

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**UNIVERSITE Dr. TAHAR MOULAY SAIDA**

**Faculté de la Technologie**  
**Département d'Electronique**



**Mémoire de fin d'étude**  
**Pour l'obtention du Diplôme de Master En Télécommunications**  
**Option : Systèmes de Télécommunications**

# **Couplage Turbo égaliseur-Turbo décodeur**

**Présenté par :**

- Hakim Fatima Zohra
- Zidoun Fatima Zohra

**Encadré par:**

❖ **Dr. OUARDI Aissa**

**Devant les jurys composé de :**

- ✓ **Dr. BOUYEDDOU Benamer**      **Président**
- ✓ **Dr. TAMI Abdelkader**      **Examineur**

**Année universitaire :**  
**2019/2020**

## Remercîment

*Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce travail.*

*Le première personne que nous tenons à remercier est notre encadrant **Dr. Ouardi. A.** 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*

*Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de s'enrichir par leurs propositions.*

*Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail*



The background of the page is a soft, artistic illustration. It features several white lilies with bright orange stamens and green leaves. Interspersed among the flowers are several butterflies with white wings and black spots. The overall style is delicate and elegant, with a light green and white color palette.

# *Dédicace*

*A Mes chers parents*

*Vous avez su m'apprendre modestement les vraies valeurs de la vie. Votre soutien et vos encouragements ont toujours été sans faille tout au long de ce parcours de longue haleine. La confiance que vous m'accordez me permet d'avancer.*

*A mon frère et mes chères sœurs*

*A ma chère amie*

*Éidoun Fatima Zohra*

*Hakim Fatima Zohra*



# Dédicace

*A Dieu, tout puissant, qui m'a donné la force, la santé et le courage de  
réaliser ce précieux travail.*

*Mes chers parents*

*Pour tous leurs efforts, leur soutien sans faille, leur amour et leurs  
encouragements. Je voudrais leur exprimer ma reconnaissance et ma profonde  
affection.*

*Ma sœur et mes frères*

*A ma chère amie : Hakim Fatima*

*Et à Toutes la famille*

*Lidoun Fatima Lohra*



## Résumé

Le travail présenté dans ce mémoire se rapporte au couplage turbo égaliseur-turbo décodeur. Il décrit la façon de combiner l'égaliseur et les décodeurs pour traiter les interférence entre symboles IES et le bruit conjointement.

Le canal est supposé gaussien, sélectif en fréquence. L'égaliseur et le turbo décodeur utilise l'algorithme Maximum A Posteriori MAP. Les deux partagent entre eux les Log Rapport de Vraisemblance LLR. Une conversion des sorties de l'égaliseur est nécessaire pour adapter les informations extrinsèques à l'entrée du turbo décodeur MAP.

**Mots clés :** Egalisation, turbo décodage MAP, Turbo égalisation MAP, Algorithme MAP-BCJR, entrelacement.

## **Abstract**

The work presented in this thesis relates to the combined turbo equalizer-turbo decoder receiver. It describes how to combine the equalizer and decoders to deal with IES symbol interference and noise together.

The channel is assumed to be gaussian, frequency selective. The equalizer and turbo decoder uses the Maximum A Posteriori algorithm MAP. Both share the Log Likelihood Ratios LLR with each other. A conversion of the equalizer outputs is necessary to adapt the extrinsic informations to the input of the MAP turbo decoder.

