

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 سعيادة -
Université Saida Dr Tahar Moulay –
Faculté de TECHNOLOGIE



MEMOIRE

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

En : Télécommunications

Spécialité : Réseaux et Télécommunications

Par : BOUZIDI Hanane & MOSTEFAI Chaimaa

Sujet

Critère de L'entropie Croisée et son application au turbo décodage

Soutenue publiquement en Juin 2023, devant le jury composé de :

Dr B.BOUYEDOU	Univ. Saida	Président
Dr A.OUARDI	Univ. Saida	Rapporteur
Dr A.TAMI	Univ. Saida	Examineur

Année universitaire 2022/2023

Dédicace Hanane

Au nom de Dieu, le clément, le tout-puissant, le très miséricordieux,

je dédie ce travail à :

*À mes chers parents qui n'ont jamais cessé de nous donner l'amour
nécessaire pour être ce que nous sommes aujourd'hui.*

*À mes frères, qui sont une source constante de soutien et
d'encouragement dans tous nos projets.*

À toute ma famille.

*À mes amis fidèles, ceux qui ont été présents à nos côtés tout au long
de notre parcours.*

À toutes les personnes que j'aime et qui m'aiment.

À tous ceux qui ont partagé notre joie et nos peines.

À tous ceux qui nous estiment et croient en nous.

Dédicace Chaima

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ
(قل اعملوا فیسری اللہ عملکم ورسولہ و المؤمنون)
صدق اللہ العظیم

*A qui je le préfère à moi-même et pourquoi pas, elle s'est sacrifiée pour moi, et n'a ménagé aucun effort pour toujours me rendre heureuse ma mère bien-aimée **MOUKHTARIA**.*

Nous marchons sur les chemins de la vie, et ceux qui contrôlent notre esprit restent dans tous les chemins que nous prenons.

*Propriétaire d'une bonne face et de bonnes actions, il ne m'a pas épargné toute sa vie mon cher père, **MILOUD***

*Pour mes frères **MOHAMED, KHEIRE EL DINNE** et mes sœurs **FOUZIA, LATIFA, FATIMA, BOUCHRA**, ils ont servi de soutien pour compléter la recherche.*

À mes amis et à tous ceux qui m'ont soutenu et m'ont aidé avec tout ce qu'ils avaient et de plusieurs façons.

Je vous présente cette recherche et j'espère qu'elle vous satisfera.

Remerciements

Nous remercions Dieu Tout-Puissant qui nous a aidés à mener à bien cette recherche scientifique et qui nous a inspiré la santé, le bien-être et la détermination,

Dieu soit loué,

Nous adressons nos sincères remerciements et notre appréciation au Prof. Dr.

Supervisor (**Dr A.OUARDI**) pour tous les conseils et les informations précieuses qu'il nous a fournis et qui ont contribué à l'éloge du sujet de notre étude dans ses divers aspects, et nous adressons également nos sincères remerciements aux membres du comité de discussion estimé.

Et n'oubliez pas de présenter la grande forme à tous les professeurs respectés et les professeurs de l'Université de Dr Tahar Moulay .

Nous vous disons merci beaucoup pour tous vos efforts.

Résumé

Résumé

Le critère EC d'entropie croisée, également appelé CE, est utilisé pour arrêter le turbo décodage lorsque le turbo décodeur fournit une trame décodée sans erreur. Ce critère utilise une information extrinsèque, généralement notée L_e , pour calculer la quantité $CE(i)$ à chaque itération pour prendre une décision.

L'objectif principal du projet est d'appliquer le critère d'entropie croisée EC au turbo décodage et de vérifier son bon fonctionnement. Nous évaluons également ses performances et sa capacité à réduire les itérations. Une comparaison est faite avec une référence idéale (théorique) GENIE.

Table des Matières

Table des Matières

Dédicace	i
Remerciements	iii
Resume	iv
Table des Matières	v
Liste des figures.....	ix
Listes des acronymes et abréviation	x
Introduction générale.....	1

Chapitre 1 les turbocodes

Introduction:.....	2
1.1.Code convolutif:	2
1.1.1.Definition :	2
1.2. Principe de décodage :	3
1.3. L'algorithme de décodage de Viterbi :.....	3
1.4. L'algorithme de décodage BCJR :	3
1.5. Principe de codage :.....	4
1.5.1 Représentation polynomiale :.....	4
1.5.2.b Exemple :	5
1.6. Représentations graphiques des codes Convolutif:.....	6
1.7. Les codeurs RSC:.....	7
1.7.1 Exemple :.....	8
1.8. Algorithme de Viterbi:	8
1.9. Méthode de Décodage MAP des codes convolutifs:.....	9
1.9.1 Algorithme MAP:.....	9
1.10. Les turbocodes :	11

Table des Matières

1.10.1 Introduction:.....	11
1.11. Le turbo codage des turbocodes convolutifs :	12
1.11.1 Introduction :.....	12
1.11.2 Le principe de base du turbo codage :.....	12
1.11.3 Le schéma de turbo codage :.....	12
1.12. Le turbo décodage des turbocodes convolutifs :	14
1.12.1 Introduction :.....	14
1.12.2 Le principe de base du turbo décodage :.....	14
1.12.3 le schéma de base du décodage des turbocoes convolutifs :	14

Chapitre 2 Critères d'arrêt pour le turbo décodage

Introduction :	16
2.1 Décodeur turbo :	17
2.2 Critère d'arrêt SCR :.....	19
2.3 Critère d'arrêt SDR :.....	20
2.4 Critère d'arrêt HDA :.....	21
2.5 Critère d'arrêt IHDA :	22
2.6 Critère d'arrêt GENIE :.....	22
2.7 Critère CRC :	22
2.8 Critère de la moyenne des LLR :.....	23

Chapitre 3 Critère de l'entropie croisée

Introduction :	25
3. Propriétés de convergence du décodage itératif via l'Entropie Croisée :.....	25
3.1 Introduction	25
3.2. Les propriétés de convergence de cet algorithme sont les suivantes	25
3.2.1 Convergence globale	25

Table des Matières

3.2.2 Convergence locale	25
3.2.3 Vitesse de convergence.....	26
3.3 Principe de fonctionnement du critère de l'Entropie Croisée :	28
3.3.1 Introduction :	28
3.4 Conclusion:	31

Chapitre 4 Résultats de simulation

4.1 Introduction :	32
4.2 Performances d'un Turbo décodeur [1, 5/7] _{oct}	32
4.2.1 Condition de simulation :	32
4.2.2 Résultats de simulation :	32
4.2.3 Interprétation des résultats :	33
4.3. Performances du critère GENIE appliqué au turbo décodage :	34
4.3.1. Condition de simulation :	34
4.3.2 Résultats de simulation :	34
4.3.3 Interprétation des résultats :	34
4.4. Performances du critère l'entropie croisée :	35
4.4.1. Condition de simulation :	35
4.4.2 Résultats de simulation :	36
4.4.3 Interprétation des résultats :	37
4.5 Conclusion importante :	37
Conclusion générale :	38
Bibliographie	39

Liste des figures

Liste des figures

<i>Figure 1- 1: Exemple de codeur Convolutif.....</i>	<i>6</i>
<i>Figure 1- 2: Exemple de codage d'information à travers un codeur Convolutif.</i>	<i>6</i>
<i>Figure 1- 3 Code Convolutif Récurif Systématique RSC[1, 5/7].</i>	<i>7</i>
<i>Figure 1- 4: Exemple de codage d'une séquence par le codeur de la Figure 1-3.</i>	<i>8</i>
<i>Figure 1- 5: Codeur turbo par composition parallèle et entrelacement de deux codeurs RSC... </i>	<i>13</i>
<i>Figure 1- 6: Décodage itératif d'un turbo-code</i>	<i>15</i>
<i>Figure 2- 1 : Itérations successives de décodage turbo.....</i>	<i>17</i>
<i>Figure 2- 2 : Schéma de principe d'un décodeur turbo.....</i>	<i>18</i>
<i>Figure 2- 3 : Critère d'arrêt Sign change Ratio</i>	<i>20</i>
<i>Figure 2- 4 : Critère d'arrêt SDR Sign Difference Ratio</i>	<i>21</i>
<i>Figure 2- 5 : Critère d'arrêt HDA Hard Decision Aided.....</i>	<i>21</i>
<i>Figure 2- 6 : Valeurs des LLR</i>	<i>23</i>
<i>Figure 2- 7 :Nombre d'erreurs avec les itérations pour 3 paquets.</i>	<i>23</i>
<i>Figure 2- 8 : La moyenne absolue des valeurs LLR en fonction du nombre des itérations</i>	<i>24</i>
<i>Figure 3- 1 : Turbo récepteur série</i>	<i>29</i>
<i>Figure 4- 1 : Performances d'un Turbo décodeur MAP[1, 5/7]oct.....</i>	<i>33</i>
<i>Figure 4- 2 : Performances du critère GENIE appliqué au turbo décodage</i>	<i>34</i>
<i>Figure 4- 3 : Nombre moyen des itérations du critère GENIE appliqué au turbo décodage</i>	<i>35</i>
<i>Figure 4- 4 : Performances d'un critère de l'entropie croisée EC.....</i>	<i>36</i>
<i>Figure 4- 5 : Nombre moyen des itérations du critère de l'entropie croisée EC</i>	<i>37</i>

Liste des acronymes et abréviation

Listes des acronymes et abréviation

BCJR : Bahl cocke Jelinek Raviv

MAP : Maximum A Posteriori

LLR : Logarithmic Likelihood Ratio

LRV : Log-rapport de vraisemblance

SCR : Sign Change Ratio

SDR : Sign Difference Ratio

BER : Bit Error Rate

FER : Frame Error Rate

HDA : Hard Decision Aided

IHDA : Improved Hard Decision Aided

CRC : Cyclic Redundancy Check

WCDMA : Wideband Code Division Multiple Access

EC : Cross Entropy

RSB: Rapport Signal/Bruit

RSC : Rapport Signal Convolutional

TEB : Taux d'erreur binaire

Introduction générale

Le turbo décodage est une technique de décodage itératif utilisée dans les systèmes de communication numériques, en particulier dans les systèmes de codage à correction d'erreurs. L'algorithme turbo décode les données en utilisant plusieurs itérations (échange d'informations entre des décodeurs de canal et un décodeur de contrôle). Cependant, il est inutile de fixer le nombre des itération du turbo décodage. Un critère d'arrêt est donc nécessaire. Le critère d'arrêt dans le turbo décodage détermine quand l'algorithme doit s'arrêter, c'est-à-dire combien d'itérations doivent être effectuées.

Le critère de l'Entropie Croisée EC, également connu sous le nom Cross-Entropy CE, est utilisé pour arrêter le turbo décodage lorsque le turbo décodeur fournit une trame décodée exempte d'erreurs. Ce critère utilise les informations extrinsèques notées généralement Le et calcule à chaque itération une quantité $CE(i)$ pour prendre une décision. Ce mécanisme permet d'éliminer plusieurs itérations et accélérer la vitesse de décodage. Ce qui réduit donc le temps de traitement au niveau du récepteur.

L'objectif de ce modeste travail est d'appliquer le critère de l'entropie croisée EC au turbo décodage, et vérifier son bon fonctionnement. Nous évaluons aussi ses performances et sa puissance de réduction des itérations. Une comparaison est effectuée avec le critère idéal (théorique) GENIE.

Chapitre 1

les turbocodes

Introduction:

Les codes convolutifs, également connus sous le nom de codes convolution, sont des codes correcteurs d'erreurs utilisés dans la transmission de données numériques. Ils utilisent des techniques de convolution pour coder les données en ajoutant des bits de redondance qui permettent de détecter et de corriger les erreurs de transmission.

Les codes convolutifs sont largement utilisés dans les systèmes de communication numériques, tels que les réseaux de téléphonie mobile, les modems et les satellites. Ils sont également utilisés dans les systèmes de stockage de données, tels que les disques durs et les lecteurs de CD.

La principale caractéristique des codes convolutifs est leur capacité à détecter et à corriger les erreurs de transmission en temps réel. Cela les rend très efficaces pour les applications qui nécessitent une transmission rapide et fiable de données numériques. Les codes convolutifs sont également très flexibles et peuvent être adaptés à une variété de configurations et de formats de données.

