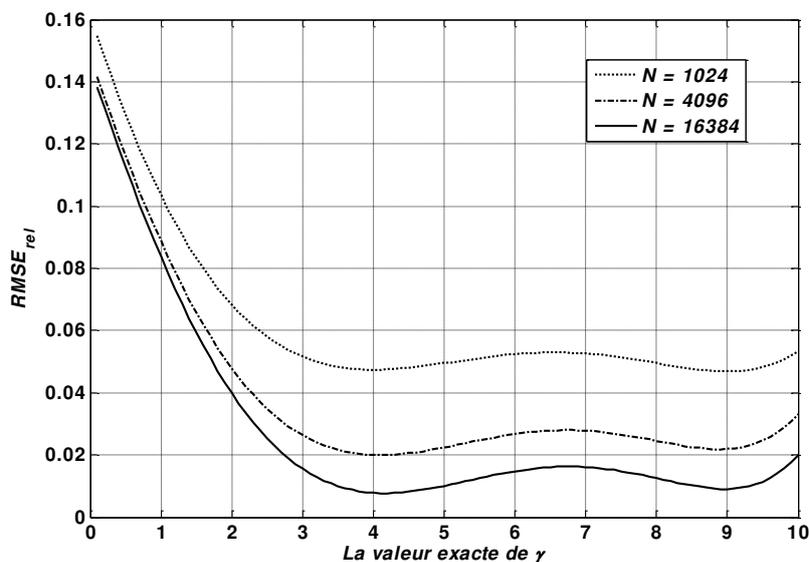


Fig. 4.31 La $RMSE_{rel}$ en fonction de la valeur exacte de γ pour trois valeurs du nombre d'échantillons N .

La figure 4.31 montre la variation de la $RMSE_{rel}$ en fonction de la valeur exacte de γ . L'analyse de cette figure montre que $RMSE_{rel}$ se diminue si nous augmentons le nombre d'échantillons, ce qui est prévisible. Avec un nombre d'échantillons $N = 4096$, et pour les grandes valeurs de γ ($\gamma \geq 2$), la $RMSE_{rel}$ est très petite et ne dépasse pas 4%, par contre, pour les petites valeurs ($\gamma < 2$), la $RMSE_{rel}$ se multiplie mais elle ne dépasse pas 15%.

Cependant, d'après ce que nous avons vu dans le paragraphe 4.3.4.1, le décodeur itératif est sensible à la disparité du paramètre γ si ce dernier est petit, par conséquent, nous devons étudier sans équivoque l'effet de l'erreur d'estimation de notre approche sur les performances du décodeur itératif. Pour ce faire, nous étudions les $RMSE$ et $RMSE_{rel}$ du rapport $LLR_c(R_k)$ donné par (4.45) et la variation du BER en fonction de $1/\gamma$ pour différentes estimations de γ , y compris l'estimation par notre méthode.

Nous rappelons que les $RMSE$ et $RMSE_{rel}$ sont définies par les formules (4.12) et (4.13) respectivement. Ces erreurs de $LLR_c(R_k)$ sont calculées entre les valeurs de $LLR_c(R_k)$ pour γ exacte et γ estimé noté $\hat{\gamma}$.

La figure 4.32 illustre les variations des $RMSE$ et $RMSE_{rel}$ de $LLR_c(R_k)$ en fonction de $1/\gamma$ (en dB) pour $\hat{\gamma}$ calculé par notre approche avec un nombre d'échantillons $N = 4096$, et pour trois autres mauvaises valeurs : $\hat{\gamma} = 2\gamma$, $\hat{\gamma} = 4\gamma$ et $\hat{\gamma} = 6\gamma$.

Pour l'étude du BER nous considérons le turbo code de la figure 4.5.c qui possède un rendement de $1/3$, un polynôme générateur (35,23) et avec un entrelaceur 15 –aléatoire de taille 64×64 . Le décodage itératif est effectué avec trois itérations (#3) en utilisant l'algorithme MAP.

La figure 4.33 montre les performances en termes de BER du décodeur itératif en fonction de $1/\gamma$ (en dB) pour les trois contextes : une mauvaise estimation de γ où $\hat{\gamma}$ est supposé égal à 2γ , une estimation de γ par notre approche et pour $\hat{\gamma}$ égal à la valeur exacte de γ .

La première remarque tirée de la figure 4.32.a est que pour les petits rapports $1/\gamma$ (grandes valeurs de γ), une mauvaise estimation de γ influe légèrement sur le rapport $LLR_c(R_k)$ ou sur les performances du décodeur itératif. En revanche, pour les grands rapports $1/\gamma$ (petites valeurs de γ), une estimation médiocre de γ agit considérablement sur le rapport $LLR_c(R_k)$ c'est-à-dire sur les performances du décodeur itératif, et plus cette estimation est mauvaise plus ces performances se dégradent. Ce résultat est déjà trouvé dans le paragraphe 4.3.4.1.

La deuxième remarque ôtée de la même figure est que notre approche présente une erreur très petite qui ne dépasse pas 0.16 quelque soit le rapport $1/\gamma$. Par contre, pour la mauvaise estimation de γ où $\hat{\gamma} = 2\gamma$ par exemple, l'erreur est très proche de 1. Pour donner une mesure comparative de ces erreurs, nous avons tracé la $RMSE_{rel}$ dans la figure 4.32.b. Cette figure montre clairement que notre approche est très efficace même pour les grandes valeurs du rapport $1/\gamma$, et elle présente une erreur relative de $LLR_c(R_k)$ qui ne dépasse jamais 7%. Par contre, pour le cas où $\hat{\gamma} = 2\gamma$ par exemple, l'erreur relative touche 55%.

A partir de la figure 4.33 nous constatons que la disparité du paramètre de dispersion exprimée par l'erreur d'estimation de γ dégrade les performances du décodeur itératif, et cette dégradation augmente au fur et à mesure que la valeur estimée $\hat{\gamma}$ s'écarte de la valeur exacte de γ . Dans le cas où la valeur de $\hat{\gamma}$ est calculée par l'approche proposée, nous remarquons une très légère dégradation au prorata du rapport $1/\gamma$. Par exemple, a un BER de 10^{-3} et en comparant avec le cas où le décodeur itératif utilise la valeur exacte de γ , nous obtenons une perte de gain de 0.092 dB. Dans le cas d'une mauvaise estimation où $\hat{\gamma} = 2\gamma$, nous observons une dégradation importante en termes du rapport $1/\gamma$. Au même BER de 10^{-3} , nous trouvons une perte de gain de 0.55 dB.

A la fin, pour le décodage itératif avec l'algorithme MAP sous l'effet d'un bruit impulsif $S\alpha S$ de **Cauchy**, nous concluons que l'approche d'estimation semi-aveugle proposée prodigue des performances très proches des celles où le paramètre γ du bruit est connu exactement.

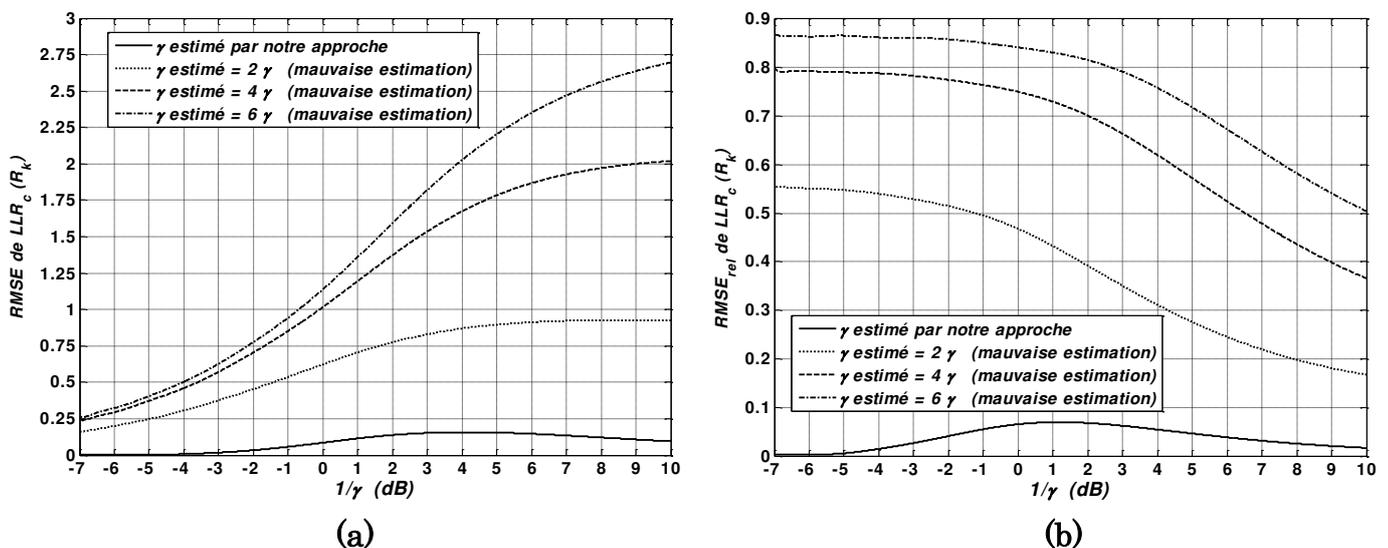


Fig. 4.32 Les deux erreurs de $LLR_c(R_k)$ en fonction de $1/\gamma$ (en dB) pour différentes estimation $\hat{\gamma}$: $\hat{\gamma}$ calculé par l'approche proposée, $\hat{\gamma} = 2\gamma$, $\hat{\gamma} = 4\gamma$ et $\hat{\gamma} = 6\gamma$. (a) $RMSE$ (b) $RMSE_{rel}$.

