

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة سعيدة – د. الطاهر مولاي –

Université Saïda – Dr. Tahar Moulay –
Faculté de Math, Informatique et Télécommunication



MEMOIRE

Présenté pour l'obtention du **Diplôme** de **MASTER** en Télécommunications

Spécialité : Systèmes de Télécommunications

Par : Mlle. GUERROUDJ Hiba

Mlle. KHELFALLAH Bouchra

Critères d'arrêt Turbo à base des quantités souples (Soft)

Soutenue, le 17/06/ 2025, devant le jury composé de :

Mr. MANSOURI Boualem	PR	Univ.Saïda Président
Mr. OUARDI Aïssa	MCA	Univ.Saïda Rapporteur
Mr. TAMI Abdelkader	MCA	Univ.Saïda Examineur

2024 / 2025

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

إِنْ أُرِيدُ إِلَّا الْإِصْلَاحَ مَا اسْتَطَعْتُ
وَمَا تَوْفِيقِي إِلَّا بِاللَّهِ
عَلَيْهِ تَوَكَّلْتُ وَإِلَيْهِ أُنِيبُ

سورة هود - 88-

إن الحمد لله نحمده سبحانه وتعالى حمدا يليق بجلال وجهه وعظيم سلطانه ،

فقد سد الخطي وشرح الصدر ويسر الأمر ،

فله الحمد كله وإليه يرجع الفضل كله والصلاة والسلام على أشرف المرسلين سيدنا محمد ﷺ

Remerciements

Bismillah ar-Rahman ar-Rahim

Louange à Allah, Le Très Miséricordieux, qui nous a accordé la force, la patience et la persévérance pour mener à bien ce travail.

*On tient à exprimer notre profonde gratitude à notre encadreur **Mr. Aissa Ouardi** pour ses conseils précieux, son accompagnement constant et sa bienveillance tout au long de la réalisation de ce mémoire.*

Nos remerciements s'adressent également à l'ensemble du corps enseignant du département de Télécommunication pour la qualité de l'enseignement dispensé durant mon parcours universitaire.

*Un grand merci à la famille **Guerroudj**, ainsi qu'à la famille **Khelfallah**, pour leurs amours, leurs prières et leurs soutiens indéfectibles tout au long de nos études.*

On n'oublie pas nos amis et camarades de promotion, pour les bons moments partagés, l'entraide et les encouragements sincères.

*À tous ceux qui, de près ou de loin, ont contribué à la réussite de ce travail, On dit : **merci du fond du cœur.***

وَكَانَ فَضْلُ اللَّهِ عَلَيْكَ عَظِيمًا

Dédicace

Louange à Allah, Le Très Haut, source de toute lumière et force,

Je dédie ce travail avant tout à Allah Subhanahu wa Taala, qui m'a guidée et soutenue dans chaque étape de ce parcours.

*À mon père "**Mohamed**", pilier de ma vie, pour son amour inconditionnel et ses sacrifices.*

À ma mère, pour sa tendresse, ses prières et son soutien constant.

*À mes frères "**Abdelhak, Mahmoud et Abdelwaheb**", pour leurs présences et encouragements qui m'ont toujours motivée.*

*À ma binôme et chère amie "**Bouchra**", pour sa collaboration précieuse et sa solidarité tout au long de cette aventure.*

À tous ceux qui ont étudié avec moi, partageant moments d'efforts et de réussite.

Et bien sûr, un grand merci à toute ma famille, surtout mes oncles et mes tantes, pour leur soutien, leur amour et l'énergie positive qu'ils m'ont toujours apportés.

Que ce modeste travail soit le reflet de tout l'amour et le soutien que vous m'avez donnés.

Hiba



وَكَانَ فَضْلُ اللَّهِ عَلَيْكُمْ عَظِيمًا

Dédicace

Au nom de Dieu, le clément, le tout-puissant, le très miséricordieux, je dédie ce travail à :

A mon père Nacer qui m'a éclairé mon chemin

A ma mère Karima, la source de mes efforts, à celle qui mérite toute ma reconnaissance pour ses sacrifices

A à mes frères Abd el Karim, Oussama

A mes sœurs Amina, Rimasse

A mes tantes Kheira, Nawel

A mes cousins et cousines Rayen, Anes, Mohamed, Ferial, Sirine, Yara

A Mes chères copines, Lamia et Hadjer. Au nom de l'amitié qui nous réunit, et au nom de nos souvenirs inoubliables

À ma précieuse binôme Hiba,

pour son engagement, son soutien et sa belle

énergie tout au long de ce parcours.

Bouchra



Résumé

Ce mémoire s'intéresse à l'étude et l'évaluation des critères d'arrêt dans le décodage Turbo, en se focalisant principalement sur les critères basés sur les informations souples (Soft information). L'objectif est d'améliorer l'efficacité du décodage en réduisant le nombre d'itérations sans dégrader les performances en termes de taux d'erreur binaire (BER). Plusieurs critères sont étudiés, notamment ceux basés sur l'entropie croisée (EC) et la moyenne des LLR (Log-Likelihood Ratios). Des simulations en MATLAB ont été réalisées pour évaluer l'impact de chaque critère sur le compromis complexité/performance. Les résultats obtenus montrent que l'utilisation de ces critères permet une réduction significative du nombre moyen d'itérations, tout en maintenant une performance de décodage proche de celle obtenue avec un nombre fixe d'itérations.

Mots clés: Décodage Turbo, Critère d'arrêt, Informations souples, Entropie croisée, LLR, Performances BER.

Abstract

This dissertation focuses on the study and evaluation of stopping criteria in Turbo decoding, with a particular emphasis on soft-information-based techniques. The main goal is to enhance decoding efficiency by reducing the number of iterations without degrading the bit error rate (BER) performance. Several stopping criteria are explored, including those based on cross-entropy and the mean of the log-likelihood ratios (LLRs). MATLAB simulations were conducted to assess the impact of each criterion on the complexity-performance trade-off. The results demonstrate that using these soft-based stopping criteria leads to a significant reduction in the average number of iterations, while maintaining decoding performance close to that of fixed-iteration decoding.

Keywords: Turbo decoding, stopping criterion, Soft information, Cross-entropy, LLR, BER performance

ملخص

يهدف هذا البحث إلى دراسة وتقييم معايير الإيقاف في عملية فك الترميز التوربيني، مع التركيز على المعايير المعتمدة على المعلومات الناعمة (Soft Information). يتمثل الهدف الرئيسي في تقليص عدد التكرارات اللازمة خلال عملية فك الترميز ، دون التأثير سلباً على أداء النظام من حيث معدل الخطأ في البتات (BER). تم في هذا العمل دراسة عدة معايير، من بينها معيار الإنتروبيا المتقاطعة (Cross-Entropy) ومتوسط نسب الاحتمال اللوغاريتمي (LLR). وقد تم تنفيذ محاكاة باستخدام برنامج MATLAB لتقييم أداء هذه المعايير ومقارنتها من حيث التوازن بين تعقيد الحسابات وجودة الأداء. أظهرت النتائج أنّ اعتماد هذه المعايير يساهم في تقليل عدد التكرارات بشكل ملحوظ، مع الحفاظ على أداء قريب من الأداء المحقق عند استخدام عدد تكرارات ثابت.

الكلمات الدالة: فك الترميز التوربيني، معيار الإيقاف، المعلومات الناعمة ، الإنتروبيا المتقاطعة، نسب الاحتمال اللوغاريتمي، أداء BER

Table des Matières

Liste des acronymes et abréviation

Liste des figures

Liste des tableaux

Introduction Générale.....1

Chapitre I : Turbocodes et algorithme MAP

I. Les codes convolutifs3

I.1 Historique3

I.1.1 Définition3

I.1.2 Codes récursifs4

I.2 Turbo codage.....5

I.2.1 Introduction5

I.2.2 Le principe de base du turbo codage.....5

I.3 Schéma de turbo codage6

I.4 Canal gaussien : Cas général6

I.5 Canal Gaussien idéal à bande infinie.....7

I.6 Capacité d'un canal gaussien continu non contraint à bande limitée
.....8

I.6.1 Capacité d'un canal gaussien à entrée binaire AWGN.....10

I.7 Méthode de Décodage MAP de Code Convolutif.....12

I.7.1 Algorithme MAP.....13

I.7.2 Algorithme Max-Log-MAP.....15

I.7.3 Algorithme Log-MAP.....16

I.8 Le Turbodécodage MAP des Turbo codes	17
--	----

Chapitre II : Critères d'arrêt Turbo à base des décisions dures

II.Introduction.....	20
II.1 Critère CRC.....	20
II.2 Critère SCR.....	21
II.3 Critère SDR.....	23
II.4 Critère HDA.....	24
II.5 Critère IHDA.....	26
II.6 Critère GENIE.....	27

Chapitre III : Critères d'arrêt Turbo à base des quantités souples

III.Introduction.....	30
III.1 Critère de l'Entropie Croisée CE (Cross-Entropy).....	30
III.2 Critère de la moyenne des LLR.....	32
III.3 Critère Minimum LLR.....	33
III.4 Critère de la variance des LLR.....	33
III.5 Critère Sum-Alpha.....	35

Chapitre IV : Résultats de simulation

IV.1 Introduction	38
IV.2 Performances du critère d'arrêt de l'Entropie Croisée EC appliqué au turbo décodage	38
IV.2.1 Conditions de simulation.....	38
IV.2.2 Résultats de simulation.....	39
IV.2.3 Interprétations des résultats	40

IV.3 Performances du critère de la moyenne des LLR appliqué au turbo décodage	40
IV.3.1 Conditions de simulation	40
IV.3.2 Résultats de simulation.....	41
IV.4 Justification du choix du seuil du critère de la moyenne des LLR	42
IV.4.1 Conditions de simulation.....	42
IV.4.2 Résultats de simulation.....	42
IV.5 Performances du critère Minimum des LLR.....	44
IV.5.1 Conditions de simulation.....	44
IV.5.2 Résultats de simulation.....	45
IV.6 Justification du choix du seuil du critère du Minimum des LLR	46
IV.6.1 Conditions de simulation.....	46
IV.6.2 Résultats de simulation.....	46
IV.7 Performances du critère de la variance des LLR.....	48
IV.7.1 Conditions de simulation.....	48
IV.7.2 Résultats de simulation.....	48
IV.8 commentaires et comparaison.....	50
Conclusion Générale	51
Bibliographie.....	53

Liste des acronymes et abréviations

BBGA : AWGN bruit blanc gaussien additif

BER: Bit Error Rate

CRC: Cyclic Redundancy Check

EC : Cross Entropy

FEC: Forward Error Correction

FER: frame Error Rate

HDA :Hard-Decision-Aided

Log-MAP : Logarithmic Maximum A posteriori Probability

LLR : Logarithmic Likelihood Ratio

MAP : Maximum A Posteriori

NRNSC : Non Recursive Non Systematic Code

RSB : Rapport signal/bruit

RSC :Recursive Systematic Code

SCR :Sign-Change-Ratio

SCRM :Sign-Change-Ratio Modified

SDR : Sign-Difference-Ratio

SNR: Signal to Noise Ratio

SOVA :Soft-Output Viterbi Algorithm

TEB : Taux d'erreur binaire

3G-WCDMA : Wideband Code Division Multiple Access

Wimax: Worldwide interoperability for Microwave Access

Liste des figures

Figure I.1 : Exemple de codeur convolutif.....	3
Figure I.2 : Exemple d'un code convolutif.....	4
Figure I.3 : Structure de codage turbo.....	6
Figure I.4 : Modèle du canal à Bruit Blanc Additif Gaussien	7
Figure I.5 : Le canal à bruit additif blanc gaussien AWGN.....	8
Figure I.6 : Graphes des valeurs des limites de Shannon pour quelques rendements du codeur et pour les deux types de décodage Soft et Hard.....	12
Figure I.7 : Schéma de décodage d'un turbo code (Turbo décodeur).....	17
Figure II.1 : Schéma de la critère CRC.....	21
Figure II.2 : Critère d'arrêt Sign change Ratio.....	22
Figure II.3 : Courbes de performance de critère SCR, $TEB = f\left(\frac{E_b}{N_0}\right)_{dB}$	23
Figure II.4 : Critère d'arrêt SDR Sign Difference Ratio.....	24
Figure II.5 : Courbes de performance de critère SDR, $TEB = f\left(\frac{E_b}{N_0}\right)_{dB}$	24
Figure II.6 : Critère d'arrêt HDA (Hard Decision Aided).....	25
Figure II.7 : Courbes de performance de critère HDA, $TEB = f\left(\frac{E_b}{N_0}\right)_{dB}$	26
Figure II.8 : Schéma de la critère GÉNIE.....	28
Figure II.9 : Performances d'un critère GENIE appliqué au turbo décodage.....	28
Figure II.10 : Nombre moyen des itérations du critère GENIE appliqué au turbo décodage.....	29
Figure III.1 : La variance des informations extrinsèques par rapport au nombre d'itérations	34
Figure III.2 : La variance de l'information extrinsèque par rapport au BER.....	35
Figure IV.1 : Performances du critère de l'entropie croisée EC.....	39
Figure IV.2 : Nombre moyen des itérations du critère de l'entropie croisée EC	39
Figure IV.3 : Performances du critère de la moyenne des LLR.....	41

Figure IV.4 : Nombre moyen des itérations du critère de la moyenne des LLR	41
Figure IV.5 : Distribution de la moyenne des LLR à la 3 ème itération.....	42
Figure IV.6 : Distribution de la moyenne des LLR à la 4 ème itération.....	43
Figure IV.7 : Distribution de la moyenne des LLR à la 5 ème itération.....	43
Figure IV.8 : Distribution de la moyenne des LLR à la 6 ème itération.....	44
Figure IV.9 : Performances du critère Minimum des LLR.....	45
Figure IV.10 : Nombre moyen des itérations du critère Minimum des LLR.....	45
Figure IV.11 : Distribution du minimum des LLR (Seuil) à la 4 ème itération.....	46
Figure IV.12 : Distribution du minimum des LLR (Seuil) à la 5 ème itération.....	47
Figure IV.13 : Distribution du minimum des LLR (Seuil) à la 6 ème itération.....	47
Figure IV.14 : Performances d'un critère de variance des LLR.....	48
Figure IV.15 : Nombre moyen des itérations d'un critère de la variance des LLR.....	49

Liste des Tableaux

Tableau I.1 : Limites de Shannon pour quelques rendement du codeur et pour les deux types de décodage Soft et Hard.....	11
Tableau IV.1 : Comparaison des performances obtenues au rapport signal sur bruit 0.8 dB.....	50

INTRODUCTION GENERALE

INTRODUCTION GENERALE :

Les communications numériques occupent aujourd'hui une place essentielle dans notre quotidien. L'utilisation des décodeurs turbo, reconnus pour leur grande efficacité, permet d'atteindre des taux d'erreur extrêmement faibles. Le turbo décodeur est une méthode employée en transmission numérique pour améliorer la fiabilité des données transmises. Il repose sur l'association de deux ou plusieurs codes correcteurs d'erreurs fonctionnant en parallèle, ce qui lui confère des performances de décodage supérieures aux approches classiques. Cette technologie est largement adoptée dans les systèmes de communication modernes, notamment dans les réseaux mobiles et les communications par satellite, grâce à sa capacité à minimiser les erreurs de transmission et à améliorer la qualité de réception.

Les turbo codes ont été développés à l'École Nationale Supérieure des Télécommunications de Bretagne (département Électronique et département Signal et Communications). Leur efficacité en correction d'erreurs est telle qu'elle s'approche, dans des conditions optimales de codage, de la limite théorique établie par C. E. Shannon.

Le terme « critère d'arrêt turbo » désigne les conditions utilisées au niveau du récepteur pour stopper les itérations du processus de turbo décodage une fois que la trame reçue a été correctement décodée. Poursuivre le décodage au-delà de ce point serait inutile. Ces critères jouent un rôle clé dans la réduction de la complexité des calculs côté réception. En d'autres termes, ils permettent au décodeur de déterminer à quel moment le processus itératif de décodage peut être interrompu, en s'assurant que la séquence décodée est suffisamment proche de celle d'origine.

Dans le cadre du turbo décodage, l'utilisation des quantités souples (soft) joue un rôle essentiel dans l'amélioration des performances de correction d'erreurs. Contrairement aux décisions dures (hard), où les bits reçus sont directement interprétés comme 0 ou 1, les décisions souples exploitent des informations de fiabilité associées aux bits reçus. Ces informations, souvent représentées sous forme de valeurs logarithmiques de vraisemblance (LLR – Log-Likelihood Ratio), traduisent le degré de confiance du récepteur quant à la valeur binaire d'un bit donné. Par exemple, une valeur LLR élevée (positive ou négative) indique une forte certitude que le bit est respectivement 1 ou 0, tandis qu'une valeur proche de zéro reflète une incertitude élevée. Ces valeurs probabilistes permettent au décodeur d'affiner progressivement son estimation au fil des itérations, améliorant ainsi la précision du décodage. L'intégration des quantités souples dans

les critères d'arrêt vise à optimiser le compromis entre performance et complexité, en stoppant le processus de décodage dès que la convergence est atteinte, réduisant ainsi le temps de calcul et la consommation des ressources.

L'objectif principal de ce travail est d'évaluer les performances des critères d'arrêt turbo basés sur les décisions souples (soft) proposés par les chercheurs dans le domaine des turbo-communications.

Le manuscrit est organisé en quatre chapitres :

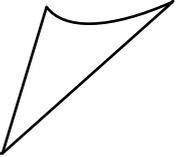
- **Le premier chapitre** est consacré aux codes convolutifs récurrents, au turbo codage, à la capacité des canaux Gaussien et binaire, au décodage MAP des codes convolutifs et au turbo décodage MAP des turbocodes convolutifs.
- **Le deuxième chapitre** présente les critères d'arrêt turbo tels que CRC, SDR, SCR, HDA, IHDA et GENIE, qui sont basés sur les décisions dures du turbo décodeur.
- **Le troisième chapitre** est dédié aux critères d'arrêt turbo basés sur les quantités souples, notamment le critère de l'entropie croisée (CE), la moyenne des LLR, le minimum LLR, la variance des LLR et le critère sum-alpha.
- **Le quatrième chapitre** expose les résultats de simulation.