**Keywords:** Equalization, turbo decoding MAP, Turbo equalization MAP, MAP-BCJR algorithm, interleaving.

# Table des matières

---

|  |           |
|--|-----------|
| Remerciements  | I         |
| Dédicace   | II        |
| Dédicace   | III       |
| Résumé   | IV        |
| Abstract   | V         |
| Table des matières   | VI        |
| Liste des Figures  | X         |
| Liste d'abréviation  | XII       |
| Introduction Générale  | 1         |
| <b>Chapitre I : Codage et égalisation</b>                        |           |
| <b>I.1. Introduction</b>   | <b>04</b> |
| <b>I.2. Les codes convolutifs</b>                                | <b>04</b> |
| <b>I.2.1. Les types de code convolutif</b>                       | <b>04</b> |
| • Les Codes systématique non récursif                            | 04        |
| • Les Codes Systématiques Récursifs (CSR)                        | 05        |
| <b>I.2.2. Principe du codage convolutif</b>                      | <b>05</b> |
| <b>I.2.3. Représentation des codes convolutifs</b>               | <b>07</b> |
| • Diagramme en arbre   | 07        |
| • Diagramme en treillis  | 08        |
| • Diagramme d'état   | 09        |
| <b>I.3. Le décodage des codes convolutifs</b>                    | <b>10</b> |
| <b>I.4. Intèrférence entre symboles</b>                          | <b>10</b> |
| <b>I.5. Egalisation</b>  | <b>12</b> |
| <b>I.6. Le système de communication numerique</b>                | <b>12</b> |
| <b>I.7. Algorithme MAP-BCJR</b>                                  | <b>13</b> |
| <b>I.8. Déroulement de L'algorithme MAP</b>                      | <b>17</b> |
| <b>I.9. Decodage avec l'algorithme MAP</b>                       | <b>18</b> |
| <b>I.10. L'algorithme MAP-BCJR appliqué à l'égalisation</b>      | <b>19</b> |
| I.10.1. Diagramme en treillis                                    | 20        |
| I.10.2. Calcul des Poids des Branches                            | 20        |
| I.10.3. Calcul des $\gamma$                                      | 21        |
| I.10.4. Calcul des $\alpha$                                      | 21        |
| I.10.5. Calcul des $\beta$                                       | 22        |
| I.10.6. Calcul des Probabilités des Symboles                     | 22        |
| I.10.6.1. Calcul de la Probabilité Du Symbole +1 à l'instant 'i' | 22        |
| I.10.6.2. Calcul de la probabilité du symbole -1 à l'instant 'i' | 22        |
| I.10.7. Calcul du Log Rapport de Vraisemblance LLR               | 23        |
| <b>I.11. Détection</b>   | <b>23</b> |
| <b>I.12. Conclusion</b>  | <b>23</b> |
| <b>Chapitre II : Les Turbo récepteurs</b>                        |           |
| <b>II.1. Introduction</b>  | <b>25</b> |
| <b>II.2. Principe turbo</b>                                      | <b>25</b> |
| II.2.1. Définition   | 25        |
| <b>II.3. Traitement Turbo et information extrinsèque</b>         | <b>27</b> |
| <b>II.4. Mécanisme Turbo</b>                                     | <b>30</b> |
| <b>II.5. Turbo codes</b>   | <b>30</b> |
| II.5.1. Turbo codeur   | 31        |

# Table des matières

---

|   |           |
|---|-----------|
| II.5.2. Entrelaceur   | 31        |
| <b>II.6. Structure de l'égaliseur</b>                         | <b>32</b> |
| <b>II.7. Principe de la turbo égalisation</b>                 | <b>33</b> |
| ➤ Egaliseur SISO  | 34        |
| ➤ Entrelacement/Désentrelacement des données                  | 35        |
| ➤ Décodeur SISO   | 35        |
| <b>II.8. Turbo égalisation MAP</b>                            | <b>35</b> |
| <b>II.9. Conclusion</b>                                       | <b>36</b> |
| <b>Chapitre III : Couplage Turbo égaliseur-Turbo décodeur</b> |           |
| <b>III.1. Introduction</b>                                    | <b>38</b> |
| <b>III.2. Modèle de système</b>                               | <b>38</b> |
| <b>III.3. Décodeur</b>  | <b>38</b> |
| <b>III.4. Modèle de canal équivalent</b>                      | <b>40</b> |
| <b>III.5. Conclusion</b>                                      | <b>41</b> |
| <b>Chapitre VI : Résultats de simulation</b>                  |           |
| <b>VI.1. Introduction</b>                                     | <b>43</b> |
| <b>VI.2. Conditions de simulation</b>                         | <b>43</b> |
| <b>VI.3. Performance du turbo décodage [1 5/7]</b>            | <b>44</b> |
| <b>VI.4. performance du turbo égaliseur - turbo décodeur</b>  | <b>45</b> |
| <b>Conclusion</b>   | <b>46</b> |
| <b>Conclusion générale</b>                                    | <b>48</b> |
| <b>Références Bibliographiques</b>                            | <b>50</b> |