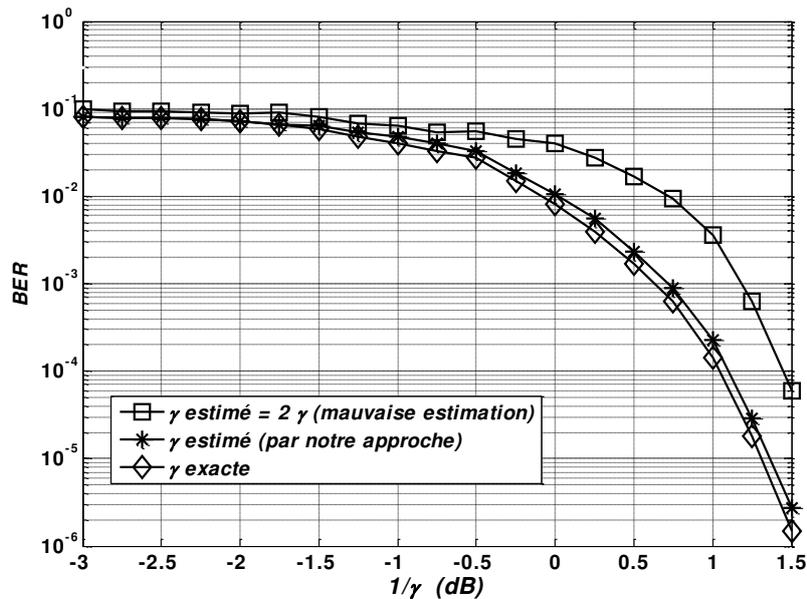


Fig. 4.33 Le *BER* du décodeur itératif pour un bruit impulsif *SaS* de Cauchy dans les trois contextes : (\square) γ estimé = 2γ (une mauvaise estimation) ($*$) γ estimé (par notre approche) (\diamond) γ exacte.

4.4 Conclusion

Dans ce chapitre nous avons présenté trois approches originales avec validation par simulation pour améliorer les performances d'un décodeur itératif.

La première approche est basée sur l'amélioration des informations extrinsèques qui sont fournies lors de chaque itération par les décodeurs élémentaires du décodeur itératif. Cette amélioration vient de la multiplication de la partie systématique du signal reçu au niveau du décodeur itératif par une fonction porte variable en fonction de l'itération.

Cette nouvelle technique est testée pour les deux algorithmes de décodage SOVA et MAP pour un bruit AWGN, et nous avons vu que le MID avec MAP présente la meilleure solution en termes de pouvoir de correction des erreurs. En plus, nous avons vu aussi que la fonction porte périodique $pp(i) = 100100 \dots$ donne une solution pratique très acceptable pour améliorer amplement le CID avec l'algorithme MAP. En outre, nous avons vu que grâce aux zéros trouvés dans la fonction porte $pp(i) = 100100 \dots$, le nombre des additions dans le processus de décodage itératif se réduit, ce qui donne un avantage en faveur de la technique proposée.

La deuxième approche permet d'estimer le paramètre de forme d'un bruit impulsif GG à partir du signal reçu directement. Pour cette raison elle est qualifiée de semi-aveugle. Cette approche est basée sur les FLOS et sur la probabilité que les deux signaux reçu et transmis soient de même signe.

Nous avons montré que l'approche proposée est plus précise que l'approche HOS proposée dans [121], et surtout lorsque le bruit est très impulsif. Numériquement, l'approche proposée donne une *RMSE* en moyenne presque cinq fois inférieure à celle de l'approche HOS.

En termes de *BER*, si le turbo décodeur utilise l'approche proposée il donne des performances très proches de celles obtenues s'il utilise les valeurs exactes des paramètres du bruit impulsif GG.

La troisième approche possède le même principe que la deuxième, mais elle s'applique au bruit impulsif *SaS* de **Cauchy** où $\alpha = 1$. Le paramètre à estimer ici est celui de dispersion du bruit γ .

La technique offerte permet aussi d'estimer le paramètre de dispersion du bruit impulsif à partir du signal reçu uniquement, comme la deuxième technique, mais la seule différence entre les deux est que le paramètre estimé par cette méthode est exprimé sous une forme implicite où il représente la solution unique de l'équation non linéaire (4.55), par contre, le paramètre estimé par la deuxième méthode possède une forme explicite donnée par l'équation (4.36).

Comme il n'y a pas une autre méthode qui traite le même problème dans la littérature jusqu'à présent, nous avons utilisé la *RMSE* relative pour évaluer cette approche. Nous avons vu que cette dernière est très efficace en termes de précision et prodigue, pour le décodage itératif avec l'algorithme MAP, des performances très proches des celles où le paramètre γ du bruit est connu exactement.

Conclusion générale et perspectives

Selon le cahier de charge exposé, notre travail concernant l'étude du décodage itératif pour un canal à bruit généralement non-gaussien ou impulsif, a englobé deux parties : une partie théorique et l'autre des contributions et simulations.

La partie théorique a pour but de dégager les fondements théoriques d'un DCS avec turbo codes et leurs décodages itératifs, à savoir : les théorèmes fondamentaux du codage de canal, la capacité du canal pour les bruits non-gaussiens de type impulsif, le principe des CTC, les différents types d'entrelaceurs utilisés dans les CTC, la notion de la modulation BPSK, les deux modèles largement utilisés pour les bruits impulsifs qui contaminent le canal : le modèle GG et le modèle α -satble, la notion de la fiabilité de décision mesurée par le LLR, la notion de l'information extrinsèque et le principe de décodage itératif, la notion des algorithmes de décodage SISO : SOVA et MAP. Les trois premiers chapitres ont été consacrés à cette partie.

La deuxième partie ne contient que le quatrième chapitre seulement. Ce chapitre était le plus intéressant dans cette thèse. Il présente trois nouvelles contributions pour améliorer les performances d'un décodeur itératif et la mise en œuvre d'une simulation permettant d'évaluer cette amélioration. Nos moyens informatiques utilisés lors de cette simulation ont limité la génération de la séquence d'information pseudo-aléatoire à envoyer à une longueur de $2^{20} = 1\,048\,576$ bits. Les performances d'un turbo décodeur itératif sont généralement évaluées en termes de taux d'erreurs binaire *BER* en fonction du *SNR* ou du rapport $1/\gamma$ (dans le cas d'un bruit *SaS* de **Cauchy**).

La première nouvelle approche présentée dans cette deuxième partie, permet de rendre plus fiables les informations extrinsèques délivrées par les décodeurs constitutifs dans chaque itération. Cette approche consiste à insérer dans l'entrée systématique du

décodeur itératif une fonction porte (binaire) variable en fonction de l'itération. Elle est conçue pour les deux algorithmes de décodage SOVA et MAP pour un bruit AWGN. La simulation numérique nous a permis de conclure que cette approche rend le décodeur itératif plus performant en termes de *BER* quelque soit l'algorithme de décodage utilisé SOVA ou MAP. Tant que l'algorithme MAP est toujours meilleur que le SOVA, parce qu'il est optimal, le décodeur itératif avec MAP utilisant l'approche proposée, présente la meilleure solution en termes de *BER*. En plus, cette simulation nous a montré que le décodeur itératif avec MAP utilisant la fonction porte périodique $pp(i) = 100100 \dots$ dans son entrée systématique, présente une solution pratique très acceptable. En outre, comme le deux-tiers des bits de la fonction porte $pp(i) = 100100 \dots$ sont des zéros, le nombre des additions dans le processus de décodage itératif se réduit de même rapport, ce qui donne un avantage de plus en faveur de l'approche proposée.

Nous avons vu dans le chapitre 3 que l'information extrinsèque d'un bit dans l'algorithme optimal MAP est la différence entre le LLR du bit considéré et LLR du canal pour ce bit. Or, ces deux LLR dépendent étroitement des paramètres du bruit qui contamine le canal. Dès lors, le décodeur itératif avec l'algorithme MAP doit estimer ces paramètres à partir de ce qu'il a reçu. Si le bruit considéré est gaussien à moyenne nulle, ce problème d'estimation ne se pose pas, car le seul paramètre à estimer est la variance du bruit qui est égale à celle du signal reçu moins un. Par contre, ce problème devient très sérieux si le bruit est impulsif (non-gaussien). Par souci de surmonter ce problème, nous avons proposé dans la deuxième partie de notre travail deux autres nouvelles approches qui sont deux méthodes d'estimation. Elles permettent de faire l'estimation des paramètres à partir du signal reçu uniquement, où elles n'ont pas besoin de séparer le bruit au signal utile. Pour cette raison elles sont qualifiées de semi-aveugles. Ces deux approches sont basées sur les FLOS et sur la probabilité que les deux signaux reçu et transmis soient de même signe.