Chapitre I

Turbocodes

et algorithme

MAP



I. Codes convolutifs

I.1 Historique

C'est en 1955 que Peter Elias introduit la notion de code convolutif. Un exemple de codeur convolutif est montré dans la figure 1.1. Il s'agit d'un codeur systématique, c'est-à-dire que le message codé contient le message information, auquel est ajouté de l'information redondante.[1]

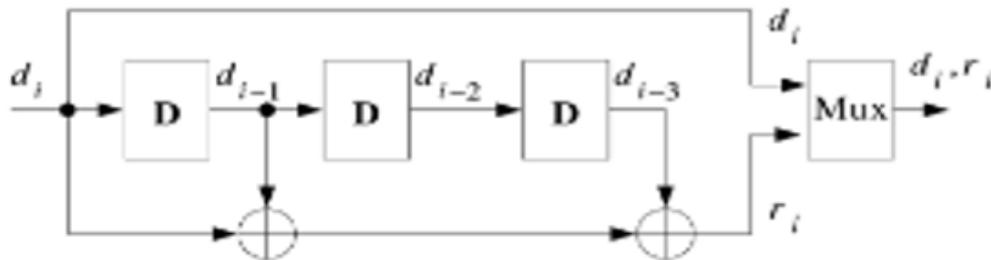


Figure I.1 : Exemple de codeur convolutif [1]

Les turbocodes sont parmi les meilleurs codes correcteurs d'erreurs utilisés dans les systèmes modernes. On va appliquer les turbocodes pour minimiser la probabilité d'erreur afin d'améliorer les performances du système à la réception du signal. [2]

I.1.1 Définition

Les principaux inconvénients du codage en bloc peuvent être contournés grâce à une approche différente de la problématique du codage : C'est notamment le cas du codage convolutif (*Convolutional Coding*). Ce type de codage introduit systématiquement de la redondance dans le message à mesure que les bits d'information sont fournis au codeur. Le message codé se construit ainsi de manière itérative, en s'appuyant sur un registre à décalage. Les codes convolutifs constituent une classe de codes correcteurs d'erreurs à la fois très souple et particulièrement efficace. Théoriquement, ils possèdent des caractéristiques similaires à celles des codes en blocs, à l'exception de leur dimension et de leur longueur (matrice génératrice). Les codes convolutifs s'appliquent en principe à des séquences infinies de symboles d'information et produisent, en retour, des séquences infinies de symboles codés. [3]

La figure I.2 montre un schéma synoptique d'un codeur convolutif non systématique. Les caractéristiques de ce codeur sont :

- Le nombre de bits d'information à l'entrée $n_e = 1$ « Code binaire ». Un code qui utilise $n_e = 2$ est appelé « Code Duo-binaire ».
- Le nombre de bits à la sortie $n_s = 2$

- Son rendement est donc $R = \frac{n_e}{n_s} = \frac{1}{2}$
- Sa longueur de contrainte $K=3$
- Les séquences génératrices sont : $[1\ 0\ 1, 1\ 1\ 1] = [5,7]_{\text{oct}}$.
- C'est un code non systématique.

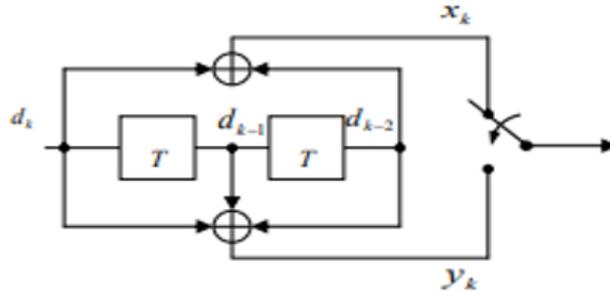


Figure I.2 : Exemple d'un code convolutif [3]

Un code convolutif est un circuit logique comprenant un registre à décalage formé de K cellules, n_s additionneurs modulo 2 (portes ou-exclusif), un ensemble de connexions entre les additionneurs (séquences génératrices), les cellules du registre à décalage et le commutateur à n_s positions (Conversion parallèle/série). [4]

I.1.2 Codes récurrents

La sortie d'un code convolutif ne dépend pas uniquement des bits d'entrée actuels, mais peut aussi être influencée par les sorties précédentes du codeur. Ce codeur est appelé code récurrent. En effet, dans le cas des codes convolutifs récurrents, une séquence d'entrée de longueur finie peut générer une séquence de sortie de longueur infinie, ce qui signifie que le codeur continue à produire des symboles codés même après la fin de la séquence d'entrée.

Cette propriété est importante car elle permet d'améliorer la performance du code en termes de correction d'erreurs surtout en faible rapport signal sur bruit, en augmentant la distance minimale entre les séquences codées. Par ailleurs, il est possible de transformer un code convolutif qui est à la fois non récurrent et non systématique, désigné sous l'acronyme NRNSC (*Non Recursive Non Systematic Code*), en un code convolutif récurrent et systématique, appelé RSC (*Recursive Systematic Code*). Cette transformation s'effectue par une opération mathématique consistant à diviser les polynômes utilisés dans le codeur par l'un de ces polynômes, ce qui modifie la structure du code pour le rendre récurrent et systématique, améliorant ainsi ses propriétés de correction.

Par exemple, on transforme ainsi un code convolutif de rendement $\frac{1}{2}$ défini par les polynômes $g_0(z)$ et $g_1(z)$, en un polynôme récuratif systématique défini par les fonctions rationnelles $[1, g_1(z)/g_0(z)]$.

Les diagrammes d'états d'un code non récuratif et du code récuratif systématique qui s'en déduit sont les mêmes. [5]

I.2 Turbo codage

I.2.1 Introduction

Le turbo codage est une technique avancée de codage de canal qui a été introduite au début des années 1990. Cette méthode repose sur l'utilisation combinée de deux codes convolutifs, placés en cascade, ce qui signifie que les données passent successivement à travers ces deux codeurs. Pour optimiser les performances, cette technique utilise une concaténation spécifique des codes ainsi qu'un procédé d'itération qui permet de répéter plusieurs fois le traitement des données codées.

Les turbocodes convolutifs, qui constituent la base de cette approche, ont été parmi les premiers types de codes à être employés dans ce cadre innovant. Ils ont rapidement démontré une supériorité significative en termes de performance, notamment dans la correction d'erreurs, par rapport aux codes en blocs traditionnels. Grâce à cette efficacité accrue, le turbo codage a révolutionné le domaine des communications numériques en améliorant la fiabilité des transmissions dans des canaux bruyants.

I.2.2 Principe de base du turbo codage

Le principe fondamental du turbo codage consiste à encoder les données à transmettre à l'aide de deux codes convolutifs distincts. Les bits de sortie générés par ces deux codeurs sont ensuite concaténés, c'est-à-dire assemblés l'un après l'autre, afin de former un flux unique de données codées. Ce flux est alors transmis sur le canal de communication.

Lors de la réception, un processus de décodage itératif est appliqué. Ce décodage utilise notamment l'algorithme de décodage MAP (*Maximum A Posteriori*), qui a pour objectif de minimiser la probabilité d'erreurs après décodage en déterminant la séquence de bits la plus probable à partir des données reçues. Le décodage itératif se caractérise par son exécution répétée : A chaque itération, il utilise les informations obtenues lors du décodage précédent et tient compte des éventuelles erreurs de transmission. Ce mécanisme d'échanges d'informations successifs permet d'affiner progressivement l'estimation des bits transmis.

Les résultats obtenus à chaque étape sont combinés pour produire une estimation finale plus précise de la séquence originale de bits, ce qui améliore considérablement la fiabilité de la transmission.

Les turbocodes convolutifs se révèlent particulièrement performants dans les environnements de transmission bruités, tels que les communications sans fil et les transmissions par satellite. Grâce à leur efficacité, ils ont été largement adoptés et intégrés dans de nombreuses normes de communication numérique modernes, notamment les réseaux 3G, 4G, ainsi que WiMax. [3]

I.3 Schéma de turbo codage

La structure d'un turbocodeur à taux de codage $1/3$ (sans poinçonnage) est présentée à la figure (Fig I.3). Cette structure consiste à transmettre le bit d'information systématique $c_0 = u$, le premier bit de redondance c_1 et le deuxième bit de redondance c_2 .

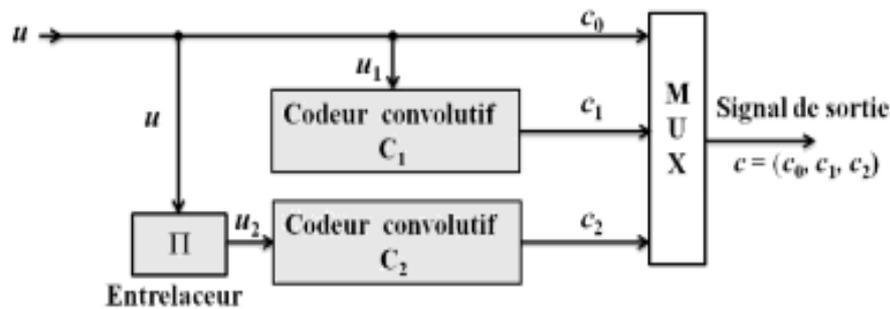


Figure I.3 : Structure de codage turbo [6]

$\Pi(t)$: Permutation ou entrelaceur de bits

Les bits d'information sont transmis trois fois sur le canal : une fois ($u \rightarrow c_0$) sans protection contre les erreurs et deux fois c_1 et c_2 qui sont des bits de redondance. Les codeurs C_1 et C_2 sont identiques $C_2 = C_1$. Le deuxième codeur C_2 reçoit en entrée une séquence d'information entrelacée par permutation pseudo aléatoire. Le multiplexeur combine alors les trois signaux en un seul. [6]

I.4 Canal gaussien : Cas général

Le bruit blanc gaussien est un processus aléatoire stationnaire, ce qui signifie que ses propriétés statistiques ne varient pas au cours du temps. Ce bruit est indépendant du signal transmis, ce qui le rend particulièrement adapté pour modéliser de façon simple et réaliste les perturbations affectant un signal durant sa transmission.

En effet, ce type de bruit représente de manière idéale l'ensemble des interférences et distorsions qui peuvent survenir aussi bien dans le canal de transmission que lors de la réception du signal. Grâce à cette modélisation, il est possible d'analyser et de concevoir des systèmes de communication capables de résister efficacement aux effets néfastes du bruit.

On considère le canal à bruit blanc additif gaussien BBAG (AWGN) comme le montre la figure I.4 et décrit par la relation :

$$r(t) = h(t) * s(t) + b(t) \quad (\text{I.1})$$

Sachant que :

$r(t)$ est le signal reçu, $s(t)$ est le signal transmis, $h(t)$ est la réponse impulsionnelle du canal, et $b(t)$ le bruit gaussien de moyenne nulle.

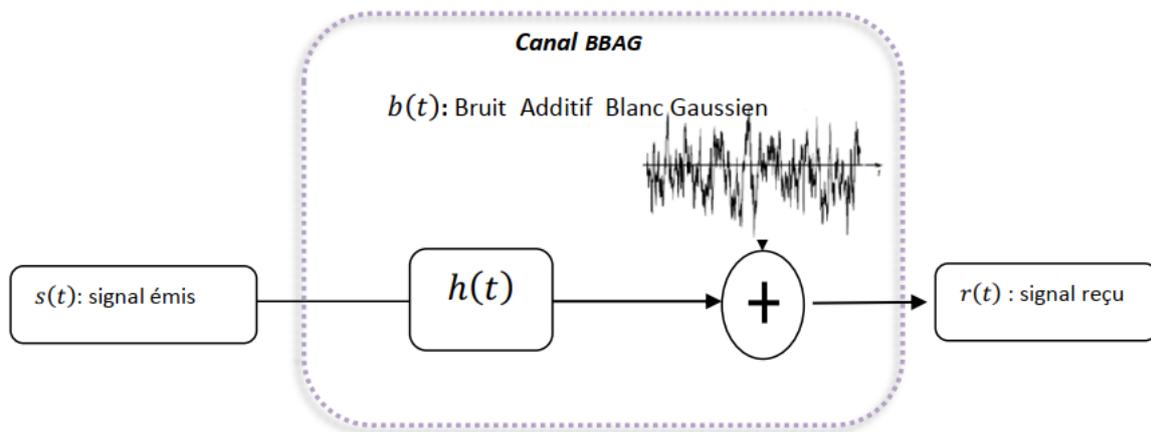


Figure I.4 : Modèle du canal à Bruit Blanc Additif Gaussien [7]

I.5 Canal Gaussien idéal à bande infinie :

Un canal de transmission avec bruit additif gaussien blanc, communément appelé AWGN (*Additive White Gaussian Noise*), est illustré sur la figure I.5. Ce canal est caractérisé par la superposition d'un bruit gaussien blanc au signal transmis, ce qui signifie que le signal reçu est la somme du signal original et d'un bruit aléatoire suivant une distribution gaussienne. Ce modèle est largement utilisé pour représenter de manière simplifiée mais efficace les perturbations subies par un signal lors de sa transmission à travers un canal, notamment dans les systèmes de communication numériques. [8]

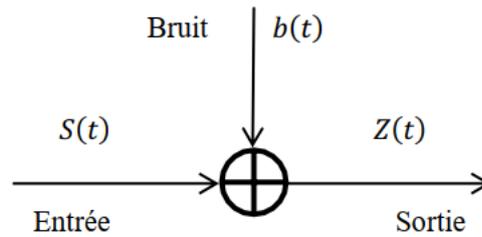


Figure I.5 : Le canal à bruit additif blanc gaussien AWGN [8]

Le canal gaussien idéal est défini par

$$h(t) = \delta(t) \quad (\text{I.2})$$

$\delta(t)$ est l'impulsion de Dirac. Sa transformée de Fourier est donnée par

$$H(f) = TF [\delta(t)] \quad (\text{I.3})$$

Sa bande est donc infinie. Ce canal est appelé canal gaussien idéal sans interférence entre symboles (sans IES).

$$H(f) = 1 \quad \forall f \quad (\text{I.4})$$

I.6 Capacité d'un canal gaussien continu non contraint à bande limitée

Pour ce canal, le bruit à la réception est un BBGA (AWGN) bruit blanc gaussien additif. Le canal est continu, le nombre de signaux à l'entrée est infini. Il est appelé canal non contraint. [9]

Sa capacité est donnée par :

$$C = W \log_2 \left(1 + \frac{P}{N} \right) \text{ bits/sec} \quad (\text{I.5})$$

W : La bande en Hz.

P : Puissance du signal reçu en watts.

N : Puissance du bruit en watts.

Donc :

$$C = W \log_2(1 + RSB) \text{ bits/sec} \quad (\text{I.6})$$

Le théorème de Shannon dit que :

Si le débit $D < C, \forall \varepsilon > 0$, il existe un code pour transmettre les informations avec une probabilité d'erreur $P_e \text{ après décodage} < \varepsilon$,

c'est-à-dire, on peut rendre $P_e \text{ après décodage} \rightarrow 0$.

Divisons le débit et la capacité par la bande W :

$$D < C \Rightarrow D < W \log_2 \left(1 + \frac{P}{N_0 W} \right) \quad (\text{I.7})$$

$$\Rightarrow \frac{D}{W} < \log_2 \left(1 + \frac{P}{N_0 W} \right) \quad (\text{I.8})$$

On appelle **Efficacité Spectrale** la quantité

$$\eta = \frac{D}{W} \text{ bits/sec/Hz} \quad (\text{I.9})$$

$$\Rightarrow \eta < \log_2 \left(1 + \frac{P}{N_0 W} \right) \quad (\text{I.10})$$

Et on a

$$E_b = P T_b = \frac{P}{D} \quad (\text{I.11})$$

Alors

$$P = E_b D \quad (\text{I.12})$$

On obtient

$$\Rightarrow \eta < \log_2 \left(1 + \frac{E_b D}{N_0 W} \right) \quad (\text{I.13})$$

$$\Rightarrow \eta < \log_2 \left(1 + \eta \frac{E_b}{N_0} \right) \quad (\text{I.14})$$

$$\Rightarrow \frac{E_b}{N_0} > \frac{2^\eta - 1}{\eta} \quad (\text{I.15})$$

E_b : Energie moyenne par bit information

N_0 : Densité spectrale de puissance du bruit

Calculons maintenant $\frac{E_b}{N_0}$ minimal :

$$\lim_{\eta \rightarrow 0} \left(\frac{E_b}{N_0} \right) = \lim_{\eta \rightarrow 0} \frac{2^\eta - 1}{\eta} = \ln 2 = 0.693 = -1.59 \text{ dB} \approx -1.6 \text{ dB} \quad (\text{I.16})$$

-1.6 dB est la valeur minimale du $\frac{E_b}{N_0}$ d'un système de communication fiable (P_e après décodage tend vers 0). Aucun système peut transmettre d'une façon fiable (P_e après décodage $\rightarrow 0$) avec $\frac{E_b}{N_0} - 1.6 \text{ dB}$. Cette valeur constitue la limite des limites. [9]

I.6.1 Capacité d'un canal gaussien à entrée binaire BI-AWGN :

La sortie de ce canal $y_i = x_i + z_i$, où :

y_i est l'observation à la sortie du canal.

x_i est le symbole transmis (Codes en ligne (NRZ, RZ, ...), modulations 2-ASK, 2-PSK, 2-FSK).

z_i est un bruit blanc gaussien échantillonné de moyenne 0 et de variance σ^2 .

La capacité de ce canal $C = \max(I(X, Y))$, est donnée par [10] :

$$C_{BI-AWGN} = -\frac{1}{2} \log_2(2\pi e \sigma^2) - \int_{-\infty}^{+\infty} \phi_\sigma(y) \log_2(\phi_\sigma(y)) dy \quad (\text{I.17})$$

Avec

$$\phi_\sigma(y) = \frac{1}{\sqrt{8\pi\sigma^2}} \left(e^{-\frac{(y+1)^2}{2\sigma^2}} + e^{-\frac{(y-1)^2}{2\sigma^2}} \right) \quad (\text{I.18})$$

$\phi_\sigma(y)$ est la densité de probabilité de sortie Y du canal BI-AWGN

Cette capacité est calculée par ordinateur en utilisant les méthodes numériques. [10]

Le tableau suivant (**Tableau I.1**) donne les limites de Shannon (E_b/N_0 minimal) pour quelques rendement du codeur et pour les deux types de décodage Soft et Hard [10].

Rate R	$(E_b/N_0)_{\text{Shannon}}$ (dB)	$(E_b/N_0)_{\text{soft}}$ (dB)	$(E_b/N_0)_{\text{hard}}$ (dB)
0.05	-1.440	-1.440	0.480
0.10	-1.284	-1.285	0.596
0.15	-1.133	-1.126	0.713
0.20	-0.976	-0.963	0.839
1/4	-0.817	-0.793	0.972
0.30	-0.657	-0.616	1.112
1/3	-0.550	-0.497	1.211
0.35	-0.495	-0.432	1.261
0.40	-0.333	-0.236	1.420
0.45	-0.166	-0.030	1.590
1/2	0	0.187	1.772
0.55	0.169	0.423	1.971
0.60	0.339	0.682	2.188
0.65	0.511	0.960	2.428
2/3	0.569	1.059	2.514
0.70	0.686	1.275	2.698
3/4	0.860	1.626	3.007
4/5	1.037	2.039	3.370
0.85	1.215	2.545	3.815
9/10	1.396	3.199	4.399
0.95	1.577	4.190	5.295

Tableau I.1 : Limites de Shannon pour quelques rendement du codeur et pour les deux types de décodage Soft et Hard. [10]

Ces valeurs sont tracées dans la figure I.6 suivante

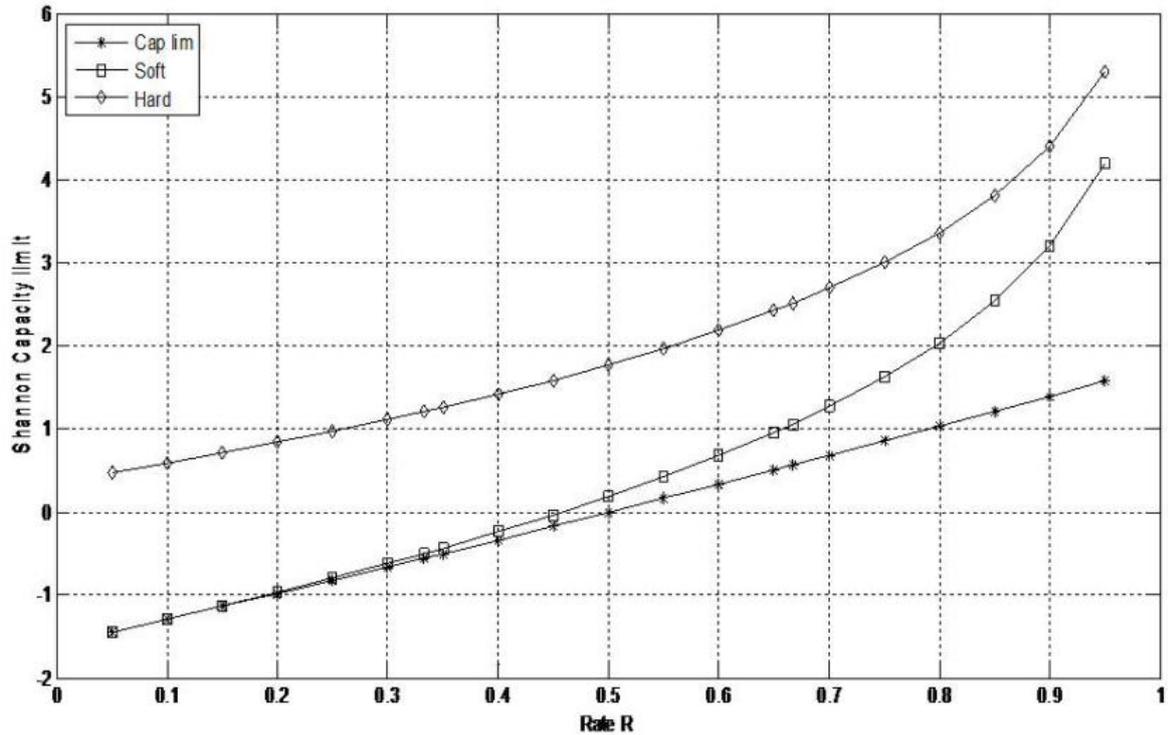


Figure I.6 : Graphes des valeurs des limites de Shannon pour quelques rendement du codeur et pour les deux types de décodage Soft et Hard. [10]

I.7 Algorithme de décodage MAP des codes convolutifs

L'algorithme Maximum A Posteriori (MAP) a été présenté formellement en 1974 dans la publication [BCJR]. Cet algorithme a été proposé comme une solution alternative au décodage classique des codes convolutifs, en se basant sur une approche probabiliste plus fine. [BCJR]

Contrairement aux méthodes de décodage traditionnelles qui cherchent généralement à minimiser la probabilité d'erreur globale sur la séquence transmise, l'algorithme MAP adopte un critère différent : Il vise à minimiser la probabilité d'erreur sur chaque symbole individuel du message. Grâce à cette approche, l'algorithme n'assure pas seulement une décision binaire sur chaque bit, mais fournit également une estimation de la fiabilité de cette décision, fondée sur la probabilité a posteriori du bit considéré.

