

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي و البحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة الدكتور الطاهر مولاي سعيدة

Université Saida Dr Tahar Moulay

Faculté de TECHNOLOGIE



MEMOIRE

Projet de recherche présenté pour l'obtention du Diplôme de MASTER

En : Télécommunications

Spécialité : Systèmes et Télécommunications

Par : BAGHDAD IBTISSEM

Sujet

Critères d'arrêt Turbo à base des décisions dures

Soutenue publiquement le 23 septembre 2024, devant le jury composé de :

Dr	TAMI Abdelkader	MCA	Univ. Saida	Président
Dr	OUARDI Aissa	MCA	Univ. Saida	Rapporteur
Dr	A.OUESSAI Asma	MCB	Univ. Saida	Examineur

Année universitaire 2023/2024

Remercîment

Je tiens à remercier tout d'abord « Allah » le tout Puissant qui m'a donné durant toutes ces années la santé, le courage et la foi.

Le première personne que j tiens à remercier est mon encadrant Dr.Ouardi.Aissa, pour l'orientation, la confiance, la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port.

Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Mes vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à ma recherche en acceptant d'examiner mon travail et de l'enrichir par leurs propositions. Enfin, je tiens également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Dédicace

*Avec l'expression de ma reconnaissance, je dédie ce modeste travail
à ceux qui, quels que soient les termes embrassés, je n'arriverais
jamais à leur exprimer mon amour sincère.*

*A l'homme, mon précieux offre du dieu, qui doit ma vie, ma réussite
et tout mon respect : mon cher père « Djillali »*

*A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit
non à mes exigences et qui n'a épargné aucun effort pour me rendre
heureuse: mon adorable mère « ASIA »*

*A ma chère sœur « Abir » et mon chère amie qui n'ont pas
cessée de me conseiller, encourager et soutenir tout au long de mes
études. Que Dieu les protège et leurs offre la chance et le bonheur.*

*A mon adorable petit frère « Mohammed » qui sait toujours comment
procurer la joie et le bonheur pour toute la famille.*

*A mes grands-mères, mes oncles et mes tantes. Que Dieu leur donne
une longue et joyeuse vie.*

*A tous les cousins et les amis que j'ai connu jusqu'à
maintenant Merci pour leurs amours et leurs encouragements.*

Résumé :

Dans ce travail on va étudier les critères d'arrêt turbo à base des décisions dures qui explorent les méthodes utilisées dans les systèmes de communication pour améliorer la détection d'erreurs. Ces approches reposent sur l'analyse des décisions binaires fermes prises par les décodeurs turbo. En se concentrant sur ces décisions, le critère d'arrêt vise à optimiser les performances du système en termes de vitesse de convergence et de réduction de la consommation d'énergie. L'étude examine comment ces critères peuvent être implémentés efficacement pour améliorer la robustesse et l'efficacité des décodeurs turbo.

Le critère d'arrêt dans un algorithme de type turbo, comme les codes turbo utilisés en télécommunications, fait référence aux conditions qui déterminent quand l'algorithme doit cesser ses itérations. Dans le contexte des décisions dures, cela signifie que l'algorithme prend des décisions binaires (0 ou 1) plutôt que des estimations probabilistes.

En utilisant ces critères, on peut optimiser le temps de calcul tout en maintenant une performance acceptable du décodage.

Abstract:

In this work we will study the turbo stopping criterion based on hard decisions which explores a method used in communication systems to improve error detection. This approach relies on the analysis of firm binary decisions made by turbo decoders. By focusing on these decisions, the stopping criterion aims to optimize system performance in terms of convergence speed and energy consumption.

The study examines how this criterion can be effectively implemented to improve the robustness and efficiency of turbo decoders.

The stopping criterion in a turbo algorithm, such as the turbo codes used in telecommunications, refers to the conditions that determine when the algorithm should stop iterating. In the context of hard decisions, this means that the algorithm makes binary decisions (0 or 1) rather than probabilistic estimates. Using these criteria, one can optimize the computation time while maintaining acceptable decoding performance.

ملخص

في هذا العمل سوف نقوم بدراسة معيار التوقف التورييني المبني على القرارات الصعبة والذي يستكشف الطريقة المستخدمة في أنظمة الاتصالات لتحسين اكتشاف الأخطاء. يعتمد هذا النهج على تحليل القرارات الثنائية الثابتة التي تتخذها أجهزة فك التشفير التوريينية. ومن خلال التركيز على هذه القرارات، يهدف معيار التوقف إلى تحسين أداء النظام من حيث سرعة التقارب وتقليل استهلاك الطاقة.

تبحث الدراسة في كيفية تنفيذ هذا المعيار بشكل فعال لتحسين متانة وكفاءة أجهزة فك التشفير التوريينية في بيئات الاتصال المختلفة.

يشير معيار التوقف في خوارزمية من النوع التورييني، مثل رموز التوريو المستخدمة في الاتصالات، إلى الشروط التي تحدد متى يجب أن تتوقف الخوارزمية عن تكراراتها. في سياق القرارات الصعبة، هذا يعني أن الخوارزمية تتخذ قرارات ثنائية (0 أو 1) بدلاً من التقديرات الاحتمالية.

باستخدام هذه المعايير، يمكننا تحسين وقت الحساب مع الحفاظ على أداء فك التشفير المقبول.

Liste des figures :

Figure 1.1	Exemple de codeur convolutif.	4
Figure 1.2	Exemple d'un code convolutif	5
Figure 1.3	Diagramme en treillis du codeur convolutif[5,7] <i>oct.</i>	6
Figure 1.4	Algorithme de viterbi	8
Figure 1.5	Structure de codage turbo	11
Figure 1.6	Echange itératif d'information probabiliste	13
Figure 2.1	Convergence de l'histogramme de LLR a posteriori pendant les itérations	17
Figure 2.2	Valeurs des LLR	18
Figure 2.3	Nombre d'erreurs avec les itérations pour 3 paquets.	18
Figure 2.4	La moyenne absolue des valeurs LLR en fonction du nombre des itérations.	19
Figure 2.5	Exemples de comportements de différents LLR	20
Figure 2.6	Exemples montrant l'évolution des LLR au fil des itérations avec les lignes pointillées pour l'instance nominale d'arrêt anticipée.	21
Figure 2.7	Effet du nombre d'itérations sur les performances de décodage.	22
Figure 2.8	ILLustration du turbo décodage d'une trame correcte.	23
Figure 2.9	Illustration du turbo décodage d'un bloc non convergent.	24
Figure 3.1	<i>Critère d'arrêt SDR Sign Difference Ratio</i>	28
Figure 3.2	<i>Critère d'arrêt Sign change Ratio</i>	29
Figure 3.3	<i>Critère d'arrêt HDA (Hard Decision Aided)</i>	30
Figure 3.4	Critère d'arrêt GENIE	31
Figure 3.5	Performances en terme de TEB de la critère CRC au turbo décodage	34
Figure 3.6	Nombre moyen des iterations par transmission du critère CRC	36
Figure 3.7	Performances d'un critère SCR	36

Figure 3.8	Nombre moyen des itérations du SCR	37
Figure 3.9	Performances d'un critère SDR	40
Figure 4.1	Nombre moyen des itérations du SDR	39
Figure 4.2	Performances d'un critère HDA	41
Figure 4.3	Nombre moyen des itérations du HDA	42

Listes des acronymes et abréviation :

ACK : Acknowledgement

BCJR : Bahl -cocke- Jelinek Raviv

BER : Bit Error Rate

BPSK : Binary phase shift keying

CRC : Cyclic Redundancy check

DVB-RCS : Digital video broadcasting- Return channel on satellite

FER : Frame Error Rate

HDA : Hard Decision Aided

IHDA : Improved Hard Decision Aided

LLR : Logarithm of likelihood Ratio

Log MAP : Logarithmic Maximum A Posteriori Probability

MAP : Maximum A Posteriori

RSC : Recursive Systematic Convolutional

SCR : Sign Change Ratio

SDR : Sign Difference Ratio

SNR : Signal to Noise Ratior

SOVA : Soft-Output Viterbi Algorithme

RSB : Rapport signal/bruit

TEB : Taux d'erreur binaire

WCDMA : Wideband Code Division Multiple Access

Sommaire

Résumé.....	
Abstract.....	
Liste des figures.....	
Listes des acronymes et abréviation.....	
Introduction générale.....	01

CHAPITRE I Turbocodes

Les codes convolutifs.....	04
1.1. Historique.....	04
1.2. Définition.....	05
1.3. Représentation en treillis.....	06
1.4 Algorithme de viterbi.....	08
1.5 Décodage MAP des codes convolutifs.....	08
1.6. Codes récursifs.....	10
1.7. Le turbo codage des codes convolutifs.....	10
1.7.1 Introduction.....	10
1.7.2 Le principe de base du turbo codage.....	10
1.7.3 Le schéma de turbo codage.....	11
1.8 Le turbo décodage des turbocodes convolutifs.....	11
1.8.1 Introduction.....	11
1.8.2 Le principe de base du turbo décodage.....	12
1.8.3 Le schéma de turbo décodage.....	13

CHAPITRE II Critères d'arrêt Turbo à base des Log-Rapports de Vraisemblances LLRs

2.1. Introduction.....	15
2.2 Critère de l'Entropie Croisée EC.....	15
2.3 Critère de la moyenne des LLR.....	16
2.4 Arrêt anticipé.....	19

CHAPITRE III Critères d'arrêt Turbo à base des décisions dures

Introduction.....	29
Critère d'arrêt SDR (Sign- Difference -Ratio).....	29
Critère d'arrêt SCR (Sign-change-Ratio).....	30
Critère d'arrêt HDA (Hard - Decision - Aided).....	31
Critère d'arrêt IHDA.....	32
Critère d'arrêt GENIE.....	31

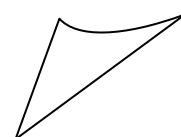
Critère d'arrêt CRC.....	32
---------------------------------	-----------

CHAPITRE IV Résultats de simulation

Introduction.....	34
Performances en terme de TEB de la critère CRC au turbo décodage.....	34
Conditions de simulation.....	34
Résultats de simulation	35
Nombre moyen des iterations par transmission du critère CRC	36
Interprétation des résultats.....	36
Critère SCR appliqué au turbo décodage.....	36
Condition de simulation.....	37
Résultats de simulation	38
Interprétation des résultats.....	39
Critère SDR appliqué au turbo décodage.....	40
Critère HDA appliqué au turbo décodage.....	41
Conclusion.....	44

Introduction

Générale



Introduction générale :

Les Communications numériques sont devenues très indispensables dans notre vie quotidienne. Des taux d'erreur très faibles sont obtenus en utilisant les décodeurs turbo qui sont très performants. Le turbo décodeur est une technique utilisée dans les communications numériques pour améliorer la fiabilité de la transmission des données. Il repose sur l'utilisation de deux ou plusieurs codes correcteurs d'erreurs en parallèle, ce qui permet d'obtenir de meilleures performances de décodage que les méthodes traditionnelles. Ils sont largement utilisés dans les systèmes de communication modernes, tels que les réseaux mobiles et les réseaux de communications par satellite, en raison de sa capacité à réduire les erreurs de transmission et à améliorer la qualité de la réception.

Les turbo codes, inventés à l'E.N.S.T de Bretagne (département Electronique et département Signal et Communications), sont des codes correcteurs d'erreurs dont le pouvoir de correction, dans des conditions idéales de codage, avoisine la limite théorique prédite par C. E. Shannon.

Le terme « Critère d'arrêt turbo » désigne les critères qui sont utilisés côté réception pour arrêter les itérations du turbo décodage lorsque la trame reçue est parfaitement décodée. Il est inutile de poursuivre le décodage si la trame est correctement décodée. Ces critères sont donc utilisés pour réduire la complexité calculatoire côté réception.

Le critère d'arrêt fait donc référence à une condition ou une règle utilisée par le décodeur pour déterminer quand arrêter le processus de décodage itératif des données. Dans le contexte des codes turbo, cela signifie décider quand considérer que la séquence de données décodée est suffisamment proche de la séquence d'origine.

Les décisions dures sont des décisions binaires, ce qui signifie qu'elles sont prises en considérant simplement deux valeurs possibles (0 ou 1). Dans le contexte du décodage des codes turbo, les décisions dures sont prises par le décodeur pour chaque bit de la séquence décodée, en attribuant soit la valeur 0, soit la valeur 1 à chaque bit. Ainsi, le "Critère d'arrêt turbo à base de décision dures" fait référence à une méthode spécifique de décision qui arrête le processus de décodage itératif des codes turbo en utilisant des décisions binaires (0 ou 1).

Dans ce travail, on s'intéresse à l'utilisation des mécanismes de vérification à base des décisions dures pour arrêter le processus turbo au bon moment. Aussi, on va évaluer les performances de ces techniques en termes de taux d'erreur binaire TEB.

Le manuscrit est organisé en quatre chapitres :

Le premier chapitre : est consacré aux différents codes convolutifs, les codes convolutifs récurrents, le turbo codage et le turbo décodage. Puis une petite description du décodage MAP des codes convolutifs.

Le deuxième chapitre : expose les critères d'arrêts turbo représentés: Critère de l'Entropie Croisée EC, Critère de la moyenne des LLR et le critère de la variance des LLR ...

Le troisième chapitre : expose les critères SDR, SCR, HDA, IHDA, GENIE et le critère CRC qui sont basées sur les décisions dures du turbo décodeur.

Le quatrième chapitre : expose les résultats de simulation et analyse les performances des critères utilisés.