Dans ce contexte, la nouvelle méthode d'estimation qui est établie pour un bruit impulsif à GGD, où le paramètre de forme à estimer est compris entre 0 et 1, représente la deuxième approche proposée dans cette deuxième partie. Nous avons vu que le paramètre estimé par cette technique possède une expression explicite en fonction des FLOS du signal reçu. En plus, nous avons montré par simulation que cette approche est plus précise que l'approche HOS proposée dans [121], notamment lorsque le bruit est très impulsif. Son erreur *RMSE* est en moyenne presque cinq fois inférieure à celle de

l'approche HOS. En termes de *BER*, si le décodeur itératif avec l'algorithme MAP utilise le paramètre de forme du bruit estimé par cette approche, il donne des performances très proches de celles obtenues s'il utilise la valeur exacte de ce paramètre.

Quant à la nouvelle méthode d'estimation conçue pour un bruit impulsif *SαS* de **Cauchy**, elle représente la troisième approche proposée dans la deuxième partie de ce travail. Le paramètre à estimer par cette approche est celui de dispersion.

A rebours de la deuxième approche, le paramètre estimé par cette méthode est exprimé sous une forme implicite où il représente la solution unique d'une équation homogène non linéaire. La méthode de **Newton** par exemple, permet de résoudre avec une grande précision cette équation.

Malheureusement, jusqu'à présent n'existe aucune méthode d'estimation dans la littérature qui traite le même problème que notre troisième approche. Pour cette raison, nous avons utilisé la *RMSE* relative pour évaluer cette nouvelle technique. Nous avons vu que cette dernière est très efficace en termes de précision, et offre au décodeur itératif avec l'algorithme MAP, des performances en termes de *BER* très voisines des celles où le paramètre de dispersion est connu exactement.

A l'issue de cette thèse, plusieurs perspectives peuvent être envisagées. Elles se résument en certains points restent à approfondir et d'autres à étudier, nous pouvons en évoquer quelques-uns

- Etendre la première approche proposée aux bruits impulsifs à GGD et à distributions α -stables ;
- Combiner la première approche avec les deux autres approches dans un même décodeur itératif ;
- Appliquer la deuxième et la troisième approches à l'algorithme SOVA en essayant de les enrichir ;
- Généraliser nos trois approches pour un autre type de modulation, comme le 4-PSK, 8-PSK, etc. Ici, il est important de signaler que la deuxième et la troisième approche se généralisent sans peine si la taille M de la constellation de modulation vérifie l'égalité $M/2 = \log_2 M$;
- Appliquer nos trois approches à d'autres types des turbo codes hors que les CTC. Par exemple les codes à contrôle de parité de faible densité LDPC ;
- Etendre la troisième approche pour un bruit à distribution *SαS* où α est différent de 1 et 2.

Annexe A

La GGD et la loi gamma

A.1 La variable aléatoire à GGD comme une transformation d'une variable aléatoire gamma

Prenons le cas où la variable aléatoire X à GGD avec une moyenne $\mu = 0$ et variance $\sigma^2 = 1$ et un paramètre de forme α . Alors la densité de probabilité PDF de X est

$$f_X(x; \alpha) = \frac{\alpha}{2} \frac{A(\alpha)}{\Gamma(1/\alpha)} \exp\{-(A(\alpha)|x|)^\alpha\} \quad x \in \mathbb{R}, \quad (\text{A.1})$$

où

$$A(\alpha) = \left[\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)} \right]^{1/2} \quad (\text{A.2})$$

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

et sa fonction de distribution cumulative CDF est

$$F_X(x; \alpha) = \int_{-\infty}^x \frac{\alpha}{2} \frac{A(\alpha)}{\Gamma(1/\alpha)} \exp(-(A(\alpha)|t|)^\alpha) dt \quad (\text{A.3})$$

Comme nous avons déjà vu, une telle fonction ne peut être écrite sous une forme fermée que pour quelques cas très spéciaux.

Supposons que nous sommes en mesure de simuler la variable aléatoire Y , qui est une transformation de X . En particulier, nous considérons la relation

$$Y = h(X) = |X| \quad (\text{A.4})$$

Le but est de simuler X compte tenu les observations aléatoires de Y . Pour ce faire il faut tout d'abord connaître les probabilités des racines x_1 et x_2 , notées $p(x_1)$ et $p(x_2)$ respectivement, de l'équation (A.4).

Soient X une variable aléatoire, avec une PDF $f_X(x)$ et une CDF $F_X(x)$, et Y une autre variable aléatoire telle que $Y = g(X)$. Nous supposons que l'équation $y = g(x)$ possède n racines x_i et la fonction dérivée g' existe toujours et que $g'(x_i) \neq 0$.

Nous considérons l'intervalle $[y - h, y + h]$ où h est une quantité positive infinitésimale. L'image inverse de cet intervalle est

$$g^{-1}([y - h, y + h]) = \bigcup_{i=1}^n [x_i - h_i, x_i + h_i]$$

où $[x_i - h_i, x_i + h_i]$ est un intervalle trop petit sur lequel la fonction g est monotone. Dans cet intervalle nous avons

$$g^{-1}(y + h) \cong g^{-1}(y) + (g^{-1})'(y) \cdot h = x_i + \frac{1}{g'(x_i)} \cdot h$$

et

$$g^{-1}(y - h) \cong g^{-1}(y) - (g^{-1})'(y) \cdot h = x_i - \frac{1}{g'(x_i)} \cdot h$$

alors $h_i = \frac{1}{g'(x_i)} \cdot h$ si g' est croissante sur $[x_i - h_i, x_i + h_i]$ et $h_i = -\frac{1}{g'(x_i)} \cdot h$ si g' est décroissante sur $[x_i - h_i, x_i + h_i]$, ce qui vaut dire que

$$h_i = \frac{1}{|g'(x_i)|} \cdot h \quad (\text{A.5})$$

Nous calculons la probabilité de $X = x_i$ où $Y = y$ notée $p(x_i)$. Nous pouvons écrire alors

$$p(x_i) = P(X = x_i | Y = y) \quad (\text{A.6})$$

Nous posons

$$\begin{aligned} p_h(x_i) &= P(X \in [x_i - h_i, x_i + h_i] | Y \in [y - h, y + h]) \\ &= \frac{P(X \in [x_i - h_i, x_i + h_i], Y \in [y - h, y + h])}{P((X \in [x_1 - h_1, x_1 + h_1]) \vee (X \in [x_2 - h_2, x_2 + h_2]) \vee \dots \vee (X \in [x_n - h_n, x_n + h_n]))} \end{aligned}$$

et comme $X \in [x_i - h_i, x_i + h_i] \Rightarrow Y \in [y - h, y + h]$ alors $P(X \in [x_i - h_i, x_i + h_i], Y \in [y - h, y + h]) = P(X \in [x_i - h_i, x_i + h_i])$ et nous avons donc

$$p_h(x_i) = \frac{P(X \in [x_i - h_i, x_i + h_i])}{\sum_{j=1}^n P(X \in [x_j - h_j, x_j + h_j])} = \left(1 + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{P(X \in [x_j - h_j, x_j + h_j])}{P(X \in [x_i - h_i, x_i + h_i])} \right)^{-1}$$

$$\begin{aligned}
&= \left\{ 1 + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{F_X(x_j + h_j) - F_X(x_j - h_j)}{F_X(x_i + h_i) - F_X(x_i - h_i)} \right\}^{-1} \\
&= \left\{ 1 + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\frac{F_X(x_j + h_j) - F_X(x_j - h_j)}{2h_j}}{\frac{F_X(x_i + h_i) - F_X(x_i - h_i)}{2h_i}} \cdot \frac{h_j}{h_i} \right\}^{-1}
\end{aligned}$$

Or (A.5) nous permet d'écrire

$$\frac{h_j}{h_i} = \left| \frac{g'(x_i)}{g'(x_j)} \right|$$

et

$$\lim_{h \rightarrow 0^+} \frac{F_X(x_j + h_j) - F_X(x_j - h_j)}{2h_j} = \lim_{h_j \rightarrow 0^+} \frac{F_X(x_j + h_j) - F_X(x_j - h_j)}{2h_j} = f_X(x_j)$$

A la fin nous pouvons écrire

$$p(x_i) = \lim_{h \rightarrow 0^+} p_h(x_i) = \left\{ 1 + \sum_{\substack{j=1 \\ j \neq i}}^n \left| \frac{g'(x_i)}{g'(x_j)} \right| \cdot \frac{f_X(x_j)}{f_X(x_i)} \right\}^{-1} \quad i = 1, \dots, n \quad (\text{A.7})$$

Dans notre cas $g(x) = |x|$ et l'équation $y = g(x)$ possède deux racines distinctes, $x_1 = -y$ et $x_2 = y$ (donc $n = 2$). Nous avons alors $g'(x_2) = -g'(x_1) = 1$ et $f_X(x_1) = f_X(x_2)$ car la densité f_X d'une GGD est paire. Par conséquent, $p(x_1) = p(x_2) = 1/2$.

Soit f_Y la PDF de la variable aléatoire Y définie par l'équation (A.4), alors

$$f_Y(y; \alpha) = \frac{\alpha}{2} \frac{A(\alpha)}{\Gamma(1/\alpha)} \exp\{-(A(\alpha)y)^\alpha\} \quad y \geq 0, \quad (\text{A.8})$$

Considérons maintenant une variable aléatoire Z avec PDF

$$f_Z(z; a, b) = \frac{b^a}{\Gamma(a)} z^{a-1} e^{-bz} \quad z > 0, a > 0 \text{ et } b > 0 \quad (\text{A.9})$$

La variable aléatoire Z est gamma distribuée avec paramètres a et b , c'est-à-dire $Z \sim \text{gamma}(a, b)$. Si nous prenons $a = 1/\alpha$ et $b = A(\alpha)^\alpha$, et nous considérons la transformation $Y = Z^{1/\alpha}$, il peut être montré que Y a la distribution d'une variable aléatoire gaussienne généralisée à valeur absolue avec PDF indiquée dans (A.8).