Les codes convolutifs peuvent être conçus pour une variété de rendements (taux de codage), qui déterminent la quantité de redondance ajoutée aux données. Les rendements les plus élevés offrent une meilleure correction d'erreur, mais nécessitent également plus de bande passante pour transmettre les données.

1.1 Code convolutif:

1.1.1 Définition :

Les principaux inconvénients du codage en bloc peuvent être évités par une approche différente de la problématique du codage : C'est le cas du codage Convolutif (Convolutional Coding). Ce type de codage ajoute systématiquement de la redondance au message codé au fur et à mesure que les bits d'information sont livrés au codeur. Le message codé se forme ainsi itérativement en utilisant un registre à décalage. Les codes convolutifs forment une classe extrêmement souple et efficace de codes correcteurs d'erreurs.. Théoriquement, ils ont les mêmes caractéristiques que les codes en blocs sauf pour la valeur de leur dimension et leur longueur (Matrice génératrice). Les codes convolutif s'appliquent théoriquement sur des séquences infinies de symboles d'information et génèrent des séquences infinies de symboles codés [1].

1.2. Principe de décodage :

Le décodage des codes convolutifs est une tâche complexe, mais il est réalisé par des algorithmes qui peuvent détecter et corriger les erreurs de transmission des données. Il existe plusieurs algorithmes de décodage de codes convolutifs, dont les plus courants sont les algorithmes de décodage Viterbi et MAP (BCJR).

Le principe de base du décodage des codes convolutifs est de comparer la séquence de données reçue avec les différentes séquences possibles qui pourraient avoir été envoyées. En d'autres termes, l'algorithme de décodage cherche la séquence de données qui est la plus proche de la séquence de données reçue.

1.3. L'algorithme de décodage de Viterbi :

Il est l'un des algorithmes de décodage les plus couramment utilisés pour les codes convolutifs. Il utilise un arbre de décision pour représenter toutes les séquences possibles de données et évalue chaque chemin possible pour trouver le chemin le plus probable. L'algorithme de Viterbi est capable de détecter et de corriger les erreurs dans la séquence de données, mais sa complexité augmente exponentiellement avec la longueur de la séquence de données [2].

1.4. L'algorithme de décodage BCJR :

Il utilise une approche statistique pour estimer la séquence de données la plus probable. Il utilise des informations sur les probabilités d'erreur de transmission pour évaluer chaque chemin possible et déterminer la séquence de données la plus probable. L'algorithme de BCJR est plus complexe que l'algorithme de Viterbi, et il nécessite l'estimation du niveau du bruit (variance).

En résumé, le décodage des codes convolutifs repose sur l'utilisation d'algorithmes de décodage qui comparent la séquence de données reçue avec les différentes séquences possibles pour déterminer la séquence la plus probable.

1.5. Principe de codage :

1.5.1 Représentation polynomiale :

A partir de la fonction génératrice ou de la transformation en z , il est possible d'associer à une suite d'éléments binaires une série formelle.

Dans le cas des codes Convolutif, les éléments binaires de la suite sont émis au cours du temps, l'indéterminée en z^{-1} sera alors notée par convention D , pour « Delay ». De plus, l'ensemble de ces séries formelles peuvent être prolongées dans le corps des séries de Laurent de la forme:

$$A(D) = \sum a_t D^t \quad (1.1)$$

En notant $g_i(D)$ les polynômes générateurs du codeur tels que :

$$g_i(D) = g_{i0} + g_{i1} D + \dots + g_{i(K-1)} D^{K-1} \quad (1.2)$$

On obtient la matrice génératrice, notée $G(D)$, définie par :

$$G(D) = \begin{pmatrix} g_{1,1}(D) & \dots & g_{1,n}(D) \\ \vdots & \dots & \vdots \\ g_{k,1}(D) & \dots & g_{k,n}(D) \end{pmatrix} \quad (1.3)$$

Il est alors possible d'écrire la relation liant les séquences d'entrée et de sortie [1] :

$$c(D) = d(D).G(D) \quad (1.4)$$

1.5.2 Exemple :

Un exemple de codeur Convolutif de rendement $R = 1 / 2$ transformant un mot

$U = [u_0 u_1 \dots u_{k-1}]$ en un mot $C = [c_0 c_1 \dots c_{n-1}] = [c_0^1 c_0^2 c_1^1 c_1^2 \dots c_{k-1}^1 c_{k-1}^2]$ est

représenté sur la figure (1-1). La valeur u_i correspond au bit qui parvient à l'entrée du codeur à l'instant k .

Les bits de sortie sont alors déterminés par : $c_k^1 = u_k + u_{k-1} + u_{k-2}$ et $c_k^2 = u_k +$

u_{k-2} (additions modulo 2). Supposons que les deux registres soient initialement à 0 et que le codeur reçoive le message $U = [1011]$

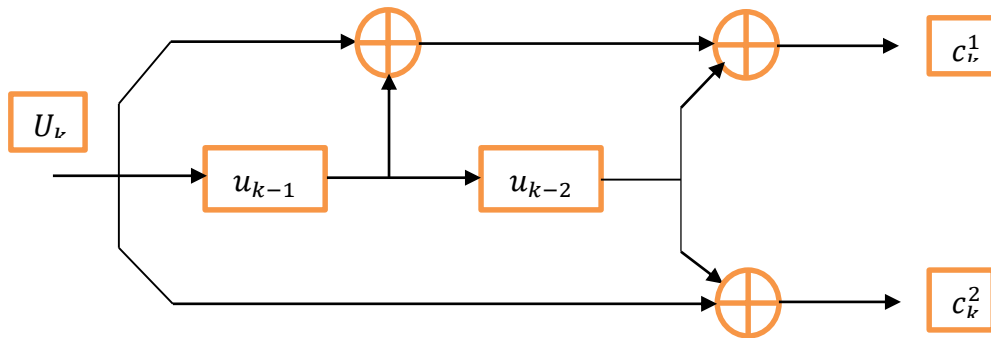


Figure 1-1 : Exemple de codeur Convolutif.

La figure (1-1) illustre le fait que la séquence codée devienne $C = [11100001]$, et que les registres à la fin soient à l'état 1 ($u_3 ; u_2 = 1$).

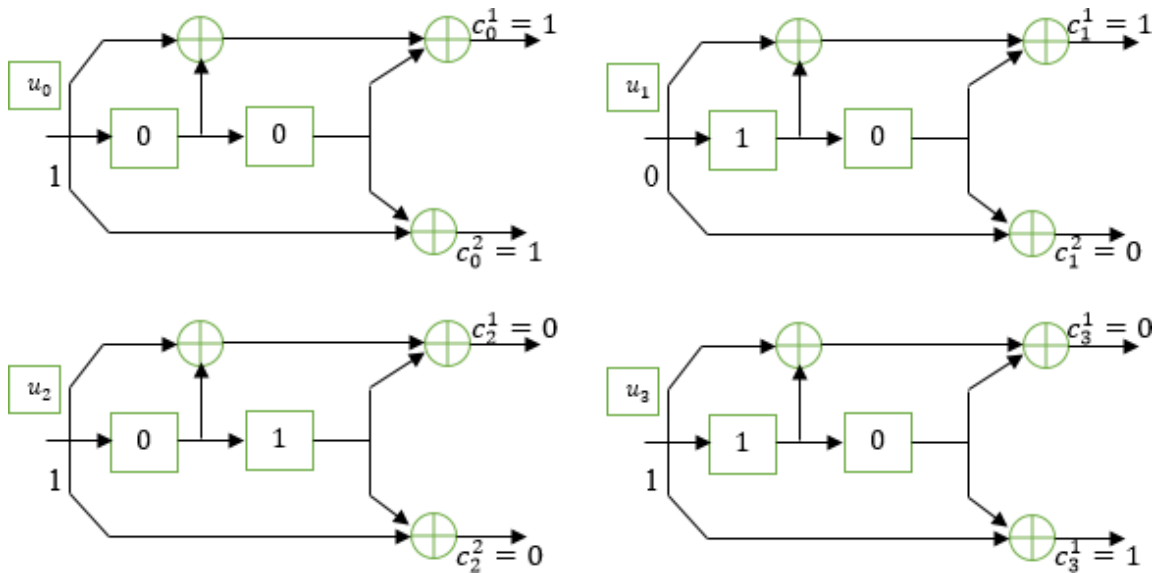


Figure 1-2: Exemple de codage d'information à travers un codeur Convolutif.

1.6. Représentations graphiques des codes Convolutif:

L'idée d'une représentation graphique d'un code Convolutif provient des caractéristiques markoviennes de la sortie du codeur. En effet, la sortie du codeur dépend de son entrée et de ses états. Les graphes équivalents à la représentation polynômiale sont souvent plus faciles à manipuler et permettent de dériver des résultats plus puissants. Tout code convolutif est représenté par trois graphes équivalents mais différents : l'arbre du code, le treillis du code et le diagramme d'états.

1.7. Les codeurs RSC:

Un codeur de forme RSC est un codeur qui est systématique et récursif. Le terme systématique signifie que les k bits d'entrée du codeur se retrouvent en sortie du codeur, donc que la matrice génératrice d'un codeur RSC sera composée d'une matrice identité. Le terme récursif signifie que les sorties du codeur peuvent dépendre des sorties précédentes, ce qui implique que certaines sorties seront rebouclées en entrée. Il en découle que les polynômes générateurs d'un codeur RSC ne seront pas de simple polynômes, mais des fractions rationnelles polynomiales.

Un codeur convolutif systématique et récursif (RSC) est obtenu à partir d'un codeur convolutif non systématique et non récursif en réinjectant une de ses sorties codées à son entrée. Figure 1-2 montre un codeur convolutif classique.

Le codeur RSC de ce codeur convolutif classique est représenté comme $G = [1, g_2/g_1]$ où la première sortie (représentée par g_1) est renvoyée à l'entrée. Dans la représentation ci-dessus, 1 désigne la sortie systématique. Figure (1-3) montre le codeur RSC résultant.

Les bons codes peuvent être obtenus en réglant le retour d'information du codeur RSC à un polynôme primitif, parce que le polynôme primitif génère des séquences de longueur maximale qui ajoute un caractère aléatoire au code turbo.

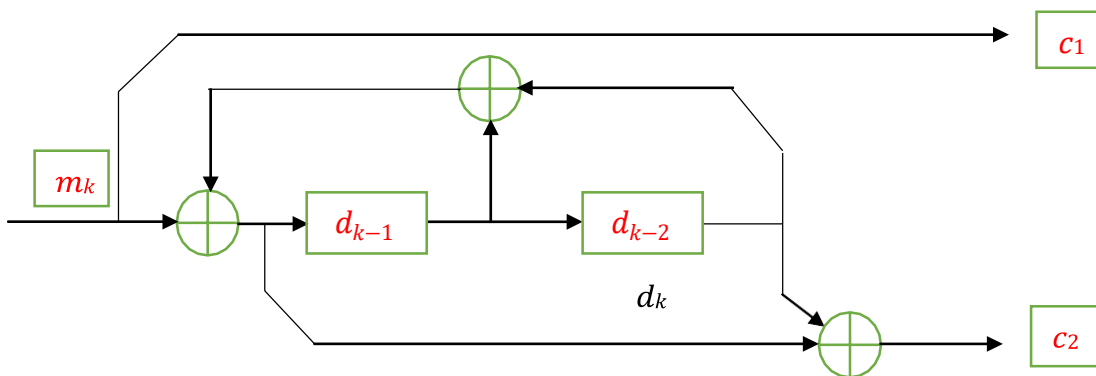


Figure 1-3 : Code Convolutif Récursif Systématique RSC[1, 5/7].

L'intérêt à utiliser la récursivité est qu'en pratique, les performances obtenues à faible rapport signal à bruit sont meilleures que celles obtenues avec un code non récursif de taille identique. C'est à partir de ce code convolutif récursif qu'ont été inventés les turbo-codes basés sur les mêmes considérations [1].

1.7.1 Exemple :

La figure 1-3 représente le schéma d'implémentation du code Convolutif RSC donc :

$$G(D) = [1, (1 + D^2)/(1 + D + D^2)] = [1, 5/7] \quad (1.5)$$

Un code convolutif est dit récursif si la séquence passant dans les registres à décalages est « alimentée » par le contenu de ces registres.