CHAPITRE I

Turbocodes

Les codes convolutifs :

1.1. 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]

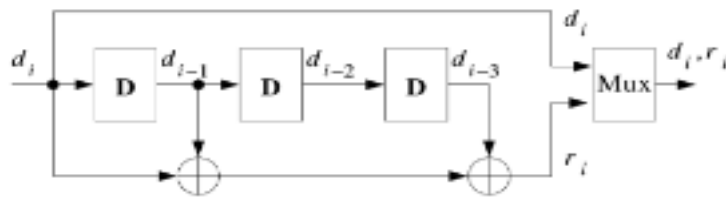


Figure 1.1 – Exemple de codeur convolutif.

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]

1.2. Définition :

Pour les codes convolutifs, chaque bloc de n_s symboles binaires en sortie, dépend non seulement des n_e symboles binaires présents en entrée mais aussi des v blocs de n_e symboles binaires précédents. Par conséquent, ces codes introduisent un effet de mémoire d'ordre v .

Chaque code convolutif est défini par les paramètres suivants :

- ♣ Rendement du code : $R = n_e / n_s$
- ♣ Longueur de contrainte : $K = v + 1$

La figure 1.2 donne un exemple d'un code convolutif non systématique de rendement $R=1/2$ et de longueur du contrainte $K=3$.

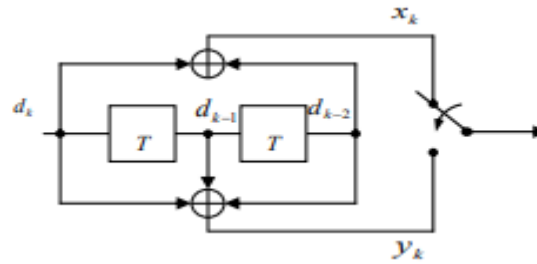


Figure 1.2- Exemple d'un code convolutif

$$(n_e = 1, n_s = 2, R = 1/2, K = 3)$$

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 et les cellules du registre à décalage et du commutateur à n_s positions. [3]

1.3. Représentation en treillis :

La représentation sous la forme d'un diagramme d'états, bien qu'elle permette de se rendre compte aisément du fonctionnement du codeur, est difficile à exploiter pour le décodage. C'est pourquoi, une représentation sous forme de treillis est privilégiée. Elle a été utilisée pour la première fois par Forney en 1973. Cette représentation permet de rendre compte à la fois de l'état interne du codeur, des transitions et de l'évolution temporelle. La Figure 3 présente le diagramme en treillis associé au codeur convolutif de la figure 3, en considérant que l'état initial du codeur est $s_0 = 00$. Comme il n'y a qu'une entrée, de chaque noeud partent 2 branches et vers chaque noeuds convergent 2 branches.

À Partir de $v+1$ unités de temps, quel que soit l'état initial, le motif du treillis se répète. Cette durée est appelée longueur de contrainte du codeur convolutif. Nous présentons sur les branches, les 8 associations entrées/sorties possibles.

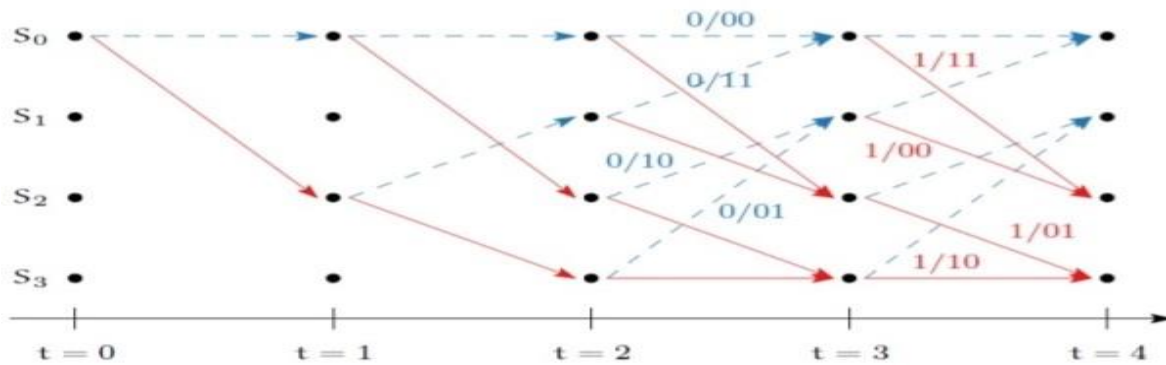


Figure 1.3 : Diagramme en treillis du codeur convolutif[5,7]*oct*.

Comme évoqué précédemment, l'un des intérêts des codeurs convolutifs est de permettre d'obtenir des mots de codes de n'importe quelle taille. Néanmoins, dans le contexte de protocoles de communications numériques, les messages ont une taille prédéfinie. Se pose alors la question de la fermeture du treillis. Trois solutions de terminaison ont été proposées dans la littérature :

1. Pas de terminaison. L'état final ne peut donc être connu par le décodeur. Ainsi, les derniers symboles encodés sont moins bien protégés. (C'est notre cas, $\beta = 1/4$)
2. Forcer l'encodeur à finir dans un état connu. Quel que soit l'état interne du codeur après que les K bits d'information aient été codés, un retour dans n'importe quel état est possible en y ajoutant v bits supplémentaires. Le message transmis est alors composé de la séquence initiale concaténée avec cette séquence supplémentaire. Une légère baisse de rendement est donc induite par cette technique.
3. Utiliser un codeur circulaire. Dans ce cas, l'état initial et l'état final sont les mêmes (mais pas forcément l'état 0). Cet état porte le nom d'état de circulation. Le treillis peut alors être vu comme un cercle. Tous les bits ont la même protection et ce sans induire une modification du rendement.

Par contre, un pré-codage du message est nécessaire afin d'identifier l'état de circulation. [4]

1.4 Algorithme de viterbi :

Actuellement, c'est l'algorithme le plus utilisé vu sa faible complexité.. L'idée de départ est de calculer la distance (au sens de Hamming, mais d'autres distances en

fonction du type de modèle de canal sont aussi utilisées) entre le mot reçu et chacun des mots codés possibles et décider de quel mot codé le mot reçu se trouve le plus près (c'est donc un décodage selon le maximum de vraisemblance).

Présentons-le à partir d'un exemple. Soit le mot émis (11 10 00 01 01 11), provenant du codage du mot d'information (1 0 1 1). Supposons que lors de la transmission, le mot soit altéré par deux erreurs et que l'on reçoive le mot (11 11 01 01 01 11). On comparera ce mot à tous les mots codés possibles.

Nous comparons le premier groupe de deux bits reçus (ici $n=2$) (11) aux deux chemins possibles (00 et 11) en calculant les distances de Hamming correspondantes.

Ensuite, il faudrait calculer les distances entre les quatre premiers bits reçus (11 11) et les quatre chemins correspondants. Mais nous avons déjà calculé les distances (2 et 0) du premier étage, il suffit donc de calculer les distances correspondant au deuxième étages et de les ajouter (propriétés des distances). Nous obtenons ainsi les distances 4 2 1 et 1.

A l'étage suivant il faudra calculer huit distances. Ici on rencontre les premiers «rebouclages » du treillis par rapport à l'arbre qui vont nous faire gagner du temps de calcul.

A partir de là, à chaque nœud de l'extrémité droite du treillis aboutissent deux chemins distincts et on pourra écarter l'un des deux, celui dont la distance est la plus grande par rapport au mot reçu. Nous appelons « chemin survivant (survivor path) », le chemin qui reste.

A chaque étape, de chaque nœud sortent deux nouvelles branches, donc il y aura de nouvelles distances à calculer. De nouveau à chaque branche d'arrivée on écarte le chemin dont la distance est la plus grande. En cas d'égalité on choisit un chemin au hasard. Le même processus est répété jusqu'à la fin de la séquence. Le chemin qui a la métrique cumulée (distance) la plus faible est retenue. Il donne la séquence d'information la plus probable. [5]

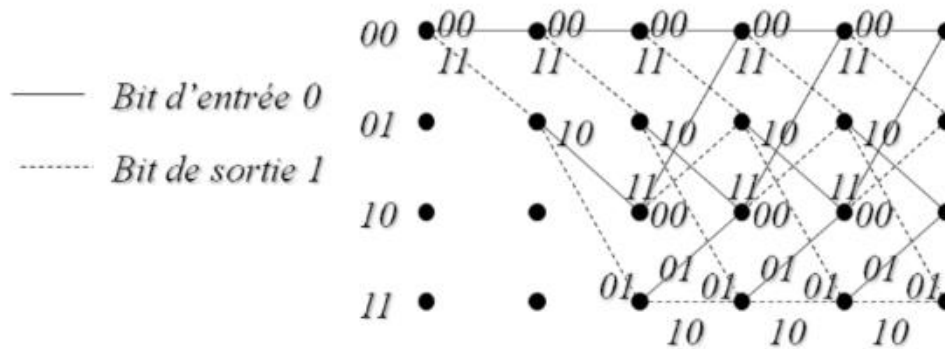


Figure 1.4 Algorithme de viterbi

1.5 Décodage MAP des codes convolutifs :

La méthode de décodage MAP (Maximum A posteriori) est une méthode de décodage qui utilise une approche probabiliste pour estimer la séquence de données la plus probable.

Dans le décodage MAP, on cherche à déterminer la séquence de données la plus probable en fonction de la séquence de données reçue. Cette méthode est souvent utilisée pour les codes convolutifs qui nécessitent une correction d'erreurs plus précise.

Le décodage MAP est basé sur le calcul de probabilité a posteriori des bits émis 0 ou 1 en fonction de la trame reçue. Cette probabilité est calculée en utilisant le théorème de Bayes, qui permet de calculer la probabilité a posteriori à partir de la probabilité a priori et de la vraisemblance.

La vraisemblance est calculée en utilisant la fonction de transition de l'état de code convolutif, qui décrit la probabilité de passer d'un état à l'autre dans le code convolutif. La probabilité a priori est déterminée en utilisant les informations sur la distribution de probabilité de la séquence de données envoyée.

Une fois que la probabilité a posteriori a été calculée, une comparaison est appliquée sur le Log-rapport de vraisemblance LLR puis une décision est prise.

Le décodage MAP est une méthode de décodage très précise, mais elle est également très complexe. Cela signifie qu'elle nécessite des ressources de calcul importantes et qu'elle est souvent utilisée dans les turbo -communication. [6]

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.

$$\text{LLR}(m_k) = \log \frac{\text{pr}(m_k=1|R_1^N)}{\text{pr}(m_k=0|R_1^N)} \quad (1.1)$$

Avec :

$$m_k \in \{0,1\} \forall k \quad (1.2)$$

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

L'algorithme définit les probabilités conjointes en termes de trois paramètres.

$$a_k(s) = \text{pr}(s_k = s | R_1^k) \quad (1.3)$$

$$\beta_k(s) = \text{pr}(R_{k+1}^N | s_k = s)$$

$$\gamma_j(S', S) = \text{pr}(m_k = j, S_k = s, R_k | s_{k-1} = s')$$

Le LLR peut maintenant être décrit en termes de :

$$\text{LLR}(k) = \log \frac{\sum_{s', s/1} \alpha_{s'}(k-1) \gamma_{s', s}(k) \beta_s(k)}{\sum_{s', s/0} \alpha_{s'}(k-1) \gamma_{s', s}(k) \beta_s(k)} \quad (1.4)$$

$\alpha_k(s)$ et $\beta_k(s)$ peuvent être calculées en avant et en arrière en utilisant les probabilités $\gamma_j(R_k, s', s)$.

$$\alpha_s(k) = \sum_{s'} \alpha_{s'}(k-1) \gamma_{s', s}(k) \quad (1.5)$$

$$\beta_s(k) = \sum_{s'} \alpha_{s'}(k-1) \beta_{s'}(k-1)$$

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 l'état final des deux codeurs sont initialisée à zéro, on aura :

$$\alpha_0 = \begin{cases} 1 & \text{si } s=0 \\ 0 & \text{autrement} \end{cases} \quad (1.6)$$

$$\beta_N = \begin{cases} 1 & \text{si } s = 0 \\ 0 & \text{autrement} \end{cases}$$

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

$$\beta_N = \frac{1}{M} \quad (1.7)$$

Où M est le nombre total des états de treillis [7].

1.6. Codes rékursifs :

La sortie d'un code convolutif peut aussi dépendre des sorties précédentes. On dit alors que le code est rékursif.

Dans le cas de codes rékursifs, à une suite d'entrée de longueur finie correspond une suite de sortie de longueur infinie.

On peut transformer un code convolutif non rékursif, non systématique (NRNSC Non recursive non systematic code) en un code rékursif systématique (RSC Recursive systematic code) en divisant les polynômes du codeur par l'un des deux.

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 rékursif systématique défini par les fonctions rationnelles $[1, g_1(z)/g_0(z)]$.

Les diagrammes d'états d'un code non rékursif et du code rékursif systématique qui s'en déduit sont les mêmes. [8]

1.7. Le turbo codage des codes convolutifs :

1.7.1 Introduction :

L'invention des turbocodes au début des années 90 a véritablement bouleversé le domaine du codage correcteur d'erreur, en exhibant des codes relativement simples à construire et à décoder, permettant d'approcher au plus près la limite théorique promise par Claude E. Shannon. Ces codes sont basés sur l'utilisation de deux codes convolutifs en parallèle, avec une technique d'entrelacement.