Nous avons $z = y^\alpha$ et $dz/dy = \alpha y^{\alpha-1}$. Par substitution dans (A.9), et compte tenu de la CDF de Z , nous obtenons

$$\begin{aligned}
 F_Z(z) &= \int_0^z \frac{b^a}{\Gamma(a)} t^{a-1} e^{-bt} dt \\
 &= \int_0^z \frac{A(\alpha)}{\Gamma(1/\alpha)} t^{1/\alpha-1} e^{-A(\alpha)t^\alpha} dt \\
 &= \int_0^y \frac{A(\alpha)}{\Gamma(1/\alpha)} u^{1-\alpha} e^{-A(\alpha)u^\alpha} \alpha u^{\alpha-1} du \\
 &= \int_0^y \frac{\alpha A(\alpha)}{\Gamma\left(\frac{1}{\alpha}\right)} e^{-(A(\alpha)u)^\alpha} du
 \end{aligned} \tag{A.10}$$

Enfin, compte tenu de la transformation (A.4), nous obtenons

$$\begin{aligned}
 F_Y(y) &= \int_0^y \frac{\alpha A(\alpha)}{\Gamma\left(\frac{1}{\alpha}\right)} e^{-(A(\alpha)u)^\alpha} du \\
 &= \int_{-\infty}^x \frac{\alpha}{2} \frac{A(\alpha)}{\Gamma\left(\frac{1}{\alpha}\right)} e^{-(A(\alpha)|\xi|)^\alpha} d\xi
 \end{aligned} \tag{A.11}$$

Par conséquent X a une distribution gaussienne généralisée avec un paramètre de forme α . Nous avons maintenant justifiée la génération d'une variable aléatoire GG à partir d'une variable aléatoire gamma.

Pour obtenir des valeurs d'une variable aléatoire X à GGD avec le paramètre α , nous pouvons utiliser la procédure à trois étapes suivante [6]

- i. Simuler Z , une variable aléatoire gamma avec PDF décelée dans (A.9), pour obtenir les réalisations z_i pour $i = 1, 2, \dots, N$;
- ii. calculer $y_i = z_i^{1/\alpha}$, qui sont des réalisations d'une variable aléatoire Y avec PDF indiquée dans (A.8) ;
- iii. prendre $X = (-1)^B Y$, où B est une variable aléatoire de **Bernoulli** de paramètre égal à $1/2$.

A.2 Simulation d'une variable aléatoire gamma

Pour appliquer la procédure à trois étapes décrite précédemment dans la section A.1, il faut primo simuler la distribution gamma. Nous savons bien, d'après les propriétés de la distribution gamma, que si $Z \sim \text{Gamma}(a, b)$ alors $bZ \sim \text{Gamma}(a, 1)$. Par conséquent, les réalisations de la distribution $\text{Gamma}(a, b)$ s'obtiennent en divisant ceux de la distribution $\text{Gamma}(a, 1)$ par b .

Le problème maintenant est de savoir comment simuler une variable aléatoire $Gamma(a, 1)$. Nous décrivons ici une procédure de simulation qui est basé sur la méthode du rejet [146].

Il est bien connu que, si $a = 1$ et $b = 1$ la distribution $Gamma(1,1)$ correspond à une distribution exponentielle de paramètre $\lambda = 1$, $Z \sim \mathbb{E}(1)$. Afin de simuler une variable aléatoire exponentielle, nous pouvons générer de variables aléatoires à partir d'une distribution uniforme et prendre la transformation $Z = -\ln(U)$ où $U \sim \mathbb{U}([0,1])$. Alors $Z \sim gamma(1,1)$. Nous avons également que si $Z = -\sum_{k=1}^N \ln(U_k)$, où les U_k sont des variables aléatoires i.i.d selon la loi uniforme $\mathbb{U}([0,1])$, alors $Z \sim Gamma(N, 1)$. Ceci résout le problème de simulation d'une distribution gamma avec un paramètre a entier. Considérons le problème de la simulation d'une distribution $Gamma(\theta, 1)$, avec $0 < \theta < 1$. Nous posons $v = \frac{e}{e+\theta}$ où $\ln(e) = 1$. La procédure suivante fait parfaitement l'affaire [6]

1. générer v_1 et v_2 à partir de deux variables aléatoires indépendantes V_1 et V_2 , uniforme sur $[0,1]$;
2. si $v_1 \leq v$, alors calculer $\xi = \left(\frac{v_1}{v}\right)^{\frac{1}{\theta}}$ et $\eta = v_2 \xi^{\theta-1}$, sinon calculer $\xi = 1 - \ln\left(\frac{v_1-v}{1-v}\right)$ et $\eta = v_2 e^{-\xi}$;
3. si $\eta > \xi^{\theta-1} e^{-\xi}$, aller à l'étape 1 ;
4. prendre ξ comme une réalisation d'une variable aléatoire à distribution $Gamma(\theta, 1)$.

Nous pouvons maintenant traiter le cas général. Nous mettons $\theta = a - [a]$, où $[a]$ est la partie entière de a (dès lors θ est la partie fractionnelle de a). Après avoir obtenu ξ tel que décrit ci-dessus, nous apposons

$$\zeta = \xi - \sum_{k=1}^{[a]} \ln(u_k) \quad (A.12)$$

où u_k sont des réalisations des variables aléatoires $U_k \sim \mathbb{U}([0,1])$, qui sont également indépendantes de V_1 et V_2 . Enfin, prendre ζ comme une réalisation d'une variable aléatoire à distribution $Gamma(a, 1)$.

Annexe B

Les séquences pseudo-aléatoires

B.1 Introduction

Les séquences pseudo-aléatoires qui sont appelées aussi les séquences pseudo-bruits PN [147], séquences à registre à décalage de longueur maximale ou m séquences) sont des séquences binaires avec une longueur $n = 2^m - 1$.

Ils ont plusieurs propriétés pratiques, une de ces propriétés, est la fonction d'autocorrélation (périodique) qui est donnée par [90] [22]

$$\varphi(0) = 1, \varphi(i) = -\frac{1}{n} \quad \text{pour } 1 \leq i \leq n - 1 \quad (B.1)$$

B.2 Les séquences pseudo-aléatoires

B.2.1 Le registre à décalage

Pour construire une séquence pseudo-aléatoire avec une longueur $n = 2^m - 1$, nous sommes besoin seulement du polynôme primitif (un polynôme irréductible) $h(x)$ de degré m [147][22]. Par exemple le polynôme

$$h(x) = x^4 + x + 1 \quad (B.2)$$

de degré $m = 4$. Ce polynôme spécifie un registre à décalage avec un feedback comme indiquer dans la figure B.1. En général, le registre à décalage est constitué de m éléments, chaque élément contient un 0 ou 1. A chaque unité de temps les contenus de ces éléments sont décalés d'une seule place vers la droite et les éléments correspondants aux termes dans $h(x)$ sont additionnés et injectés dans l'élément mémoire numéro 4 (la somme est calculée en modulo-2).

Dans l'exemple ci-dessus, si le registre contient $a_{i+3}, a_{i+2}, a_{i+1}, a_i$ à l'instant i , alors à l'instant $i + 1$ il contient

$$a_{i+4} = a_{i+1} + a_i, \quad a_{i+3}, \quad a_{i+2}, \quad a_{i+1},$$

comme il est indiqué dans la figure B.2. Autrement dit, ce registre à décalage avec feedback génère une séquence infinie $a_0, a_1, a_2, \dots, a_i, \dots$ qui satisfait à la relation de récurrence

$$a_{i+4} = a_{i+1} + a_i \quad i = 0, 1 \quad (B.3)$$

où $+$ représente ici l'addition modulo-2, en donnant les valeurs initiales a_0, \dots, a_{m-1} .

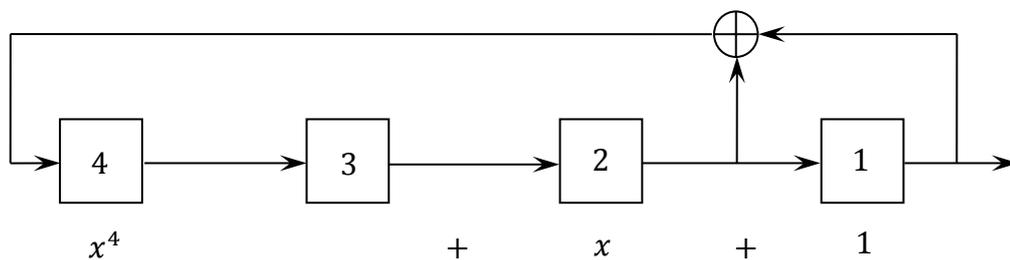


Fig. B.1 Registre à décalage avec un feedback correspondant à $h(x) = x^4 + x + 1$.