Supposons que le codeur reçoive le message $m = [1011]$, les registres étant initialement tous deux à 0.

La séquence codée devient (Figure1-3) :

$C = [11\ 01\ 10\ 10]$, et les registres seront finalement à l'état 11.

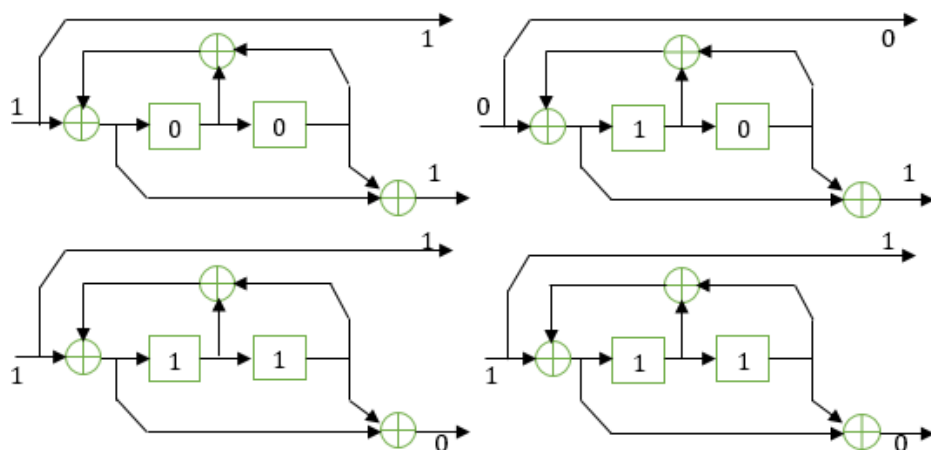


Figure 1-4: Exemple de codage d'une séquence par le codeur de la Figure 1-3.

1.8. Algorithme de Viterbi:

L'algorithme de Viterbi a été présenté en 1967 par A. J. Viterbi comme décodeur des codes convolutif [3] il fournit un moyen efficace d'estimer la séquence présente en entrée selon le critère du maximum de vraisemblance [5]. Le principe de cet algorithme consiste à estimer les transitions, qui se sont produites dans la mémoire du codeur convolutif pendant le codage. L'estimation se base sur le maximum de vraisemblance (Maximum Likelihood), fonction qui permet d'identifier le message globalement le plus probable.

L'importante contribution apportée par l'algorithme de Viterbi est la possibilité

Chapitre 1 les turbocodes

d'exploiter tout le potentiel de correction d'erreurs, mis à disposition par le code, sans devoir contrôler individuellement chacun des chemins possibles. En effet, la recherche du chemin globalement le plus probable par analyse des chemins possibles devient rapidement impraticable avec l'augmentation du nombre des symboles du message [4].

1.9. Méthode de 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 à haut débit, qui nécessitent une correction d'erreurs plus précise.

Le décodage MAP est basé sur le calcul de la probabilité a posteriori, qui est la probabilité qu'une certaine séquence de données a été envoyée en fonction de la séquence de données 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 du 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-communications [6].

1.9.1 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 .

Chapitre 1 les turbocodes

$$\text{LLR}(m_k) = \log \frac{\text{pr}(m_k=1 | R_1^N)}{\text{pr}(m_k=0 | R_1^N)} \quad (1.6)$$

Avec :

$$m_k \in \{0,1\} \forall k \quad (1.7)$$

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

L'algorithme MAP calcule le rapport de vraisemblance LLR :

$$\text{LLR}(k) = \log \frac{\text{prob } 1(i)}{\text{prob } 0(i)} \quad (1.8)$$

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

$$\begin{aligned} \alpha_k(s) &= \text{pr}(s_k = s | R_1^k) \\ \beta_k(s) &= \text{pr}(R_{k+1}^N | s_k = s) \\ \gamma_j(s', s) &= \text{pr}(m_k = j, s_k = s, R_k | s_{k-1} = s') \end{aligned} \quad (1.9)$$

Le LLR peut maintenant être décrit en termes de :

$$\text{LLR}(k) = \log \frac{\sum_{s', s/1} \alpha_{s'}(k-1) \gamma_{s', s}(k) \beta_s(k)}{\sum_{s', s/0} \alpha_{s'}(k-1) \gamma_{s', s}(k) \beta_s(k)} \quad (1.10)$$

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

$$\begin{aligned} \alpha_s(k) &= \sum_{s'} \alpha_{s'}(k-1) \gamma_{s', s}(k) \\ \beta_s(k) &= \sum_{s'} \alpha_{s', s}(k-1) \beta_s(k-1) \end{aligned} \quad (1.11)$$

Pour la mise en œuvre de l'algorithme MAP, il est nécessaire d'initialiser les valeurs de

Chapitre 1 les turbocodes

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

$$\alpha_0 = \begin{cases} 1 & \text{si } s=0 \\ 0 & \text{autrement} \end{cases} \quad (1.12)$$
$$\beta_N = \begin{cases} 1 & \text{si } s=0 \\ 0 & \text{autrement} \end{cases}$$

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

$$\beta_N(s) = \frac{1}{M} \quad (1.13)$$

Où M est le nombre total des états du treillis[1][6].

1.10. Les turbo-codes :

1.10.1 Introduction:

Les turbo-codes [7] sont une classe de codes correcteurs d'erreurs qui ont été développés dans les années 1990. Ils sont utilisés pour améliorer la qualité de transmission des données dans les systèmes de communication numérique, en particulier dans les environnements de transmission bruités.

Les turbocodes sont basés sur la technique de turbo codage, qui utilise deux codes convolutifs en cascade avec une technique de concaténation et d'itération pour améliorer la performance de décodage. Les turbocodes sont connus pour leur capacité à atteindre des performances de décodage proches des limites théoriques de Shannon pour les canaux de transmission bruités.

Le principe de base des turbocodes est de coder les données à transmettre à l'aide de deux codes convolutifs différents, puis de concaténer les bits de sortie de ces deux codes pour former un seul flux de données codées. Ce flux est ensuite transmis sur le canal de transmission. À la réception, un décodage itératif est effectué, qui utilise l'algorithme de décodage de Viterbi pour déterminer la séquence de bits la plus probable. Ce décodage est effectué à plusieurs reprises en utilisant les informations de la sortie de décodeur précédent et en prenant en compte les erreurs de transmission. Les résultats de décodage sont combinés pour produire une estimation plus précise de la séquence de bits originale.

Les turbocodes sont largement utilisés dans les normes de communication numérique

Chapitre 1 les turbocodes

telles que le 3G, le 4G, le WiMAX et le DVB-S2. Ils ont également été proposés pour les communications spatiales, y compris les communications par satellite [1].

1.11. Le turbo codage des turbocodes convolutifs :

1.11.1 Introduction :

Le turbo codage est une technique de codage de canal qui a été introduite dans les années 1990. Elle est basée sur l'utilisation de deux codes convolutifs en cascade, avec une technique de concaténation et d'itération. Les turbocodes convolutifs ont été l'un des premiers codes utilisés dans cette technique, et ils ont montré des performances supérieures à celles des codes de blocs classiques.

1.11.2 Le principe de base du turbo codage :

Le principe de base du turbo codage est de coder les données à transmettre à l'aide de deux codes convolutifs différents, puis de concaténer les bits de sortie de ces deux codes pour former un seul flux de données codées. Ce flux est ensuite transmis sur le canal de transmission.

À la réception, un décodage itératif est effectué, qui utilise l'algorithme de décodage de Viterbi pour déterminer la séquence de bits la plus probable. Ce décodage est effectué à plusieurs reprises en utilisant les informations de la sortie de décodeur précédent et en prenant en compte les erreurs de transmission. Les résultats de décodage sont combinés pour produire une estimation plus précise de la séquence de bits originale.

Les turbocodes convolutifs sont particulièrement efficaces dans les environnements de transmission bruités, tels que les communications sans fil et les transmissions par satellite. Ils ont été largement utilisés dans les normes de communication numérique telles que le 3G, le 4G et le WiMAX [1].

1.11.3 Le schéma de turbo codage :

Le schéma de turbo codage est basé sur deux codes convolutifs en cascade, avec une technique de concaténation et d'itération. voici le schéma de base des turbocodes convolutifs :

Données à coder → Encodeur 1 → entre relation → Encodeur 2 → Canal de transmission

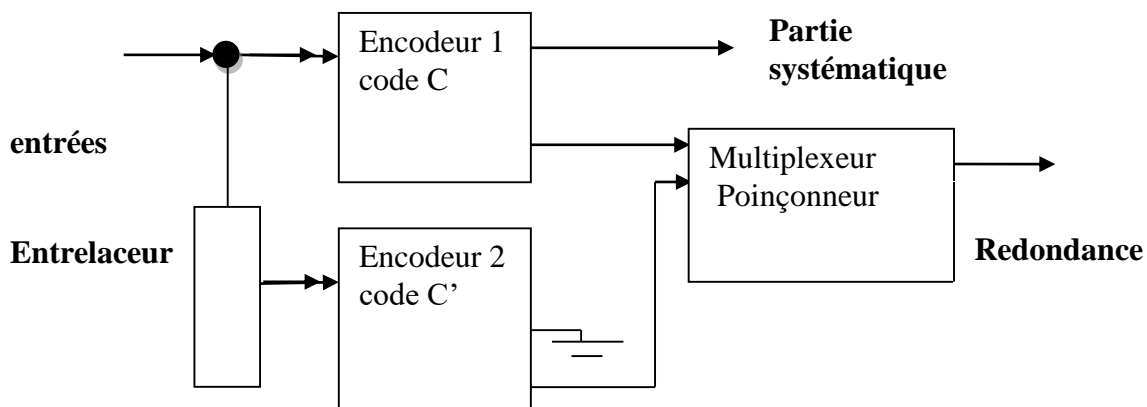


Figure 1-5 : Codeur turbo par composition parallèle et entrelacement de deux codeurs RSC.

Le processus de codage commence par les données à coder, qui sont introduites dans le premier encodeur convolutif. Les bits de sortie de cet encodeur sont ensuite envoyés dans un interleaver qui réorganise les bits pour minimiser l'impact des erreurs de transmission consécutives. Les bits réorganisés sont ensuite introduits dans le deuxième encodeur convolutif.

Le flux de bits de sortie des deux encodeurs est ensuite concaténé et transmis sur le canal de transmission. À la réception, le flux de bits est décodé itérativement en utilisant l'algorithme de décodage de Viterbi, en prenant en compte les informations de décodage précédentes et en utilisant un système de rétroaction pour améliorer la précision de décodage.

Le processus de décodage est itéré plusieurs fois jusqu'à ce qu'une estimation précise de la séquence de bits originale soit obtenue.

Il convient de noter que les détails précis du schéma de turbo codage peuvent varier en fonction des normes de communication numérique spécifiques ou des implémentations particulières [8].

1.12. Le turbo décodage des turbocodes convolutifs :

1.12.1 Introduction :

Le turbo décodage des turbocodes convolutifs est une méthode avancée de décodage utilisée pour les turbocodes convolutifs, qui sont des codes convolutifs à plusieurs composants. Les turbocodes convolutifs sont des codes correcteurs d'erreurs qui utilisent plusieurs composants, chaque composant étant un code convolutif différent.

1.12.2 Le principe de base du turbo décodage :

Le principe de base du turbo décodage est de réutiliser l'information redondante dans les différentes composantes du turbocode pour améliorer la correction d'erreurs. Le codage turbo est basé sur l'utilisation de deux codes convolutifs identiques, qui sont utilisés pour encoder la même séquence de données. Les deux codes convolutifs sont ensuite entrelacés pour produire la séquence de données envoyée.

Lorsque la séquence de données est reçue, elle est décodée en utilisant un algorithme de décodage itératif. L'algorithme de décodage itératif utilise des informations de la composante de code convolutif qui n'a pas été utilisée dans l'encodage pour améliorer la correction d'erreurs.

Le processus de décodage itératif consiste en plusieurs itérations, où la séquence de données est décodée plusieurs fois en utilisant les informations de la composante de code convolutif qui n'a pas été utilisée dans l'encodage. Les informations obtenues lors d'une itération sont utilisées pour améliorer le décodage lors de l'itération suivante.