En 1993, l'algorithme BCJR a été revisité et adapté par Berrou et al., dans le contexte du développement du turbo codage. Cette version modifiée a été communément appelée algorithme MAP (Maximum A Posteriori). Par la suite, plusieurs variantes de cet algorithme ont été introduites, exprimées notamment dans le domaine logarithmique, pour réduire la complexité de

calcul et faciliter l'implémentation. Parmi ces variantes, on distingue les algorithmes [Log-MAP] et [Max-Log-MAP] que nous allons détailler ultérieurement [11].

I.7.1 Algorithme MAP

L'algorithme de décodage MAP est une technique récursive qui calcule le Log rapport de vraisemblance - (LLR) de chaque bit basé sur l'ensemble du bloc de données observées de longueur N .

L'algorithme MAP permet de calculer la probabilité a posteriori de chaque bit d'information ou de chaque symbole transmis. Sur cette base, le décodeur associé sélectionne, à chaque instant, le bit ou le symbole ayant la plus forte probabilité d'avoir été émis. Cependant, bien que cette approche soit théoriquement plus précise, elle est restée longtemps inutilisée en pratique dans le décodage des codes convolutifs. En effet, elle n'apportait pas de gain significatif en performance par rapport à l'algorithme de Viterbi, qui était plus simple et plus efficace à implémenter sur le plan matériel et logiciel. La complexité computationnelle de l'algorithme MAP représentait alors un frein à son adoption [11].

La situation a toutefois changé radicalement en 1993, avec l'apparition des turbocodes. Le processus de décodage des turbocodes repose sur l'utilisation de décodeurs élémentaires à sorties pondérées, aussi appelées décisions souples (*soft outputs*), ce qui nécessite une estimation fine du degré de certitude associé à chaque bit. Contrairement à l'algorithme de Viterbi, qui fournit des décisions dures (*hard decisions*), l'algorithme MAP est particulièrement bien adapté à ce type de décodage souple, car il permet d'associer naturellement une pondération (ou une mesure de confiance) à chaque décision prise.

C'est donc dans ce nouveau contexte que l'algorithme MAP a trouvé toute son utilité et est devenu un élément clé du décodage itératif des turbocodes [11].

Le Log rapport de vraisemblance LLR du bit m_k est donné par

$$LLR(m_k) = \log \frac{P_r(m_k=1|R_1^N)}{P_r(m_k=0|R_1^N)} \quad (I.19)$$

Avec : $m_k \in \{0, 1\} \quad \forall k$

$R_1^N = \{R_1, \dots, R_k, \dots, R_N\}$ Représente la trame des échantillons reçus.

L'algorithme MAP calcule le rapport de vraisemblance LLR :

$$LLR(m_k) = \log \frac{\sum_s \sum_{s'} P_r(m_k=1, S_k=s, S_{k-1}=s', R_1^N)}{\sum_s \sum_{s'} P_r(m_k=0, S_k=s, S_{k-1}=s', R_1^N)} \quad (I.20)$$

S_k et S_{k-1} Sont les états du treillis aux instants k et $k - 1$ [BCJR].

L'algorithme MAP-BCJR définit ces probabilités conjointes en termes de trois paramètres.

$$\alpha_K(s) = P_r(S_k = s | R_1^N)$$

$$\beta_K(s) = P_r(R_{k+1}^N | S_k = s)$$

$$\gamma_j(R_K, s', s) = P_r(m_K = j, S_k = s, R_k | S_{k-1} = s')$$

Le LLR peut maintenant être décrit en termes de : [BCJR]

$$\Lambda(m_k) = \log \frac{\sum_s \sum_{s'} \gamma_1(R_K, s', s) \alpha_{K-1}(s') \beta_K(s)}{\sum_s \sum_{s'} \gamma_0(R_K, s', s) \alpha_{K-1}(s') \beta_K(s)} \quad (I.21)$$

$\alpha_K(s)$ et $\beta_K(s)$ peuvent être calculées en avant et en arrière en utilisant les probabilités de transition du treillis du codeur $\gamma_j(R_K, s', s)$. [BCJR]

$$\alpha_K(s) = \sum_{s'} \sum_{j=0}^1 \gamma_j(R_K, s', s) \alpha_{K-1}(s') \quad (I.22)$$

$$\beta_K(s) = \sum_{s'} \sum_{j=0}^1 \gamma_j(R_{K+1}, s, s') \beta_{K+1}(s') \quad (I.23)$$

$\gamma_j(R_K, s', s)$ Est donné par :

$$\begin{aligned} \gamma_j(R_K, s', s) &= P_r(R_k | m_K = j, S_k = s, S_{k-1} = s') \\ &\times P_r(m_k = j | S_k = s, S_{k-1} = s') \times P_r(S_k = s | S_{k-1} = s') \end{aligned} \quad (I.24)$$

Pour la mise en œuvre de l'algorithme MAP, il est nécessaire d'initialiser les valeurs de récurrences α et β .

Dans le cas où l'état initial et final des deux codeurs sont initialisée à zéro, on aura :

$$\alpha_0(s) = \begin{cases} 1 & \text{si } s = 0 \\ 0 & \text{autrement} \end{cases} \quad (I.25)$$

$$\beta_N(s) = \begin{cases} 1 & \text{si } s = 0 \\ 0 & \text{autrement} \end{cases} \quad (I.26)$$

Mais, si la terminaison vers l'état zéro est inconnue pour les deux codeurs alors l'initialisation de β deviendra :

$$\beta_N(s) = \frac{1}{M} \text{ pour tous les états } s \quad (I.27)$$

Où M est le nombre total des états du treillis. L'initialisation ci-dessus est basée sur la terminaison de treillis du bloc turbo dans l'état arbitraire [12].

I.7.2 Algorithme Max-Log-MAP

Le décodage basé sur le critère MAP présente une complexité de calcul relativement élevée, ce qui rend son intégration directe dans un circuit électronique difficile. Cette complexité provient principalement des opérations mathématiques nécessaires, telles que les multiplications, les divisions ainsi que les fonctions exponentielles (la densité gaussienne du bruit), qui sont coûteuses en termes de ressources matérielles et de temps de traitement.

Afin de surmonter ces difficultés, plusieurs algorithmes sous-optimaux ont été proposés en pratique. Leur objectif est de réduire la complexité de mise en œuvre tout en maintenant une performance de décodage acceptable. L'une des approches les plus efficaces pour simplifier les calculs consiste à transférer le problème dans un espace logarithmique.

Cette transformation permet de remplacer les multiplications par des additions, les divisions par des soustractions, et les exponentiations par des multiplications, ce qui facilite considérablement l'implémentation du décodage MAP dans les systèmes numériques.

Pour éviter ces problèmes, nous pouvons calculer le logarithme naturel de toutes ces variables, à savoir, $\gamma_j(R_k, s', s)$, $\alpha_k(s)$ et $\beta_k(s)$. Étant donné que $\gamma_j(R_k, s', s)$ est le résultat de la multiplication de trois facteurs, le logarithme de $\gamma_j(R_k, s', s)$, $\tilde{\gamma}_j(R_k, s', s)$ est l'addition du logarithme de ces trois facteurs [12][13].

$$\begin{aligned} \tilde{\gamma}_j(R_k, s', s) &= \ln \gamma_j(R_k, s', s) = \\ &= \ln \left(\Pr \left((R_k | m_k = j, S_k = s, S_{k-1} = s') \right) \right) + \ln \left(\Pr (m_k = j | S_k = s, S_{k-1} = s') \right) + \\ &= \ln \left(\Pr (S_k = s | S_{k-1} = s') \right) \end{aligned} \quad (\text{I.28})$$

Pour $\alpha_k(s)$, nous avons :

$$\tilde{\alpha}_k(s) = \ln(\alpha_k(s)) \approx \max_{\text{tous les } s'} \left(\tilde{\gamma}_j(R_k, s', s) + \tilde{\alpha}_{k-1}(s') \right) \quad (\text{I.29})$$

Avec

$$\tilde{\alpha}_k(s) = \ln(\alpha_k(s)) \quad (\text{I.30})$$

Avec : $\ln(A + B + C + \dots) \approx \max(A, B, C, \dots)$

La condition d'initialisation devient :

$$\tilde{\alpha}_0(s) = \begin{cases} 0 & \text{Si } s = 0 \\ -\infty & \text{autrement} \end{cases}$$

Et pour $\beta_k(s)$, nous avons :

$$\tilde{\beta}_k(s) = \ln(\beta_N(s)) = \max_{\text{tous les } s'} \left(\tilde{\gamma}_j(R_{k+1}, s, s') + \tilde{\beta}_{k+1}(s') \right) \quad (\text{I.31})$$

Et

$$\tilde{\beta}_N(s) = \ln\left(\frac{1}{M}\right) \quad \text{Pour tous les } s \quad (\text{I.32})$$

La sortie souple des données décodées pour cette approche est :

$$\text{LLR}(m_k) \approx \max_{\text{tous les } s'} \left(\tilde{\gamma}_1(R_k, s', s) + \tilde{\alpha}_{k-1}(s') + \tilde{\beta}_k(s) \right) - \max_{\text{tous les } s'} \left(\tilde{\gamma}_0(R_k, s', s) + \tilde{\alpha}_{k-1}(s') + \tilde{\beta}_k(s) \right) \quad (\text{I.33})$$

I.7.3 Algorithme Log-MAP

La réduction de la complexité de calcul obtenue avec l'algorithme Max-Log-MAP repose sur l'emploi d'approximations mathématiques qui permettent de simplifier considérablement les opérations arithmétiques nécessaires au décodage. Cependant, cette simplification a un inconvénient majeur : Elle entraîne une dégradation de la qualité de l'estimation de la vraisemblance LLR, ce qui peut affecter la précision du décodage.

Pour remédier à cette limitation, une version améliorée appelée algorithme Log-MAP a été développée [13]. Cette version intègre la notion de Jacobian Algorithm, une technique mathématique permettant d'améliorer la qualité des approximations logarithmiques utilisées lors du décodage.

Grâce à cette méthode, l'algorithme Log-MAP parvient à conserver une meilleure précision dans le calcul des probabilités a posteriori, tout en restant plus facile à mettre en œuvre que l'algorithme MAP d'origine, plus complexe. Il représente ainsi un bon compromis entre performance de décodage et simplicité de calcul.

L'algorithme Jacobian comporte les relations suivantes :

$$\ln(e^{\delta_1} + e^{\delta_2}) = \max(\delta_1, \delta_2) + \ln(1 + e^{-|\delta_2 - \delta_1|}) = \max(\delta_1, \delta_2) + f_c(|\delta_1 - \delta_2|) \quad (\text{I.34})$$

Et

$$\ln\left(\frac{e^{\delta_1} + e^{\delta_2} + \dots + e^{\delta_n}}{e^{\delta}}\right) = \ln(e^{\delta} + e^{\delta_n}) = \max(\delta, \delta_n) + f_c(|\delta - \delta_n|) \quad (\text{I.35})$$

Où $f_c(\dots)$ est une fonction de correction

I.8 Le Turbodécodage MAP des Turbo codes

En sortie du canal, trois vecteurs bruités $r^{(0)} = \tilde{u}$, $r^{(1)}$ et $r^{(2)}$ sont observés. Afin de retrouver à partir de ces vecteurs l'information \hat{u} , le décodage suit le schéma représenté sur la figure I.7. Les grandeurs considérées à chaque nœud du schéma sont des LLR. Deux décodeurs convolutifs sont placés en parallèle afin d'utiliser simultanément les informations données par chacun d'eux. [14]

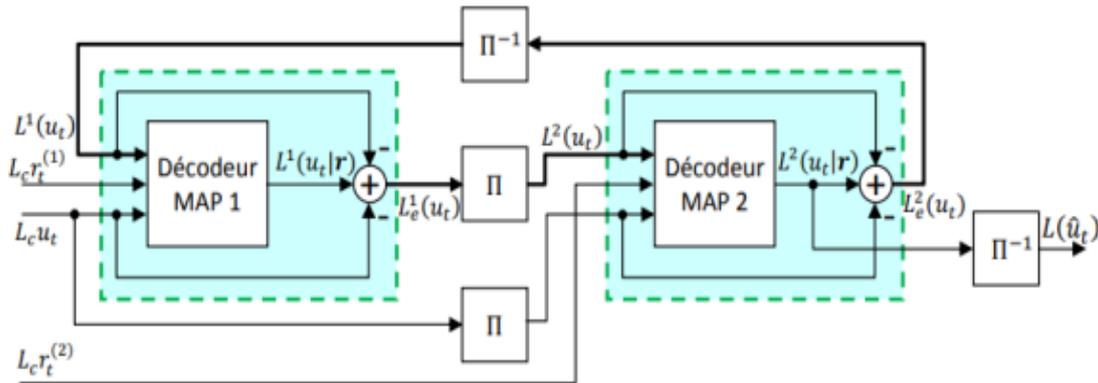


Figure I.7 : Schéma de décodage d'un turbo code (Turbo décodeur) [14]

Comme on le voit sur la figure, chaque décodeur prend trois entrées : les bits de sortie de canal systématiques \tilde{u} , les bits de parité transmis par le codeur RSC associé $r^{(i)}$, $i = 1, 2$ et les informations provenant du décodeur de l'autre composant sur les valeurs probables des bits concernés. $(L(u_t))$ appelé aussi l'information à priori)

Cette information provenant de l'autre décodeur est appelée information a priori. Les composants décodeurs doivent également fournir ce qu'on appelle les sorties souples (Soft-output) pour les bits décodés sous forme de LLR. La polarité du LLR détermine le signe du bit, tandis que son amplitude quantifie la probabilité d'une décision correcte. [14]

Deux types de décodeurs sont adéquats, soit l'algorithme SOVA (Soft-Output Viterbi Algorithm) proposé par Hagenauer et Höher [15] et l'algorithme MAP (Maximum A posteriori Probability) de Bahl et al. [16]. Dans ce travail on n'utilisera que le deuxième type pour des raisons de performances et de complexité, ainsi que pour sa popularité dans les applications pratiques actuelles.

En utilisant la définition du LLR, nous pouvons écrire :

$$L(u_t | \mathbf{r}) = L_c u_t + L(u_t) + L_e(u_t) \tag{I.36}$$

Nous donnons ci-dessous les définitions des termes a priori, extrinsèque et a posteriori, que nous utilisons tout au long de cette section. L'information a priori relative à un bit est une information connue avant le début du décodage, à partir d'une source autre que la séquence reçue ou les contraintes du code. Elle est également appelée information intrinsèque pour faire contraste avec l'information extrinsèque. [12]

L'information extrinsèque fournie par un décodeur relative à un bit u_t , est l'information basée sur la séquence reçue et sur l'information a priori, mais excluant les bits systématiques reçus et l'information a priori $L(u_t)$ relative au bit u_t . Typiquement, le composant décodeur fournit cette information en utilisant les contraintes imposées à la séquence transmise par le code utilisé.

L'information a posteriori relative à un bit est l'information que le décodeur génère en prenant en compte toutes les sources d'informations disponibles concernant u_t . C'est le LLR a posteriori, c'est-à-dire $L(u_t | r)$, que l'algorithme MAP génère en sortie [12].

Dans le turbodécodeur, les deux composants décodeurs partagent leurs informations extrinsèques sur les bits de message en tant qu'informations a priori introduites dans chaque décodeur. La principale différence pour les décodeurs MAP à l'intérieur d'un turbodécodeur est la source de leurs informations a priori. La deuxième différence dans un turbodécodeur est que chaque décodeur MAP est utilisé plus d'une fois. L'information extrinsèque est mise à jour et transmise entre les décodeurs sur plusieurs itérations.

Une itération du turbodécodage consiste en une application de l'algorithme MAP pour chaque composant décodeur. Un turbodécodeur peut utiliser n'importe quel nombre d'itérations. Cependant, augmenter le nombre d'itérations augmente le temps de décodage. [12]

Le turbodécodeur dont le schéma bloc est illustré dans la figure I.7, a comme entrée $L_c u_t, L_c r_t^{(1)}$ et $L_c r_t^{(2)}$ les LLR des séquences $r^{(0)} = \tilde{u}, r^{(1)}$ et $r^{(2)}$ en sortie du canal respectivement.

Dans la première itération $l = 1$, le décodeur DEC1 n'a que $L_c u_t$ et $L_c r_t^{(1)}$ à son entrée, et pas d'information a priori ($L_1^1(u_t) = 0$). A la sortie il donne l'information a posteriori $L_1^1(u_t | r)$. La nouvelle information extrinsèque qu'il a créé à propos des bits u_t de la première itération, indiquée par l'indice 1, est donc $L_{e,1}^1(u_t) = L_1^1(u_t | r) - L_c u_t$.

Les entrées du décodeur DEC2 provenant du canal sont $\Pi(L_c u_t)$ et $L_c r_t^{(2)}$, les valeurs $L_c u_t$ étant entrelacées pour être dans le même ordre que celui des bits de message lorsqu'ils sont entrés dans le codeur RSC2.