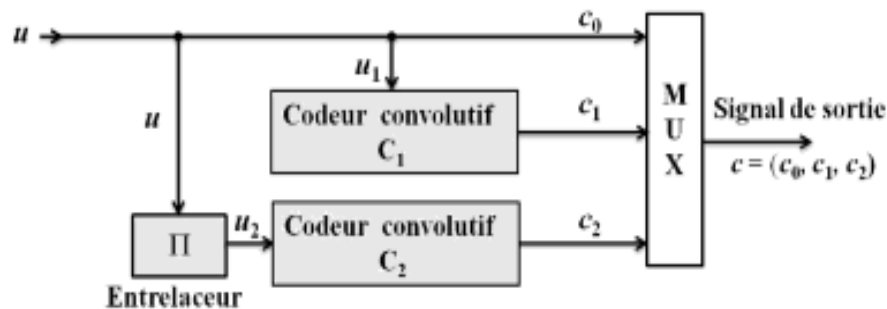
1.7.2 principe de base du turbo codage :

Ce principe consiste pour l'essentiel à diviser le problème a priori très complexe en une cascade de traitements élémentaires, plus simples à mettre en œuvre. Toutefois,

on sait aujourd'hui que lorsque l'on ne prend pas de précautions particulières à cet égard, cette séparation conduit naturellement à une perte d'information. Pour remédier à ce problème, le principe Turbo préconise l'instauration d'un échange itératif d'information probabiliste entre ces différents traitements, de manière à ce que l'intégralité de l'information disponible soit bien prise en compte dans la résolution du problème global.

L'application du principe Turbo a démontré des gains de performances impressionnants par rapport aux systèmes traditionnels. Son utilisation s'est donc rapidement popularisée au sein de la communauté scientifique. [9]

1.7.3 schéma de turbo codage :



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

Figure 1.5 Structure de codage turbo

La structure d'un turbocodeur à taux de codage 1/3 (sans perforation) est présentée à la figure (fig 1.5). Cette structure sert à affiner et améliorer l'estimation des bits d'information originaux dans le décodeur.

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) avec la protection contre les erreurs d'un code convolutif (C_1 et C_2), l'un des blocs de codage recevant en entrée une séquence d'informations entrelacée par permutation aléatoire. Le multiplexeur combine alors les trois signaux en un seul. [10]

1.8 turbo décodage des turbocodes convolutifs :**1.8.1 Introduction :**

Les turbocodes sont une classe de codes correcteurs d'erreurs approchant la limite théorique de capacité formulée par Claude Shannon. Conjointement à leurs excellentes performances de décodage, la complexité calculatoire modérée des turbo décodeurs a permis leur inclusion dans de nombreux standards de communications numériques. Une des métriques permettant la caractérisation de codes correcteurs d'erreurs est l'évolution du taux d'erreurs binaires en fonction du rapport signal sur bruit .

1.8.2 principe de base du turbo décodage :

Dans le cadre des turbocodes, une courbe de performance de décodage comprend deux zones principales. Dans la première zone, une faible amélioration de la qualité du canal de transmission entraîne de grandes améliorations au niveau des performances de décodage. En revanche dans la seconde, une amélioration de cette qualité ne résulte qu'en une amélioration marginale des performances de décodage. Cette seconde région est nommée zone du plancher d'erreurs. Elle peut empêcher l'utilisation de turbo codes dans des contextes nécessitant de très faibles taux d'erreurs. C'est pourquoi la communauté scientifique a proposé différentes optimisations favorisant la construction de turbo codes atténuant ce plancher d'erreurs. Cependant, ces approches ne peuvent être considérées pour des turbocodes déjà standardisés. Dans ce contexte, il s'adresse le problème de la réduction du plancher d'erreurs en s'interdisant de modifier la chaîne de communications numériques du côté de l'émetteur. Pour ce faire, un état de l'art de méthodes de post-traitement de décodage est dressé pour les turbocodes. Il apparaît que les solutions efficaces sont coûteuses à mettre en œuvre car elles nécessitent une multiplication des ressources calculatoires ou impactent fortement la latence globale de décodage. Dans un premier temps, deux algorithmes basés sur une supervision de l'évolution de métriques internes aux décodeurs, sont proposés. L'un d'eux permet d'augmenter la convergence du turbo décodeur. L'autre ne permet qu'une réduction marginale du plancher d'erreurs. Dans un second temps, il est observé que dans la zone du plancher d'erreurs, les trames décodées par le turbo décodeur sont très proches du mot de code originellement transmis. Ceci est démontré par une proposition de

prédiction analytique de la distribution du nombre d'erreurs binaires par trame erronée. Cette dernière est réalisée grâce au spectre de distance du turbo code. Puisque ces erreurs binaires responsables du plancher d'erreurs sont peu nombreuses, une métrique permettant de les identifier est mise en œuvre. Ceci mène alors à l'établissement d'un algorithme de décodage permettant de corriger des erreurs résiduelles. Cet algorithme, appelé algorithme Flip-and-Check se base sur un principe de création de mots candidats et de vérifications successives par un code détecteur d'erreurs.

Grâce à cet algorithme de décodage, un abaissement du plancher d'erreurs d'un ordre de grandeur est obtenu pour les turbo codes de différents standards (LTE, CCSDS, DVB-RCS et DVB-RCS2), ce, tout en conservant une complexité calculatoire raisonnable.[11]

1.8.3 schéma de turbo décodage :

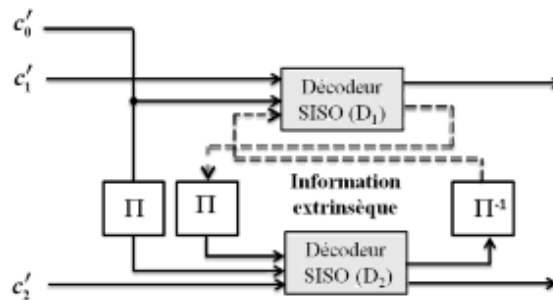


Figure 1.6 Echange itératif d'information probabiliste

Pour le décodage des codes convolutifs (systématiques ou non- systématique), on peut prendre des décisions sur la base de la probabilité à posteriori maximale, autrement dit sur la base de la distance minimale entre le mot reçu et un mot-code. [12]

CHAPITRE II

Critères d'arrêt Turbo à base des Log-Rapports de Vraisemblances LLRs

2.1. INTRODUCTION :

Les critères d'arrêt turbo basés sur les log-rapports de vraisemblance (LLRs) sont essentiels pour les décodeurs turbo, car ils déterminent quand arrêter le processus itératif. Les LLRs représentent la fiabilité des décisions des bits décodés. Les critères d'arrêt sont conçus pour garantir une performance optimale tout en minimisant la consommation de ressources. En pratique, ces critères se basent souvent sur des seuils qui assurent la convergence de la décision de décodage.

2.2 Critère de l'Entropie Croisée EC :

Soit X une variable aléatoire discrète de distribution p :

On appelle entropie de X la moyenne des valeurs de $S(p_i)$:

$$H(X) = \sum_{i=1}^n p_i S(p_i) \quad (2.1)$$

$$= - \sum_{i=1}^n p_i \log_2(p_i) \quad (2.2)$$

L'entropie est exprimée en bits. Elle correspond à la taille minimale du message qu'il faut pour transmettre la valeur de la variable aléatoire.

Si $p_1 = p_2 = 0.5$, alors $H(p) = 1$, IL faut un bit pour transmettre l'information.

On définit l'entropie croisée de la façon suivante :

$$H(p, q) = - \sum_i p_i \log_2(q_i) \quad (2.3)$$

$$\text{Si } p = q, \text{ alors } H(p, q) = H(p) \quad (2.4)$$

$$\text{Sinon } H(p, q) > H(p) \quad (2.5)$$

La différence $H(p, q) - H(p)$ est appelée divergence de **kullback –Leibler** :

$$H(p, q) = H(p) + D_{KL}(p||q) \quad (2.6)$$

Elle mesure la différence entre les distributions de probabilité p et q . [13]

La entropie croisée CE (Cross-Entropy) peut être approximée comme suit

$$T(i) = \sum_k \frac{|\Delta Le_2^{(i)}(\hat{u}_k)|^2}{e^{|L_1^{(i)}(\hat{u}_k)|}} \quad (2.7)$$

Avec

$$\Delta Le_2^{(i)}(\hat{u}_k) = Le_2^{(i)}(\hat{u}_k) - Le_2^{(i-1)}(\hat{u}_k) \quad (2.8)$$

Après chaque itération, la technique CE calcule l'entropie croisée approximative entre les rappor

ts de log-vraisemblance (LLR) des décodeurs composants comme dans (2.1)

L'itération est arrêté si :

$$T(i) \leq (10^{-2} \sim 10^{-4})T(1) \quad (2.9)$$

$T(1)$ est l'entropie croisée EC de la première itération.

2.3 Critère de la moyenne des LLR :

Selon le principe du décodage itératif, le LLR de l'entrée a priori d'un composant décodeur provient du LLR de la sortie extrinsèque de l'autre composant. L'observation des LLRs d'un composant décodeur permet de caractériser la convergence du turbo décodeur vers la trame décodée finale. Dans un algorithme de décodage Soft-In-Soft-Out (MAP), le LLR a posteriori en sortie d'un composant décodeur peut être modélisé par une variable aléatoire gaussienne indépendante, où sa variance correspond à la variance du canal AWGN, et sa valeur moyenne au bit systématique transmis. C'est le signe de LLR qui détermine le symbole binaire, négatif pour le bit 0 et positif pour le bit 1, et la valeur absolue quantifie la certitude de la décision dure du décodeur. Ainsi, lorsque les valeurs LLR sont proches de zéro, la décision du décodeur est ambiguë.

Les histogrammes des deux sorties a posteriori $(L_l^1(u_t|r), L_l^2(u_t|r))$ des composants décodeurs sont représentés sur la Figure 2.1 en prenant l'indice d'itération comme paramètre. Il est évident que l'histogramme converge vers la distribution gaussienne bilatérale centrée sur deux moyennes, qui représentent les deux symboles binaires, de façon proportionnelle au nombre d'itérations.

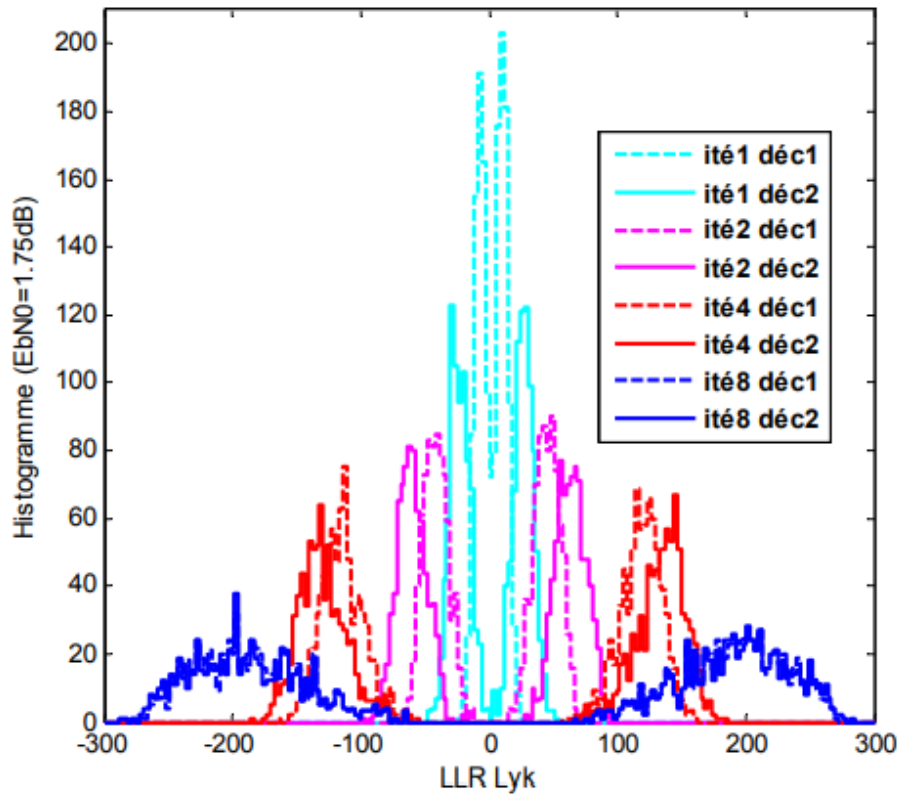


Figure 2.1 Convergence de l'histogramme de LLR a posteriori pendant les itérations

En observant le nombre de bits erronés par trame aux sorties des deux composants décodeur, pour des taux SNR élevées et après chaque itération, on peut conclure que la plupart des trames sont correctement décodées (zéro bits erronés) juste après quelques itérations sans atteindre la limite supérieure des itérations I_{max} .

D'autre part, pour des taux SNR faibles, la limite supérieure des itérations n'est pas suffisante pour atteindre zéro bits erronés. De plus, des itérations supplémentaires n'améliorent pas la situation et les valeurs LLR sont saturées. [14]

La figure 2.2 affiche les valeurs LLR typiques pour dix trames de bits de données à la sortie d'un décodeur après dix itérations [15].

Nous pouvons immédiatement voir de grandes différences dans les valeurs moyennes absolues des LLR. On peut utiliser ceci afin de déterminer le bon moment d'arrêt du processus de décodage Turbo. Cette approche est basée sur le suivi de la

moyenne des valeurs absolues des LLR's $M_{|L|}$. Les résultats de simulation [16] ont montré qu'au fil des itérations, $M_{|L|}$ augmente à mesure que le nombre d'erreurs dans le paquet diminue. La moyenne $M_{|L|}$ et le nombre d'erreurs correspondant des trois trames typiques en fonction des itérations sont montrés à la Figure 2.2 et 2.3. Sur la base de ce constat, Zhai and Fair [16] ont proposé le critère d'arrêt de la moyenne des LLRs.