L'algorithme de décodage itératif est souvent basé sur l'algorithme de décodage MAP, qui utilise une approche probabiliste pour estimer la séquence de données la plus probable. Les informations de la composante de code convolutif qui n'a pas été utilisée dans l'encodage sont utilisées pour améliorer la précision de l'estimation de la séquence de données [8].

1.12.3 le schéma de base du décodage des turbocodes convolutifs :

le processus de décodage des turbocodes convolutifs est basé sur l'utilisation d'un décodeur itératif, également appelé décodage de type « turbo ». voici le schéma de base du décodage des turbocodes convolutifs

Canal de transmission → Décodeur 1 → entrelaceur → Décodeur 2 → LLR

Chapitre 2

Critères d'arrêt pour le turbo décodage

Introduction :

Le décodage turbo se fait d'une manière itérative. Le nombre d'itérations est limité puisque après un certain nombre, le décodage sera saturé et les performances ne s'améliorent plus. Afin de contrôler et arrêter ce processus et gagner par la suite plus de temps de traitement, une condition appelée critère d'arrêt devrait être vérifiée après chaque itération. Dans la littérature, nous trouvons plusieurs critères d'arrêt [9]. Nous allons dans la suite présenter quelques techniques.

De nombreuses méthodes d'arrêt du décodage itératif ont été discutées dans la littérature. Le critère du rapport de changement de signe (SCR) a été proposé dans [10]. Il utilise les informations extrinsèques $Le_2^{(i)}(k)$ et $Le_2^{(i-1)}(k)$ des itérations i et $i - 1$. En comptant le nombre de changements de signes $C(i)$ de son information extrinsèque calculée par le second décodeur entre les itérations i et $i - 1$, il sera possible d'arrêter les itérations si :

$$\frac{C(i)}{N} \leq \tau \quad (2.1)$$

Avec

$$0.0005 N \leq \tau \leq 0.03 N \quad (2.2)$$

Le même article propose un deuxième critère appelé HardDecision-Aided (HDA). Ce critère arrête le décodage turbo lorsque la trame décodée de l'itération i et celle de l'itération $i - 1$ sont identiques.

La référence [11] propose d'utiliser un code de redondance cyclique (CRC) pour vérifier si la trame a été corrigée. Le CRC est placé avant le turbocodeur du côté des émissions. Une vérification CRC est appliquée après chaque itération. Cette technique ajoute une redondance qui augmente le débit à la sortie du turbo encodeur. Un réseau de neurones est utilisé dans [12] pour arrêter le décodage turbo. Une autre règle appelée Sign Difference Ratio (SDR) a été proposée dans [13]. Pour cette règle, on utilise les signes de l'information a priori $La_m^{(i)}(u(k))$ et de l'information extrinsèque $Le_m^{(i)}(u(k))$ ($m=1, 2$)

Chapitre 2 Critères d'arrêt pour le turbo décodage

du même décodeur. En comptant le nombre de fois D_{ji} que leurs signes diffèrent à l'itération i , il est recommandé d'arrêter les itérations lorsque:

$$D_{ji} \leq pN \quad (2.3)$$

Où, p est un seuil qui représente la différence de signe. La Improved Hard Decision Aiding Rule (IHDA) [14] utilise, à l'itération i , les bits décodés des décodeurs SISO1 et SISO2. Si les trames décodées sont les mêmes, le décodage turbo peut être arrêté. On peut aussi citer les contributions [15-27]. Dans ce mémoire, nous nous sommes intéressés au critère de l'entropie croisée CE (Cross-Entropy) [28-29]. Il utilise la différence des informations extrinsèques ΔL_e entre les itérations i et $i-1$ (Figure 2.1). Le chapitre 3 est consacré à cette règle d'arrêt Turbo.

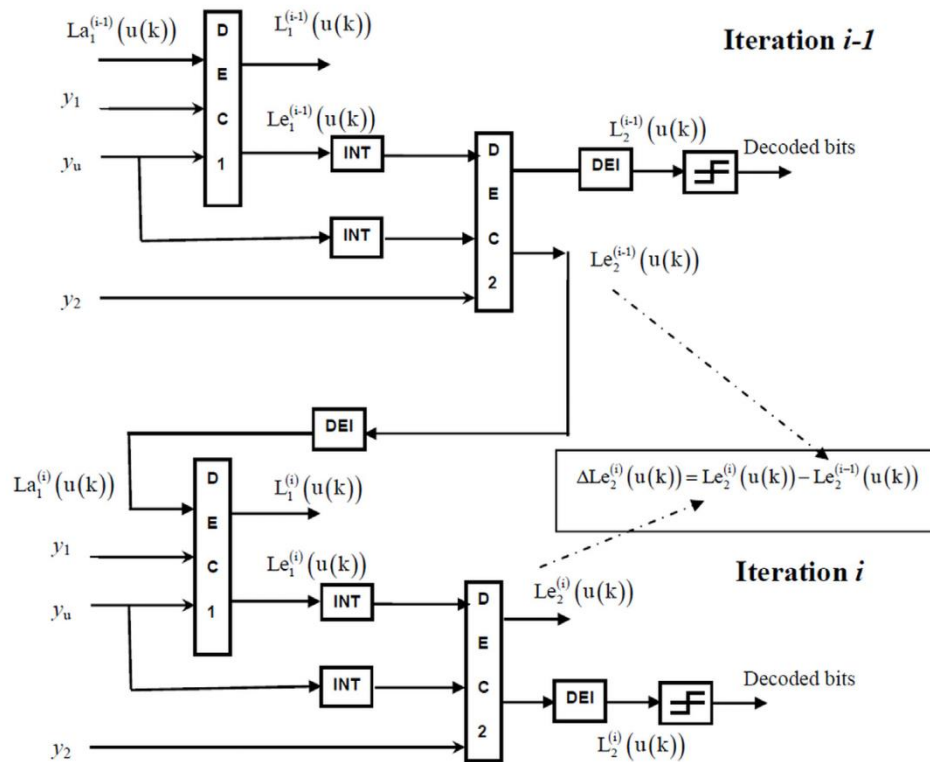


Figure 2-1 : Itérations successives de décodage turbo

2.1 Décodeur turbo :

Nous avons défini l'information extrinsèque et l'algorithme de décodage (Chapitre 1). Nous pouvons maintenant préciser plus en détail le processus de décodage des codes turbo. La

Chapitre 2 Critères d'arrêt pour le turbo décodage

figure2-2 illustre le schéma de principe du décodage itératif dans un décodeur turbo. Notons que sur cette figure apparaît ce qu'on appelle un entrelaceur / désentrelaceur. L'entrelacement permet de décorréler les données traitées. De plus, il casse les paquets d'erreurs en séparant leurs positions.

Dans la suite, nous appelons DEC1 le décodeur avec l'entrée de parité 1 et DEC2 le décodeur avec l'entrée de parité 2. Chaque décodeur génère, pour chaque symbole, le log-rapport de vraisemblance LLR (Log Likelihood Ratio) qui représente le Log-rapport de la probabilité du 1 sur la probabilité du 0.. Cette procédure se répète jusqu'à atteindre une condition prédéterminée qui représente le critère d'arrêt de la procédure de décodage.

A la $i^{\text{ème}}$ itération, le DEC1 reçoit l'information systématique r_k^s (y) dans la figure 2.1), l'information de parité r_k^{1p} et l'information a priori $La_k^1 = Lex_k^2(i-1)$, pour générer les informations extrinsèque $Lex_k^1(i)$ qui seront entrelacées avant d'être transmises au DEC2 comme entrée a priori $La_k^2(i)$.

A la première itération, nous n'avons pas besoin d'utiliser l'information a priori La_k^1 au niveau du premier décodeur. En effet, comme nous l'avons mentionné, les bits sont équiprobables et donc l'information de la probabilité a priori est nulle à la première itération.

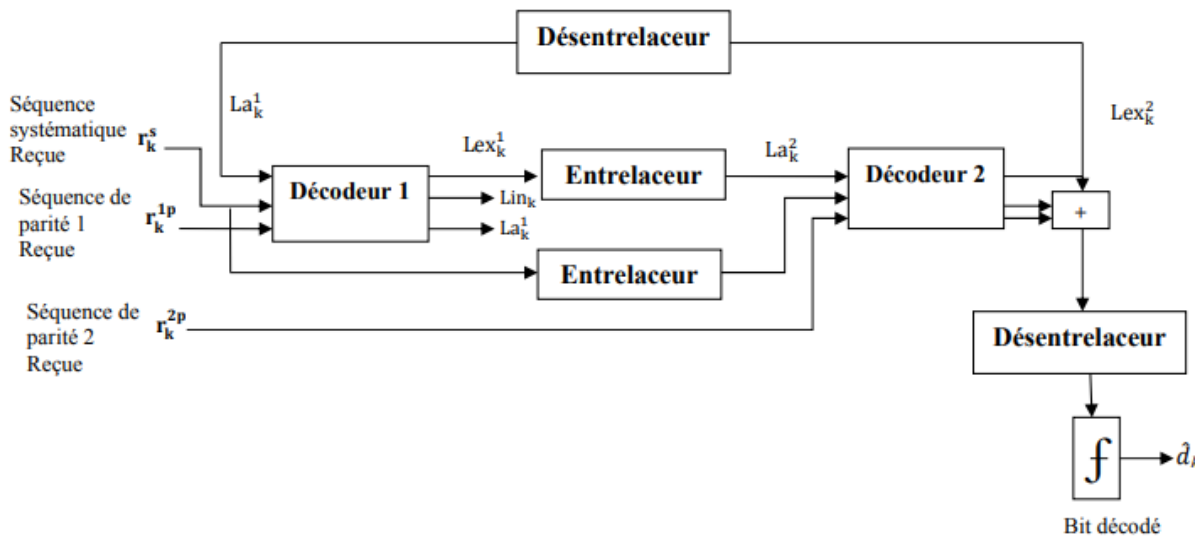


Figure 2-2 : Schéma de principe d'un décodeur turbo

Après décodage des symboles par le premier décodeur MAP, la sortie à l'instant k est égale à :

$$Dj_i \leq LLR_k^1 = Lin_k + La_k^1(i) + Lex_k^1(i) \quad (2.4)$$

Chapitre 2 Critères d'arrêt pour le turbo décodage

Lin_k n'est que le LLR du symbole systématique fourni par le canal ($L_c r_k, L_c = 2/\sigma^2, \sigma^2$ est la variance du bruit).

Cette donnée est ensuite entrelacée afin de correspondre aux symboles entrelacés par le deuxième codeur. Le deuxième décodeur MAP reçoit donc les informations suivantes : $Lex_k^1(i)$ qui représente l'information extrinsèque entrelacée, l'information systématique entrelacée r_k^S et la parité du deuxième codeur r_k^{2P} pour générer l'information extrinsèque $Lex_k^2(i)$. Cette dernière est ensuite désentrelacée à son tour, puis retournée, via la boucle de retour vers le DEC1. Le but de l'entrelacement est de disperser les erreurs arrivées en rafale qui sont difficilement décodables par les décodeurs MAP. Plus la taille de l'entrelaceur est grande plus les erreurs en bloc sont dispersées.

Après avoir acquis une information sur le bit qui va être décodé, le deuxième décodeur MAP va pouvoir corriger un nombre d'erreurs que le premier décodeur n'a pas pu faire. La sortie du deuxième décodeur est égal à :

$$Dji \leq LLR_k^2(i) = Lin_k + La_k^2(i) + Lex_k^2(i) \quad (2.5)$$

Après un nombre fixe d'itérations, la décision finale sur les bits du message est basée sur les signes des LLR aux sorties des décodeurs. Il est important de noter que seules les valeurs extrinsèques sont mises à jour à chaque itération, les valeurs du canal correspondant aux bits du message restent toujours fixes.

2.2 Critère d'arrêt SCR :

Soit $C^2(i)$ le nombre de changements de signe entre les éléments de la séquence de l'information extrinsèque $Le^2(i-1)$ à la $(i-1)^{ème}$ itération et les éléments de la séquence l'information extrinsèque $Le^2(i)$ de l'itération i du deuxième décodeur (avec $i > 1$).

Il s'agit de vérifier le rapport $\frac{C^2(i)}{N}$. N est la longueur du paquet d'information avant codage. Lorsque ce rapport est inférieur à $\leq (0.005 \sim 0.03)$, le décodage peut-être arrêté sans presque aucune dégradation des performances [10].