Pour l'itération l , le décodeur DEC2 utilisera les informations extrinsèques provenant du premier décodeur $L_{e,l}^1(u_t)$ après entrelacement, en tant qu'information a priori supplémentaire

$$L_l^2(u_t) = \Pi \left(L_{e,l}^1(u_t) \right) \text{ sur les bits du message } u_t.$$

Le décodeur DEC2 sortira les LLR a posteriori qu'il calcule pour les bits du message en tant que $L_l^2(u_t | \mathbf{r})$. La nouvelle information extrinsèque créée par le décodeur DEC2 sur les bits de message à l'itération l est donc :

$$L_{e,l}^2(u_t) = L_l^2(u_t | \mathbf{r}) - \Pi(L_c u_t) - L_l^2(u_t) \tag{I.37}$$

A partir de la seconde itération, le décodeur DEC1 répétera l'algorithme MAP mais maintenant avec une information a priori provenant de information extrinsèque du décodeur DEC2 calculée à l'itération précédente. Ainsi, dans la $l^{\text{ème}}$ itération, l'information extrinsèque provenant du décodeur DEC2 est désentrelacée de manière à être dans le même ordre que les bits de message de DEC1 : $L_l^1(u_t) = \Pi^{-1} \left(L_{e,l-1}^2(u_t) \right)$.

La nouvelle information extrinsèque sur les bits du message fournie par le décodeur DEC1 est ainsi :

$$L_{e,l}^1(u_t) = L_l^1(u_t | \mathbf{r}) - L_c u_t - L_l^1(u_t) \tag{I.38}$$

Notons que les rapports de vraisemblance logarithmique du canal restent inchangés tout au long du turbodécodage, et seules les informations extrinsèques changent à chaque itération.

Le turbodécodeur peut être arrêté après un nombre fixe d'itérations I_{max} ou lorsque des critères prédéterminés ont été atteints (Critères d'arrêt , chapitre 2 et 3). [12]

Chapitre II

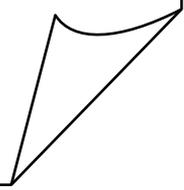
Critères

d'arrêt Turbo

à base des

décisions

dures



Introduction

Le décodage turbo est une technique puissante qui repose sur un processus itératif permettant d'améliorer progressivement la qualité de la correction des données transmises. Ce processus repose sur des échanges d'informations entre plusieurs décodeurs, et il est répété plusieurs fois afin d'atteindre une meilleure fiabilité. Cependant, il est important de noter que le nombre d'itérations ne peut pas être augmenté indéfiniment. En effet, après un certain seuil, le processus de décodage atteint une saturation où les performances cessent de s'améliorer de manière significative.

Dans le but d'optimiser le temps de traitement et d'éviter des calculs inutiles, il devient alors essentiel d'introduire un mécanisme de contrôle pour décider du moment adéquat pour arrêter les itérations. C'est dans ce contexte qu'intervient ce qu'on appelle un critère d'arrêt, une condition spécifique que l'on vérifie à la fin de chaque itération pour déterminer s'il est pertinent de continuer ou non le décodage. [17]

La littérature scientifique propose plusieurs critères d'arrêt, chacun avec ses avantages, ses inconvénients et son domaine d'application. Dans ce qui suit, nous allons présenter quelques-unes des techniques les plus couramment utilisées à cet effet.

II.1 Critère CRC

Il s'agit ici d'une règle d'arrêt basée sur la détection d'erreurs dans les séquences décodées, en utilisant un code de redondance cyclique (CRC) externe. Ce type de code est appliqué après la prise de décision dure (hard decision) sur les bits décodés, ce qui permet d'évaluer la validité de la séquence obtenue à chaque itération du processus de décodage.

Plus précisément, un code de détection d'erreurs, indépendant du turbo code, tel qu'un code CRC, est concaténé avec ce dernier en tant que code externe. Ce code cyclique est utilisé pour identifier les séquences erronées issues du turbo décodage. Lorsqu'une séquence incorrecte est produite, le CRC génère un syndrome non nul, indiquant la présence d'une erreur. En revanche, lorsque le syndrome CRC est nul, cela signifie que la séquence est considérée comme correcte selon le code CRC, et donc le critère d'arrêt est atteint. À ce moment, les itérations de décodage peuvent être arrêtées en toute sécurité.

Cette méthode d'arrêt par somme de contrôle CRC est bien établie et reconnue pour son efficacité. Elle est d'ailleurs adoptée dans certaines normes de communication sans fil,

notamment dans la norme 3G-WCDMA (Wideband Code Division Multiple Access), ce qui témoigne de sa fiabilité dans des systèmes réels. [18]

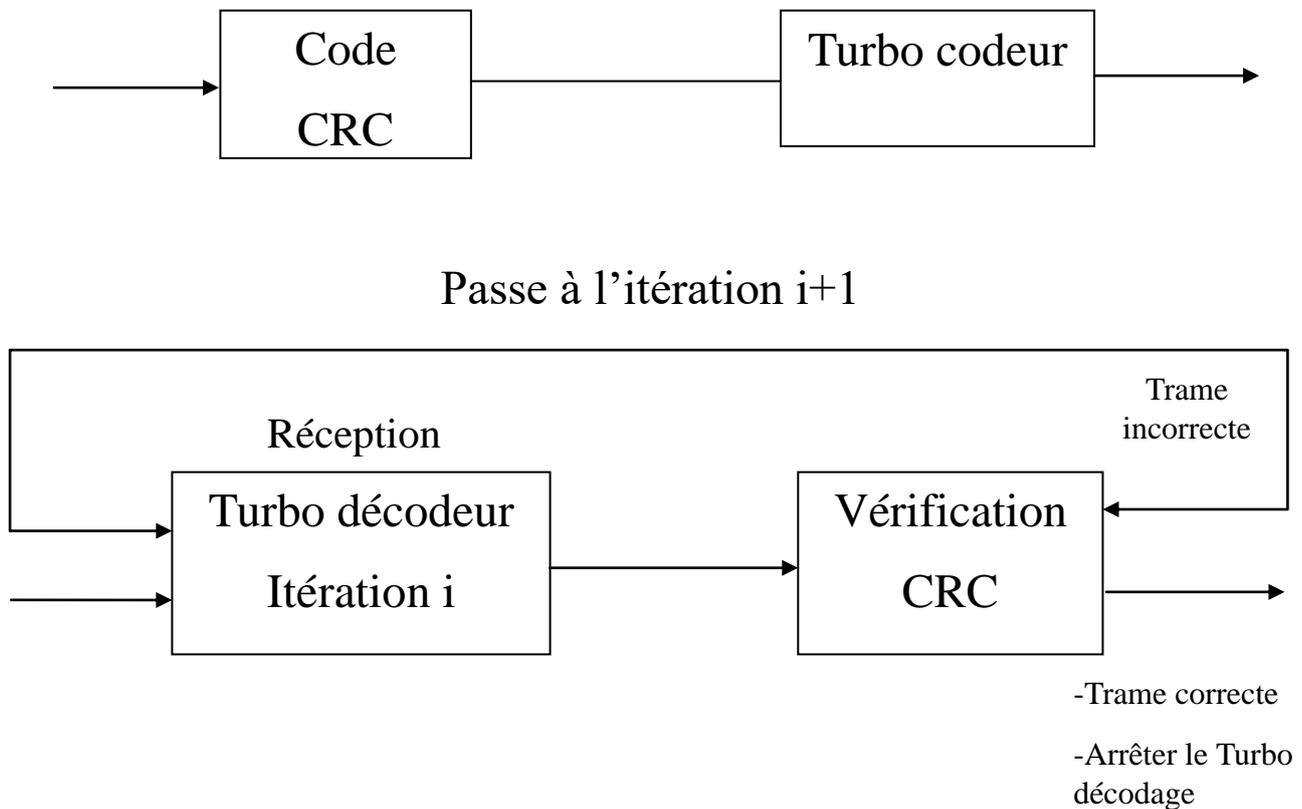


Figure II.1 : Schéma de la critère CRC [19]

II.2 Critère SCR

Soit $c^2(i)$ le nombre de changements de signe détectés entre les éléments de la séquence d'information extrinsèque $Le_2(i - 1)$ à la $(i - 1)^{ème}$ itération et ceux de la séquence d'information extrinsèque $Le_2(i)$ calculée à la $i^{ème}$ itération avec $i > 1$.

Ce changement de signe constitue l'élément central du critère de rapport de changement de signe appelé Sign-Change-Ratio (SCR). Il s'agit d'un critère simple à implémenter, qui repose sur le calcul du rapport $\frac{c^2(i)}{N}$, où N représente la longueur de la séquence considérée.

Ce critère permet de détecter la stabilité du processus de décodage : Lorsque le rapport des changements de signe devient suffisamment faible, cela signifie que les informations extrinsèques n'évoluent plus de manière significative, suggérant une convergence du processus. Ainsi, S'il est

inférieur à $\leq (0.005 \sim 0.03)$ alors le décodage peut être interrompu sans presque aucune dégradation de la performance [20]. Ce seuil indique que moins de 0.5 % à 3 % des bits changent encore de signe entre deux itérations consécutives, ce qui est considéré comme négligeable.

Une variante de ce critère a été proposée pour affiner davantage l'analyse de la convergence. Elle consiste à appliquer la même méthode, mais cette fois-ci sur la séquence d'information extrinsèque $Le_1(i - 1)$ et $Le_1(i)$. Par conséquent, le nouveau rapport à évaluer devient alors $\frac{c^n(i)}{N}$ qui doit être $\leq (0.005 \sim 0.03)$ pour arrêter le décodage. Le critère résultant de cette double évaluation est désigné par SCRM (Sign-Change-Ratio Modified).

Il est important de noter que, bien que le critère SCRM offre une meilleure précision pour détecter la convergence du processus de décodage, il en résulte une complexité accrue. En effet, sa mise en œuvre nécessite deux fois plus de comparaisons que le SCR classique, ce qui double approximativement la charge de calcul. [3]

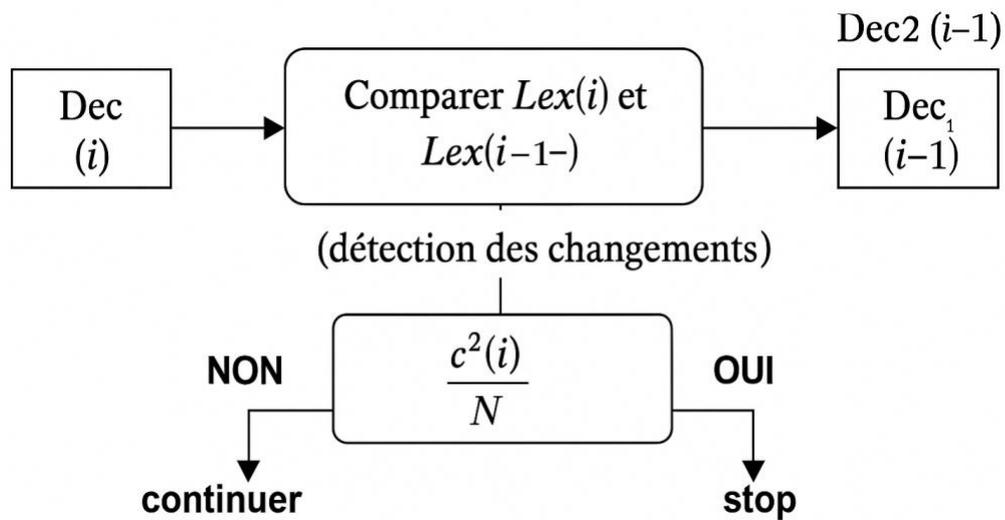


Figure II.2: Critère d'arrêt Sign change Ratio [3]

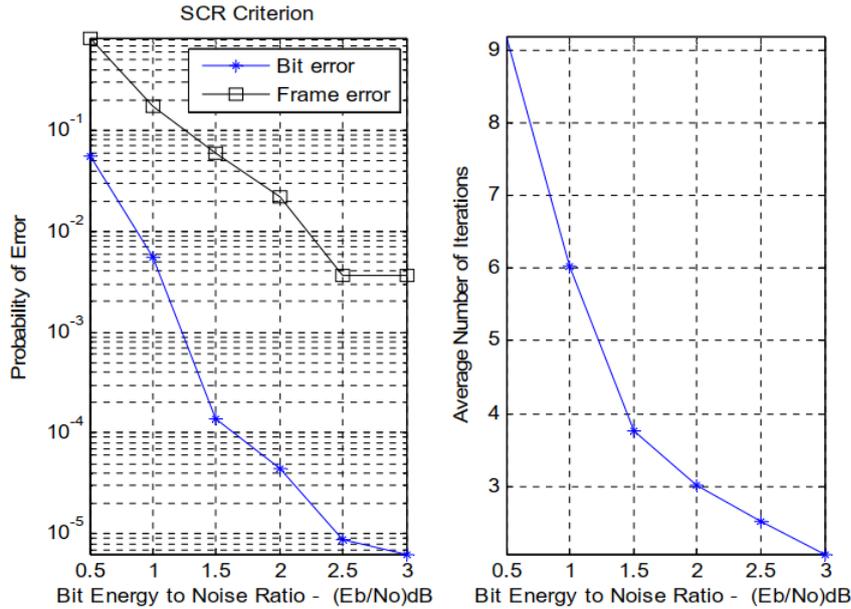


Figure II.3 : Courbes de performance de critère SCR, $TEB = f\left(\frac{E_b}{N_0}\right)_{dB}$ [21]

II.3 Critère SDR

Le critère d'arrêt Sign-Difference-Ratio (SDR) est une variante du SCR. Dans ce cas, on compte le nombre de fois D_{ji} où les signes de l'information a priori et de l'information extrinsèque d'un même décodeur diffèrent à l'itération 'i'.

Le processus de décodage turbo est arrêté si la condition suivante est satisfaite

$$D_{ij} \leq P N \tag{II.1}$$

Où 'P' est un seuil qui représente le rapport de différence de signe SDR, 'N' est la longueur de la trame d'information.

Ce seuil 'P' est généralement compris entre :

$$10^{-3} \leq P \leq 10^{-2} \tag{II.2}$$

De plus, D_{ji} représente aussi le nombre de différences de signe entre les informations extrinsèques des deux décodeurs.

Le SDR atteint des performances similaires au SCR en termes de BER, FER et de nombre moyen d'itérations, tout en nécessitant une complexité moindre. Toutefois, les méthodes SCR et SDR peuvent nécessiter en moyenne une itération supplémentaire comparée à un arrêt parfaitement optimal (Critère GENIE). [3]

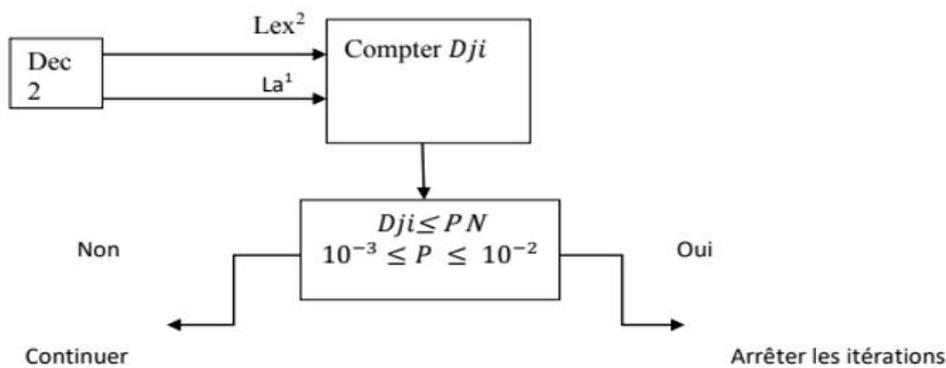


Figure II.4 : Critère d'arrêt SDR Sign Difference Ratio [3]

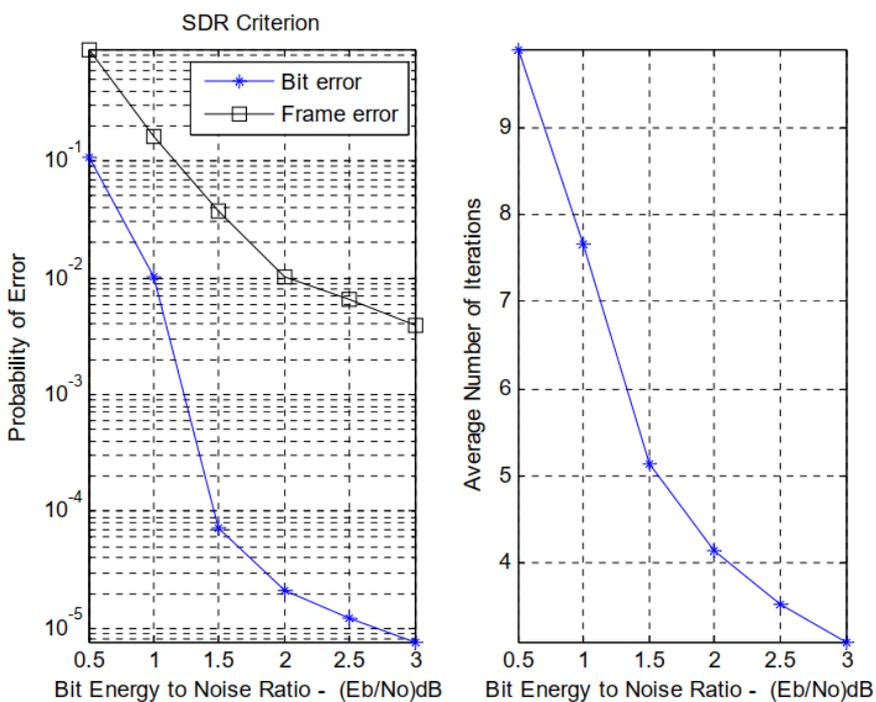


Figure II.5 : Courbes de performance de critère SDR, $TEB = f\left(\frac{E_b}{N_0}\right)_{dB}$ [21]

II.4.Critère HDA

Le critère d'arrêt HDA (Hard-Decision-Aided) est une méthode performante et efficace qui tire parti des principes de l'analyse statistique appliquée aux décisions dures. Cette approche permet d'évaluer la qualité des décisions prises lors du décodage en se basant sur des mesures précises de confiance et de cohérence des données. En s'appuyant sur ces concepts, le critère HDA facilite une détection plus rapide de la convergence du décodage, ce qui permet d'arrêter le processus de manière optimale tout en préservant les performances du système.

L'information de fiabilité (LLR) à la fin de chaque itération peut fournir une information concernant la convergence du processus de décodage. A la $(i - 1)$ éme itération, on sauvegarde la séquence d'information de fiabilité $LLR_2(i - 1)$ dans une mémoire tampon. Lors de la i éme itération, on compare la séquence d'information décodée de fiabilité $LLR_2(i)$ à celle de $LLR_2(i - 1)$. Si les éléments de ces deux dernières séquences présentent les mêmes signes, alors le décodage pourrait s'arrêter.

Nous pouvons appliquer la même procédure à la sortie du DEC1. C'est à dire comparer $LLR_1(i - 1)$ et $LLR_1(i)$. Nous désignons par HDA le critère d'arrêt résultant [22] [5].