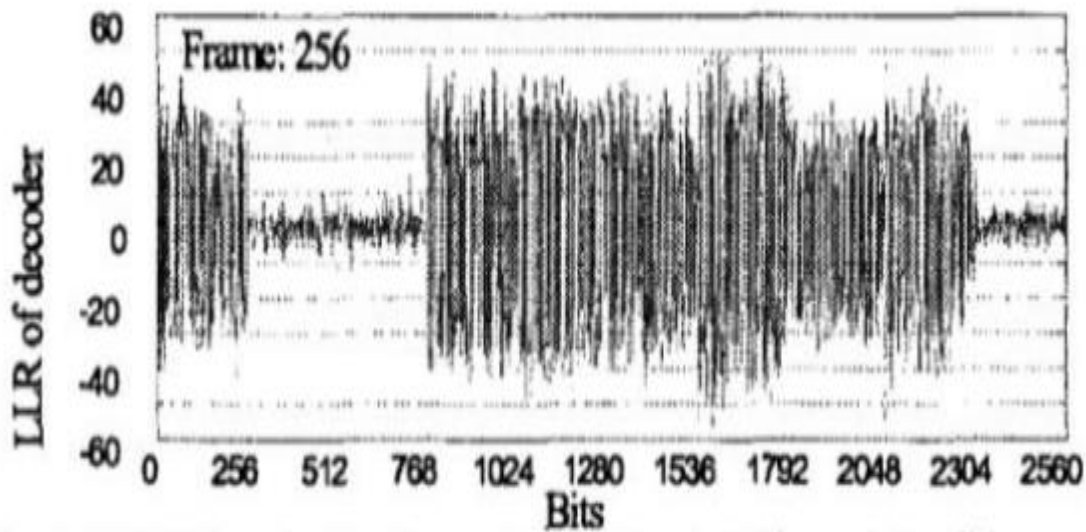


Figure 2.2 : Valeurs des LLR [15]

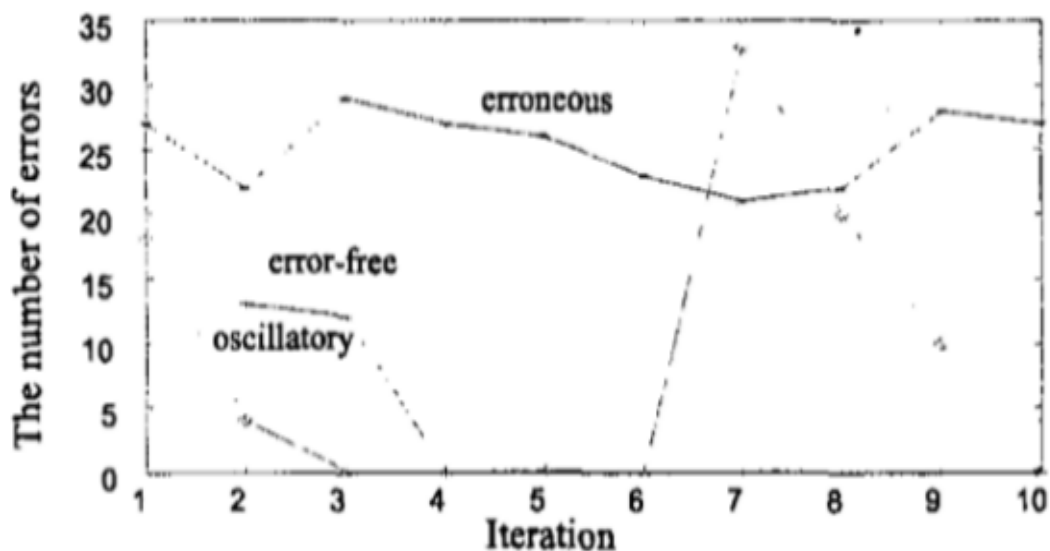


Figure 2.3 : Nombre d'erreurs avec les itérations pour 3 paquets. [14]

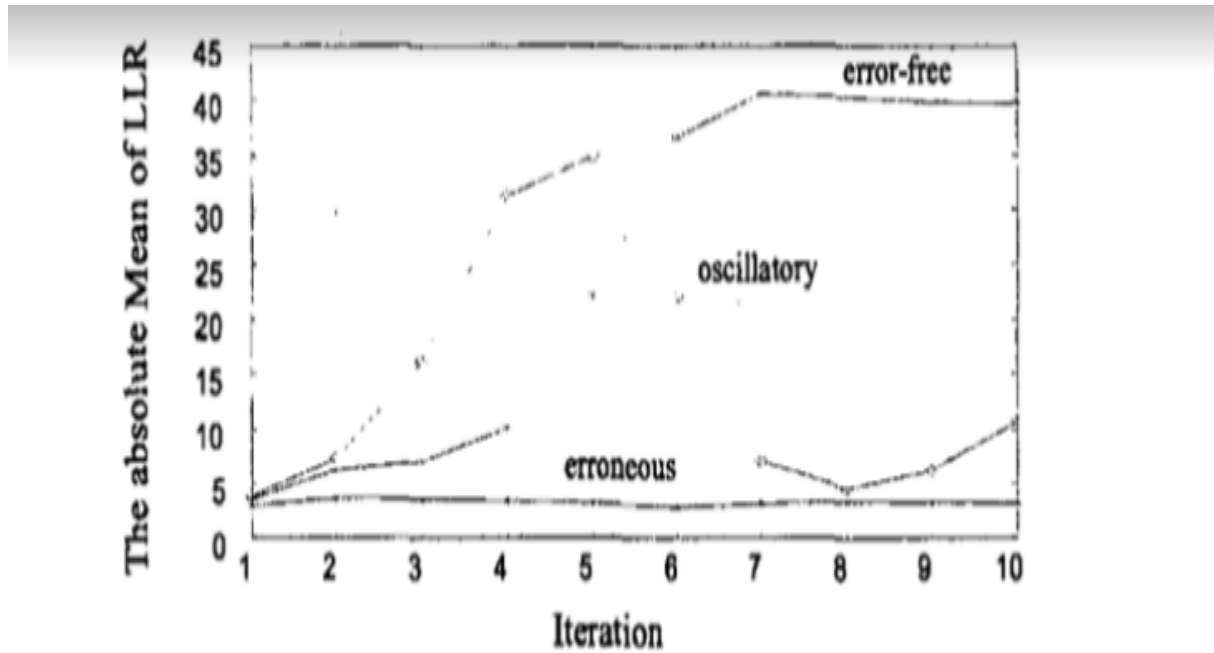


Figure 2.4: La moyenne absolue des valeurs LLR en fonction du nombre des itérations. [16]

Si $M_{|L|}$ est supérieur à un seuil bien déterminé $Th(M_{|L|} > Th)$, alors, arrêtez le processus itératif et considérez que la trame est exempte d'erreurs, sinon continuez à itérer.

Si $M_{|L|} < Th$ pour MAX_{it} , le nombre maximal d'itérations autorisées, considérez que la trame est indécodable. Ce schéma simple offre de bonnes performances et est simple à mettre en œuvre [16].

2.4 Arrêt anticipé :

Les changements dans les LLR après avoir atteint la convergence deviennent insignifiants et les décodeurs SISO reproduisent presque les mêmes valeurs/signes dans les itérations suivantes. Cela ressort clairement de l'évolution des LLR représentée sur les figures fig2.5, fig3.2, fig3.3. Il est possible de terminer le processus de décodage à un moment donné sans affecter les performances. Surtout à des SNR élevés, le décodeur atteint la convergence à un stade précoce et, par

conséquent, les itérations suivantes peuvent être jugé inutiles. Le décodage turbo est très coûteux en termes de calculs, de latence et de consommation électrique. Il est pratiquement judicieux de terminer ce processus dès que le résultat est suffisamment fiable. Beaucoup des méthodes d'arrêt anticipé sont proposées dans la littérature. Le but est de faciliter le décodage en supprimant les itérations inutiles [11] [17].

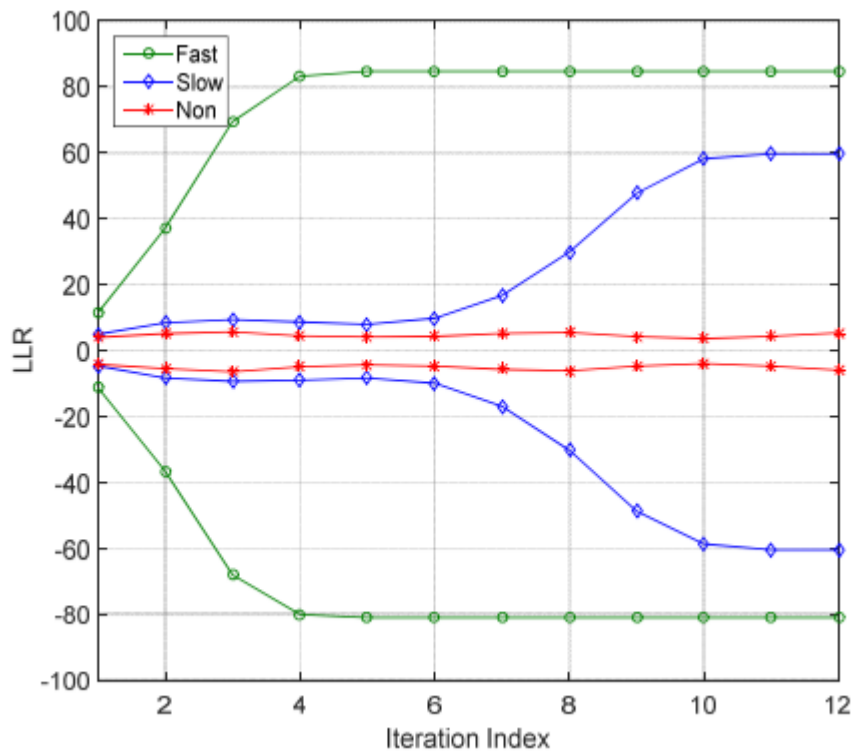


Figure 2.5 Exemples de comportements de différents LLR

La figure 2.6 montre différents modèles possibles pour les traces des LLR. L'objectif d'une règle d'arrêt anticipé est de déterminer la meilleure instance pour arrêter le traitement. De nombreuses méthodes sont proposées dans la littérature.

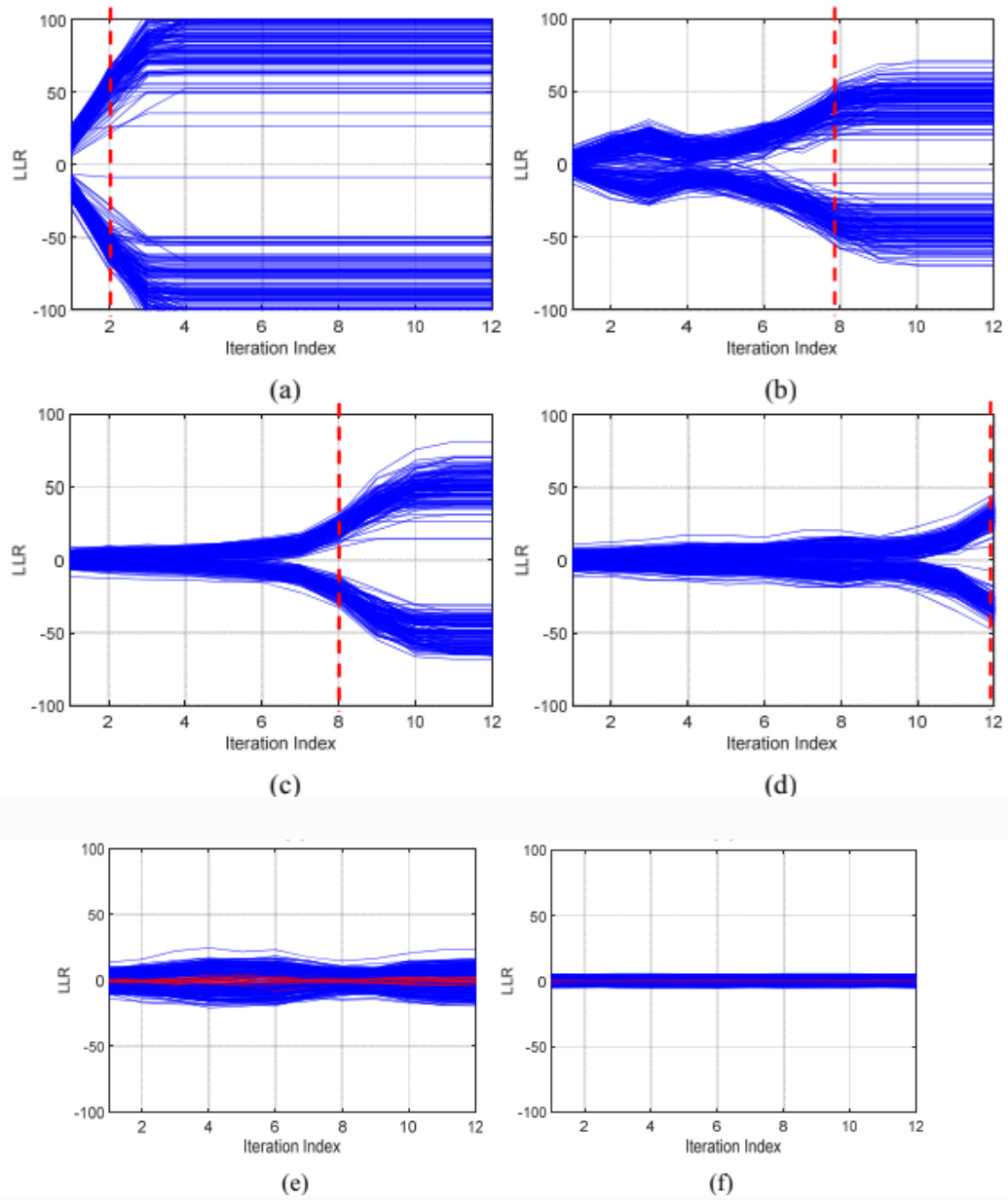


Figure 2.6 Exemples montrant l'évolution des LLR au fil des itérations avec les lignes pointillées pour l'instance nominale d'arrêt anticipée.