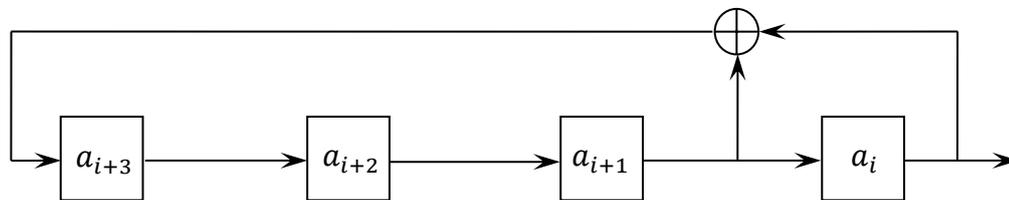


Fig. B.2 Registre à décalage défini par une relation de récurrence.

B.2.2. Séquences pseudo-aléatoires

Puisque chaque élément mémoire parmi les m éléments mémoires contient un 1 ou 0, donc il y a 2^m états possibles. Ainsi la séquence a_0, a_1, \dots doit être périodique. Mais l'état $00 \dots 0$ ne peut pas se produire à moins que la séquence soit toute zéros, donc la période maximale possible est $2^m - 1$.

Pour le polynôme primitif $h(x)$ de l'équation (B.2), sa séquence de sortie illustrée dans la figure B.1 ayant une période $2^4 - 1 = 15$.

Nous acceptons que pour chaque entier strictement positif m il existe un polynôme primitif de degré m [22]. Dans le tableau B.1, nous donnons les polynômes primitifs de degré $m \leq 40$ [90]. Si $h(x)$ un polynôme primitif de degré m , le registre à décalage passe

par tous les états distincts (non nuls) avant la répétition, et produit une séquence de sortie de période $2^m - 1$. A partir de chaque série pseudo-aléatoire $a_i, a_{i+1}, \dots, a_{i+2^m-2}$ de longueur $2^m - 1$, nous pouvons tirer $2^m - 1$ séquences pseudo-aléatoires différentes.

<i>Deg m</i>	$h(x)$	<i>Deg m</i>	$h(x)$
1	$x + 1$	21	$x^{21} + x^2 + 1$
2	$x^2 + x + 1$	22	$x^{22} + x + 1$
3	$x^3 + x + 1$	23	$x^{23} + x^5 + 1$
4	$x^4 + x + 1$	24	$x^{24} + x^4 + x^3 + x + 1$
5	$x^5 + x^2 + 1$	25	$x^{25} + x^3 + 1$
6	$x^6 + x + 1$	26	$x^{26} + x^8 + x^7 + x + 1$
7	$x^7 + x + 1$	27	$x^{27} + x^8 + x^7 + x + 1$
8	$x^8 + x^6 + x^5 + x + 1$	28	$x^{28} + x^3 + 1$
9	$x^9 + x^4 + 1$	29	$x^{29} + x^2 + 1$
10	$x^{10} + x^3 + 1$	30	$x^{30} + x^{16} + x^{15} + x + 1$
11	$x^{11} + x^2 + 1$	31	$x^{31} + x^3 + 1$
12	$x^{12} + x^7 + x^4 + x^3 + 1$	32	$x^{32} + x^{28} + x^{27} + x + 1$
13	$x^{13} + x^4 + x^3 + x + 1$	33	$x^{33} + x^{13} + 1$
14	$x^{14} + x^{12} + x^{11} + x + 1$	34	$x^{34} + x^{15} + x^{14} + x + 1$
15	$x^{15} + x + 1$	35	$x^{35} + x^2 + 1$
16	$x^{16} + x^5 + x^3 + x^2 + 1$	36	$x^{36} + x^{11} + 1$
17	$x^{17} + x^3 + 1$	37	$x^{37} + x^{12} + x^{10} + x^2 + 1$
18	$x^{18} + x^7 + 1$	38	$x^{38} + x^6 + x^5 + x + 1$
19	$x^{19} + x^6 + x^5 + x + 1$	39	$x^{39} + x^4 + 1$
20	$x^{20} + x^3 + 1$	40	$x^{40} + x^{21} + x^{19} + x^2 + 1$

Tab. B.1 Polynômes primitifs pour $m \leq 40$.

B.2.3 Propriétés des séquences pseudo-aléatoires [90][22]

Soit $h(x)$ un polynôme primitif de degré m et soit δ_m l'ensemble constitué de ces séquences pseudo-aléatoires générées à partir de $h(x)$.

Propriété 1 : La propriété de décalage : Si $b_0, b_1, \dots, b_{2^m-2}$ est une séquence pseudo-aléatoire quelconque dans δ_m , alors tout décalage cyclique de b , noté $b_j b_{j+1} \dots b_{2^m-2} b_0 \dots b_{j-1}$ est aussi dans δ_m .

Propriété 2 : La récurrence : Nous supposons que $h(x) = \sum_{i=0}^m h_i x^i$ avec $h_0 = h_m = 1$, $h_i = 0$ ou 1 pour $0 < i < m$. Toute séquence pseudo-aléatoire $b \in \delta_m$ satisfait la relation de récurrence

$$b_{i+m} = h_{m-1}b_{i+m-1} + h_{m-2}b_{i+m-2} + \dots + h_1b_{i+1} + b_1 \text{ pour } i = 0, 1, \dots \quad (B.4)$$

Réciproquement toute solution de (B.4) est dans δ_m . Il y a m solutions de (B.4) linéairement indépendantes, d'où il y a m séquences linéairement indépendantes dans δ_m .

Propriété 3 : Propriété de fenêtre : Toute fenêtre de longueur m (glissée en avant) d'une séquence pseudo-aléatoire dans δ_m est apparaître exactement une seule fois (Figure B.3).

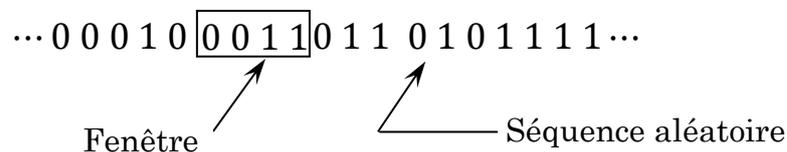


Fig. B.3 Fenêtre d'une séquence pseudo-aléatoire.

Propriété 4 : Demi 0 et demi 1 : Chaque séquence pseudo-aléatoire dans δ_m contient 2^{m-1} « 1 » et $2^{m-1} - 1$ « 0 ». Ce qui signifie que la probabilité d'avoir un 1 est la même que celle d'avoir un 0.

Propriété 5 : Propriété de l'addition : La somme de deux séquences dans δ_m est une autre séquence dans δ_m (la somme de deux solutions de (B.4) est aussi une solution).

Propriété 6 : Propriété du décalage et de l'addition : La somme d'une séquence pseudo-aléatoire et un décalage cyclique de lui-même est une autre séquence pseudo-aléatoire et m décalage cyclique de lui-même est une autre séquence pseudo-aléatoire (à partir des propriétés 1 et 2).

B.2.4 La fonction d'autocorrélation

Nous arrivons maintenant à la propriété la plus importante. La fonction d'autocorrélation $\varphi(i)$ d'une séquence réelle (ou complexe) $s_0 s_1 \dots s_{n-1}$ de longueur n est définie par [90][22]

$$\varphi(i) = \frac{1}{n} \sum_{j=0}^{n-1} s_j \bar{s}_{i+j} \quad i = 0, \pm 1, \pm 2, \dots \quad (B.5)$$

C'est une fonction périodique : $\varphi(i) = \varphi(i + n)$. La fonction d'autocorrélation d'une séquence binaire $a_0 a_1 a_2 \dots a_{n-1}$ doit être égale à la fonction d'autocorrélation de séquence réel $(-1)^{a_0}, \dots, (-1)^{a_{n-1}}$ obtenu, en remplaçant les « 1 » par « - 1 » et les « 0 » par « + 1 », ainsi

$$\varphi(i) = \frac{1}{n} \sum_{j=0}^{n-1} (-1)^{a_j + a_{i+j}} \quad (B.6)$$

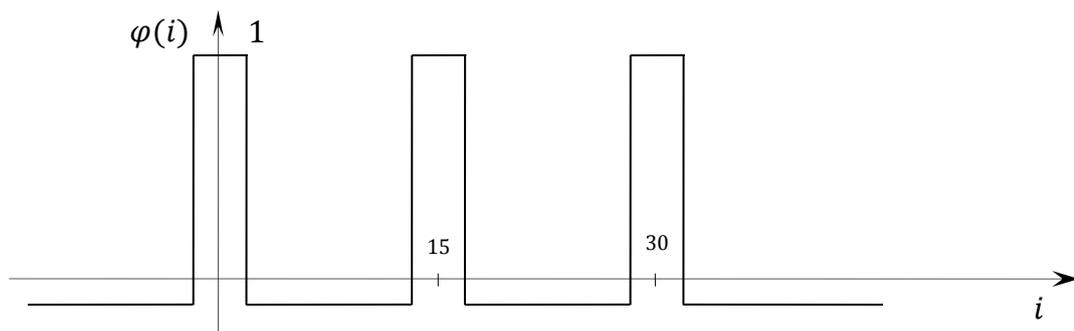


Fig. B.4 Fonction d'autocorrélation d'une séquence pseudo-aléatoire.

Propriété 7 : La fonction d'autocorrélation : La fonction d'autocorrélation d'une séquence pseudo-aléatoire de longueur $n = 2^m - 1$ est donnée par

$$\varphi(0) = 1, \quad \varphi(i) = -\frac{1}{n} \quad \text{pour } i = 1, \dots, 2^m - 2$$

D'après la figure B.4 nous remarquons que la fonction d'autocorrélation d'une séquence pseudo-aléatoire de longueur $n = 2^m - 1$ est une pseudo impulsion de **Dirac** (dans une période) ce qui signifie que la séquence est effectivement pseudo-aléatoire.