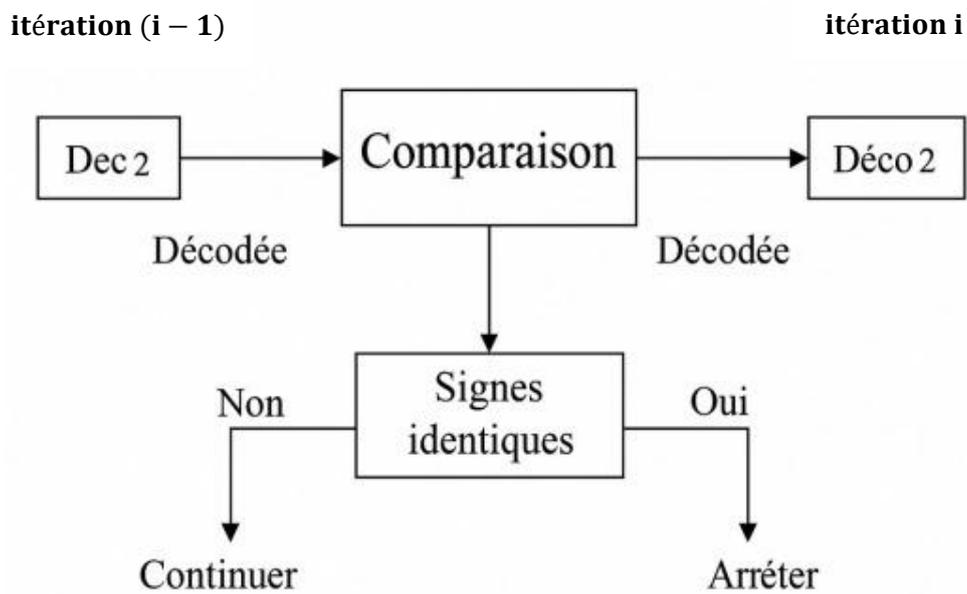


Figure II.6: Critère d'arrêt HDA (Hard Decision Aided) [5]

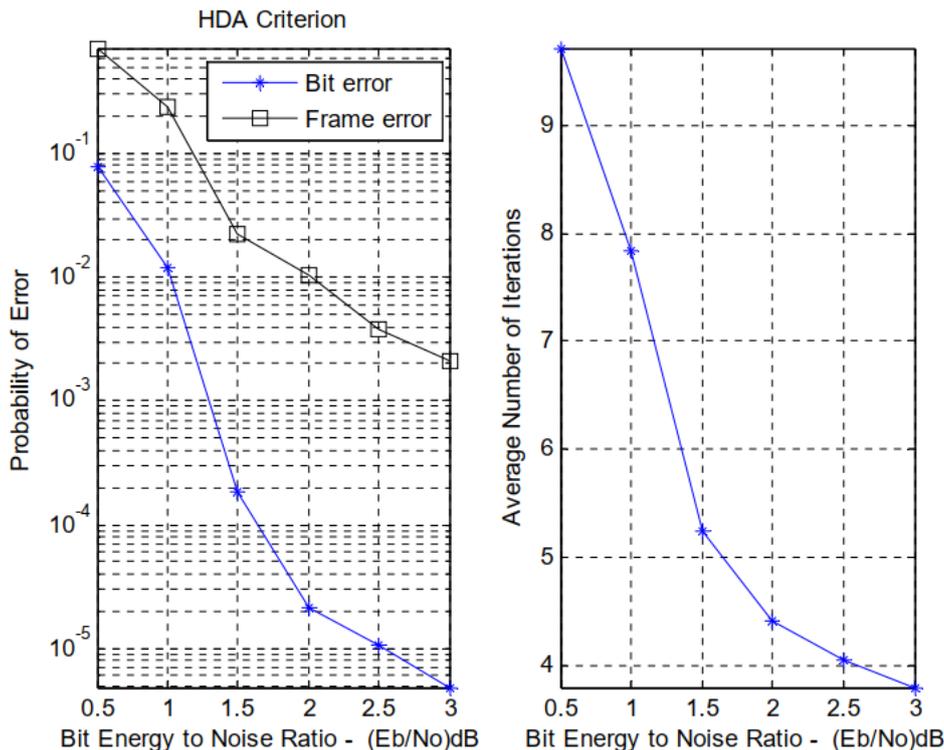


Figure II.7 : Courbes de performance de critère HDA, $TEB = f\left(\frac{E_b}{N_0}\right)_{dB}$ [21]

II.5.Critère IHDA

Cette règle repose sur l’observation que les informations extrinsèques tendent à augmenter progressivement avec le nombre croissant d’itérations dans le processus de décodage. Cette augmentation continue contribue à un raffinement progressif des valeurs des log-likelihood ratios (LLR), ce qui améliore la fiabilité des décisions prises par le décodeur au fil des itérations. Le critère IHDA exploite donc cette propriété pour détecter la convergence du décodage, permettant ainsi d’arrêter le processus de manière efficace lorsque les améliorations deviennent marginales. [5]

Son principe est de comparer les décisions dures basées sur $\left(\left(\frac{2}{\sigma^2}\right) y_d^2(k) + L_{e1}^i(d(k))\right)$ avec les décisions basées sur le LLR $L_2^i(d(k))$ des bits d’information à la sortie du second décodeur.

Elle consiste à :

1. Décoder l’itération i (décodeur 1, décodeur 2).
2. Appliquer une décision sur $\left(\left(\frac{2}{\sigma^2}\right) y_d^2(k) + L_{e1}^i(d(k))\right)$
3. Appliquer une décision sur $L_2^i(d(k))$
4. Comparez les deux décisions
 - Si les deux décisions sont identiques, arrêtez le décodage Turbo.
 - Sinon, passez à l’itération $i+1$. [5]

II.6. Critère GENIE

La règle d'arrêt GENIE constitue une référence essentielle et largement reconnue pour mesurer la performance des différents critères d'arrêt utilisés dans le domaine du turbo-décodage. Cette règle, bien que purement théorique, joue un rôle fondamental en établissant une limite imbattable de performance, servant ainsi de point de comparaison incontournable pour toutes les autres méthodes. En effet, grâce à une connaissance parfaite et préalable de la séquence de bits transmise, le critère GENIE est capable de reconnaître immédiatement le paquet d'information décodé correct. Cette capacité lui permet d'arrêter le processus itératif précisément au moment où le paquet correct est retrouvé, optimisant ainsi le nombre d'itérations nécessaires. [23]

Dans la mise en œuvre de la technique d'arrêt GENIE, on suppose que le décodeur dispose de la connaissance complète de tous les bits transmis. À chaque itération, le décodage est vérifié en comparant la trame décodée à la trame d'origine. Lorsque toutes les parties de la trame sont correctement décodées, le processus est interrompu, sauf si le nombre maximal d'itérations prédéfini est atteint. Cette approche garantit ainsi que le décodage s'arrête au plus tôt, tout en assurant les meilleures performances possibles en termes de taux d'erreur binaire (BER) et de taux d'erreur sur trame (FER). [24]

Ainsi, le critère GENIE est considéré comme un critère théorique idéal, une sorte de limite de référence qui permet d'évaluer et de comparer l'efficacité des autres critères d'arrêt basés sur les trames, en mettant en évidence leur proximité ou leur écart avec cette limite optimale. [25]

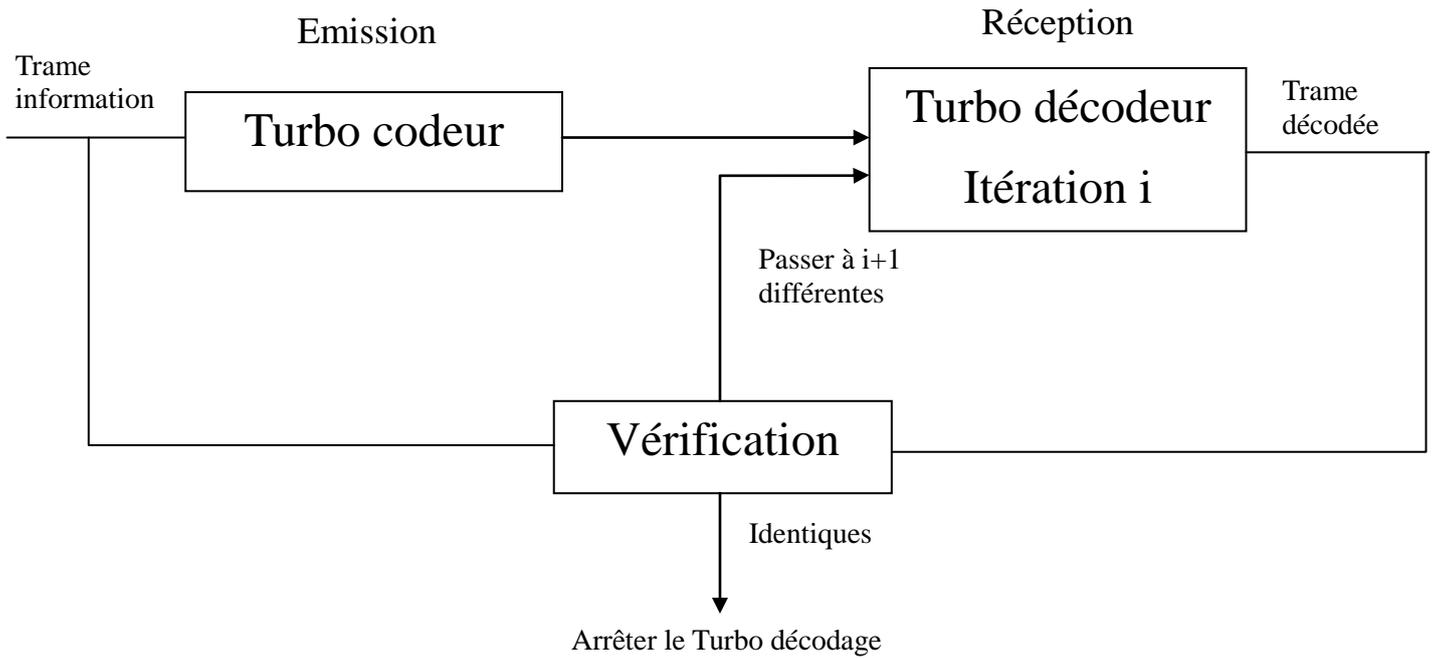


Figure II.8 : Schéma du critère GÉNIE [23]

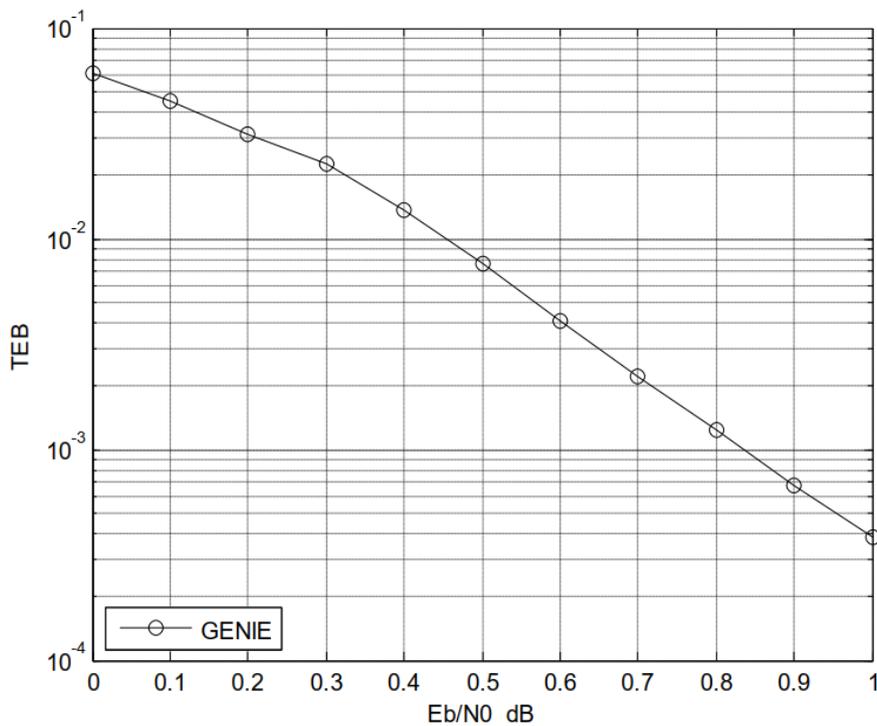


Figure II.9 : Performances d'un critère GENIE appliqué au turbo décodage [12]

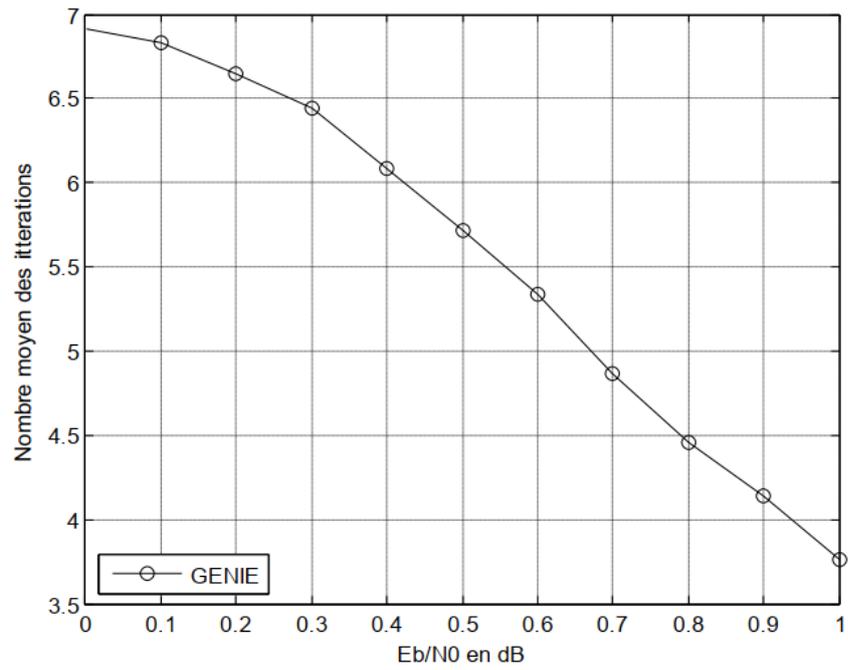


Figure II.10 : Nombre moyen des itérations du critère GENIE appliqué au turbo décodage [12]

Chapitre III

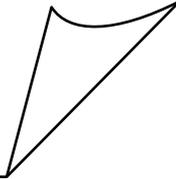
Critères

d'arrêt Turbo

à base

des quantités

souples



Introduction

Dans les systèmes de décodage turbo, l'amélioration des performances se fait à travers un processus itératif. Toutefois, pour éviter une consommation excessive de ressources, il est essentiel de savoir quand arrêter ces itérations. Les critères d'arrêt jouent un rôle clé dans cette gestion, permettant de déterminer le moment optimal pour arrêter de décodage sans compromettre la qualité des résultats. Ces critères peuvent être classés en deux catégories principales: Les critères hard et les critères soft. Les critères hard sont généralement basés sur la stabilité des décisions binaires (0 ou 1), tandis que les critères soft reposent sur la fiabilité des valeurs de Log-Likelihood Ratio (LLR), qui indiquent le degré de confiance dans chaque bit décodé. Ce chapitre a pour objectif de présenter les techniques à base des décisions souples (Soft) et d'analyser leurs fonctionnements.

III.1 Critère de l'Entropie Croisée CE (Cross-Entropy)

Une fois que le turbodécodage a convergé, les probabilités a posteriori $L(u_t|r)$ de chaque composant décodeur (DEC1, DEC2) devraient être en accord. D'où une mesure utile de la convergence du décodeur est la similarité de ces distributions. Une façon de mesurer la similarité de deux fonctions de distribution de probabilité est de calculer la divergence de Kullback-Leibler également connue sous le nom de l'entropie croisée.

L'entropie croisée CE (Cross-Entropy) est une mesure de la différence entre deux distributions de probabilité, P la distribution réelle et Q la distribution de référence pour la variable aléatoire X. Lorsque P et Q sont des distributions discrètes, l'entropie croisée CE est donné par :

$$D(P, Q) = \sum_j p_j \log_2 \frac{p_j}{q_j} \quad (\text{III.1})$$

Si les distributions sont les mêmes, l'entropie croisée CE est nul. Autrement-dit plus elles sont proches, plus la valeur de $D(P, Q)$ est petite. Hagenauer et al. [26] a pris l'idée du critère de l'entropie croisée pour détecter la convergence entre les sorties de chaque composant décodeur du turbodécodeur. En d'autres termes, le critère d'entropie croisée (CE) est utilisé pour mesurer la proximité des deux distributions de sortie $L_i^1(u_t|r)$ et $L_i^2(u_t|r)$ des deux composants décodeurs soumis à une séquence de données indépendantes U_t . A l'itération i, l'entropie croisée T(i) de $L_i^2(u_t|r)$ par rapport à la distribution de référence $L_i^1(u_t|r)$:

$$T(i) \approx \sum_t \frac{|\Delta L_{e,i}(u_t)|^2}{\exp(|L_i^1(u_t|r)|)} \quad (\text{III.2})$$

Sachant que $\Delta L_{e,i}(u_t) = L_{e,2}^2(u_t) - L_{e,1}^1(u_t)$

$$L_{e,i}^2(u_t) = L_i^2(u_t|r) - \Pi(L_c u_t) - L_{a,i}^2(u_t) \quad (\text{III.3})$$

$$L_{e,i}^1(u_t) = L_i^1(u_t|r) - (L_c u_t) - L_{a,i}^1(u_t) \quad (\text{III.4})$$

En pratique, nous déterminons que la convergence s'est produite lorsque l'entropie croisée devient suffisamment petite. Il est pris comme critère d'arrêt si : [18]

$$T(i) < \text{un seuil} \quad (\text{III.5})$$

Les deux décodeurs fournissent les informations extrinsèques $Le_1^{(i)}(u(k))$ et $Le_2^{(i)}(u(k))$ des bits $u(k)$ à l'itération ' i '. Le critère de l'entropie croisée utilise les LLR à la sortie des décodeurs pour fournir une information sur la fin du décodage [3] [26]

On définit la différence des informations extrinsèques entre l'itération i et $i-1$ à la sortie du deuxième décodeur par [26] :

$$\Delta Le_2^{(i)}(u(k)) = Le_2^{(i)}(u(k)) - Le_2^{(i-1)}(u(k)) \quad (\text{III.6})$$

La même quantité peut être calculée par [26] :

$$\Delta Le_2^{(i)}(u(k)) = L_2^{(i)}(u(k)) - L_1^{(i)}(u(k)) \quad (\text{III.7})$$

Selon [26], l'entropie croisée de l'itération (i) est donnée par :

$$CE(i) \approx \sum_n \frac{|\Delta Le_2^{(i)}(u(k))|^2}{e^{|L_1^{(i)}(u(k))|}} \quad (\text{III.8})$$

La trame est jugée correcte si:

$$CE(i) < \varepsilon \quad (\text{III.9})$$

Il est recommandé [26] que le seuil ε soit dans l'intervalle :

$$10^{-2} CE(1) \leq \varepsilon \leq 10^{-4} CE(1) \quad (\text{III.10})$$

$CE(1)$ Est l'entropie croisée de la première itération.

Pour un turbo récepteur en série [27], prenons par exemple un Turbo égaliseur. Le traitement T_1^{-1} est un égaliseur. T_2^{-1} est un décodeur MAP. Dans ce cas, T_2^{-1} peut fournir les

informations extrinsèques $Le_2^{(i)}(u(k))$ des bits d'information $u(k)$. Tandis que T_1^{-1} se charge de l'estimation des LLR $L(c(n))$ des symboles codés et mappés $c(n)$ mais ne peut pas avoir une information sur les bits $u(k)$. On peut appliquer le critère de l'Entropie Croisée aux symboles mappés codés $c(n)$ puisque le décodeur MAP peut aussi estimer les Log-rapports de vraisemblance LLR des bits codés. Dans ce cas, le critère devient :

Pour chaque trame traitée, calculer l'Entropie Croisée des symboles $c(n)$ par [27] [3] [26] :

$$CE(i) = \sum_n \frac{|\Delta Le_2^{(i)}(c(n))|^2}{e^{|L_1^{(i)}(c(n))|}} \quad (\text{III.11})$$

Où :

$$\Delta Le_2^{(i)}(c(k)) = Le_2^{(i)}(c(k)) - Le_2^{(i-1)}(c(k)) \quad (\text{III.12})$$

La trame est décidée correcte si :

$$CE(i) < \varepsilon \quad (\text{III.13})$$

Avec toujours [27]:

$$10^{-2}CE(1) \leq \varepsilon \leq 10^{-4}CE(1) \quad (\text{III.14})$$

III.2 Critère de la moyenne des LLR

Dans le décodage turbo, le critère basé sur la moyenne des LLR (Log-Likelihood Ratios) permet d'évaluer la fiabilité globale des bits estimés à chaque itération. Ce critère calcule la moyenne des valeurs absolues des LLR des bits d'information. Si cette moyenne dépasse un seuil prédéfini θ , cela signifie que les bits sont globalement fiables et que le décodeur a probablement convergé. On peut alors interrompre le processus d'itération afin de réduire le temps de calcul sans compromettre la qualité du décodage.