On peut encore modifier ce critère en évaluant le nombre de changements de signe entre la séquence de l'information extrinsèque $Le^1(i-1)$ et $Le^1(i)$ du premier décodeur. Par conséquence, nous devons spécifier le rapport $\frac{C^n(i)}{N}$, ($n = 1, 2$) qui doit être $\leq (0.005 \sim 0.03)$ pour arrêter le décodage.

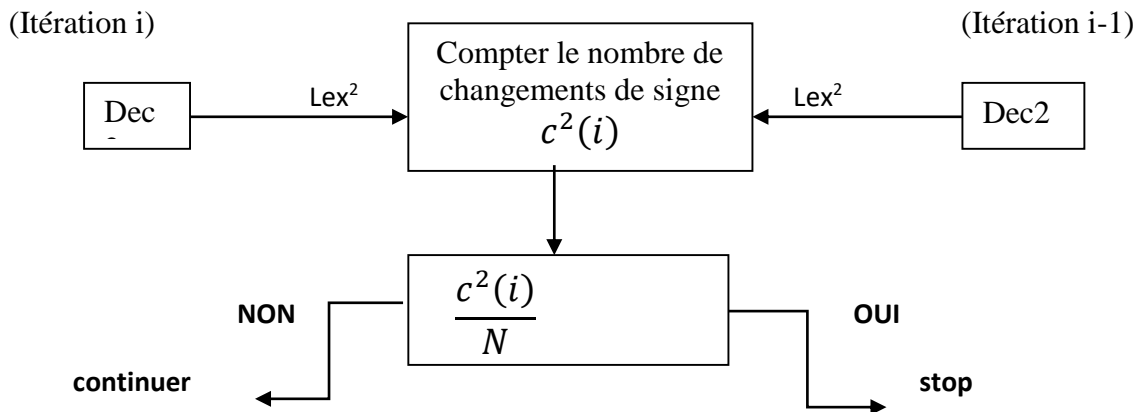


Figure 2-3 : Critère d'arrêt Sign change Ratio

2.3 Critère d'arrêt SDR :

Le critère d'arrêt Sign Difference Ratio (SDR) [13] est une variante du SCR. Dans ce cas on compte combien de fois D_{ji} les signes de l'information intrinsèque (a priori) et de l'information extrinsèque (a posteriori) du même décodeur diffèrent dans l'itération ' i '. Le processus turbo est arrêté lorsque :

$$D_{ji} \leq PN$$

Où " P " est un seuil représentant le rapport de différence de signe SDR [13] et :

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

D_{ji} est aussi le nombre de différences de signe entre les informations extrinsèques des deux décodeurs [13].

Le SDR atteint les performances similaires de la SCR en termes de taux d'erreur binaire BER (Bit Error Rate), de taux d'erreur par paquet FER (Frame Error Rate) et de nombre moyen d'itérations, mais nécessite moins de complexité. En moyenne, les méthodes SCR et SDR nécessitent une itération de plus que le critère GENIE idéal (Section 2.7).

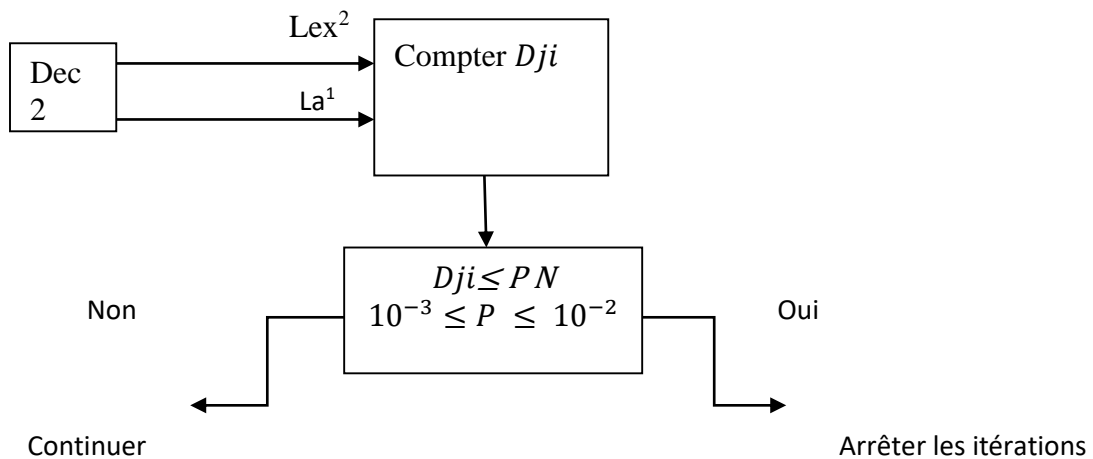


Figure 2-4 : Critère d'arrêt SDR Sign Difference Ratio

2.4 Critère d'arrêt HDA :

Le critère d'arrêt HDA [10] est une méthode efficace dérivée du concept de l'entropie croisée CE. A la $(i - 1)^{\text{ème}}$ 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. Au cours de la $i^{\text{ème}}$ itération, la séquence d'informations décodée $A^2(i)$ est comparée à la séquence résultante de l'itération précédente $A^2(i - 1)$. Si les deux séquences décodées sont identiques, le décodage peut-être arrêté [10].

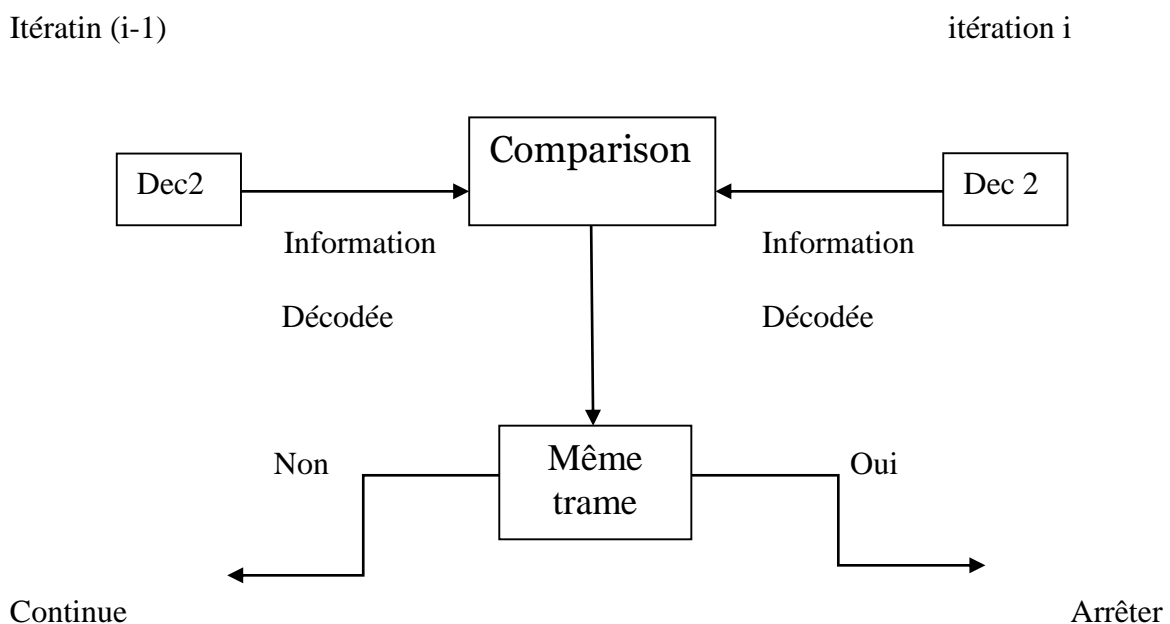


Figure 2-5 : Critère d'arrêt HDA Hard Decision Aided

Chapitre 2 Critères d'arrêt pour le turbo décodage

2.5 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[14]. Son principe est de comparer les décisions dures basées sur $\left(\left(\frac{2}{\sigma^2}\right) y_d^2(\mathbf{k}) + L_{e_1}^i(d(\mathbf{k}))\right)$ avec les décisions basées sur le LLR $L_2^i(d(\mathbf{k}))$ 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 $\left(\left(\frac{2}{\sigma^2}\right) y_d^2(k) + L_{e_1}^i(d(k))\right)$.
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 $i + 1$ [14].

2.6 Critère d'arrêt GENIE :

La règle d'arrêt GENIE est une référence pour les performances des critères d'arrêt existants. Cette règle d'arrêt est une règle théorique qui peut être utilisée comme un repère de comparaison des performances avec les autres règles. Pour cette règle, le critère GENIE compare le paquet décodé avec le paquet émis supposé connu par le récepteur. Si les deux séquences sont identiques, le turbo décodage peut être arrêté.

Par conséquent, GENIE est considéré comme un critère idéal théorique qui peut être utilisé comme un repère de comparaison[12] .

2.7 Critère CRC :

C'est une règle d'arrêt basée sur la détection des trames erronées à l'aide d'un code externe de redondance cyclique (CRC) appliqué aux bits décodés (Turbo) après une décision dure [11]. Un code de détection d'erreur séparé, tel qu'un code CRC, peut être combiné en série en tant que code externe avec un turbocode interne pour détecter les séquences erronées à la sortie du turbo décodeur. La condition d'arrêt avec cette règle est satisfaite chaque fois que le syndrome de code CRC est égal à zéro. Cette technique CRC fait partie de la norme sans fil 3G-WCDMA (Wideband Code Division Multiple Access)[1] .

2.8 Critère de la moyenne des LLR :

La figure 2-5 affiche les valeurs LLR typiques pour dix trames de bits de données à la sortie d'un décodeur après dix itérations. 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'arrêter 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 [30] 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 de trois trames typiques en fonction des itérations sont montrés à la Figure 2-6 et 2-7. Sur la base de ce constat, Zhai and Fair [30] ont proposé le critère d'arrêt de la moyenne des LLRs.

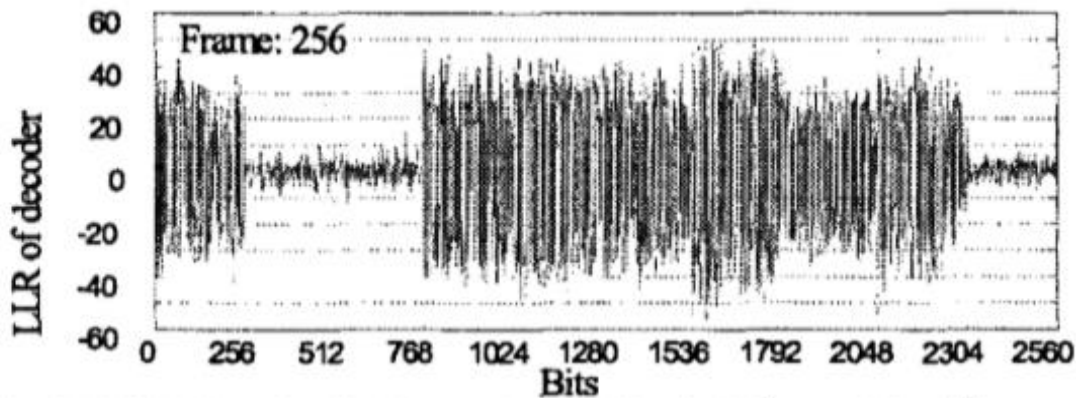


Figure 2-6 : Valeurs des LLR

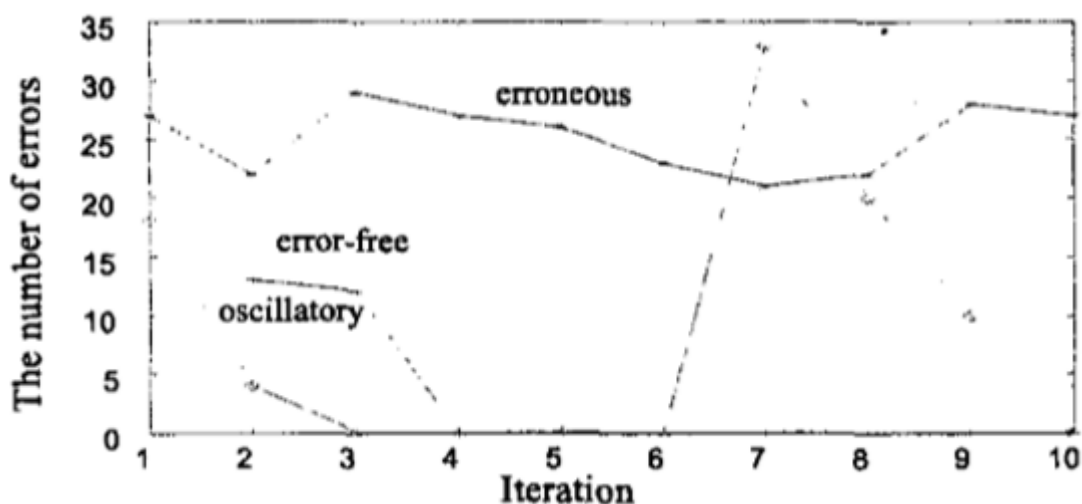


Figure 2-7: Nombre d'erreurs avec les itérations pour 3 paquets.