## Listes des figures

---

### Liste des figures


|                     |   |           |
|---------------------|---|-----------|
| <b>Figure I. 1</b>  | Codeur convolutionnel systématique non récursif                           | <b>05</b> |
| <b>Figure I. 2</b>  | Codeur convolutionnel systématique récursif                               | <b>05</b> |
| <b>Figure I. 3</b>  | principe d'un codeur convolutif   | <b>06</b> |
| <b>Figure I. 4</b>  | Exemple de codeur convolutifs non systématique<br>( $R=1/2 ; m=2$ )       | <b>06</b> |
| <b>Figure I.5</b>   | Diagramme en arbre du codeur convolutif                                   | <b>08</b> |
| <b>Figure I. 6</b>  | Diagramme en treillis du codeur convolutif                                | <b>09</b> |
| <b>Figure I.7</b>   | Diagramme d'état du codeur convolutif                                     | <b>10</b> |
| <b>Figure I. 8</b>  | Intèrference entre symboles   | <b>11</b> |
| <b>Figure I.9</b>   | Technique d'égalisation   | <b>12</b> |
| <b>Figure I.10</b>  | Principe d'une chaine de transmission numérique                           | <b>13</b> |
| <b>Figure I.11</b>  | Source de MarKov à travers un canal bruité                                | <b>14</b> |
| <b>Figure I.12</b>  | Structure transversale du filtre RIF                                      | <b>19</b> |
| <b>Figure I.13</b>  | Diagramme en treillis d'un canal de longueur 3                            | <b>20</b> |
| <b>Figure II.1</b>  | Principe turbo et émetteur parallèle ou serie                             | <b>26</b> |
| <b>Figure II.2</b>  | Principe turbo récepteur  | <b>26</b> |
| <b>Figure II.3</b>  | Principe itératif du turbo égaliseur                                      | <b>27</b> |
| <b>Figure II.4</b>  | Traitement itératif   | <b>27</b> |
| <b>Figure II.5</b>  | Différentes informations générées par un traitement<br>SISO               | <b>29</b> |
| <b>Figure II.6</b>  | Turbo codeur  | <b>31</b> |
| <b>Figure II.7</b>  | Fonction d'entrelacement  | <b>32</b> |
| <b>Figure II.8</b>  | Principe de turbo égaliseur   | <b>32</b> |
| <b>Figure II.9</b>  | Architecture de base du recepteur itératif utilisant le<br>principe turbo | <b>34</b> |
| <b>Figure II.10</b> | Turbo égaliseur utilisant le critère MAP                                  | <b>35</b> |
| <b>Figure III.1</b> | Un bloc MAP   | <b>39</b> |
| <b>Figure III.2</b> | Structure générale du décodeur combiné                                    | <b>39</b> |
| <b>Figure III.3</b> | Structure du décodeur turbo   | <b>39</b> |

# Listes des abréviations

---

## Liste des abréviations

|      |  |
|------|--|
| APP  | A Posterior Probability                |
| AWGN | Additive White Gaussian Noise          |
| BBGA | Bruit Blanc Gaussien Additif           |
| BCJR | Bahl, Cocke, Jelinek, and Raviv        |
| BER  | Bit Error Rate                         |
| BPSK | Binary Phase Shift Keying              |
| IES  | Interférence Entre Symboles            |
| LLR  | Logarithm of Likelihood Ratio          |
| LRV  | Logarithme de Rapport de Vraisemblance |
| MAP  | Maximum A Posteriori                   |
| RSB  | Rapport Signal sur Bruit               |
| SISO | Soft Input Soft Output                 |
| SOVA | Soft Output Viterbi Algorithm          |
| TEB  | Taux d'Erreur Binaries                 |



# Introduction générale

# Introduction Générale

---

## Introduction Générale :

Les télécommunications font partie des technologies qui ont révolutionné notre mode de vie au vingtième siècle. Du télégraphe à l'Internet, les progrès établis en la matière sont spectaculaires. Les informations transmises étaient tout d'abord codées en morse, puis des techniques de modulation et de codages analogiques ont permis de transmettre du son, puis des images. Ensuite la venue des techniques numériques a considérablement augmenté le débit et la qualité des informations à transmettre d'un point à un autre.

Parallèlement, le développement rapide de la microélectronique et des capacités de miniaturisation permet aujourd'hui la mise en oeuvre de techniques complexes dans des appareils de taille réduite.

Dans les systèmes de transmission numériques, le fait de vouloir passer un flot important d'informations à travers un canal dont la bande est souvent limitée a tendance à créer de l'Interférence Entre Symboles (IES). Cette interférence peut dégrader très fortement le signal reçu. Il est donc nécessaire de concevoir des récepteurs permettant de combattre efficacement l'IES.