L'idée ici est de mesurer la fiabilité globale des bits transmis en calculant la moyenne des valeurs absolues des LLR (Log-Likelihood Ratio) tel que décrit dans [28] et présenté de manière similaire dans [29] après chaque itération i . La formule indique que l'on additionne la valeur absolue de chaque LLR pour chaque bit, puis on divise par le nombre total de bits K . Cette approche permet d'obtenir une estimation de la fiabilité moyenne des bits à chaque itération. Si cette moyenne dépasse un seuil T , cela signifie que les bits sont suffisamment fiables et l'itération

peut être arrêtée. C'est une méthode simple pour savoir quand le processus de décodage a convergé. [30]

$$\mu_i = \frac{1}{K} \sum_{k=1}^K |LLR_k^i| \quad (\text{III.15})$$

Avec : k : nombre totale des bits

$|LLR_k^i|$: La fiabilité (LLR) du bit à l'instant k de l'itération i.

- **Critères d'arrêt** : Si la moyenne μ_i dépasse un seuil T (valeur fixée à l'avance), on arrête les itérations.

Cela signifie que les bits sont considérés comme assez fiables.

- **Formule d'arrêt** :

$$\mu_i > T \quad (\text{III.16})$$

III.3 Critère Minimum LLR

Dans cette méthode, l'accent est mis sur le bit le moins fiable parmi tous les bits. Ici, l'on cherche à déterminer quel est le LLR le moins plus faible en valeur absolue, ce qui correspond au bit pour lequel l'incertitude est la plus grande. Si ce minimum dépasse également un seuil T, cela signifie que même le bit le moins fiable est suffisamment fiable pour arrêter les itérations. Cette approche est particulièrement utile pour identifier rapidement les moments où tous les bits sont devenus suffisamment fiables, même dans les cas où un seul bit pourrait être considéré comme incertain. [30]

- **Critères d'arrêt** : si ce minimum dépasse aussi le seuil T, ça veut dire même le pire bit est fiable, donc on arrête.

- **Formule d'arrêt** :

$$\min (LLR(k)) > T \quad (\text{III.17})$$

III.4 Critère de la variance des LLR extrinsèques

Un critère d'arrêt efficace repose sur l'analyse de la variance des LLR extrinsèques (Log-Likelihood Ratios) produits à chaque itération du décodage turbo. Lorsque cette variance dépasse un certain seuil, cela indique que les informations échangées entre les décodeurs n'apportent plus de gain significatif, ce qui justifie l'arrêt prématuré du processus itératif. D'après les résultats

expérimentaux de [31], plus le nombre d'itérations est élevé, plus la variance de l'information extrinsèque augmente.

La variance est définie comme l'espérance du carré de l'écart entre une LLR extrinsèque L_{e_m} et sa moyenne μ , selon la formule :

$$\text{Var}(L_{e_m}) = E[(L_{e_m} - \mu)^2] \quad (\text{III.18})$$

Où L_{e_m} représente la LLR extrinsèque issue du décodeur $m = 1$ ou 2 , et μ est sa moyenne

Même après avoir soustrait la moyenne, la variance continue d'augmenter car les LLR extrinsèques sont des valeurs réelles dont la moyenne est généralement faible. Par ailleurs, l'amplitude de ces LLR augmente avec le nombre d'itérations.

Les résultats sont présentés dans les Figures III.1 et III.2. $\text{Var}(L_{e2})$ Représente la variance de l'information extrinsèque issue du deuxième décodeur composant. Après un certain nombre d'itérations, les variances $\text{Var}(L_{e2})$ pour chaque valeur de SNR se stabilisent à certaines valeurs, et les performances en termes de BER se détériorent à mesure que les variances augmentent. Par conséquent, le seuil pour ce critère devrait être déterminé et ajusté en fonction des différentes valeurs de SNR. [21]

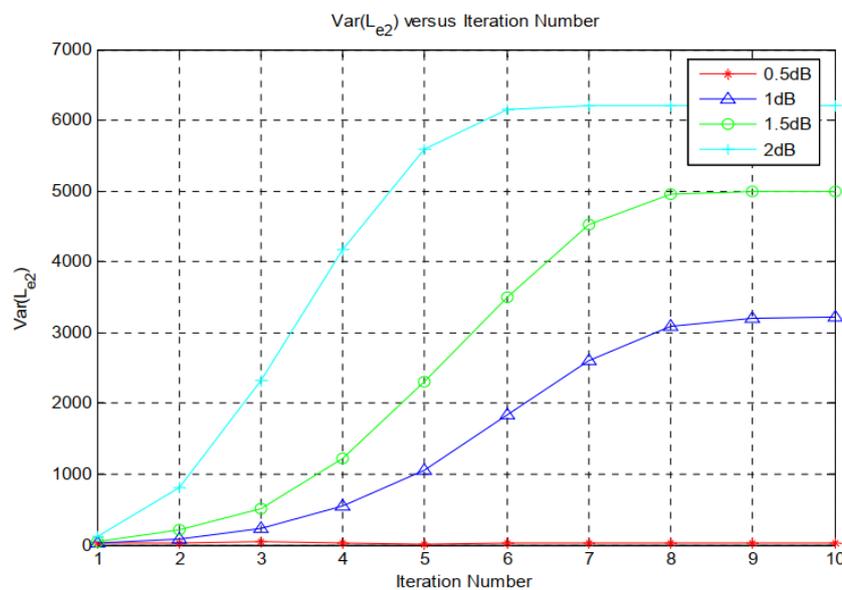


Figure III.1 : La variance des informations extrinsèques par rapport au nombre d'itérations [21]

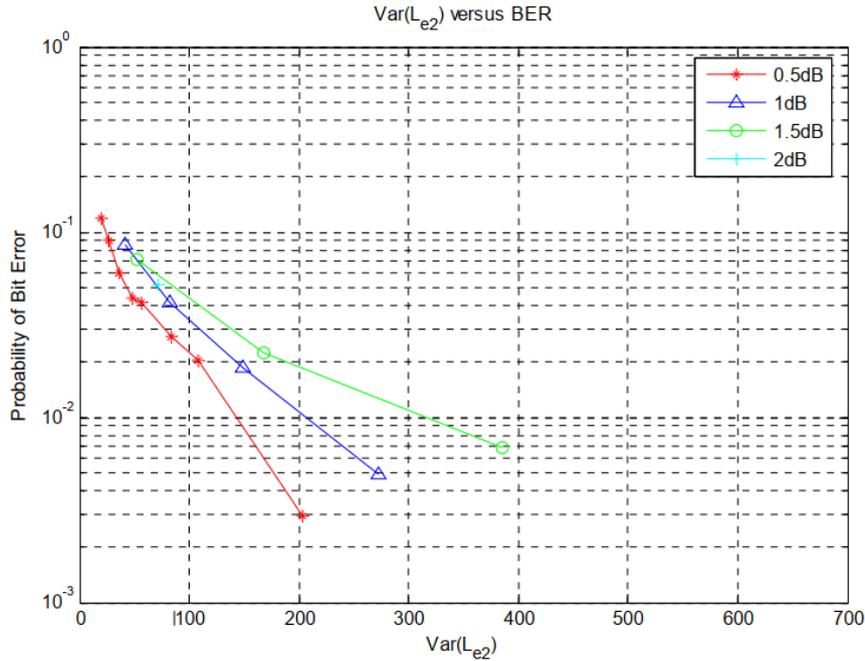


Figure III.2 : La variance de l'information extrinsèque par rapport au BER [21]

Cela justifie l'utilisation d'un seuil d'arrêt ajusté en fonction du SNR. La stratégie d'arrêt peut être formulée ainsi :

$$Var(L_{e2}^{(1)}) \cdot k < Var(L_{e2}^{(i)}) \tag{III.19}$$

Où k est une constante empirique, et i le numéro d'itération.

III.5 Critère Sum-Alpha

[32] propose un critère d'arrêt très simple pour le décodage Turbo, basé sur le comportement des valeurs probabilistes $\alpha_m(t)$ calculées lors de la récursion directe (forward recursion). Ce critère n'utilise pratiquement aucun espace mémoire.

À chaque instant t , on appelle $\alpha_{max}(t)$ la probabilité de l'état ayant la valeur alpha la plus élevée. Cela signifie que :

$$\alpha_{max}(t) = \max(\alpha_m(t)) \tag{III.20}$$

On appelle les autres valeurs alpha des concurrentes.

Les observations suivantes ont été faites lors de simulation répétées [32] :

- Les instants corrects sont caractérisés par une forte valeur de $\alpha_{max}(t)$ pour l'état le plus probable. Les probabilités alpha des autres états sont très faibles.

- Les instants erronés sont caractérisés par des valeurs significatives des alphas concurrentes, ce qui diminue la valeur de $\alpha_{max}(t)$.
- Les trames correctes sont donc caractérisées par une $\alpha_{max}(t)$ élevée et des alphas concurrentes très faibles. La somme des alphas concurrents reste faible.
- Les trames erronées sont caractérisées par des alphas concurrents ayant des valeurs de probabilité importantes. L'augmentation de la valeur des alphas concurrents diminue la probabilité de l'état le plus probable $\alpha_{max}(t)$.
- La diminution des valeurs concurrentes peut être utilisée pour signaler des trames correctes.

À partir de ces observations, nous pouvons formuler un critère permettant de détecter les trames correctes et ainsi d'arrêter le processus de décodage Turbo. Ce critère est appelé "Sum- α ". Il consiste à :

1. À chaque instant t , après le calcul des alphas dans la direction avant de l'algorithme MAP, calculer la somme des alphas concurrentes, c'est-à-dire :

$$Sum_c(t) = \sum_{m=1}^M \alpha_m(t) - \alpha_{max}(t) \quad (\text{III.21})$$

On peut également écrire :

$$Sum_c(t) = 1 - \alpha_{max}(t) \quad (\text{III.22})$$

Car la somme de tous les alphas à un instant donné est égale à 1 :

$$\sum_{m=1}^M \alpha_m(t) = 1 \quad (\text{III.23})$$

2. Calculer la somme de tous les $Sum_c(t)$ sur l'ensemble des instants et la normaliser par la longueur de la trame N :

$$Sum_\alpha = \frac{1}{N} \sum_{t=1}^N Sum_c(t) \quad (\text{III.24})$$

3. Si Sum_α est inférieure à un seuil T , alors on arrête le processus Turbo, c'est-à-dire :

Si $Sum_\alpha < T$, alors arrêter les itérations.

Concernant le seuil T , une valeur élevée peut considérablement réduire le nombre moyen d'itérations, car elle provoquera un arrêt rapide du décodage Turbo. Cependant, cela entraînera une dégradation du taux d'erreur binaire (BER).

Pour un entrelaceur de taille $N = 5120$, [32] montre que $T = 0.001$ est une valeur adéquate.

Cette technique n'utilise qu'une seule mémoire (ce qui est un grand avantage), car la somme jusqu'à l'instant l est calculée par :

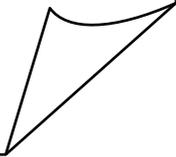
$$\sum_{t=1}^l Sum_c(t) = \sum_{t=1}^{l-1} Sum_c(t) + Sum_c(l) \quad (\text{III.25})$$

L'implémentation du critère d'arrêt $Sum-\alpha$ est plus simple en termes d'espace mémoire requis. [32] montre aussi qu'il offre les mêmes performances que celles du critère d'arrêt basé sur l'entropie croisée (Cross-Entropy, CE). De plus, le critère $Sum-\alpha$ est considéré comme un critère précoce [32] car il offre une information sur la qualité du décodage même avant le calcul des log-rapport de vraisemblance LLR.

Chapitre IV

Résultats de

Simulation



IV.1 Introduction :

Ce chapitre vise à évaluer les performances d'un Turbo décodeur MAP $[1, 5/7]_{oct}$ à l'aide des simulations MATLAB. Il explore également l'application du critère d'arrêt à base des décisions souples (Soft) dans le processus de décodage Turbo MAP, ainsi que l'évaluation de ses performances.

IV.2 Performances du critère d'arrêt de l'Entropie Croisée EC appliqué au turbo décodage :**IV.2.1 Conditions de simulation :**

Turbo codeur : $[1, 5/7]_{oct}$

Rendement : $R = 1/3$

Nombre de paquets = 1000

Rapport signal sur bruit RSB minimal = 0 dB.

Rapport signal sur bruit RSB maximal = 0.8dB

Longueur des trames $L = 1024$

Algorithme de décodage : MAP

Canal de transmission : Canal gaussien AWGN sans interférences entre symboles IES.

IV.2.2 Résultats de simulation :

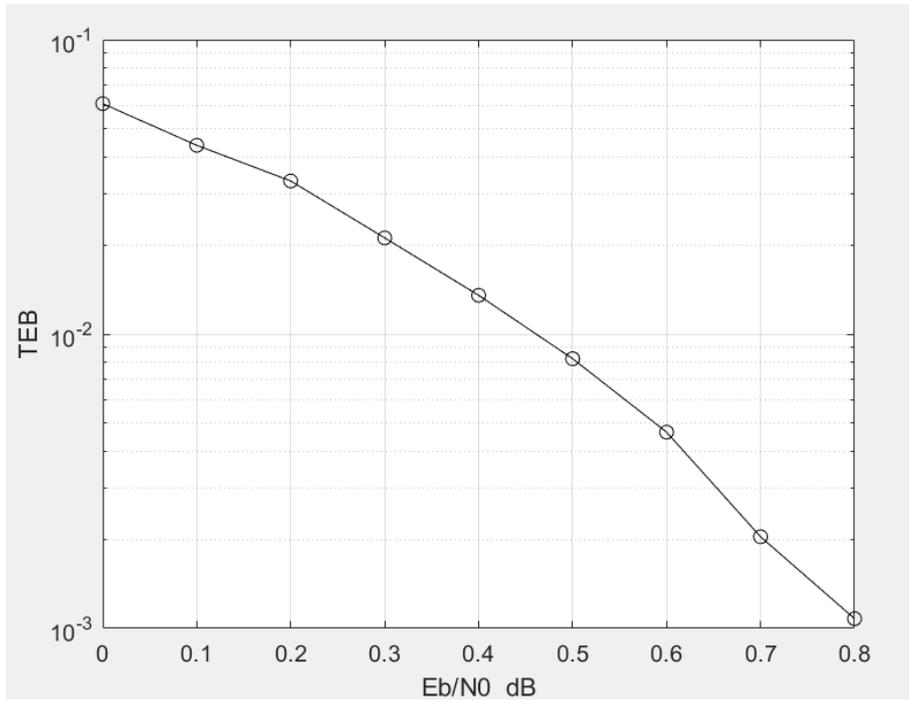


Figure IV.1: Performances du critère de l'entropie croisée EC

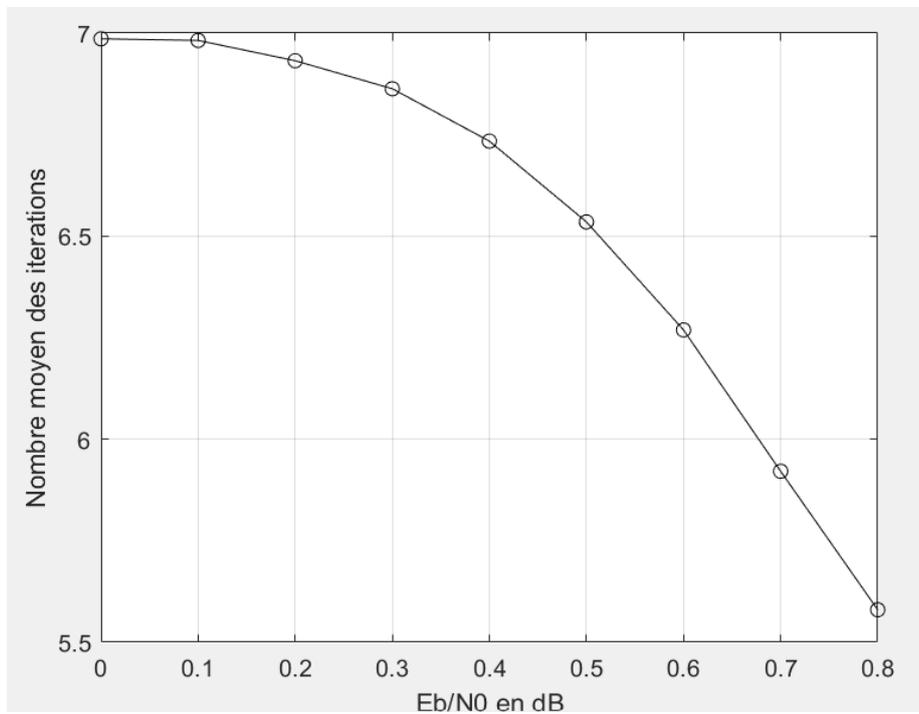


Figure IV.2: Nombre moyen des itérations du critère de l'entropie croisée EC

IV.2.3 Interprétations des résultats

Les résultats montrent que le critère d'arrêt basé sur l'entropie croisée permet de maintenir de bonnes performances de décodage. Le TEB diminue progressivement avec l'augmentation de E_b/N_0 , ce qui confirme l'efficacité du décodage MAP avec ce critère.

En parallèle, le nombre moyen d'itérations baisse quand la qualité du signal s'améliore, passant d'environ 7 à moins de 6. Cela prouve que le critère EC permet de stopper les itérations dès que la convergence est atteinte, évitant ainsi des calculs inutiles.

Ce critère offre donc un bon compromis entre performance, rapidité de décodage et réduction de la complexité, ce qui est particulièrement avantageux dans les systèmes à ressources limitées.