La figure 2.7 montre le fait que plus les SISO itèrent, meilleures sont les performances attendues, cependant, la latence de décodage et la consommation

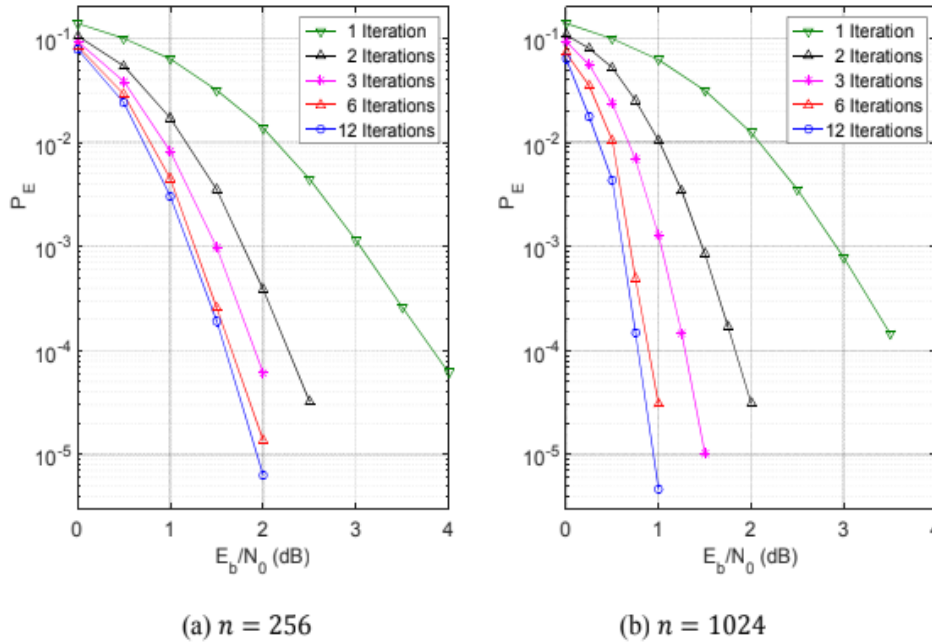


Figure 2.7 Effet du nombre d'itérations sur les performances de décodage.

augmentent également. Dans certains cas, le décodeur atteint la convergence avant d'atteindre le nombre maximum des itérations. Une autre raison pour laquelle il est recommandé de terminer le décodage au bon moment (trames correctes) [11] [17].

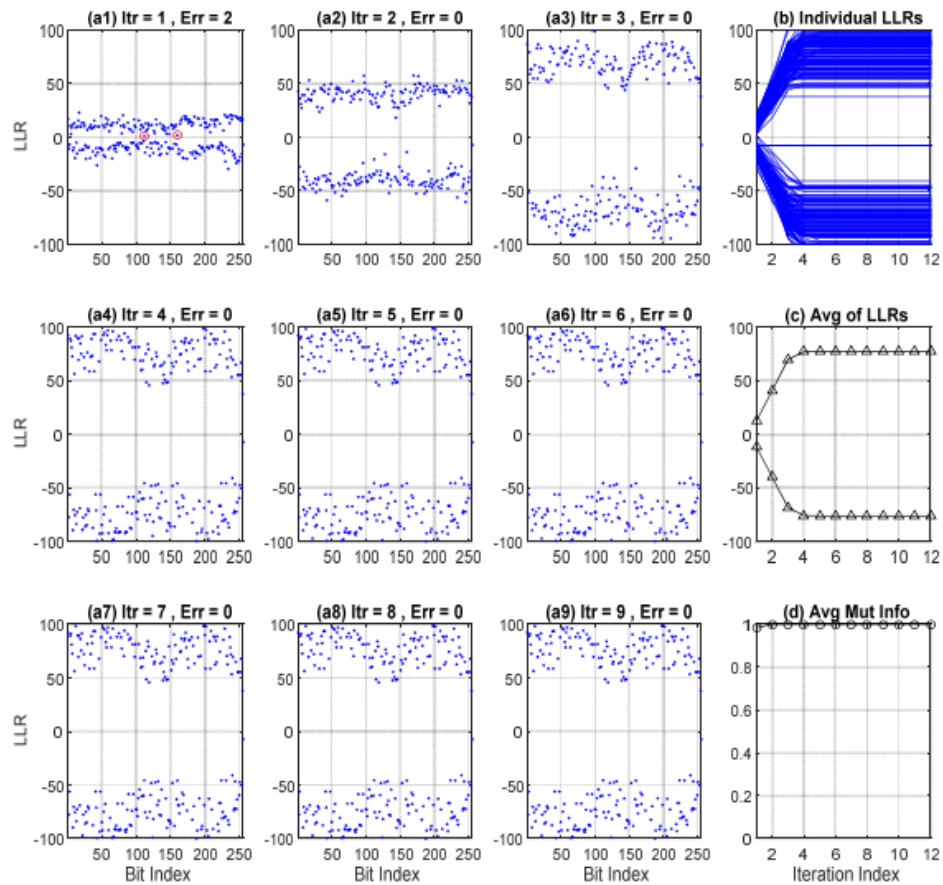


Figure 2.8 ILLustration du turbo décodage d'une trame correcte.

Le nombre d'itérations suffisant pour atteindre le niveau de fiabilité requis varie pour chaque bloc reçu, même pour le même SNR. Ce nombre dépend du nombre d'erreurs. Pour économiser les ressources de décodage, en arrêtant la boucle de décodage, les méthodes d'arrêt précoces utilisent des mesures personnalisées pour vérifier la convergence. En principe, ils calculent un ou plusieurs paramètres après chaque itération et associer les valeurs de ces métriques au niveau de fiabilité des LLR calculés. La boucle d'itération se termine une fois qu'une valeur combinée de ces métriques dépasse un seuil prédéfini. Les seuils en vigueur sont soit déterminés par simulation, ou ils sont ajustés de manière adaptative. De nombreuses méthodes sont présentées dans la littérature pour estimer le meilleur instant pour terminer le décodage. Ces méthodes peuvent être regroupés en deux catégories principales : ils

s'appuient soit sur la sortie Soft, soit sur la sortie Hard. Basé sur la sortie Soft, les méthodes de cette catégorie dépendent de la sortie douce (Soft) du code turbo. Ils surveillent par exemple la moyenne des LLR, la variance [18], la moyenne et la variance [19] [20], le minimum [21][22], à la fois le maximum ou la somme des valeurs absolues des LLR après chaque itération. Ces méthodes reflètent essentiellement l'amélioration de leurs métriques surveillées comme mesure des progrès dans le décodage. La convergence est supposée être atteinte lorsque les métriques utilisées dépassent certaines valeurs « seuil ».

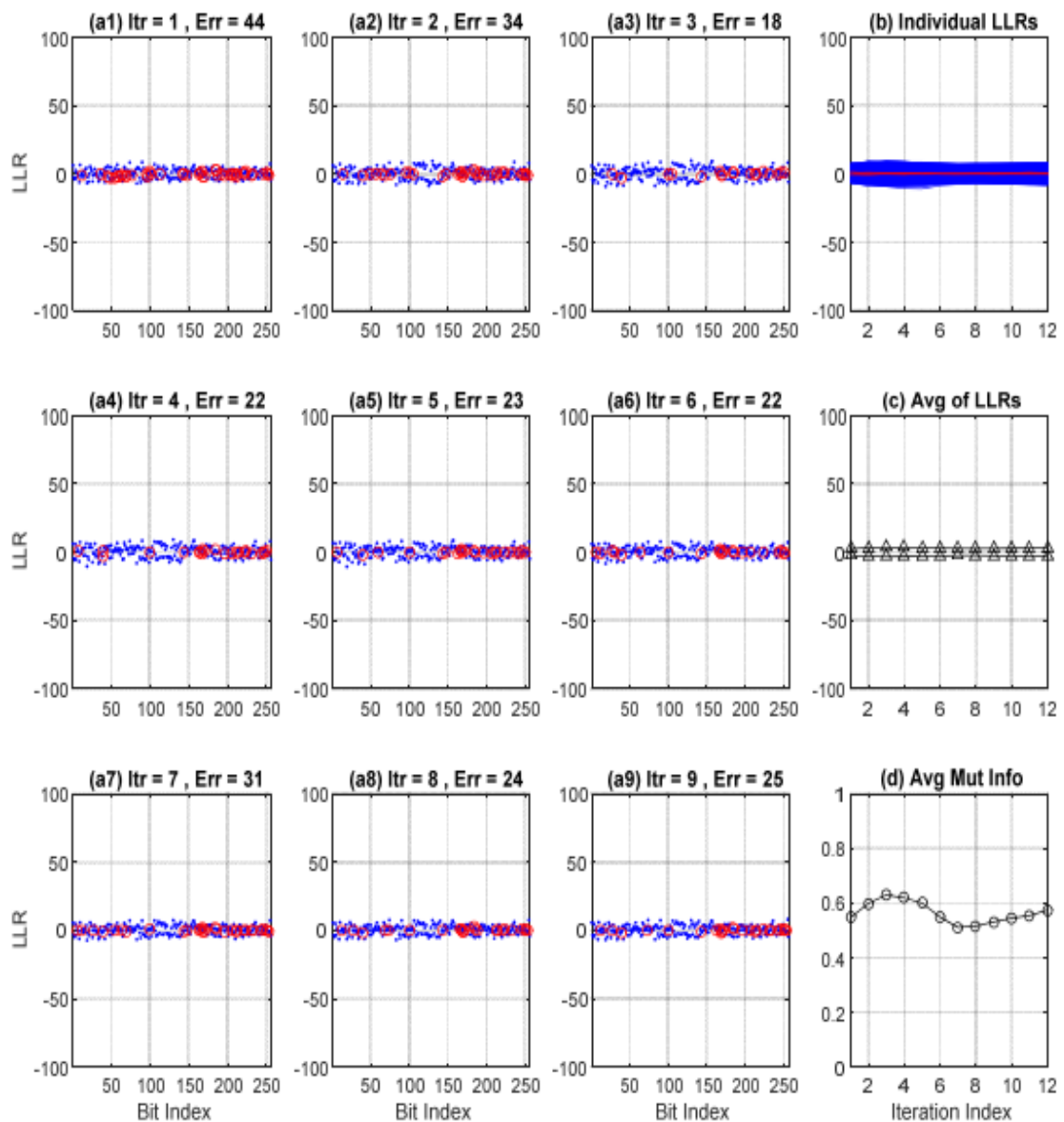


Figure 2.9 Illustration du turbo décodage d'un bloc erroné.

D'autres méthodes utilisent également les sorties Soft pour calculer une mesure décisive, comme l'entropie croisée [23][24], l'information mutuelle [25], la corrélation croisée. [26]

L'entropie moyenne est utilisée [27] comme condition d'arrêt : elle est calculée après chaque itération et l'itération s'arrête si sa valeur devient inférieure à un seuil. Dans une autre proposition, un réseau de neurones est utilisé [28].

Autres méthodes basées sur la sortie dure (Hard)

D'un autre côté, d'autres méthodes utilisent à la place la sortie dure du turbo décodeur. Parmi ces méthodes, le Sign-Change-Ratio (SCR)[29] , le Sign-Difference-Ratio (SDR) [29], le Hard Decision Aided (HDA) [30-32] et le HDA amélioré (IHDA)[33] .

Méthodes basées sur une redondance supplémentaire

Une redondance supplémentaire est utilisée pour détecter le point de convergence dans certaines propositions. Le contrôle de redondance cyclique (CRC) [34] et Bose Chaudhuri- Des codes Hocquenghem (BCH) [35] sont déployés au sein du turbo code pour arrêter le décodage.

En raison de la dégradation du débit après insertion de la redondance, ces méthodes ne sont pas bien adaptées à de nombreuses applications.

Méthodes à faible SNR

Les méthodes mentionnées ci-dessus recherchent l'instant de convergence pour déclencher la commande de terminaison. Cependant, à un SNR inférieur, la réception a moins de chances d'atteindre ce niveau. Par conséquent, de telles méthodes permettent essentiellement au décodeur de continuer à itérer jusqu'au nombre maximal des itérations.

Les méthodes discutées détectent uniquement le point de convergence. Elles permettent au décodeur de continuer à itérer les blocs non décodables.

CHAPITRE III
Critères d'arrêt Turbo à base
des décisions dures

3.2 INTRODUCTION :

Dans les codes turbo, le décodage itératif entre les deux décodeurs constituent le cœur du processus

de décodage .Le critère d'arrête joue un role crucial pour déterminer quand le décodage itératif doit

S'arrêter et fournir la séquence de données décodée finale.

"Le critère d'arrêt turbo à base des décisions dures" fait référence à une stratégie d'arrêt qui se base sur des décisions binaires (dures) plutôt que sur des informations souples (soft).

Concrètement, cela signifie que le décodeur s'arrête lorsqu'il détecte que les bits décodés ont

atteint un certain niveau de fiabilité, sans tenir compte des informations de confiance intermédiaires.

Cette approche peut être plus simple à mettre en œuvre que des critères d'arrêt plus élaborés, mais elle peut aussi être moins performante dans certains cas.

Le choix du critère d'arrêt est donc un élément clé dans la conception des décodeurs turbo, avec un impact direct sur les performances du système de communication.