B.2.5 La Série

Nous définissons une série comme une chaîne maximale des symboles consécutifs identiques. Par exemple, pour la séquence 000100110101111 contient des séries de quatre 1, trois 0, deux 1, deux 0, deux séries simples de 1 et deux séries simples de 0, la totale est de 8 séries [90].

Propriété 8 : La Série : Dans chaque séquence pseudo-aléatoire $1/2$ des séries ayant une longueur de 1, $1/4$ ayant une longueur de 2, $1/8$ ayant une longueur de 3, et ainsi de suite, jusqu'à $1/2^{m-1}$. Dans chaque cas, le nombre de séries de 0 est égal au nombre de séries de 1.

B.2.6 Séquences pseudo-aléatoires q -aires

Les séquences pseudo-aléatoires peuvent être définies sur des corps de Galois q -éléments avec q premier ou sur des corps de **Galois** étendus.

Ces séquences sont très utilisables dans les simulations de transmissions numériques à q -états [22].

Bibliographie

- [1] C. Berrou, *codes et turbocodes*, Springer-Verlag, France, 2007.
- [2] C. Berrou, A. Glavieux, *turbo codes : principe et application*, département électronique, département signal et communications, ENST, France. 1995.
- [3] 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](#).
- [4] 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](#).
- [5] 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](#).
- [6] A. Monir , *Contribution à la modélisation et à la synthèse des signaux aléatoires : signaux non gaussiens, signaux à corrélation non exponentielle*, Thèse de doctorat, Novembre 2003, université d'Angers, Angers, France.
- [7] H. Ben Maad, *Optimisation des stratégies de décodage des codes LDPC dans les environnements impulsifs : application aux réseaux de capteurs et ad hoc*, Thèse de doctorat de l'Université de Reims Champagne-Ardenne, Juin 2011.
- [8] L. Anthony, *Blancheur et non-gaussianité pour la déconvolution aveugle de données bruitées : application aux signaux sismiques*, Thèse de Doctorat de L'INPG, Institut National Polytechnique de Grenoble, 13 Septembre 2006.
- [9] B. Mandelbrot, « The paul Lévy I knew, Lévy flights and related topics in physics, » *Lecture Notes in Phys*, Berlin, 450 : pp. ix-xii, 1995.
- [10] J. L. Chambers, C. L. Mallows and B. W. Stuck, « A methode for simulating stables random

- variables, » J. Amer. Stat. Assoc., 71 : pp. 304-344, 1976.
- [11] R. Kapoor, A. Banerjee, G. A. Tsihrintzis and N. Nandhakumar, « Uwb radar detection of targets in foliage using alpha-stable clutter models, » IEEE Transactions on Aerospace and Electronic Systems, 35 : pp. 819-834, 1999.
- [12] S. Porcarelli, G. Procissi, R. G. Garroppo and S. Giordano, « Testing alpha-stable processes in modeling broadband teletraffic, » Proc. of ICC 2000, New Orleans, Louisiana, USA, pp. 18-22, 2000.
- [13] C. Nikias and M. Shao, *Signal Processing with Alpha Stable Distributions and Applications*, Wiley, New York, 1995.
- [14] J. G. Gonzalez, J. L. Paredes, and G. R. Arce, « Zero-order statistics : A mathematical framework for the processing and characterization of very impulsive signals, » IEEE Trans. Sig. Process., vol. 54, no. 10, pp. 3839–3851, Octobre 2006.
- [15] J. G. Gonzalez, *Robust techniques for wireless communications in non-Gaussian environments*, Ph.D. dissertation, Dept. Elec. Eng., Univ. of Delaware, Newark, DE, 1997.
- [16] S. A. Kassam, *Signal Detection in Non-Gaussian Noise*, Springer, New York, NY, USA, 1988.
- [17] 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.
- [18] D. Roviras, *Communications numériques 1* (version 10), CNAM, Département EASY Electronique, Automatique et Systèmes, 2014-2015. Paris.
- [19] 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.
- [20] 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.
- [21] Y. Mori, *Théorie de l'information et du codage : signal analogique, signal numérique et application et télécommunications*, vol 5, Ed. Lavoisier 2006, Paris.
- [22] A. Spataru, *Fondements de la théorie de la Transmission de l'information*, Presse Polytechnique Romandes. Lausanne, Suisse, 1987.
- [23] F. Flitti et Y. Omarouyoub, « codes convolutifs : Etude, Simulation et évaluation, » Projet de fin d'étude, ENP, département d'électronique, Septembre 1996.
- [24] K. Banzouzi, *Mesures de dépendance pour une modélisation alpha-stable. Application aux séries chronologiques*

- stables*, Thèse Doctorat en statistique, Août 2011, Université Toulouse III - Paul Sabatier, Toulouse, France.
- [25] A. L. 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.
- [26] C. Poulliat, A. Goupil, M. Colas, G. Gelle, « Performances des codes LDPC pour l'Ultra-Large Bande (ULB) impulsionnelle sous approximation gaussienne généralisée des interférences, » GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 2009.
- [27] S. Haykin, *Digital Communication*, John Wiley & Sons, New York, 1988.
- [28] F. Menezla, *Etude et évaluation d'un turbo-code dans son approche pragmatique : Turbo-code*, Editions universitaires européennes EUE, 2012.
- [29] Y. Mori, *Codage de source et de canal*, vol 6, Ed. Lavoisier, Paris-France, 2006.
- [30] V. T. Zarasoa, « Codage de sources distribuées : codage symétrique et adaptatif en débit de sources corrélées, » Rapport de stage, université de Rennes 1, cedex, France, Septembre 2007.
- [31] J. L. Ramsey. « Realization of optimum interleavers, » IEEE Trans. Inform. Theory. vol.16. no. 3, pp. 338-345, May 1970.
- [32] S. A. Barbulescu, and S. S. Pietrobon, « Interleaver Design for Turbo codes, » Electronic Letters, 30(25), pp. 2107-2108, 1994.
- [33] G. D. Forney, « Burst-correcting codes for the class bursty channel, » IEEE Trans. On Commun. Technol. vol. COM-19, no. 5, pp. 772-780, October 1971.
- [34] P. Csillag, *Introduction aux codes correcteurs*, Edition Marketing, 1990.
- [35] 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.
- [36] S. Dolinar, and D. Divsalar, « Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations, » JPL TDA Progress Report, pp. 42-122, August, 1995.
- [37] C. Heegard and S. Wicker, *Turbo Coding*, Kluwer Academic Press, January 1999.
- [38] A. H. Osseiran, *Sur le décodage des codes turbo*, Mémoire de maîtrise des sciences appliquées (Génie électrique), Université de Montréal, Ecole Polytechnique de Montréal, Canada, Octobre 1999.
- [39] G. Royer, *Evaluation des entrelaceurs au sein des Codes Turbo par simulations*, Mémoire de Maîtrise des sciences appliquées, Ecole polytechnique de Montréal, Novembre 2000.

- [40] J. Briffa, *Interleavers for Turbo Codes*, Master dissertation of Philosophy of the University of Malta, Faculty of Engineering, Octobre 1999.
- [41] 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.
- [42] L. Dinoi and S. Benedetto, « Design of prunable S-random interleavers, » in Proc. International Symposium on Turbo codes, Brest, pp. 279–282, France, 2003.
- [43] M. Ferrari, F. Scalise, and S. Bellini, « Prunable S-random interleavers, » in Proc. IEEE Intl. Commun. Conference (ICC), vol. 3, pp.1711 –1715 , 2002 .
- [44] H. R. Sadjadpour, N. J. A. Sloane, M. Salehi and G. Nebe, « Interleaver design for turbo codes, » IEEE Journal on Selected Areas in Communications 19(5) : pp. 831-837, 2001.
- [45] N. Jaoua, *Estimation Bayésienne non Paramétrique de Systèmes Dynamiques en présence de Bruit Alpha-stables*, Thèse doctorat, Laboratoire d'Automatique, Génie Informatique et Signal UMR CNRS 8219, Université de Lille, France, 06 Juin 2013.
- [46] T.Y. Al-Naffouri, A. A. Quadeer and G. Caire, « Impulse Noise Estimation and Removal for OFDM Systems , » IEEE Trans. on communications, vol. 62, no. 3, pp. 976-989, March 2014.
- [47] W. Dan, « Improvement for LDPC Coded OFDM Communication System over Power Line, » Master of Science Thesis performed at the Radio Communication Systems Group, KTH. Stockholm, Sweden, May 2013.
- [48] M. Sumi and M. Prasanth, « Periodic impulsive noise reduction in OFDM based power line communication, » International Journal of Research in Engineering and Technology (IJRET), vol. 03, no. 05, pp. 517-522, May 2014.
- [49] M. R. Ahadiat, P. Azmi and A. Haghbin, « Impulsive noise estimation and suppression in OFDM systems over in-home power line channels, » International Journal of Communication Systems. Published online in Wiley online Library : 9 July 2014.
- [50] J.H. Miller and J.B. Thomas, « Detectors for discrete-time signals in non-Gaussian noise, » IEEE Transactions of Information Theory, IT-18(2), pp. 241-250, 1972.
- [51] K. Kokkinakis, A. K. Nandi, « Exponent parameter estimation for a generalized Gaussian probability density functions with application to speech modeling, » Signal Processing, 85, pp. 1852-1858, 2005.
- [52] R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey and D.E. Knuth, « On the Lambert W function, » Advances in Computational Mathematics, 5, pp. 329-359, 1996.