IV.3 Performances du critère de la moyenne des LLR appliqué au turbo décodage :

IV.3.1 Conditions de simulation :

Turbo codeur : $[1,5/7]_{oct}$

Rendement : $R = 1/3$

Nombre de paquets = 1000

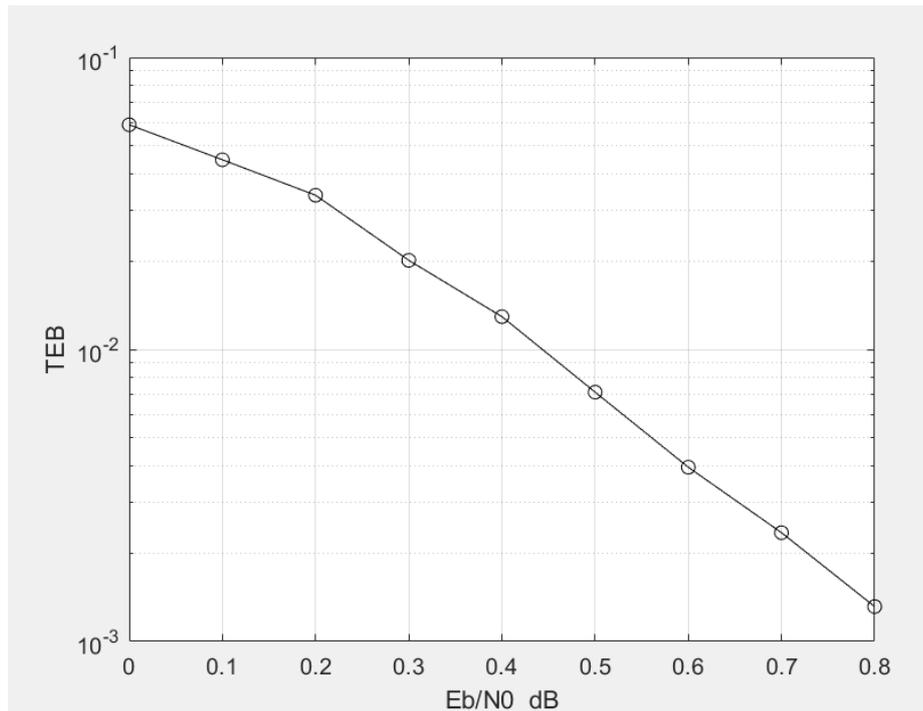
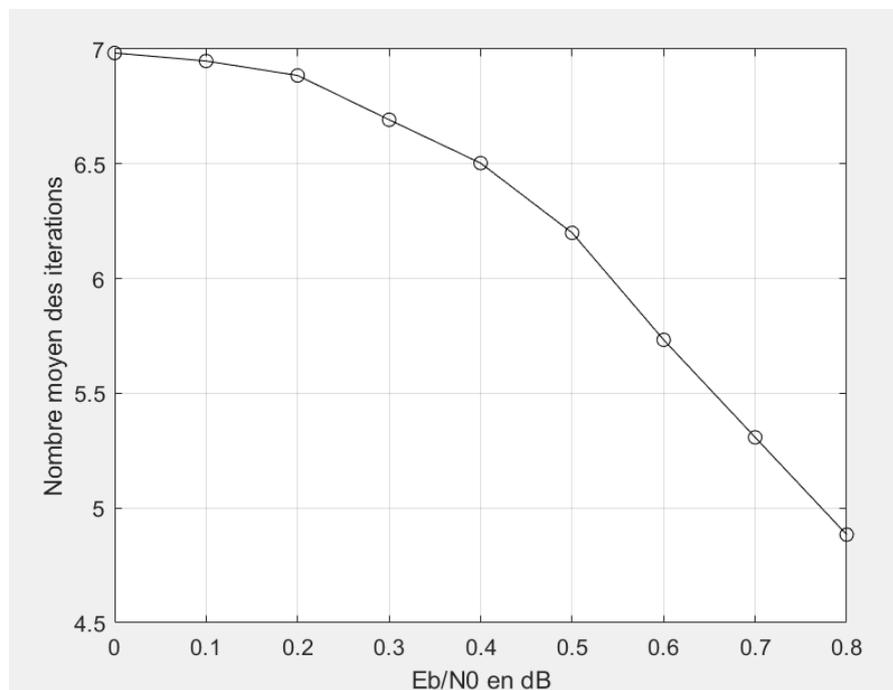
RSB minimal = 0dB

RSB maximal = 0.8dB

Longueur des trames $L = 1024$

Seuil du critère d'arrêt de la moyenne des LLR : Seuil = 20

IV.3.2 Résultats de simulation :

**Figure IV.3 :** Performances du critère de la moyenne des LLR**Figure IV.4 :** Nombre moyen des itérations du critère de la moyenne des LLR

IV.4 Justification du choix du seuil du critère de la moyenne des LLR :

Pour justifier le choix du seuil du critère de la moyenne des LLR [29], nous proposons d'estimer les densités de probabilité des quantités calculées par ce critère pour chaque itération en fixant un rapport signal sur bruit moyen. Les trames ciblées sont les trames correctes. Les seuils dépendent fortement de la taille L de l'entrelaceur.

IV.4.1 Conditions de simulation :

Nombre de paquets = 5000

RSB minimal = 0dB

RSB maximal = 0.8dB

Longueur des trames L= 1024

Seuil du critère d'arrêt de la moyenne des LLR : Seuil = 20000000000 (+ ∞)

IV.4.2 Résultats de simulation :

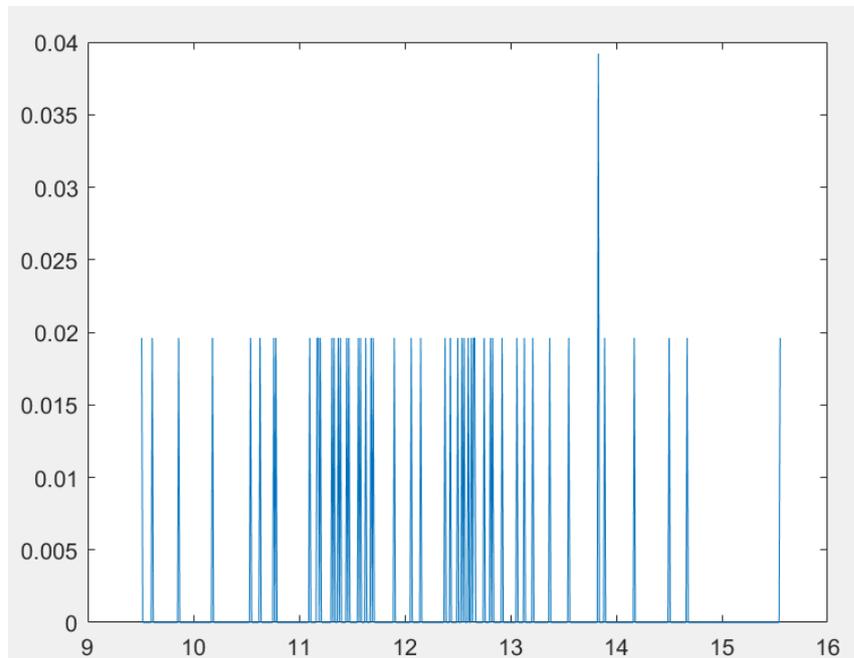


Figure IV.5 : Distribution de la moyenne des LLR à la 3ème itération

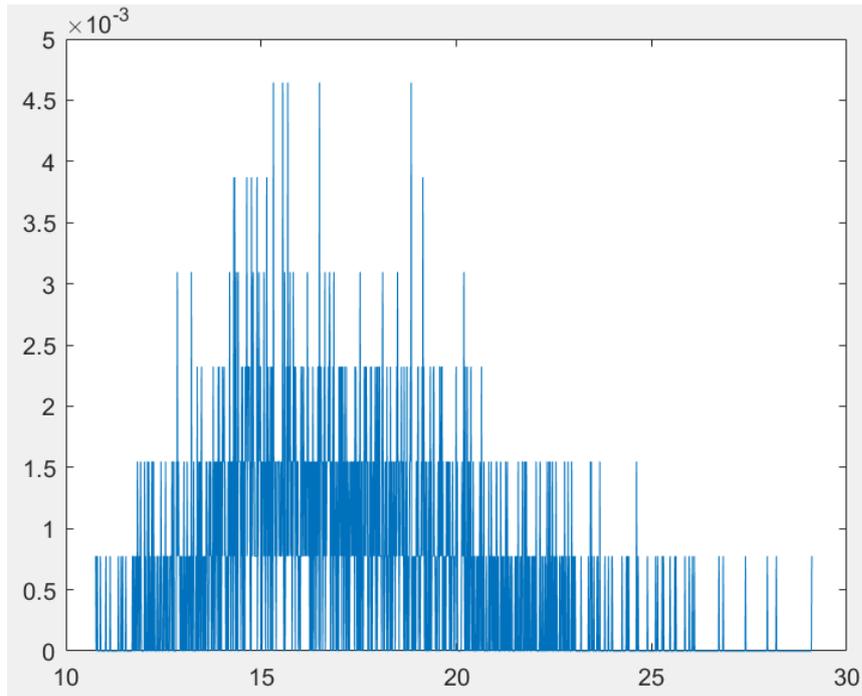


Figure IV.6 : Distribution de la moyenne des LLR à la 4 ème itération

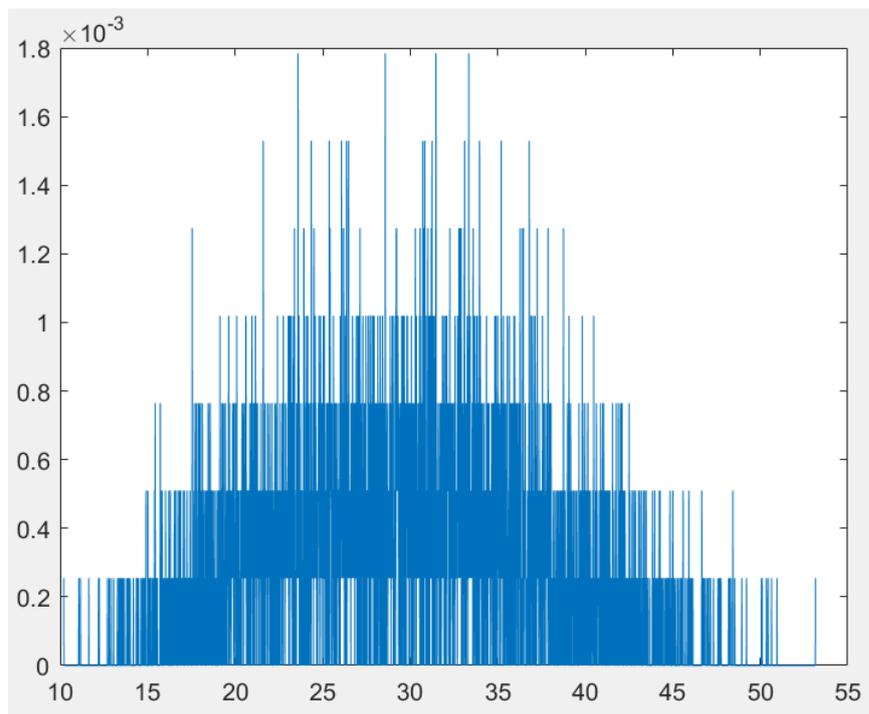


Figure IV.7 : Distribution de la moyenne des LLR à la 5 ème itération

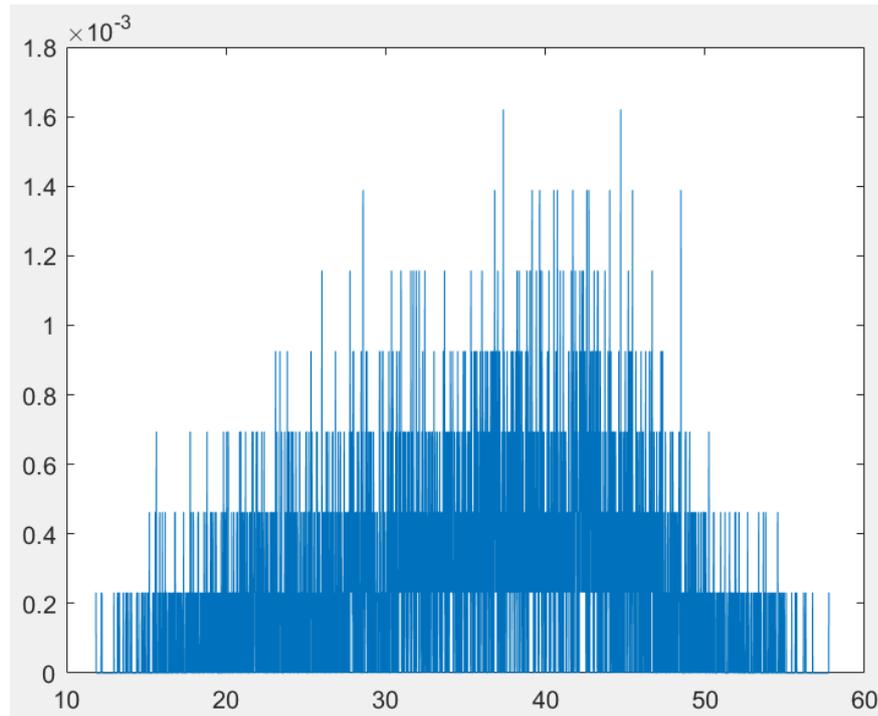


Figure IV.8 : Distribution de la moyenne des LLR à la 6ème itération

Ces résultats montrent que le seuil doit être supérieur à 10 et inférieur à 32. Nous avons choisi la valeur 20.

IV.5 Performances du critère Minimum des LLR :

IV.5.1 Conditions de simulation :

Nombre de paquets = 1000

RSB minimal = 0 dB

RSB maximal = 0.8 dB

Longueur des trames $L = 1024$

Seuil du critère d'arrêt Minimum des LLR : Seuil = 7

IV.5.2 Résultats de simulation :

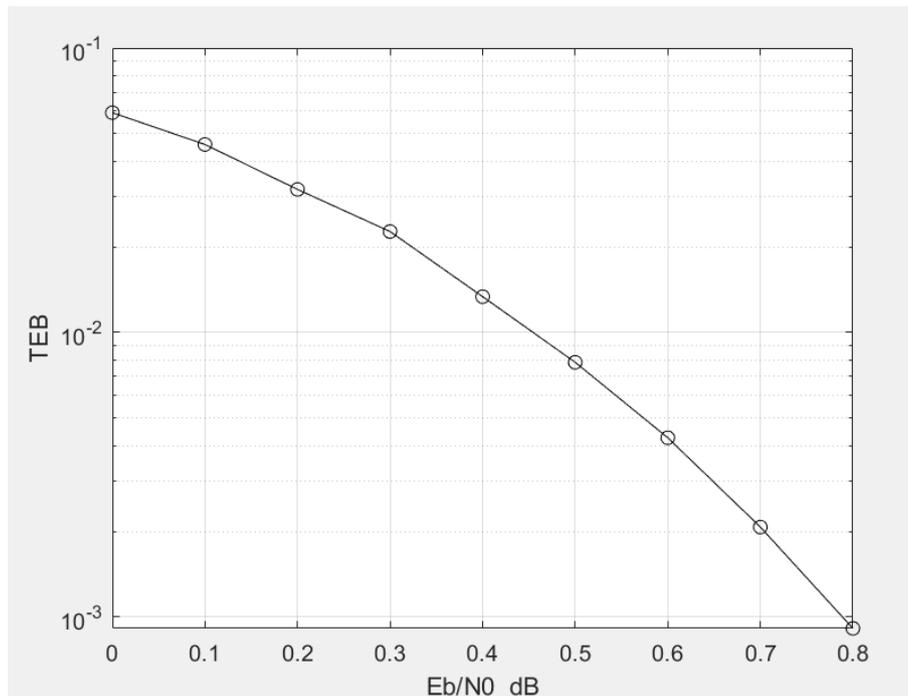


Figure IV.9 : Performances du critère Minimum des LLR

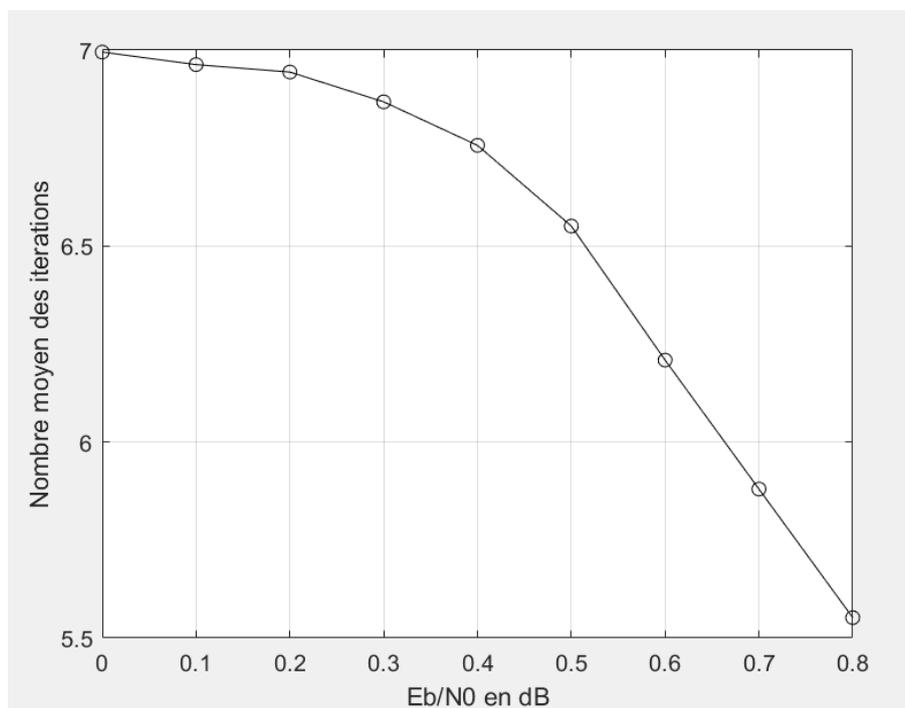


Figure IV.10 : Nombre moyen des itérations du critère Minimum des LLR

IV.6 Justification du choix du seuil du critère du Minimum des LLR :

De même, dans cette partie, nous traçons les densités de probabilité des quantités calculées par le critère du minimum des LLR pour chaque itération en fixant un rapport signal sur bruit $RSB = 0.6$ dB.

IV.6.1 Conditions de simulation :

Nombre de paquets = 5000

RSB minimal = 0.6dB

RSB maximal = 0.6dB

Longueur des trames $L = 1024$

Seuil du critère d'arrêt Minimum des LLR : Seuil = $20000000000 (+ \infty)$.