3.3 Critère d'arrêt SDR :

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

$$D_{ij} \leq PN \quad 3.1$$

Où ' P ' est un seuil qui représente le rapport de différence de signe ratio SDR, ' N ' est la longue de bit et :

$$10^{-3} \leq P \leq 10^{-2} \quad 3.2$$

D_{ji} est 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. Les méthodes SCR et SDR nécessitent en moyenne une itération supplémentaire par

rapport au critère GENIE idéal. [7]

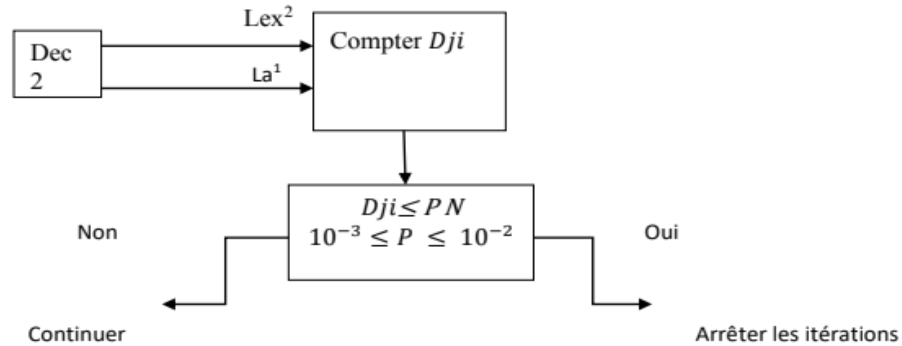


Figure 3.1 : Critère d'arrêt SDR Sign Difference Ratio [6] [29]

Critère d'arrêt SCR (Sign-change-Ratio)

Soit $C^2(i)$ le nombre de changement de signe 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) à la $i^{ème}$ itération avec $i > 1$.

Le critère SCR est très simple. IL s'agit de contrôler le rapport $\frac{C^2(i)}{N}$.

S'il est inférieur à $\leq (0.005 \sim 0.03)$ alors le décodage peut être arrêté sans presque aucune dégradation de la performance. [29]

Nous pouvons encore modifier ce critère par l'évaluation du nombre de changements de signe entre Les éléments de la séquence d'information extrinsèque Le 1 $(i - 1)$ et Le 1 (i) par conséquent, nous devons déterminer le rapport $\frac{C^n(i)}{N}$ qui doit être $\leq (0.005 \sim 0.03)$ pour arrêter le décodage.

Nous désignons par SCRM le critère d'arrêt résultat.

Dans ce cas la complexité est le double de celle du SCR classique.

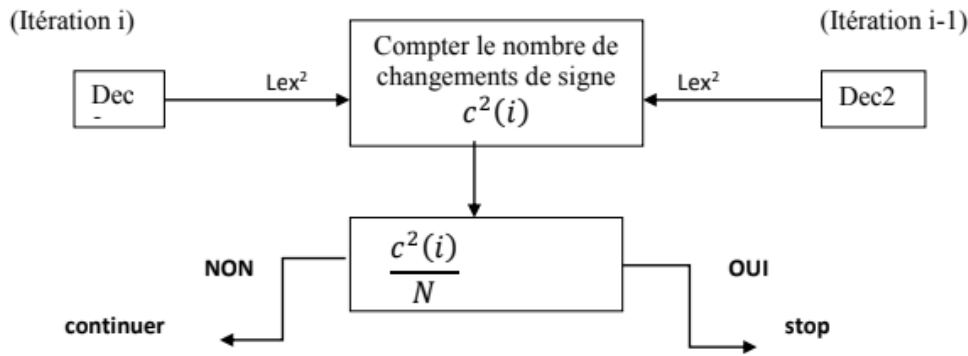


Figure 3.2: Critère d'arrêt Sign change Ratio [6] [29]

Critère d'arrêt HDA (Hard - Decision - Aided)

Le critère d'arrêt HDA [10] est une méthode efficace dérivée du concept de l'entropie croisée CE. A la $(i - 1)^{ème}$ itération, la séquence d'informations décodée $A^2(i - 1)$ à la sortie du deuxième décodeur de l'itération $(i - 1)$ est stockée dans une mémoire.[29]

L'information de fiabilité (LLR) à la fin de chaque itération peut fournir une information concernant la Convergence du processus de décodage. A la $(i - 1)^{ème}$ itération, on sauvegarde la séquence d'information de fiabilité $\Lambda_2(i - 1)$ dans une mémoire tampon. Lors de la $i^{ème}$ itération, on compare la séquence d'information décodée de fiabilité $\Lambda_2(i)$ à celle de $\Lambda_2(i - 1)$. 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 $\Lambda_1(i - 1)$ et $\Lambda_1(i)$. Nous désignons par IHDA le critère d'arrêt résultant. Dans ce cas. IL est clair que nous avons besoin de deux mémoires tampon. Donc la complexité double de celle du SCR.

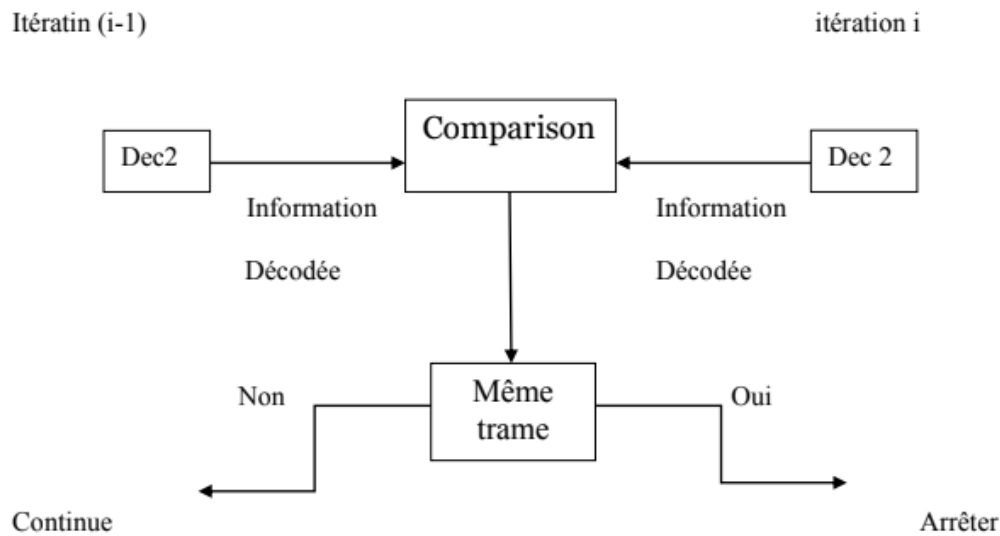


Figure 3.3: Critère d'arrêt HDA (Hard Decision Aided) [6]

Critère d'arrêt IHDA :

Cette règle utilise le fait que les informations extrinsèques ne cessent d'augmenter avec l'augmentation du nombre d'itérations, ce qui entraîne un raffinement des LLR.

Son principe est de comparer les décisions dures basées sur $((\frac{2}{\sigma^2}) y^2 d(k) + L_{e1}^i(d(k)))$ 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 $((\frac{2}{\sigma^2}) y^2 d(k) + L_{e1}^i(d(k)))$
 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 $i+1$. [6]

Critère d'arrêt GENIE :

La règle d'arrêt GENIE est une référence pour la performance de critères d'arrêt existants.

Cette règle d'arrêt est une règle théorique, qui est utile pour créer une imbattable référence de Performance par rapport à laquelle les autres règles sont mesurées. Pour cette règle, le GENIE Immédiatement reconnaît le mot décodé correct, basé sur la prescience de la séquence de bits transmise, et arrête le processus itératif avec précision et le nombre minimum d'itérations nécessaires pour trouver le mot de code correct. [6]

Dans la technique d'arrêt du GENIE, le décodeur est supposé connaître tous les bits transmis et arrête le processus de décodage lorsque tous les morceaux sont correctement décodés, à moins que le nombre d'itérations atteigne le nombre maximal d'itérations autorisé. Donc, GENIE assure les meilleures performances BER et le plus petit nombre d'itérations qui peut être atteint par n'importe quel critère d'arrêt basé sur les trames.

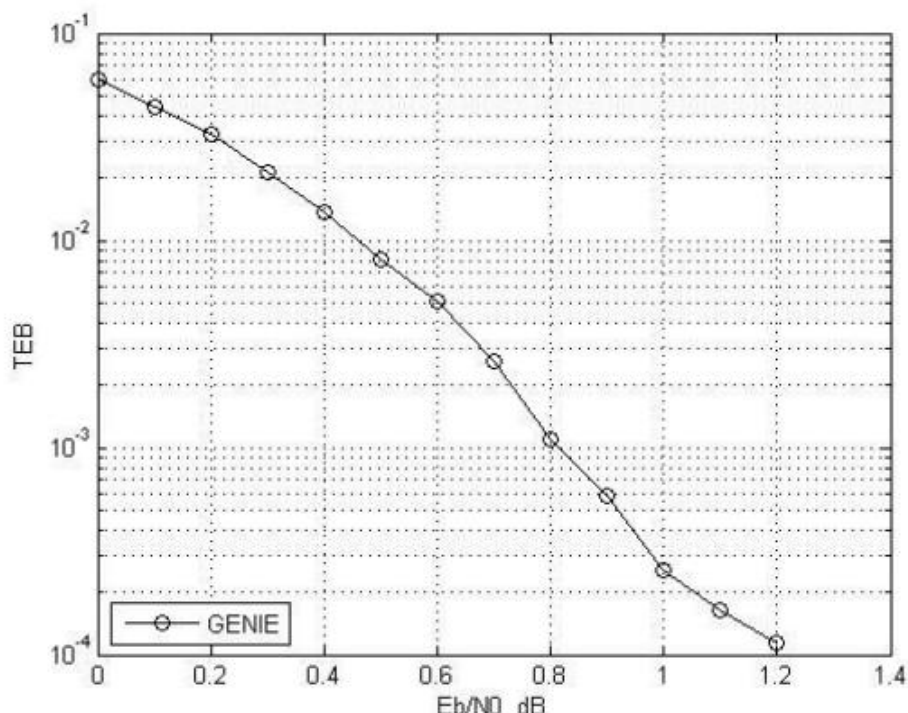


Figure 3.4 Performances du critère GENIE appliqué au turbo décodage[7]

Critère d'arrêt CRC :

IL s'agit d'une règle d'arrêt fondée sur la détection de séquences décodées erronées à l'aide d'un Code de redondance cyclique externe (CRC) appliqué à des bits décodés après la décision dure.

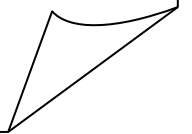
Un code de détection d'erreur séparé, tel qu'un code CRC, peut être concaténé en tant que code externe avec un turbo code interne afin de marquer les séquences erronées du turbo décodage.

La condition d'arrêt avec cette règle est satisfaite chaque fois que le syndrome du code CRC est nul.

Cette technique des sommes de contrôle CRC fait partie de la norme sans fil 3G-WCDMA (Wideband code Division Multiple Access). [34]

CHAPITRE IV

Résultats de simulation



INTRODUCTION :

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

Conditions de simulation :

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

Rendement : $R = 1/3$

Nombre de paquets = 1000

RSB minimal = 0 dB

RSB maximal = 1.1 dB

Longueur des trames $L=1024$

Algorithme de décodage : MAP

Canal de transmission :AWGN

Performances en terme de TEB de la critère CRC au turbo décodage**Condition de simulation (Mémoire Master [7]):**

Nombre de paquets = 1000

RSB minimal = 0 dB

RSB maximal = 1 dB

Longueur des trames $L=1000$

Nombre de transmission $NT=10$

Algorithme de décodage : MAP

Canal de transmission : AWGN

La partie redondance CRC a une taille de 16 bits.