- [53] M. Nardon and P. Pianca, « Simulation techniques for generalized Gaussian densities, » Working Papers 145, Department of Applied Mathematics, Università Ca' Foscari Venezia, 2006.
- [54] J.A. Domínguez-Molina, G. González-Farías, « Sobre la estimación del parámetro de forma de la distribución gaussiana generalizada, » Aportaciones Matemáticas. Serie comunicaciones, 30, pp. 117-120, 2002.
- [55] A. Saha Aditya and G. V. Anand, « Design of detectors based on stochastic resonance, » Signal Processing, 83: pp. 1193-1212, 2003.
- [56] G. Samorodnitsky and M. S. Taqqu, *Stable Non-Gaussian Random Processes*, Chapman & Hall, New York, 1994.
- [57] C. Moler, *Numerical Computing with MATLAB*, MathWorks, www.mathworks.com/moler, USA, February 15, 2008.
- [58] L. R. Burden and J. D. Faires, *Numerical Analysis*, Ninth Edition Youngstown State University, USA, 2010.
- [59] F. Chapeau-Blondeau, A. Monir, « Numerical evaluation of the Lambert W function and application to generation of generalized Gaussian noise with exponent 1/2, » IEEE Transactions on Signal Processing, 50(9), pp. 2160-2165, 2002.
- [60] F. N. Fritsch, R. E. Shafer, and W. P. Crowley, « Algorithm 443 : Solution of the transcendental equation $we^w = x$, » Communications of the ACM, 16:123-124, 1973.
- [61] D. A. Barry , P. J. Culligan-Hensley and S. J. Barry, « Real values of the W-function, » ACM Transactions on Mathematical Software, 21: pp. 161-171, 1995.
- [62] P. J. Nolan, « Stable Distributions : Models for Heavy Tailed Data, », American University, Processed July 28, 2014.
- [63] W. Feller, *An introduction to probability theory and its applications*, vol. II. John Wiley & Sons Inc., 2 éd, 1971.
- [64] V.V. Uchaikin, V.M. Zolotarev, *Chance and Stability. Stable Distributions and their Applications*, Ser. Modern Probability and Statistics, Netherlands, Utrecht, VSP, Moscow 1999.
- [65] V.M. Zolotarev, « One-dimensional stable distributions, » volume 65 of Translations of Mathematical Monographs. American Mathematical Society, Providence, 1986.
- [66] A. Janicki and A. Weron, « Can one see α –Stable Variables and Processes, » Statistical Science, vol. 9, no. 1, pp. 109-126, February 1994.
- [67] M. Laurence, *Lois α –stables*, d'après Nikias et Shao, 4 mars 2010.

- [68] P. Čížek, W. Härdle and R. Weron, « Statistical Tools for Finance and Insurance, » Springer Science & Business Media, 2005.
- [69] B. Mandelbrot , « 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. pp. 3968–3970.
- [70] R. D. Christopher and E. E. Kuruoğlu, « Estimation of the Parameters of Skewed α –Stable Distributions, » Conference in « Applications of Heavy Tailed Distributions in Economics, Engineering and Statistics, American University, Washington, DC, 3-5 June 1999.
- [71] R. A. Gonzalo, *Nonlinear Signal Processing : A Statistical Approach*, A John Wiley & Sons, Inc. Publication, November 2004.
- [72] C. J. Chambers, C. L. Mallows and B. W. Stuck, « A method for simulating stable random variables, » Journal of the American Statistical Association, vol. 71, n 354, pp. 340–344, 1976.
- [73] 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, January 2011.
- [74] C. Berrou and A. Glavieux, « Near optimum error correcting coding and decoding : turbo-codes, » IEEE Trans. Commun., vol. 44, no. 10, pp. 1261–1271, 1996.
- [75] C. E. Shannon, « A mathematical theory of communication, » Bell Systems Technical Journal, vol. 27, pp. 379-423, pp. 623-56, 1948.
- [76] J. Christophe, P. Adde and C. Leroux, « Architecture de turbo décodeur en blocs entièrement parallèle pour la transmission de données au-delà du Gbit/s », Annales des Télécommunications, 62 (1-2), pp.214-239, 2007
- [77] A. Kazem, *Particules déterministes généralisées en filtrage non-linéaire : Applications défense et télécommunications*, Thèse de Doctorat, l'université Toulouse III - Paul Sabatier, France, 17 septembre 2008.
- [78] L. A. Perișoară and R. Stoian, « The Decision Reliability of MAP, Log-MAP, Max-Log-MAP and SOVA Algorithms for Turbo Codes, », International Journal of Communications, Issue 1, vol. 2, 2008.
- [79] M. Et-Tolba, *Transmission de données pour le HSDPA et étude des algorithmes HARQ*, Thèse de Doctorat, ENST de Bretagne, France, 20 Décembre 2006.
- [80] J. C. Marc, P. C. Fossorier, S. Lin and C. Xu, « Bi-Directional SOVA Decoding for Turbo-Codes, » IEEE Communications letters, vol. 4, no. 12, December 2000.
- [81] F. H. Huang, *Evaluation of Soft Output Decoding for Turbo Codes*, Master thesis, Blacksburg, Virginia,

May 29, 1997

- [82] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*, Second Edition, John Wiley & Sons, Ltd, England, 2006.
- [83] F. J. Gravetter and L. B. Wallnau, *Statistics for the Behavioral Sciences*, 9th Edition, Wadsworth, Cengage Learning, USA, 2013.
- [84] R. Fano, « A heuristic discussion of probabilistic decoding, » *Information Theory, IEEE Transactions on*, vol. 9, no. 2, pp. 64–74, 1963.
- [85] A. J. Viterbi, « Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, » *IEEE Transactions on Information Theory*, vol. IT-13, April, 1967, pp. 260–269.
- [86] GD. Forney, « The Viterbi Algorithm, » *Proc. of the. IEEE*, vol. 61, no. 3, pp. 268–278, Mars 1973.
- [87] J. Hagenauer and P. Hoeher, « A viterbi algorithm with soft-decision outputs and its applications, » in *Global Telecommunications Conference, 1989, and Exhibition. Communications Technology for the 1990s and Beyond, GLOBECOM 89.*, IEEE, vol.3., pp. 1680–1686, 1989
- [88] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, « Optimal decoding of linear codes for minimizing symbol error rate (corresp.), » *Information Theory, IEEE Transactions on*, vol. 20, no. 2, pp. 284–287, 1974.
- [89] J. G. Proakis, M. Salehi, *Digital Communications*, 5th edn. McGraw Hill, New York, 2008.
- [90] A. Chemsas, K. Ghrissi, « Etude et évaluation des systèmes de communication numérique utilisant la modulation par codes en Treillis TCM », *Projet de Fin d'Etudes*, Ecole Nationale Polytechnique d'Alger, Octobre 1997.
- [91] A. F. Mondragón-Torres, K. R. Narayanan and E. Sánchez-Sinencio, « Floating Gate Analog Implementation of the Additive Soft-Input Soft-Output Decoding Algorithm, » supported in part by Fulbright-CONACYT and Texas Instruments. July 5, 2001.
- [92] F. 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.
- [93] M. Borda, *Fundamentals in Information Theory and Coding*, Springer -Verlag Berlin Heidelberg, 2011.
- [94] D. Bera, T. Chakravarty and S. Chakrabarti, « Reliable Wireless Communication for Medical Devices Using Turbo Convolution Code, » *Int. J. Communications, Network and System Sciences*, vol. 3, pp. 703–710, August 2010.