Chapitre 2 Critères d'arrêt pour le turbo décodage

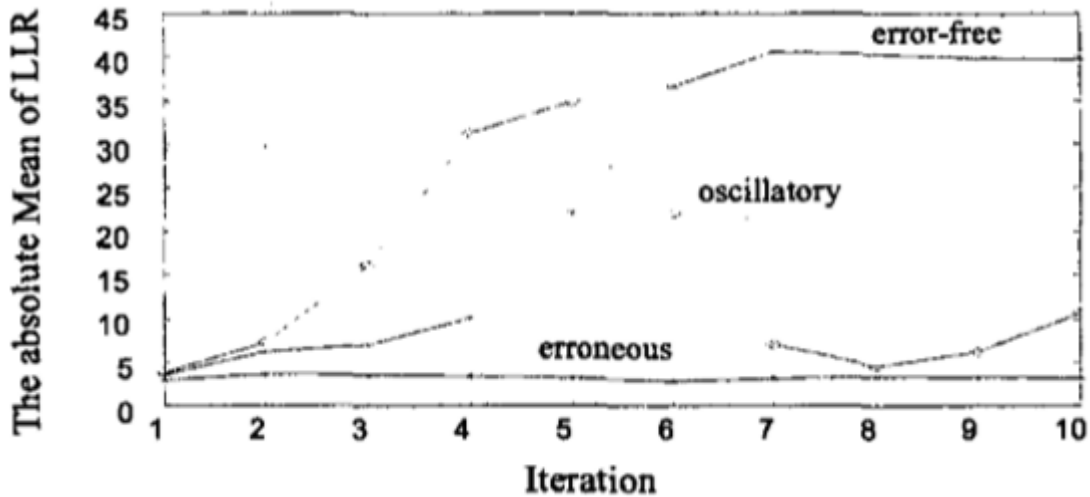


Figure 2-8 : La moyenne absolue des valeurs LLR en fonction du nombre des itérations .

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[30].

Chapitre 3

Critère de l'entropie croisée

Introduction :

La notion d'entropie croisée (cross-entropy en anglais) est couramment utilisée dans le domaine de l'apprentissage automatique pour mesurer la divergence entre deux distributions de probabilités.

Plus précisément, l'entropie croisée entre deux distributions de probabilités p et q est définie comme suit [28]:

$$H(P, Q) = -\sum_x P(x) \log(Q(x))$$

où x représente une variable aléatoire, $p(x)$ est la probabilité associée à cette variable dans la distribution P , et $Q(x)$ est la probabilité associée à cette variable dans la distribution Q .

Intuitivement, l'entropie croisée mesure le nombre moyen de bits nécessaires pour encoder les données générées à partir de la distribution p lorsque nous utilisons une distribution de probabilité q pour les prédire. Si les distributions p et q sont identiques, l'entropie croisée est minimale, sinon elle est plus grande [28].

En apprentissage automatique, l'entropie croisée est souvent utilisée comme fonction de coût pour entraîner des modèles de classification. L'objectif est de minimiser l'entropie croisée entre les prédictions du modèle et les étiquettes réelles des données d'entraînement [28].

3. Propriétés de convergence du décodage itératif via l'Entropie Croisée :

3.1. introduction

Le décodage itératif via l'entropie croisée est une technique de décodage utilisée dans les systèmes de communication numériques, en particulier dans les systèmes de communication par canal bruité. Il s'agit d'un algorithme itératif qui est utilisé pour décoder les données transmises en utilisant la probabilité de distribution des erreurs dans le canal de transmission [28].

3.2. Les propriétés de convergence de cet algorithme sont les suivantes

3.2.1. Convergence

Le décodage itératif via l'entropie croisée converge globalement vers la solution optimale, c'est-à-dire la solution qui minimise l'entropie croisée entre les données transmises et les données reçues, même si la condition initiale est éloignée de la solution optimale.

3.2.3. Vitesse de convergence

La vitesse de convergence de l'algorithme dépend de plusieurs facteurs, tels que la qualité du canal de transmission et le nombre d'itérations. En général, plus le canal de transmission est bruité et plus le nombre d'itérations est élevé, plus la vitesse de convergence est lente.

Complexité de calcul : La complexité de calcul de l'algorithme dépend également de plusieurs facteurs, tels que la taille de la matrice de codage, la qualité du canal de transmission et le nombre d'itérations. En général, plus la taille de la matrice de codage est grande, plus la qualité du canal de transmission est faible et plus le nombre d'itérations est élevé, plus la complexité de calcul est élevée.

On peut montrer que l'entropie croisée est un critère utile pour le décodage itératif [28]. Nous montrerons ici que l'entropie croisée est un critère d'arrêt utile pour le décodage itératif. L'entropie croisée de deux distributions $P(u)$ et $Q(u)$ est définie comme suit :

$$CE = E \left\{ \log \frac{P(\hat{u})}{Q(\hat{u})} \right\} \quad (3.1)$$

C'est une mesure de la proximité de deux distributions. Ici, E désigne l'opérateur d'espérance mathématique sur le $\log \frac{P(\hat{u})}{Q(\hat{u})}$. En supposant l'indépendance statistique, nous obtenons :

$$\log \frac{P(\hat{u})}{Q(\hat{u})} = \sum_k \log \frac{P(\hat{u}_k)}{Q(\hat{u}_k)} \quad (3.2)$$

Maintenant, regardons deux itérations suivantes ($i - 1$) et (i). prenons par exemple le décodage d'un turbo code produit. Une itération consiste en le décodage dans la direction « horizontale » et la direction « verticale ». Nous définissons [28] :

$$L_Q^{(i)}(\hat{u}_k) = L_c \cdot y_k + L_e^{-(i-1)}(\hat{u}_k) + L_e^{(i)}(\hat{u}_k) \quad (3.3)$$

$$L_P^{(i)}(\hat{u}_k) = L_c \cdot y_k + L_e^{(i)}(\hat{u}_k) + L_e^{-(i)}(\hat{u}_k)$$

$L_P^{(i)}(\hat{u}_k)$ et $L_Q^{(i)}(\hat{u}_k)$ sont les LLR calculés dans les directions horizontale et verticale.

La différence des sorties douces est égale à :

$$L_P^{(i)}(\hat{u}_k) - L_Q^{(i)}(\hat{u}_k) = L_e^{-(i)}(\hat{u}_k) - L_e^{-(i-1)}(\hat{u}_k) = \Delta L_e^{-(i)}(\hat{u}_k) \quad (3.4)$$

En utilisant la propriété:

Chapitre 3 Critère de l'entropie croisée

$$\hat{u}_k^{(i)} = \text{sign}(L_P^{(i)}(\hat{u}_k)) \quad (3.5)$$

Et :

$$|L_P^{(i)}(\hat{u}_k)| = \hat{u}_k^{(i)} \cdot L_P^{(i)}(\hat{u}_k) \quad (3.6)$$

On peut montrer que [28] :

$$\begin{aligned} & E_{p\left\{\log\frac{P(\hat{u}_k)}{Q(\hat{u}_k)}\right\}} \quad (3.7) \\ & \approx -\hat{u}_k^{(i)} \Delta L_e^{- (i)}(\hat{u}_k) \frac{1}{1 + \exp\left(\left|L_P^{(i)}(\hat{u}_k)\right|\right)} + \log \frac{1 + \exp\left(-\left|L_Q^{(i)}(\hat{u}_k)\right|\right)}{1 + \exp\left(-\left|L_P^{(i)}(\hat{u}_k)\right|\right)} \end{aligned}$$

La dernière approximation est valable, lorsque les décisions ne changent plus, c'est-à-dire lorsque :

$$\text{Sign}(L_P^{(i)}(\hat{u}_k)) = \text{sign}(L_Q^{(i)}(\hat{u}_k)) = \hat{u}_k^{(i)}. \quad (3.8)$$

De plus, si les fiabilités sont suffisamment importantes, et en utilisant la propriété $\log(1+z) \approx z$, on aura :

$$\begin{aligned} \text{Ep}\left\{\log\frac{P(\hat{u}_k)}{Q(\hat{u}_k)}\right\} &= \exp\left(-\left|L_Q^{(i)}(\hat{u}_k)\right|\right) \cdot (1 - \exp(-\hat{u}_k^{(i)} \Delta L_e^{- (i)}(\hat{u}_k)))(1 + \\ & \hat{u}_k^{(i)} \Delta L_e^{- (i)}(\hat{u}_k)). \quad (3.9) \end{aligned}$$

Finalement, on peut montrer que [28] :

$$E_{p\left\{\log\frac{P(\hat{u}_k)}{Q(\hat{u}_k)}\right\}} \approx \sum_k \frac{\left|\Delta L_e^{- (i)}(\hat{u}_k)\right|^2}{\exp\left(\left|L_Q^{(i)}(\hat{u}_k)\right|\right)} \quad (3.10)$$

Bien sûr, l'hypothèse d'indépendance statistique entre les valeurs de vraisemblance n'est pas exactement vraie après quelques itérations. Néanmoins, on pourrait utiliser le critère [28]

$$T(i) = \sum_k \frac{\left|\Delta L_e^{- (i)}(\hat{u}_k)\right|^2}{\exp\left(\left|L_Q^{(i)}(\hat{u}_k)\right|\right)} < \text{seuil} \quad (3.11)$$

Comme un critère d'arrêt des itérations. C'est-à-dire, lorsque $T(i)$ est inférieur au seuil « threshold », alors, arrêter le turbo décodage.

Chapitre 3 Critère de l'entropie croisée

Les résultats de la simulation de [Hagenauer] ont montré que $T(i)$ chute d'un facteur de 10^{-2} à 10^{-4} une fois qu'aucune autre erreur ne sera corrigée et qu'une valeur seuil de $T(1) \cdot 10^{-3}$ est appropriée pour arrêter les itérations [28].

3.3.Principe de fonctionnement du critère de l'Entropie Croisée appliqué en turbo égalisation :

3.3.1.Introduction :

Le critère de l'Entropie Croisée utilise donc les LLR aux sorties des deux traitements Turbo pour fournir une information sur la fin du décodage. Soit le Turbo récepteur de la figure (3-1).

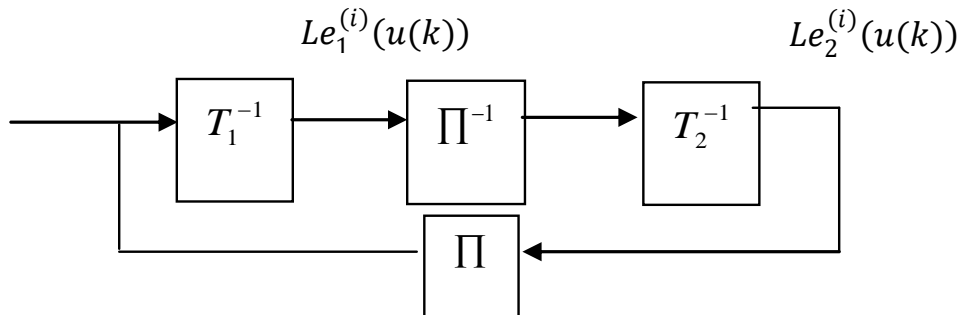


Figure 3-1. Turbo récepteur série

T_1^{-1} et T_2^{-1} sont les traitements inverses de l'émission. Π et Π^{-1} représentent respectivement l'entrelaceur et le désentrelaceur associé. Les données sont supposées entrelacées à l'émission. Soit $Le_1^{(i)}(u(k))$, et $Le_2^{(i)}(u(k))$ les informations extrinsèques des bits d'information $u(k)$ fournies par les deux traitements.