Résultats de simulation :

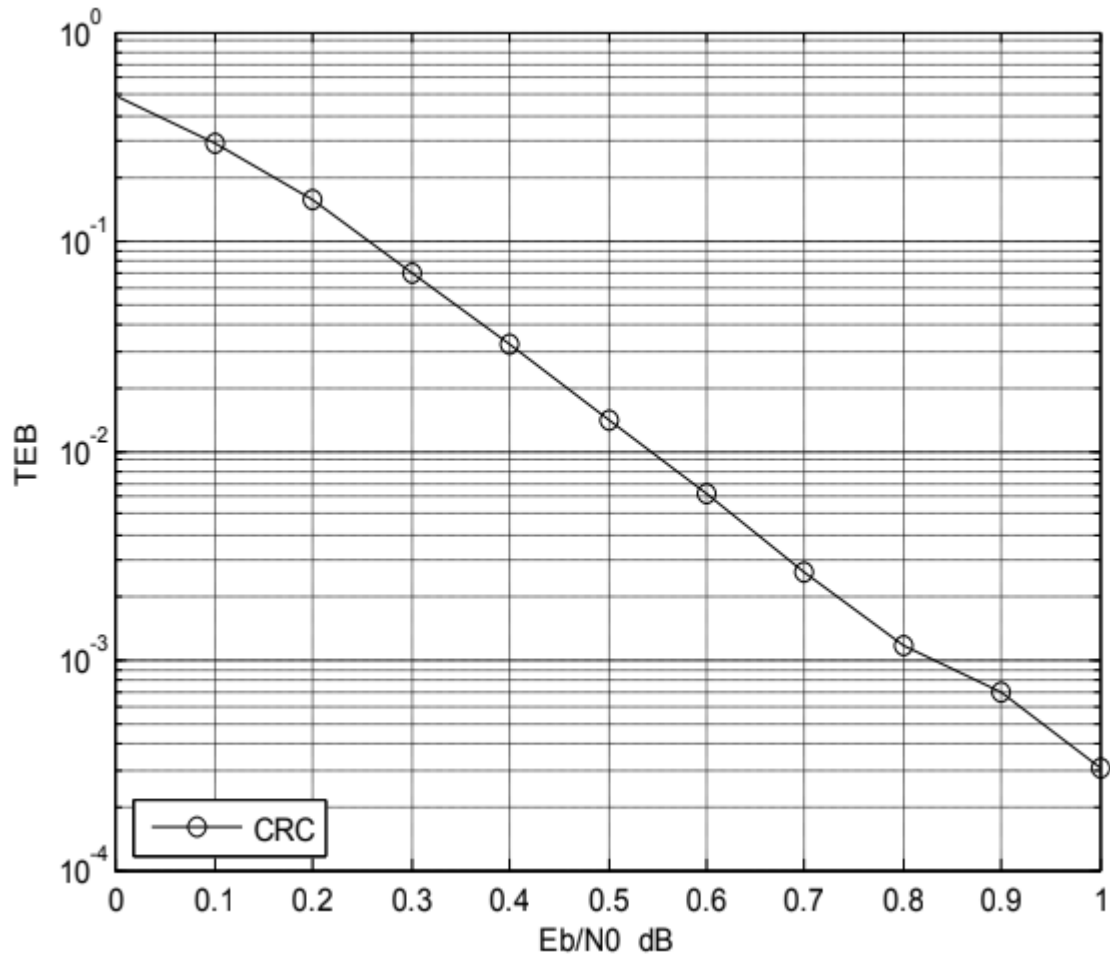


Figure 3.5 Performances en terme de TEB de la critère CRC au turbo décodage [7]

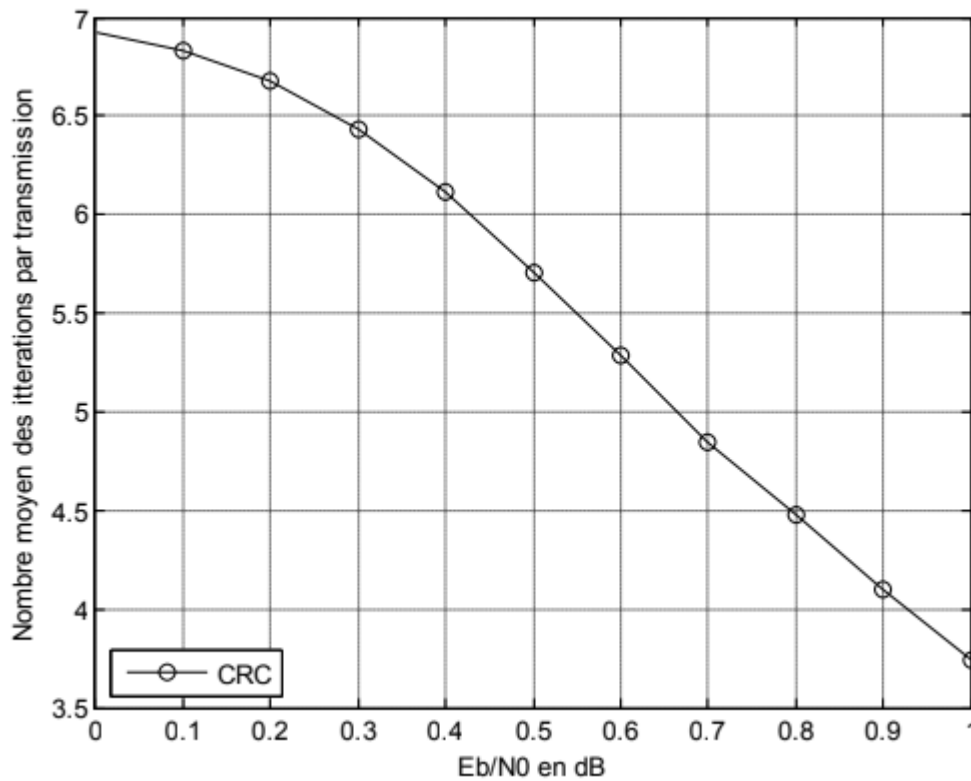


Figure 3.6 Nombre moyen des itérations par transmission du critère CRC [7]

Interprétation des résultats :

Performances comparatives : le système utilisant le CRC en double utilisation (comme critère d'arrêt et d'acquiescement) offre des performances similaires au turbo décodeur avec 7 itérations, ce qui est un résultat intéressant.

Taux d'erreur binaire (TEB) : Au rapport signal sur bruit (RSB) de 1 dB, le TEB est 3×10^{-4} .

Itérations : Le nombre moyen d'itérations a considérablement diminué avec l'augmentation du RSB, passant de 6.9 itérations à 0 dB à 3.7 itérations à 1 dB. Ceci indique une amélioration de l'efficacité du décodage avec un meilleur rapport signal sur bruit [7].

Critère SCR appliqué au turbo décodage :

Condition de simulation :

Nombre de paquets = 1000

RSB minimal = 0 dB

RSB maximal = 1.1dB

Longueur des trames $L=1024$

Algorithme de décodage : MAP

Seuil=0.05

Canal de transmission :AWGN

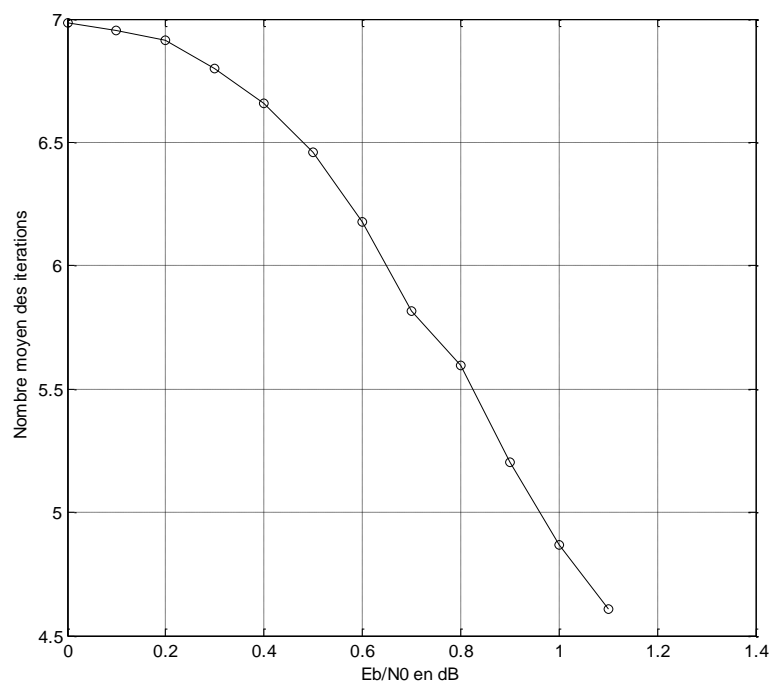


Figure 3.7 Nombre moyen des itérations du SCR

Interprétation des résultats :

La figure 3.7 montre le nombre moyen des itérations de SCR. A un rapport signal sur bruit de 1 dB le nombre moyen des itérations est 4.9 iteration

A 1.1 dB le nombre moyen des itérations est diminue à 4.4 iteration. Donc un SCR plus bas peut suggérer que les gradients ou les résidus sont relativement stables ,

ce qui peut faciliter une convergence plus rapide et réduire le nombre moyen des itérations.

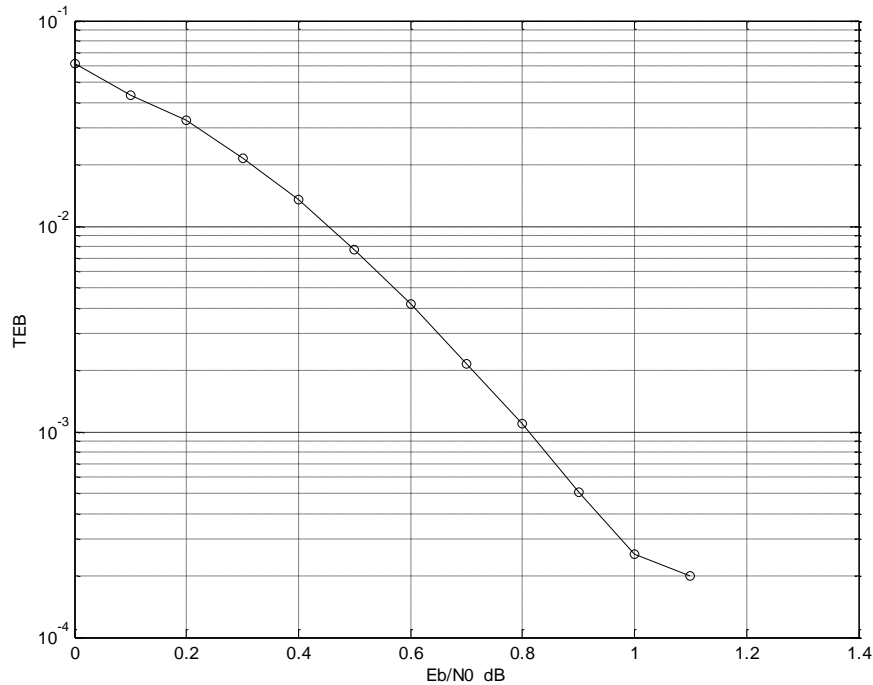


Figure 3.8 Performances d'un critère SCR

Interprétation des résultats :

La figure 3.8 présente le taux d'erreur binaire d'un critère SCR appliqué au turbo décodage .

A un rapport signal sur bruit 0.5 dB le taux d'erreur binaire est $8 \cdot 10^{-3}$

A un rapport signal sur bruit de 1 dB, le taux d'erreur binaire est $2 \cdot 10^{-4} < TEB \leq 3 \cdot 10^{-4}$

Nombre moyen des itérations du SDR

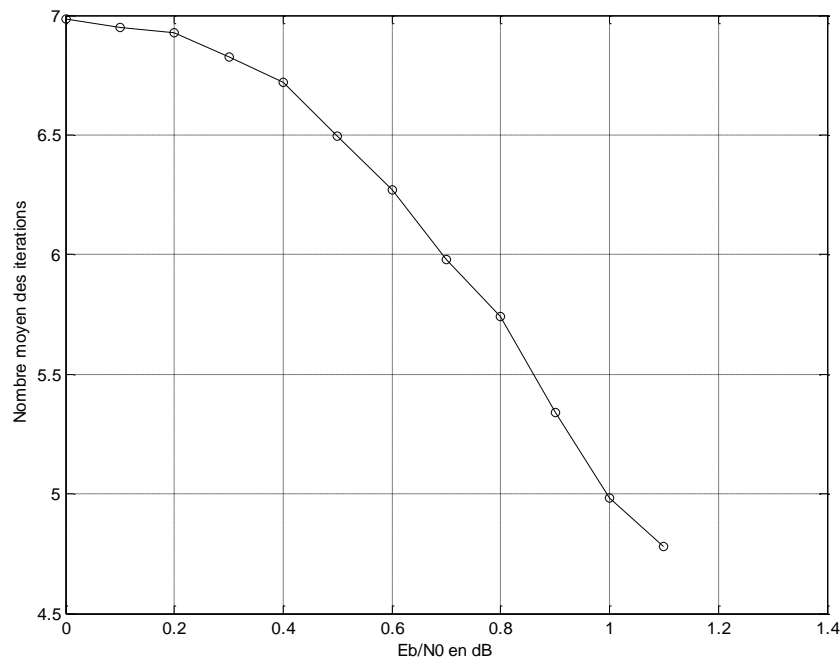


Figure 3.9 Nombre moyen des itérations du SDR

Interprétation des résultats :

La figure 3.9 présente le nombre moyen d'itération de SDR. Le nombre d'itérations nécessaires diminue en fonction du rapport signal sur bruit..

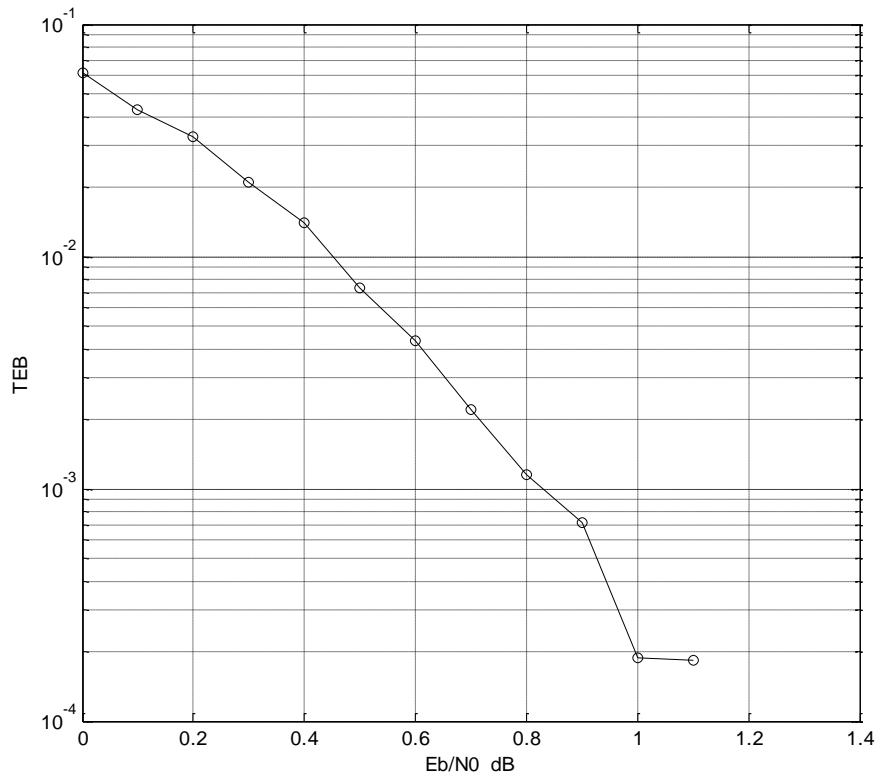


Figure 4.1 performance d'un critère SDR

Interprétation des résultats :

La figure 4.1 montre le taux d'erreur binaire d'un critère SDR appliqué au turbo décodage. Ceci montre le bon fonctionnement de ce critère.