- [95] S. S. Pietrobon, « Implementation and performance of a turbo/map decoder », 1st Int. J. Satellite Commun., February 1997.
- [96] C. Berrou, A. Glavieux and P. Thitimajshima, « Near Shannon limit error-correcting coding and decoding : turbo-codes, » Proceedings of ICC'93, Genève, pp. 1064-1070, Mai 1993.
- [97] J. G. Proakis, M. Salehi and G. Bauch, *Contemporary Communication Systems Using MATLAB*, Cengage Learning, Stamford USA, 2013.
- [98] W. E. Ryan and S. Lin, *Channel Codes : Classical and Modern*, Cambridge University Press 2009.
- [99] A. Neubauer, J. Freudenberger and V. Kühn, *Coding Theory : Algorithms, Architectures, and Applications*, John Wiley & Sons Ltd. 2007.
- [100] J. P. Woodard and L. Hanzo, « Comparative study of turbo decoding techniques : An overview, » IEEE Transactions on Vehicular Technology, 49(6) : pp. 2208-2233, November 2000.
- [101] F. J. MacWilliams and N. J. A. Sloane, « Pseudo-Random Sequences and Arrays, » Proceedings of the IEEE, vol. IT-64, pp. 1715-1728, December 1989.
- [102] J. C. B. 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.
- [103] A. M. Lakhdar, R. Méliani, S. Azzedine, M. Kandouci, « Etude de l'effet de l'entrelacement sur les turbo codes, » Colloque National sur le Traitement du signal et ses Applications CNTSA'2005, Université 08 mai 45, Guelma, 19-20 Septembre 2005.
- [104] Small World Communications, *Iterative Decoding of Parallel Concatenated Convolutional Codes*, Application Note (Version 1.4), 13 January 1999.
- [105] B. J. T. Morgan, *Elements of Simulation*, Chapman & Hall/CRC Texts in Statistical Science, Octobre 1984.
- [106] V. T. Vakili, D. A. Moghaddam and K. Keyghobad, « Secure Turbo Codes in Impulsive Noise Channel, » Telecommunications in Modern Satellite, Cable and Broadcasting Services, 2007. TELSIS, Serbia, Nis, pp. 135-138, 8th International Conference on, 26-28 September 2007.
- [107] J. Vogt and A. Finger, « Improving the max-log-map turbo decoder, », Electronics Letters, vol. 36, no. 23, pp.1937-1939, 2000.
- [108] P. Ould-Cheikh-Mouhamedou, Y. Guinand and P. Kabal, « Enhanced max-log-app and Enhanced log-app decoding for dvb-rcs, », in 3rd International Symposium on Turbo codes, 2003.
- [109] 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.
- [110] S. Papaharalabos, P. Sweeney, B. G. Evans, « A new method of improving SOVA turbo decoding for AWGN, Rayleigh and Rician fading channels, » IEEE 59th Vehicular Technology Conf. 2004. VTC 2004- Spring., vol. 5, pp. 2862-2866, 17-19 May 2004.
- [111] M. G. Kendall, and A. Stuart, A., *The Advanced Theory of Statistics, Volume 1: Distribution Theory*, (Chapter 3), 3rd Edition, Griffin, 1969.
- [112] 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, pp. 2067-2071, May 2009.
- [113] A. Chemsas, D. Saigaa and A. Taleb-Ahmed, « Nouvelle approche pour optimiser un décodeur itératif », Communication dans le colloque CORESA 2012, 15^{ème} édition, Lille, 24 et 25 Mai 2012.
- [114] H. Wymeersch, *Iterative Receiver Design*, Cambridge University Press, New York, 2007.
- [115] C. B. Schlegel, L. C. Pérez, *Trellis and Turbo Coding*, John Wiley & Sons, 2004.
- [116] S. Chaoui, H. Ben Cheikha, « Etude et résultats du comportement de convergence des codes turbo systématiques et partiellement systématiques, » 4th International Conference SETIT, Tunisie 2007.
- [117] X. M. Zhang, Y. Q. Shi, H. Chen, A. Haimovich, A. Vetro, and H. Sun, « Successive packing based interleaver design for turbo codes, » in Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium. vol n° 1, pp. 17-20, 2002.
- [118] L. Kocarev, F. Lehmann, G. Maggio, B. Scanavino, Z. Tasev, and A. Vardy, « Non linear dynamics of iterative decoding systems : analysis and applications, » Information Theory, IEEE Transactions on, vol. 52, no. 4, pp.1366-1384, 2006.
- [119] L. A. Perisoara, R. Stoian, « The Decision Reliability of MAP, Log-MAP, Max-Log-MAP and SOVA Algorithms for Turbo Codes, » International Journal of Communications, Issue 1, vol. 2, pp. 65–74, 2008.
- [120] P. U. Chuan Hsian, « Iterative SOVA decoding over symmetric alpha stable channels », Journal of Engineering Science and Technology, vol. 7, no. 3, pp. 360 – 378, School of Engineering, Taylor's University, 2012.
- [121] T. Majoul, F. Raouafi, and M. Jaïdane, « Semi-blind turbo decoding in impulsive noise channels, » IEEE , ISCCSP 2008, Malta, pp. 810-813, 12-14 March 2008.
- [122] A. Chemsas, D. Saigaa, H. Ghodbane and A. Taleb-Ahmed, « Novel semi-blind estimation for

- turbo decoding in impulsive noise channel, », *International Journal of System Assurance Engineering and Management*, Springer, 22 Jan 2015.
- [123] L. Xifeng and X. Yongle, « State Estimation Based On Generalized Gaussian Distributions, » *Metrol. Meas. Syst.*, vol. XX, no. 1, pp. 65–76, 2013.
- [124] G. Mallat, « A theory for multiresolution signal decomposition: the wavelet representation, » *IEEE Trans. Pattern Anal. Machine Intel.*, 7(11), pp. 674–693, 1989.
- [125] Y. U. Shaoquan, A. Zhang and L. I. Hongwei, « A Review of Estimating the Shape Parameter of Generalized Gaussian Distribution, » *Journal of Computational Information Systems* 8: 21, pp. 9055–9064, 2012.
- [126] A.A. Roenko, V. V. Lukin, I. Djurovic and M. Simeunovic, « Estimation of parameters for generalized Gaussian distribution, » *Proc. IEEE. Int. Symp. Conf. on Communications, Control and Signal Processing (ISCCSP)*, Athens, pp. 376–379, 21–23 May 2014.
- [127] G. F. Graciela, D. M. J. Armando and M. R. D. Ramón, « A practical procedure to estimate the shape parameter in the generalized Gaussian distribution, » *Technique report I-01-18_eng. pdf*, available through [http://www.cimat.mx/reportes/enlinea/I-01-18_eng. Pdf](http://www.cimat.mx/reportes/enlinea/I-01-18_eng.Pdf), vol. 1, 2003.
- [128] T. Der-Feng T. Tsung-Ru and Y. S. Han, « Robust turbo decoding in impulse noise channels, » *IEEE , Proc. IEEE Int. Symp. Conf. on Power Line Communications and Its Applications (ISPLC)*, Johannesburg, pp. 230 – 235, 24–27 March 2013.
- [129] T.A. Summers and S.G. Wilson, « SNR Mismatch and Online Estimation in Turbo Decoding, » *IEEE Trans. on Communications*, vol. 46, no. 4, pp. 421–423, April 1998.
- [130] H. Xiaoling and P. Nam, « Turbo Decoders Which Adapt to Noise Distribution Mismatch, » *IEEE Communications Letters*, , vol. 2, no. 12, pp. 321–323, 1998.
- [131] C. S. Regazzoni, C. Sacchi, A. Teschioni and S. Giulini, « Higher Order Statistics based sharpness evaluation of a generalized Gaussian PDF model in impulsive noisy environments, » *Proc. of 9th IEEE Workshop on Statistical Signal and Array Processing*, pp. 411–414, 1998.
- [132] A. Walter, *Mathematics for Physics and Physicists*, Published by Princeton University Press, 41 William Street, Princeton, New Jersey , USA, 2007.
- [133] K. Sharifi and A. Leon-Garcia, « Estimation of shape parameter for generalized Gaussian distributions in subband decompositions of video, » *IEEE Trans. on Circuits and Syst. for Video Technology*, vol. 5, no. 1, pp. 52–56, February 1995.
- [134] M. K. Varanasi and B. Aazhang, « Parametric generalized Gaussian density estimation, » *J. Acoust. Soc. Amer.*, vol. 86, pp. 1404–1415, 1989.

- [135] W. Taiyue, L. Xiusheng, D. Yanqing, and L. Hongwei, « Locally optimum detection of a Noise model based on generalized gaussian distribution, » in Proc. of IEEE Int. Conf. on MultiMedia and Information Technology, pp. 253-256, 2008.
- [136] G. F. Graciela, D. M. J. Armando and M. R. D Ramón, « Efficiency of the approximated shape parameter estimator in the generalized gaussian distribution, » IEEE Transactions on Vehicular Technology, vol.58, no. 8, pp. 4214-4223, October 2009.
- [137] E. F. Fama and R. Roll, « Some Properties of Symmetric Stable Distributions, » Journal of the American Statistical Association, vol. 63, no. 323 pp. 817-836, Septembre 1968.
- [138] J. H. McCulloch, « Simple consistent estimators of stable distribution parameters, » Communications on Statistics - Simulation, 15(4) : pp. 1109–1136, 1986.
- [139] I. A. Koutrouvelis, « Regression-type Estimation of the Parameters of Stable Laws, » Journal of the American Statistical Association, 75 (372), pp. 918-928. 1980.
- [140] S. Kogon and D. Williams, « On the characterization of impulsive noise with alpha-stable distributions using fourier techniques, » In Proceedings of the 29th Asilomar Conference of Signals, Systems and Computing, 1995.
- [141] X. Ma and C. L. Nikias, « Parameter estimation and blind channel identification in impulsive signal enviroments, » IEEE Transactions on Signal Processing, 43(12) : pp. 2884 – 2897, December 1995.
- [142] G. A. Tsihrintzis and C. L. Nikias, « Fast estimation of the parameters of alpha-stable impulsive interference, » IEEE Transactions on Signal Processing, 44(6) : pp. 1492–1503, June 1996.
- [143] S. Bates and S. McLaughlin, « The estimation of stable distribution parameters from teletraffic data, » IEEE Trans. Signal Process. 48, 3, pp. 865 – 870. March 2000.
- [144] S. Chakraborty et G. Bhattacharya , *Proceedings of the International Symposium on Engineering under Uncertainty : Safety Assessment and Management (ISEUSAM - 2012)*, Springer India 2013.
- [145] X. Wang, K. Li, P. Gao and S. Meng « Research on Parameter Estimation Methods for Alpha Stable Noise in a Laser Gyroscope's Random Error », Sensors, vol. 15, pp. 18550-18564, 2015.
- [146] P. Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer-Verlag, New York, 2004.
- [147] P. Bogaert, *Probabilités pour scientifiques et ingénieurs : Introduction au calcul des probabilités*, De Boeck & Larcier s.a., 2006.