IV.6.2 Résultats de simulation :

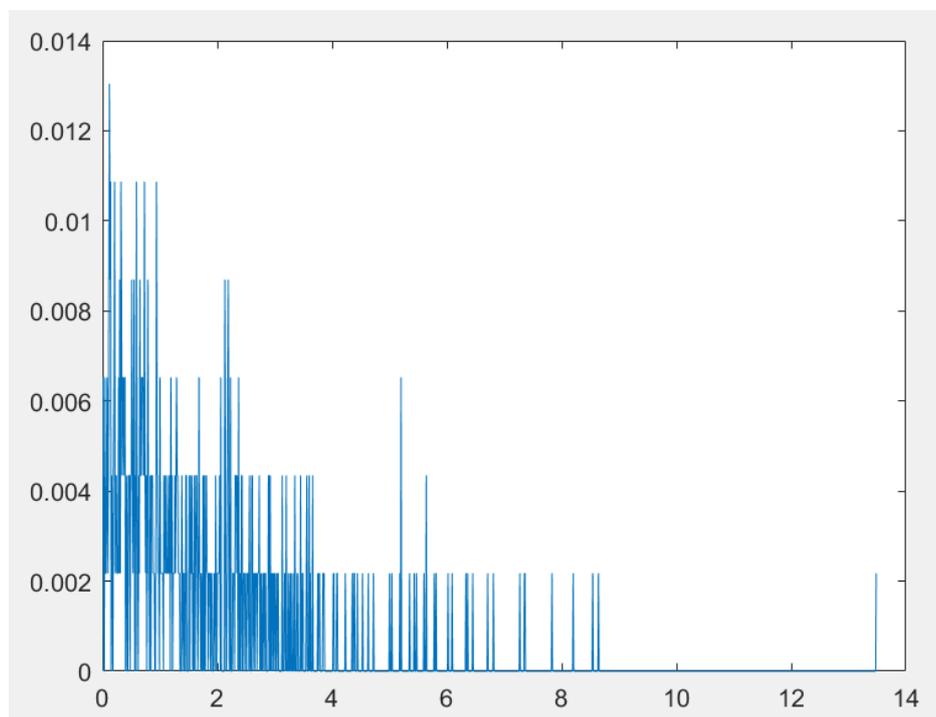


Figure IV.11 : Distribution du minimum des LLR (Seuil) à la 4ème itération

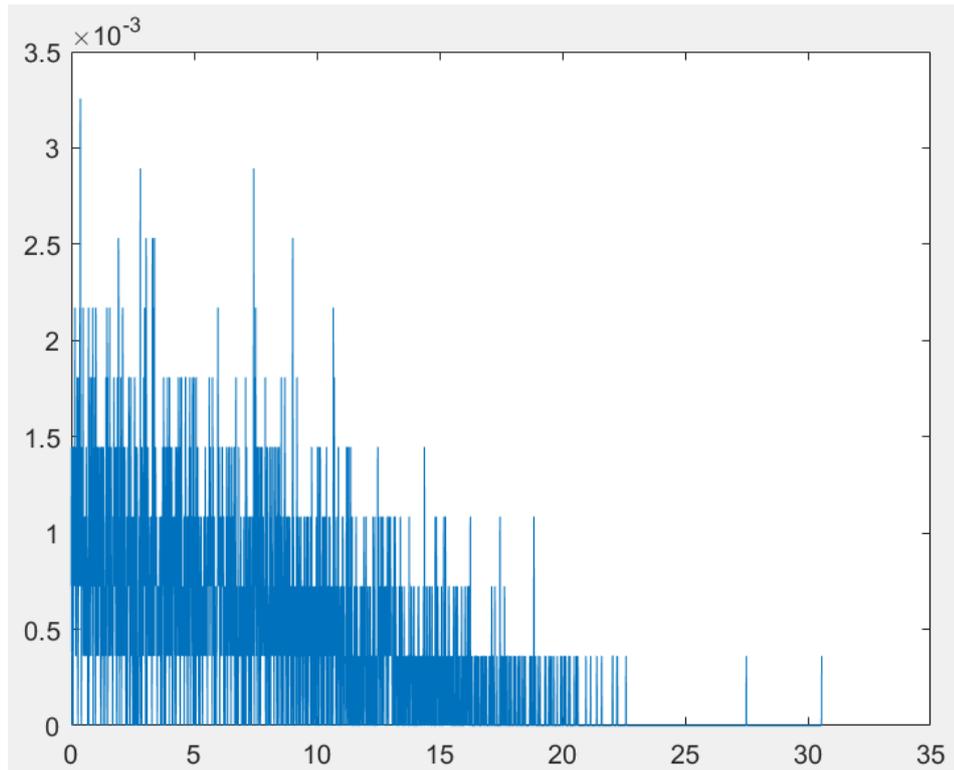


Figure IV.12 : Distribution du minimum des LLR (Seuil) à la 5ème itération

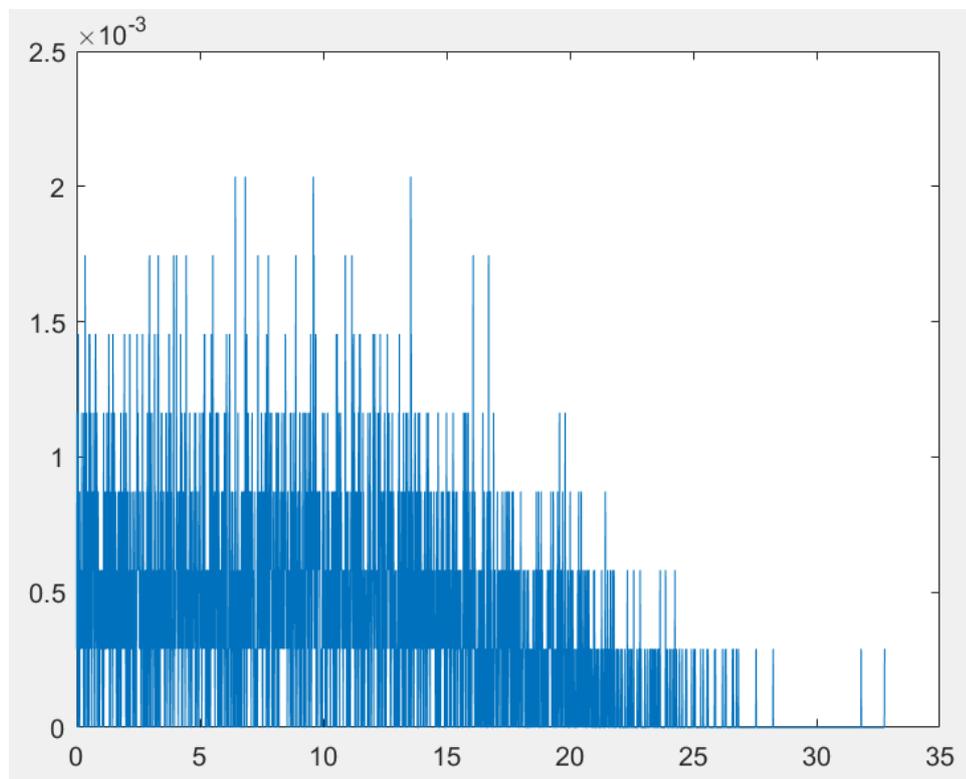


Figure IV.13 : Distribution du minimum des LLR (Seuil) à la 6ème itération

Ces résultats montrent que le seuil doit être supérieur à 0 et inférieur à 25. Nous avons choisi la valeur 7.

IV.7 Performances du critère de la variance des LLR extrinsèques :**IV.7.1 Conditions de simulation :**

Nombre de paquets = 1000

RSB minimal = 0 dB

RSB maximal = 0.8 dB

Longueur des trames $L= 1024$

Seuil du critère d'arrêt de la variance des LLR extrinsèques : Seuil=27

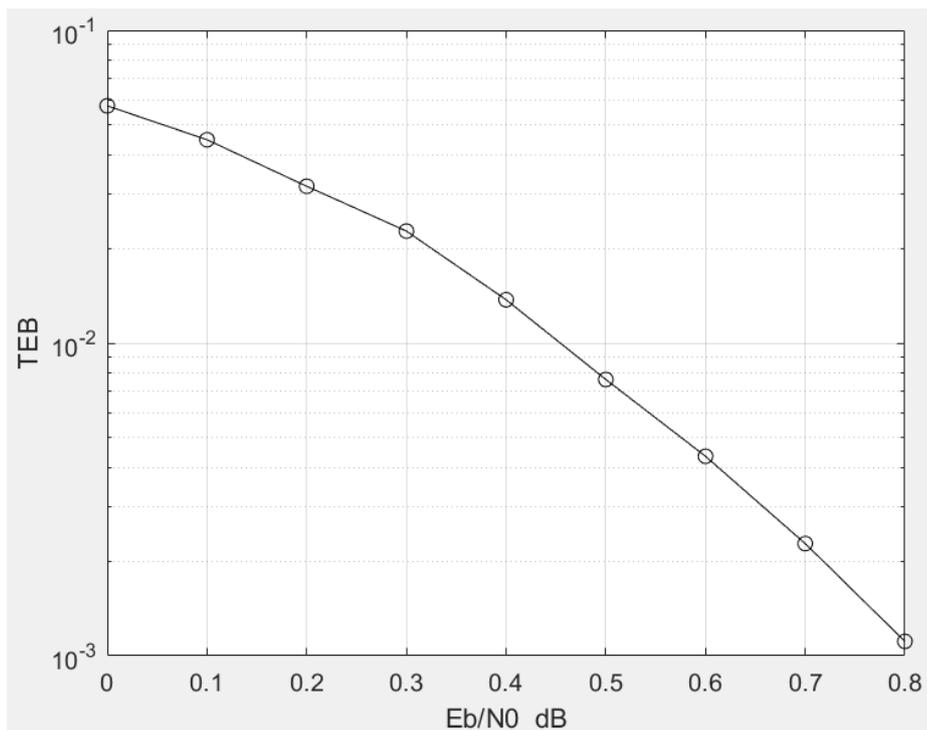
IV.7.2 Résultats de simulation :

Figure IV.14 : Performances du critère de la variance des LLR extrinsèques

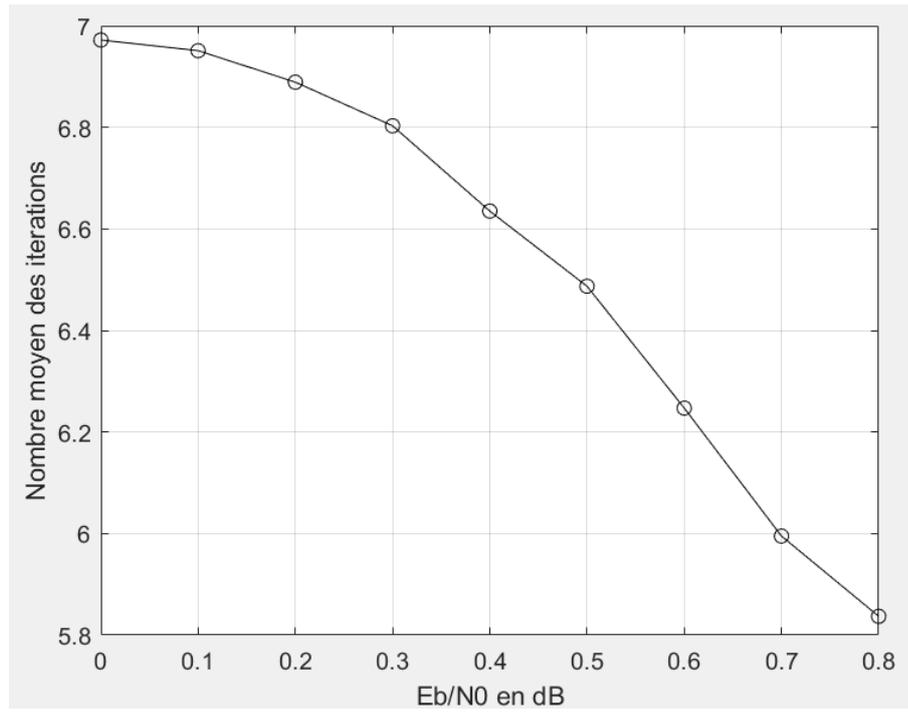


Figure IV.15 : Nombre moyen des itérations du critère de la variance des LLR extrinsèques

IV.8 commentaires et comparaison :

Les résultats de simulation ont montré un bon fonctionnement des quatre critères d'arrêt souples : Entropie croisée, Moyenne des LLR, Minimum des LLR et Variance des LLR. Le nombre moyen des itérations est inférieur au nombre maximal fixé qui est égal à 7. Pour faire une comparaison, nous choisissons le critère EC comme une référence. Prenons par exemple le rapport signal sur bruit 0.8 dB. En utilisant la figure IV.1, le taux d'erreur binaire TEB est égal à $TEB = 10^{-3}$. Le turbo décodeur utilise à la moyenne 5.6 itérations. Le tableau suivant compare ces performances avec celles des autres critères.

Critère d'arrêt	EC	Moyenne des LLR	Minimum des LLR	Variance des LLR
TEB	10^{-3}	$2 \cdot 10^{-3}$	10^{-3}	$1.1 \cdot 10^{-3}$
Nombre moyen des itérations	5.6	4.9	5.51	5.82

Tableau IV.1 : Comparaison des performances obtenues au rapport signal sur bruit 0.8 dB

Ces résultats montrent que pour un entrelaceur de longueur 1024 et aux forts rapports signal sur bruit, le critère de la moyenne des LLR donne la plus faible complexité (4.9 itérations).

CONCLUSION GENERALE

CONCLUSION GENERALE :

Dans un contexte où les communications numériques ne cessent d'évoluer, assurer la fiabilité de la transmission des données demeure un enjeu majeur. Les turbocodes, reconnus pour leur efficacité remarquable en correction d'erreurs, ont largement contribué à améliorer les performances des systèmes de communication modernes. Grâce à leur capacité à s'approcher de la limite théorique de Shannon, ils sont devenus un pilier incontournable dans plusieurs applications, notamment dans les domaines des réseaux mobiles et des communications par satellite.

Ce travail s'est principalement concentré sur l'étude des critères d'arrêt du turbo décodage, et plus particulièrement sur ceux basés sur les quantités souples (Soft). Ces critères permettent de mettre fin au processus itératif de décodage de manière intelligente, en tenant compte du niveau de fiabilité des bits décodés. Contrairement aux critères classiques, basés sur des décisions dures, les critères soft exploitent des mesures probabilistes comme les LLR (Log-Likelihood Ratios), ce qui offre un meilleur compromis entre complexité et performance.

À travers l'analyse des différents critères existants et la mise en œuvre de simulations, nous avons pu observer l'impact significatif de ces approches sur la réduction du nombre d'itérations tout en maintenant une qualité de décodage satisfaisante. Les résultats obtenus confirment l'intérêt d'intégrer ces critères souples 'Soft' dans les systèmes de turbo décodage pour une optimisation globale des ressources.

En perspective, plusieurs axes peuvent être envisagés pour enrichir cette étude : l'implémentation et l'évaluation du critère récent Sum-Alpha présenté dans le chapitre 3, la proposition de critères hybrides combinant soft et hard, l'adaptation dynamique des seuils en fonction de la longueur de l'entrelaceur et du canal, ou encore l'exploration de techniques d'intelligence artificielle pour prédire la convergence du décodage et arrêter le traitement. Ces

pistes pourraient contribuer à rendre le turbo décodage encore plus performant et adaptable aux futures exigences des systèmes de communication.

Chapitre I

- [1]: Claude Berrou, "Code et Turbocodes". Springer Paris Berlin Heidelberg New York. Paris, "Ecole nationale supérieure des Télécommunications des Bretagne", springer-verlag France (2007) France.
- [2]: Hendaoui Mounira et Benakcha Abdelhamid. "Courrier du savoir scientifique et technique". Applications Des turbocodes dans un système multi usagers Wcdma, 2014
- [3]: Bouzidi Hanane, Mostefai chaimaa, "Critère de L'entropie croisée et son application au turbo décodage", Mémoire Master, Université Dr. Tahar Moulay, 2023.
- [4]: LATIFA Mostari, " Codage de canal ", Polycoié de cours et travaux dirigés, Université Hassiba Ben Bouali, Chlef, 2016/2017.
- [5]: Baghdad Ibtissem, "Critères d'arrêt Turbo à base des décisions dures", Mémoire Master, Université Dr. Tahar Moulay, 2024.
- [6]: Abdelghafour Berraissoul, "codage de canal : une introduction aux codes correcteurs ». Ellipses, (2021)
- [7]: Hamidane Assia, Lomani Cherifa, "Evaluation de la capacité des canaux Rayleigh et UWB pour un système SISO et MIMO ", Mémoire de Master, Université Saad Dahlab de blida, 2011/2012
- [8]: Zeggai Zeyd, Zerrouk Nassim, "Conception d'une chaîne de transmissions de données cryptées sur logiciel LABVIEW", Mémoire de Master, Université Abou Bekr Belkaid de Tlemcen, 2020/2021
- [9]: Dr Ouardi.A , Cours communication numérique Licence 3 , Université Dr Tahar Moulay saida
- [10]: W. E. Ryan and S. Lin, Channel Codes: Classical and Modern. Cambridge, UK: Cambridge University Press, 2009.
- [11]: C. Berrou, Codes et turbocodes. New York: Springer-Verlag France, 2007
- [12]: Bouazza Rabie Lakhdar, Kaoudj salima" Critères d'arrêt CRC pour les turbocodes et son double intérêt pour les systèmes H-ARQ ", Mémoire Master, Université Dr Tahar Moulay Saïda ,2021/2022
- [13]: Djaouhara Rahmouni, "Codes convolutifs et décodage MAP "
- [14]: Achour Chahinez, Chabane Nawel, "Les Turbo Décodeurs TBCJR-LOG MAP", Mémoire de Master, Université Dr Tahar Moulay Saïda 2022/2023.

[15]: J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," Global Telecommunications Conference and Exhibition 'Communications Technology for the 1990s and Beyond' (GLOBECOM), 1989. IEEE, Dallas, TX, 1989, pp. 1680-1686 vol.3.

[16]: L. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate ", IEEE Transactions on Information Theory, vol. 20, no. 2, pp. 284-287, 1974.

Chapitre II

[17]: A. Ouardi, "Sum-Log Stopping Criterion for Log-MAP Turbo Decoding," Advances in Electrical and Computer Engineering, vol. 23, no. 1, 2023

[18]: Imed Amamra, «Codage canal et techniques efficaces de décodage itératif ». Thèse de doctorat, Université 20 Août 1955 de Skikda, 2018.

[19]: K. R. Narayanan et G. L. Stuber, "A novel ARQ technique using the turbo coding principle," in IEEE Communications Letters, vol. 1, no. 2, pp. 49-51, March 1997

[20]: Lajnef.K, "Étude des performances des codes turbo" Mémoire de maîtrise, École Polytechnique de Montréal, PolyPublie (2001).

[21]: Kai-Jen Cheng, "Comparison and Analysis of Stopping Rules for Iterative Decoding of Turbo Codes", Master of Science, the faculty of the Russ College of Engineering and Technology of Ohio University, June 2008

[22]: R. Y. Shao, S. Lin, and M. P. C. Fossorier, "Two simple stopping criteria for turbo decoding," IEEE Transactions on Communications, vol. 47, no. 8, , Aug. 1999.

[23]: Y. Wu, B. D. Woerner and W. J. Ebel, "A simple stopping criterion for turbo decoding," in *IEEE Communications Letters*, vol. 4, no. 8, pp. 258-260, Aug. 2000, doi: 10.1109/4234.864187.

[24]: Chun Ling Kei and Wai Ho Mow, "Improved stopping criteria for iterative decoding of short-frame multi-component turbo codes", 2002, vol. 1

[25]: M. E. Buckley and S. B. Wicker, "A neural network for predicting decoder error in turbo decoders", in IEEE Communications Letters, vol. 3, no. 5, 1999

Chapitre III

- [26]: J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary block and convolutional codes", IEEE Transactions on Information Theory, vol. 42, no. 2, Mar. 1996.
- [27]: A. Ouardi, B. S. Bouazza, "Complexity Reduction of Turbo Equalization Using Cross-Entropy Stopping Criterion", JTEC, vol. 13, no. 2, Jun. 2021.
- [28]: A. Matache, S. Dolinar, and F. Pollara, "Stopping Rules for Turbo Decoders," TMO Progress Report 42-142, Aug. 2000, [http://tda.jpl.nasa.gov/progress report/](http://tda.jpl.nasa.gov/progress%20report/), Jet Propulsion Laboratory, Pasadena, California
- [29]: F. Zhai and I. J. Fair, "New Error Detection Techniques and Stopping Criteria for Turbo Decoding," in Proc. 2000 IEEE Canadian Conference on Electrical and Computer Engineering, Halifax, Canada, Mar. 2000.
- [30]: Frank Gilben, Frank Kienle, Norbert Wehn, "Low Complexity Stopping Criteria for UMTS Turbo-Decoders" , Microelectronic System Design Research Group, University of Kaiserslautern Erwin-Schroedinger-Strasse, 67663 Kaiserslautern, Germany.
- [31] : B. Kim and H. S. Lee, "Reduction of the Number of Iterations in Turbo Decoding Using Extrinsic Information," in Proceedings of IEEE TENCON 99, Inchon, South Korea, 1999.
- [32]: Kai-Jen Cheng, "Comparison and Analysis of Stopping Rules for Iterative Decoding of Turbo Codes", Master of Science, the faculty of the Russ College of Engineering and Technology of Ohio University, June 2008
- [33]: A. Ouardi, "Sum- α stopping criterion for turbo decoding," International Journal of Electronics and Telecommunications, vol. 67, no. 3, 2021.