Critère HDA appliqué au turbo décodage

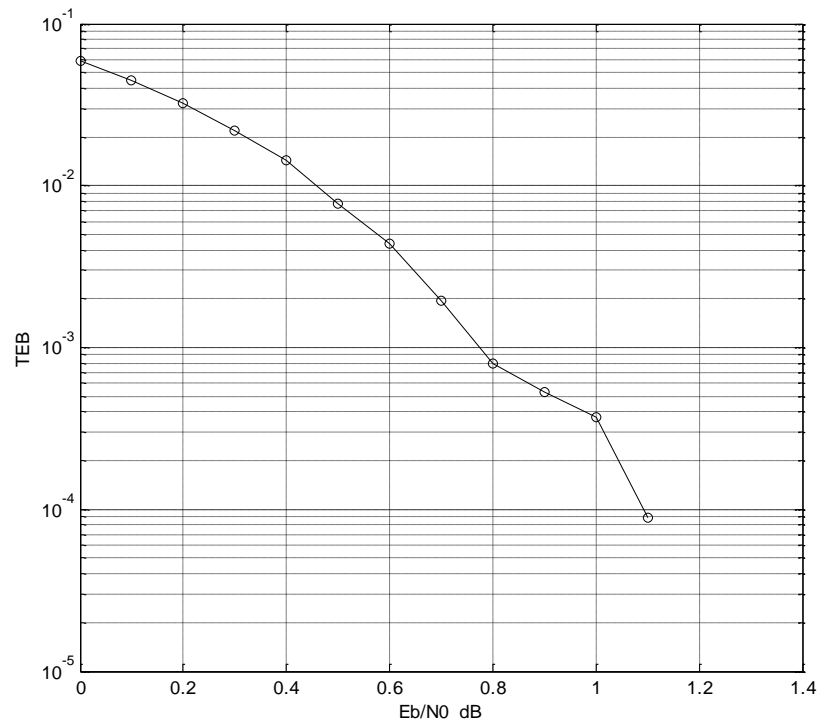


Figure 4.2 Performances d'un critère HDA

Interprétation des résultats :

La figure 4.2 montre le taux d'erreur binaire TEB d'un critère HDA appliqué au turbo décodage. la diminution du taux d'erreur binaire confirme l'efficacité du critère HDA.

A un rapport signal sur bruit (RSB) de 0.8 db le $TEB < 10^{-3}$

a un rapport signal sur bruit (RSB) de 1.1 dB le TEB est de 10^{-4}

cela signifie que : une augmentation du rapport E_b/N_0 améliore la qualité du signal et réduit le TEB, ce qui se traduit par une communication plus fiable.

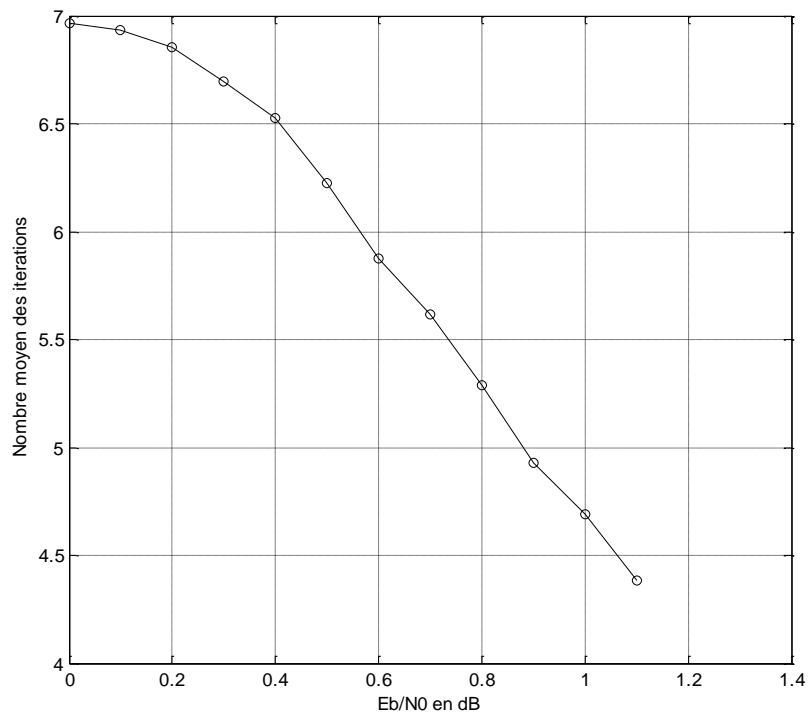


Figure 4.3 Nombre moyen des itérations du HDA

Interprétation des résultats :

La figure 4.3 présente le nombre moyen des itérations de HDA .le nombre moyen des itérations est réduit de 7 itérations à 4.4 itérations.

Conclusion Générale

Conclusion Générale :

Durant tout ce mémoire, nous avons étudié des critères d'arrêt Turbo à base des décisions dures.. Les codes convolutifs et les turbo codes sont en effet des techniques importantes en télécommunications pour améliorer la fiabilité de la transmission des données. Les algorithmes de décodage Viterbi et MAP (Maximum A Posteriori) sont utilisés pour décoder ces codes efficacement. Évaluer les performances des décodeurs turbo MAP implique généralement d'analyser des mesures telles que le taux d'erreur binaire (BER) du rapport signal sur bruit (SNR). Cela permet de comprendre comment ces décodeurs se comportent dans des conditions réelles de transmission. Les résultats de simulation montrent une diminution des taux d'erreurs binaires (TEB) après chaque itération. Cependant, il est noté que fixer un nombre précis d'itérations pourrait être inutile, car certaines trames sont correctement décodées après seulement quelques décodages. Cela suggère que le processus itératif peut être adaptatif et que le système peut converger vers une solution même sans un nombre fixe prédéterminé d'itérations. La mise en œuvre d'un critère d'arrêt turbo basé sur les décisions dures représente une approche efficace pour améliorer les performances des décodeurs turbo. En utilisant les critères à base des décisions dures, il a été démontré que les taux d'erreurs binaires (TEB) diminuent de manière significative après chaque itération, ce qui indique une convergence rapide du processus de décodage. De plus, aux forts rapport signal sur bruit, le turbo décodeur utilise un nombre faible des itérations.

En conclusion, l'approche basée sur les décisions dures représente une contribution significative pour optimiser les décodeurs turbo dans divers contextes d'application des communications numériques.

Références

Bibliographie

- [1] CLaude Berrou et.al. "Code et Turbocodes". springer paris Berlin Heidelberg New York . paris, "Ecole nationale supérieure des Télécommunications des bretagne" , springer-verlag france (2007)france ,page165.
- [2] Hendaoui Mounira et Benakcha Abdelhamid. "courrier du savoir scientifique et technique". Applications Des turbocodes dans un système multi usagers Wcdma, 12 (12),pages15-19. 2014
- [3] LATIFA Mostari, " Codage de canal ", Polycopié de cours et travaux dirigés, Université Hassiba Ben Bouali, Chlef, (2016-2017). (p. 36)
- [4] G.D.Forney,"The viterbi algorithm,"proceedings of the IEEE, vol.61, no.3, pp.268278, March 1973.
- [5] Mori.yvon,"Théorie de l'information et du codage", hermes science, vol.5, pages(138-140), (2006) paris.
- [6] Bouzidi Hanane, Mostefai chaimaa,"critère de L'entropie croisée et son application au turbo décodage", Mémoire Master, Université Dr. Tahar Molay, 2023.
- [7] Bouazza Rabie Lakhdar, Kaoudj salima" Critères d'arrêt CRC pour les turbo codes et son double intérêt pour les systèmes H-ARQ ", Mémoire Master, Université Dr. Tahar Molay, 2022.
- [8] Geneviève Baudoin "Radiocommunications numériques/ 1", principes, modélisation et simulation, Dunod.page(375), (2002) paris.
- [9] **Raphaël Le Bidan**, "Nouvelles formes d'onde en imagerie, localisation et communications", **Département Signal et Communications (98040)**. Journée d'études SEE, Université européenne de Bretagne, 2005 paris
- [10] Abdelghafour Berraissoul, "codage de canal : une introduction aux codes correcteurs ".ellipses, (2021) pages (p 421-422)
- [11] Thibaud Tonnelier, "Contribution à l'amélioration des performances de décodage des turbo codes : algorithmes et architecture. Electroniques. Université de Bordeaux, 2017. English.
- [12] A. H. Osseiran, " Sur le décodage des codes turbo ", Mémoire de maîtrise des sciencesappliquées (Génie électrique), Oct 1999, Université de Montréal, Ecole Polytechnique de Montréal, Canada.
- [13] Alexis Nasr, Entropie croisée .Introduction à l'apprentissage automatique ,Master Sciences cognitives, Aix Marseille Université(2018).
- [14] AMAMRA Imed, « Codage Canal et Techniques Efficaces de Décodage Itératif », Thèse Présentée en Vue de l'Obtention du Diplôme de Doctorat ès Sciences, sous la direction de Dr. DEROUCHE Nadir, Skikda, Université 20 Août 1955 de Skikda, 2018.

- [15] Bouzidi Hanane, Mostefai chaimaa,"critère de L'entropie croisée et son application au turbo décodage", Mémoire Master, Université Dr. Tahar Molay, 2023,page(35-36).
- [16] F. Zhai and I. J. Fair, "New error detection techniques and stopping criteria for turbo decoding, " in 2000 Canadian Conference on Electrical and Computer Engineering, Halifax, NS, Canada, 2000, pp. 58–62.
- [17] A. C. Reid, T. A. Gulliver, and D. P. Taylor, "Convergence and errors in turbodecoding," IEEE Transactions on Communications, vol. 49, no. 12, pp. 2045–2051, Dec 2001.
- [18] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," in 1994 IEEE GLOBECOM. Communications: The Global Bridge, 1994, vol. 3, pp. 1298–1303.
- [19] W. J. E. Sr, "Turbo decoder stopping based on mean and variance of extrinsics," US6879648 B2, 12-Apr-2005.
- [20] T. D. Wolf and W. J. E. Sr, "Turbo decoder stopping criterion improvement," US6898254 B2, 24-May-2005.
- [21] N. Y. Yu, M. G. Kim, Y. S. Kim, and S. U. Chung, "Efficient stopping criterion for iterative decoding of turbo codes," Electronics Letters, vol. 39, no. 1, pp. 73–75, Jan. 2003.
- [22] Z. Wang and K. K. Parhi, "On-line extraction of soft decoding information and applications in VLSI turbo decoding," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 49, no. 12, pp. 760–769, Dec. 2002.
- [23] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," IEEE Transactions on Information Theory, vol. 42, no. 2, pp. 429–445, Mar. 1996.
- [24] R. Mohamad, H. Harun, M. Mokhtar, W. A. W. Adnan, N. M. Anas, and K. Dimyati, "Early stopping turbo iteration at low SNR for CE-based stopping criteria," in TENCON 2014 - 2014 IEEE Region 10 Conference, 2014, pp. 1–4.

- [25] F. Alberge, "On Some Properties of the Mutual Information Between Extrinsic with Application to Iterative Decoding," *IEEE Transactions on Communications*, vol. 63, no. 5, pp. 1541–1553, May 2015.
- [26] L. Li, Q. Wang, and C. Y. Lu, "A Novel Stopping Criterion for Turbo Decoding," in *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, 2006, vol. 1, pp. 201–205.
- [27] J. Y. Chen, L. Zhang, and J. Qin, "Average-entropy variation in iterative decoding of turbo codes and its application," *Electronics Letters*, vol. 44, no. 22, pp. 1314–1315, Oct. 2008.
- [28] M. E. Buckley and S. B. Wicker, "A neural network for predicting decoder error in turbo decoders," *IEEE Communications Letters*, vol. 3, no. 5, pp. 145–147, May 1999.
- [29] 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, pp. 1117–1120, Aug. 1999.
- [30] Y. Wu, B. D. Woerner, and W. J. Ebel, "A simple stopping criterion for turbo decoding," *IEEE Communications Letters*, vol. 4, no. 8, pp. 258–260, Aug. 2000.
- [31] R. Y. Shao, M. Fossorier, and S. Lin, "Two simple stopping criteria for iterative decoding," in *Proceedings. 1998 IEEE International Symposium on Information Theory (Cat. No.98CH36252)*, 1998, p. 279-.
- [32] Chun Ling Kei and Wai Ho Mow, "Improved stopping criteria for iterative decoding of short-frame multi-component turbo codes," 2002, vol. 1, pp. 42–45.
- [33] T. M. N. Ngatched and F. Takawira, "Simple stopping criterion for turbo decoding," *Electronics Letters*, vol. 37, no. 22, pp. 1350–1351, Oct. 2001.
- [34] A. Shibutani, H. Suda, and F. Adachi, "Reducing average number of turbo decoding iterations," *Electronics Letters*, vol. 35, no. 9, pp. 701–702, Apr. 1999.
- [35] A. Shibutani, H. Suda, and F. Adachi, "Complexity reduction of turbo decoding," in *Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th*, 1999, vol. 3, pp. 1570–1574 vol.3.