Pour des traitements en parallèle, la figure suivant montre les informations extrinsèques calculées par les deux traitements

L'Entropie Croisée de l'itération 'i' est donné selon [28] par :

$$CE(i) \approx \sum_n \frac{|\Delta Le_2^{(i)}(u(k))|^2}{e^{|L_1^{(i)}(u(k))|}} \quad (3.12)$$

Chapitre 3 Critère de l'entropie croisée

Où :

$$\Delta Le_2^{(i)}(u(k)) = Le_2^{(i)}(u(k)) - Le_2^{(i-1)}(u(k)) \quad (3.13)$$

Ceci est équivalent aussi à :

$$\Delta Le_2^{(i)}(u(k)) = L_2^{(i)}(u(k)) - L_1^{(i)}(u(k)) \quad (3.14)$$

$L_1^{(i)}(u(k))$ est le Log-rapport de vraisemblance LLR des bits d'information $u(k)$ calculés par le premier traitement T_1^{-1} .

C'est-à-dire, la différence entre les informations extrinsèques de deux itérations successives.

La trame est décidée correcte si :

$$CE(i) < \varepsilon \quad (3.15)$$

Avec [28]:

$$10^{-2}CE(1) \leq \varepsilon \leq 10^{-4}CE(1) \quad (3.16)$$

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

Pour un turbo récepteur d'une concaténation parallèle, prenons par exemple un turbo décodeur MAP. Il fournit 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.

On définit la différence des informations extrinsèques entre l'itération i et $i-1$ à la sortie du deuxième décodeur par [28] :

$$\Delta Le_2^{(i)}(u(k)) = Le_2^{(i)}(u(k)) - Le_2^{(i-1)}(u(k)) \quad (3.17)$$

La même quantité peut être calculée par [28]:

$$\Delta Le_2^{(i)}(u(k)) = L_2^{(i)}(u(k)) - L_1^{(i)}(u(k)) \quad (3.18)$$

Chapitre 3 Critère de l'entropie croisée

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

$$CE(i) \approx \sum_n \frac{|\Delta Le_2^{(i)}(u(k))|^2}{e^{|L_1^{(i)}(u(k))|}} \quad (3.19)$$

Le cadre est jugé correct si:

$$CE(i) < \varepsilon \quad (3.20)$$

Il est recommandé [28] que le seuil ε soit dans l'intervalle :

$$10^{-2}CE(1) \leq \varepsilon \leq 10^{-4}CE(1) . \quad (3.21)$$

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

Pour un turbo récepteur en série, 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 bit codés. Dans ce cas, le critère devient :

Pour chaque trame traitée, calculer l'Entropie Croisée des symboles $c(n)$ par [28]:

$$CE(i) = \sum_n \frac{|\Delta Le_2^{(i)}(c(n))|^2}{e^{|L_1^{(i)}(c(n))|}} \quad (3.22)$$

Où :

$$\Delta Le_2^{(i)}(c(k)) = Le_2^{(i)}(c(k)) - Le_2^{(i-1)}(c(k)) \quad (3.23)$$

La trame est décidée correcte si :

$$CE(i) < \varepsilon \quad (3.24)$$

Avec toujours [28]:

$$10^{-2}CE(1) \leq \varepsilon \leq 10^{-4}CE(1) \quad (3.25)$$

3.4 Conclusion:

L'entropie croisée est une mesure de la divergence entre deux distributions de probabilités P et Q . Elle est souvent utilisée en apprentissage automatique comme fonction de coût pour entraîner des modèles de classification. L'entropie croisée mesure le nombre moyen de bits nécessaires pour encoder les données générées à partir de la distribution p lorsque nous utilisons une distribution de probabilité q pour les prédire. L'objectif est de minimiser l'entropie croisée entre les prédictions du modèle et les étiquettes réelles des données d'entraînement. Si les distributions p et q sont identiques, l'entropie croisée est minimale, sinon elle est plus grande.

Chapitre 4

Résultats de simulation

4.1 Introduction :

L'objectif de ce chapitre est l'évaluation par simulation MATLAB des performances d'un Turbo décodeur MAP $[1, 5/7]_{\text{oct}}$ et l'application du critère d'arrêt de l'entropie croisée au turbo décodage MAP ainsi que l'évaluation de leur performances.

4.2 Performances d'un Turbo décodeur $[1, 5/7]_{\text{oct}}$

4.2.1 Conditions de simulation :

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

Rendement : $R = 1/3$

Nombre de paquets = 3000

RSB minimal = 0 dB

RSB maximal = 1.2 dB

Longueur des trames $L=1000$

Algorithme de décodage : MAP

4.2.2 Résultats de simulation :

Les performances du turbo décodeur sont largement déterminées par l'ensemble des paramètres suivants : le nombre de paquets (trames), la longueur de trame (nombre de bits transmis), le rendement (ici $1/3$), la longueur de contrainte et l'algorithme de décodage. Afin d'analyser les performances de ces paramètres, une suite de simulations a été réalisées afin d'évaluer le taux d'erreur binaire en fonction du rapport signal à bruit RSB (E_b/N_0).

La figure 4.1 montre les performances en termes de Taux d'Erreur Binaire TEB du turbo décodeur MAP qui utilise 7 itérations. Le turbo codeur utilisé est formé de deux codeur récursifs systématiques RSC de séquences génératrices $[1, 5/7]_{\text{oct}}$ en parallèle.

4.3. Performances du critère GENIE appliqué au turbo décodage :

4.3.1. Conditions de simulation :

Nombre de paquets = 3000

RSB minimal = 0 dB

RSB maximal = 1.2 dB

Longueur des trames $L=1000$

4.3.2 Résultats de simulation :

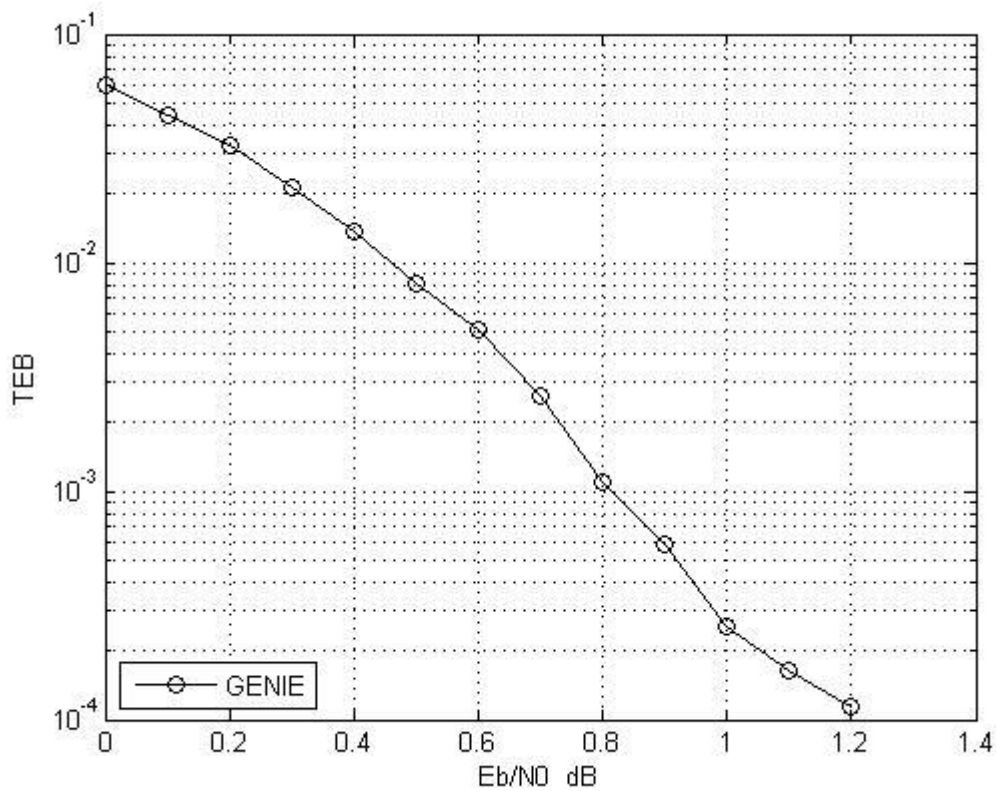


Figure 4.2 Performances du critère GENIE appliqué au turbo décodage

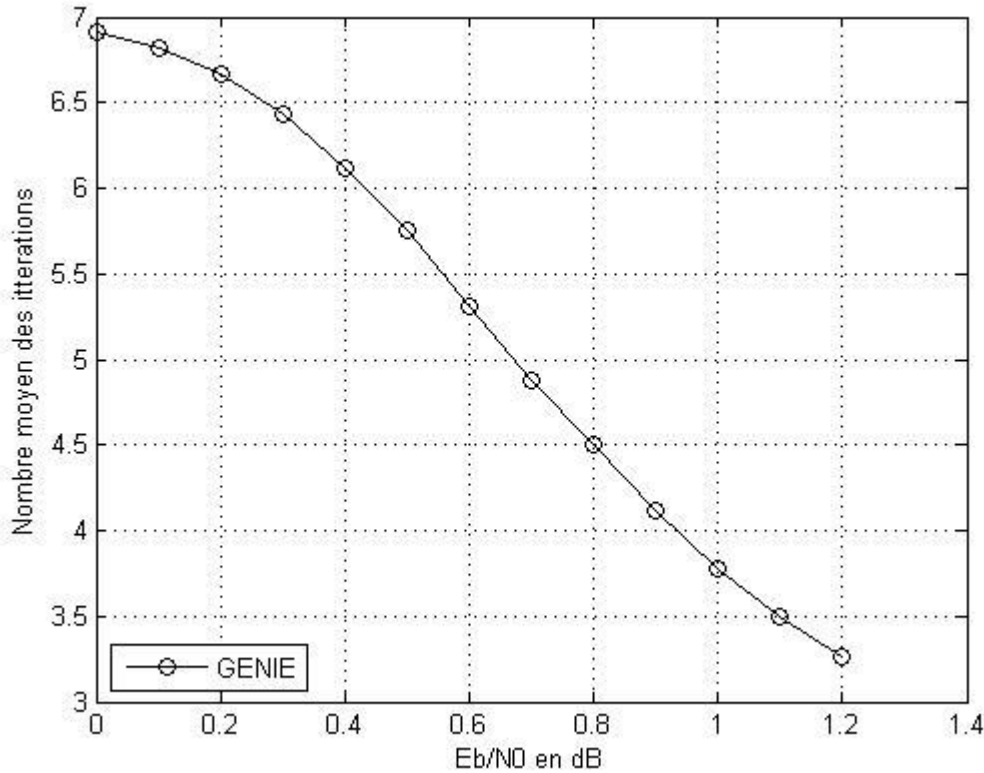


Figure 4.3 Nombre moyen des itérations du critère GENIE appliqué au turbo décodage

4.3.3 Interprétation des résultats :

La figure 4.2 montre le taux d'erreur binaire TEB d'un turbo décodeur $[1, 5/7]$ supervisé par le critère d'arrêt GENIE qui arrête le turbo décodage après l'obtention d'une trame correcte. Au RSB = 1 dB, le TEB = $2.7 \cdot 10^{-4}$. Au RSB = 1.2 dB, le nombre moyen des itérations est réduit de 7 jusqu'à 3.25 itérations. C'est-à-dire, à la moyenne 3.75 itérations/trame ($7 - 3.25 = 3.75$ itérations /trames) est éliminées, ce qui accélère la vitesse du décodage. Pour 1000 mille trames, on a éliminé $3750 \times 2 = 7500$ décodages MAP (Chaque itération est constituée de 2 décodage MAP et des entrelaceurs/désentrelaceurs).

4.4. Performances du critère de l'entropie croisée :

4.4.1. Condition de simulation :

Nombre de paquets = 3000

RSB minimal = 0 dB

RSB maximal = 1.2 dB

Longueur des trames $L=1000$.

Seuil du critère d'arrêt de l'entropie croisée : seuil = 10^{-3} CE(1).

Chapitre 4 Résultats de simulation

CE(1) est l'entropie croisée de la première itération. C'est-à-dire, on commence la vérification du critère à partir de la deuxième itération. On note que le seuil doit vérifier la condition $10^{-2} CE(1) < \text{seuil} < 10^{-4} CE(1)$.

4.4.2 Résultats de simulation :

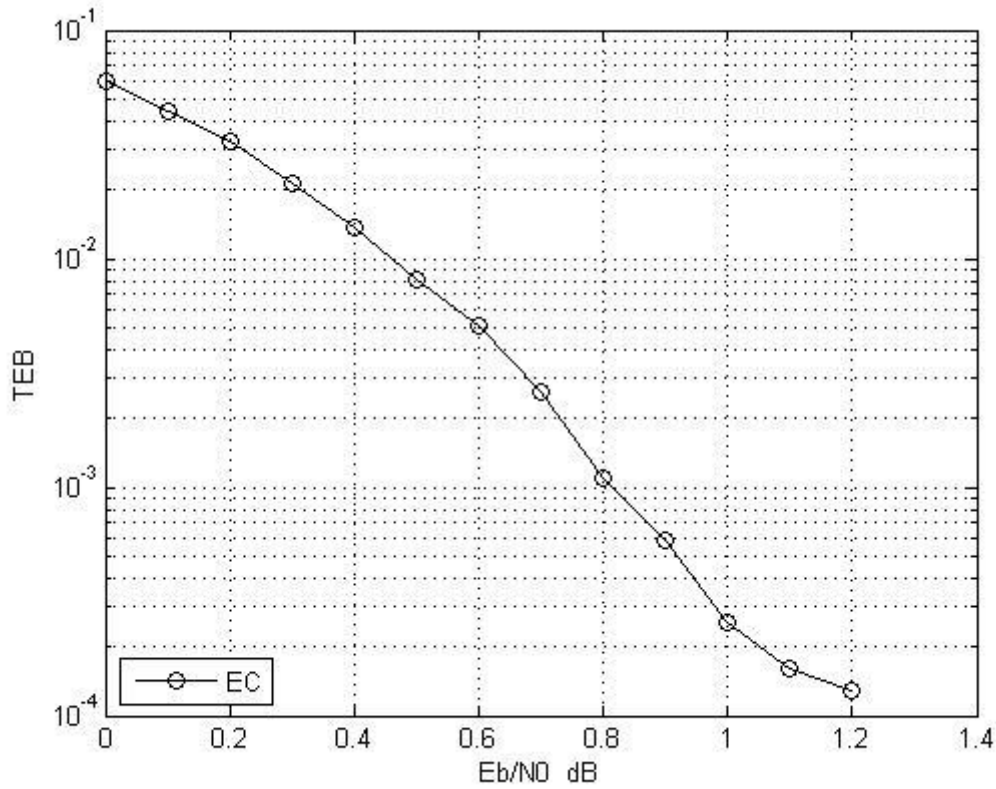


Figure 4.4 Performances d'un critère de l'entropie croisée EC

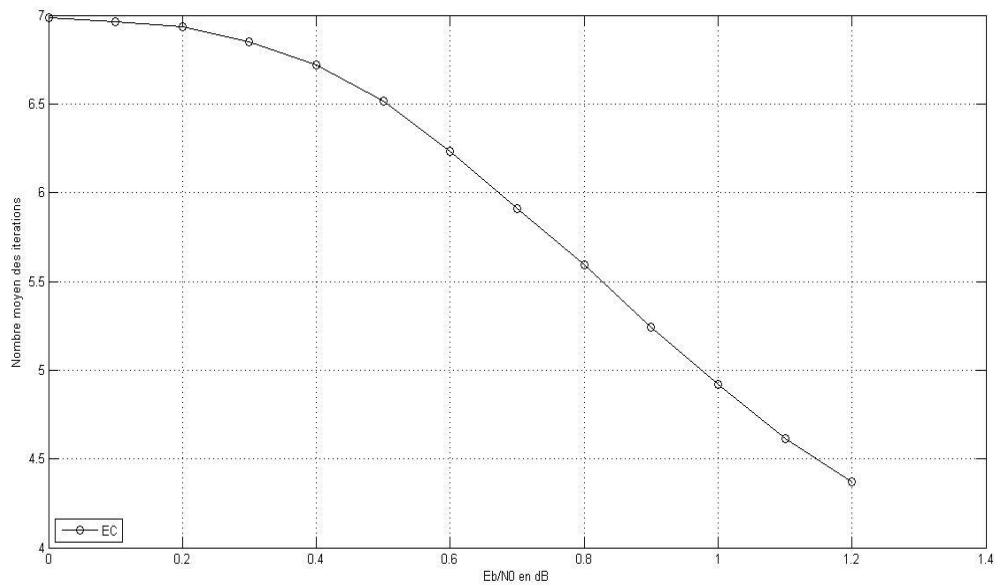


Figure 4.5 Nombre moyen des itérations du critère de l'entropie croisée EC

4.4.3 Interprétation des résultats :

La figure 4.4 montre le taux d'erreur binaire TEB d'un turbo décodeur [1, 5/7] supervisé par le critère d'arrêt de l'entropie croisée EC qui arrête le turbo décodage après l'obtention d'une trame correcte. Au RSB = 1 dB, le TEB = $2.7 \cdot 10^{-4}$.

Au RSB = 1.2 dB, le nombre moyen des itérations est réduit de 7 jusqu'à 4.35 itérations. C'est-à-dire, à la moyenne 2.65 itérations/trame ($7 - 4.35 = 2.65$ itérations /trames) est éliminées, ce qui accélère la vitesse du décodage. Pour 1000 mille trames, on a éliminé 2650 itérations, qui correspond à $2650 \times 2 = 5300$ décodages MAP.

4.5 Conclusion importante :

Le critère de l'entropie croisée EC a ajouté (à la moyenne) $4.35 - 3.25 = 1.1$ itération par rapport au critère théorique (idéal) GENIE. Notons que le critère GENIE reste irréalisable en pratique parce qu'il suppose la connaissance des bits informations. Cependant il (le critère GENIE) représente l'optimalité.

Conclusion générale :

Conclusion générale :

Tout au long de ce mémoire, nous avons étudié **les codes convolutifs, les turbocodes** et leurs algorithmes de décodage **Viterbi** et **MAP**. Ensuite, nous avons évalué les performances des turbo décodeurs MAP et leurs qualités de décodage.

Les résultats de simulation ont montrés les taux d'erreurs binaires TEB diminuent après chaque itération. Cependant, il est inutile de fixer le nombre d'itérations car certaines trames sont décodées correctement après quelques décodages (itérations).

L'objectif de ce travail consiste à évaluer les performances du critère d'arrêt de l'entropie croisée EC. Les résultats de simulation ont montré que le critère de l'entropie croisée EC a arrêté le turbo décodage au bon moment. Il a éliminé plusieurs itérations ce qui réduit la complexité calculatoire et accélère la vitesse de décodage coté réception.

La comparaison avec le critère idéal GENIE montre que le critère de l'entropie croisée ajoute à la moyenne 1.1 itération au forts rapports signal sur bruit RSB. Le critère GENIE reste un critère théorique de comparaison. Il est impossible d'utiliser ce critère dans des systèmes réels car il suppose la connaissance des trames d'information de l'émetteur (avant codage).

Pour terminer ce travail, nous proposons de vérifier l'optimalité du seuil utilisé par le critère EC. **Peut-on trouver un seuil optimal qui dépend fortement de l'entropie croisée de la première itération CE(1) et de la longueur des trames ??**

Bibliographie

Bibliographie

- [1] Bouazza Rabie Lakhdar, Kaoudj Salima " Critère d'arrêt CRC pour les turbo codes et son double intérêt pour les systèmes H-ARQ ",Mémoire Master, Université Dr. Tahar Molay ,2022.
- [2] A. J. Viterbi, " Error bounds for convolutional codes and an asymptotically optimum decoding algorithm ", IEEE Trans. Inform. Théorie, pp. 260-269, Avril 1967.
- [3] Mohamed El Amine M'Sir. "Conception d'architectures rapides pour codes convolutifs en télécommunications : application aux turbo-codes ". Université Paul Verlaine - Metz, 2003.
- [4] G. Baudoin, " Radiocommunications numériques 1 ", DUNOD, 2002.
- [5] J .G Proakis, "Digital Communication ", (3 rd édition), McGraw-Hill, 1995.
- [6] L. R. Bahl et al, " Optimal decoding of linear codes for minimizing symbol error rate ", IEEE Trans. Inf. Th, pp 284-287, March 1974.
- [7] C. Berrou and A. Glavieux, " Near Optimum Error Correcting Coding And Decoding : Turbo-codes ", IEEE Transactions on Communications, Vol. 44, N°10, pp.1262-1271, Octobre 1996.
- [8] C. Berrou " Codes et turbocodes ", École Nationale Supérieure des Télécommunications de Bretagne, Springer-Verlag France 2007.
- [9] A. Ouardi, "Sum-Log Stopping Criterion for Log-MAP Turbo Decoding," Advances in Electrical and Computer Engineering, vol. 23, no. 1, pp. 43-50, 2023.
- [10] 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.
- [11] A. Shibutani, H. Suda, and F. Adachi, "Reducing average number of turbo decoding iterations," *Electronics Letters*, vol. 35, no. 9, pp. 701–702, 1999.
- [12] 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, pp. 145–147, 1999.
- [13] 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, 2000.

Bibliographie

- [14] 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, 2000.
- [15] 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, 2003.
- [16] A. Taffin, "Generalised stopping criterion for iterative decoders," *Electronics Letters*, vol. 39, no. 13, pp. 993–994, 2003.
- [17] F. Zhai and I. J. Fair, "Techniques for early stopping and error detection in turbo decoding," in *IEEE Transactions on Communications*, vol. 51, no. 10, pp. 1617–1623, 2003.
- [18] B. Scanavino, G. M. Maggio, Z. Tasev, and L. Kocarev, "A novel stopping criterion for turbo codes based on the average a posteriori entropy," *IEEE Global Telecommunications Conference (GLOBECOM '03)*, San Francisco, CA, USA, 2003, vol. 4, pp. 2051–2055.
- [19] F. M. Li and A. Y. Wu, "On the new stopping criteria of iterative turbo decoding by using decoding threshold," in *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5506–5516, 2007.
- [20] 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, 2008.
- [21] S. Asoodeh, "New stopping criterion for turbo code in the presence of SNR mismatch," in *2010 International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT)*, 2010, pp. 182–186.
- [22] A. A. Chaudhry and M. A. Shah, "Comparison and analysis of ECSS based turbo decoder stopping rules," in *2012 International Conference on Emerging Technologies (ICET)*, Islamabad, Pakistan, 2012, pp. 1–6.
- [23] J. Wu, B. R. Vojcic, and J. Sheng, "Stopping criteria for iterative decoding based on mutual information," *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2012, pp. 183–187.
- [24] A. Savin, L. Trifina, and M. Andrei, "Threshold based iteration stopping criterion for turbo codes and for scheme combining a turbo code and a golden space-time block code," *Advances in Electrical and Computer Engineering*, vol. 14, no. 1, pp. 139–142, 2014.

Bibliographie

- [25] F. Alberge, "On some properties of the mutual information between extrinsics with application to iterative decoding," *IEEE Transactions on Communications*, vol. 63, no. 5, pp. 1541–1553, 2015.
- [26] T. P. Fowdur, Y. Beeharry, and K. M. S. Soyjaudah, "A novel scaling and early stopping mechanism for LTE turbo code based on regression analysis," *Annals of Telecommunications*, vol. 71, no. 7, pp. 369–388, 2016.
- [27] M. AlMahamdy and J. Dill, "Early termination of turbo decoding by identification of undecodable blocks," *IEEE Wireless Communications and Networking Conference (WCNC)*, San Francisco, CA, 2017, pp. 1–6.
- [28] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes", in *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429–445, 1996.
- [29] M. Y. M. Nasir, R. Mohamad, M. Kassim, N. M. Tahir and E. Abdullah, "Performance analysis of cross-entropy stopping criterion for quadrature amplitude modulation," *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, Shah Alam, Malaysia, 2019, pp. 273–27.
- [30] 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.