Durant plusieurs décennies, la limite de Shannon semblait inaccessible. En 1993, Berrou, Glavieux et Thitimajhima, ont proposé un décodage original exploitant l'algorithme MAP. Ce décodage, appelé « turbo », permet d'approcher la limite fixée par Shannon pour des canaux Gaussiens non sélectifs en fréquence. Parmi les codes connus à l'époque, les codes turbo sont les premiers à s'en approcher d'aussi près.

Le turbo code permet à un système de communication d'opérer beaucoup plus près de la limite théorique définie par Shannon et de réaliser ainsi un 'Gain de Codage' supérieur à celui déjà réalisé par l'algorithme de Viterbi. Dans son article «Near Shannon Limit Error-Correcting Coding and Decoding», Berrou, Glavieux et Thitimajhima ont utilisé deux encodeurs convolutionnels récursifs, avec sortie systématique RSC (*Recursive Systematic Convolution Code*) en parallèle, avec l'insertion d'un entrelaceur entre les deux, en utilisant une version modifiée de l'algorithme de décodage MAP dans le processus de décodage.

Le turbo égaliseur optimal utilise l'algorithme MAP qui minimise la probabilité d'erreur par symbole. Cependant, il souffre d'une grande complexité calculatoire lorsque la longueur de la réponse impulsionnelle du canal est grande.

Dans ce mémoire nous nous sommes intéressés au couplage turbo égaliseur-turbo décodeur.

Nous décrivons comment peut-on traiter conjointement les interférences entre symboles IES et le bruit. La combinaison du turbo égaliseur avec le turbo décodeur

# Introduction Générale

---

nécessite quelques modifications des sorties de l'égaliseur avant leur injection à l'entrée du turbo décodeur.

Le manuscrit est organisé en Quatre chapitres :

Dans **le premier chapitre**, nous introduisons le codage et égalisation, nous décrivons les codes convolutifs, ses types, principe et présentation. Nous décrivons aussi le problème des IES et égalisation et l'algorithme MAP.

Dans **le deuxième chapitre**, nous traitons les turbo récepteurs, leur principe, le turbo codage MAP et turbo égalisation MAP.

**Le troisième chapitre** a été consacré au couplage turbo égaliseur – turbo décodeur et les adaptations nécessaires.

**Le dernier chapitre**, est consacré aux résultats de simulation et leurs interprétations.





# Chapitre I : Codage et égalisation

## I.1. Introduction

L'essor des communications numériques pour les transmissions sur des canaux sélectifs en fréquences est à l'origine de nombreux travaux sur le traitement de l'Interférence Entre Symboles (IES). Les différentes solutions proposées utilisent généralement des procédés d'égalisation et de codage de canal.

Dans l'approche classique, l'égalisation et le décodage sont traités de manière disjointe. Dans les systèmes avec turbo égalisation, nous utilisons le traitement itératif d'égalisation et de décodage de canal conjoint.

## I.2. Les codes convolutifs

En 1955 Elias inventé les codes convolutifs. Ils forment une classe extrêmement souple et efficace de codes correcteurs d'erreurs. Ce sont les codes les plus utilisés dans les systèmes de télécommunications fixes et mobiles, où une faible probabilité d'erreur par bit est requise. Théoriquement, ils ont les mêmes caractéristiques que les codes en blocs sauf pour la valeur de leur dimension et leur longueur. Les codes convolutifs s'appliquent sur des séquences finies ou infinies de symboles d'information et génèrent des séquences finies ou infinies de symboles codés. Les codes convolutionnels sont basés sur l'introduction de la redondance à l'aide des registres à décalage [1].

### I.2.1. Les types de code convolutif

Il existe deux types de codeurs convolutionnels : codeur convolutionnel systématique non récursif et codeur convolutionnel récursif et systématique.

- **Les Codes systématique non récursif**

Un codeur convolutionnel est dit systématique si les symboles codés comportent une réplique exacte des bits d'information de l'entrée. Autrement dit, il laisse passer directement un bit d'information de l'entrée à la sortie comme symbole systématique. Les autres symboles à la sortie de l'encodeur sont appelés symboles de parités. Le principe de fonctionnement d'un codeur convolutionnel systématique non récursif est illustré à la figure (I.1).[1]

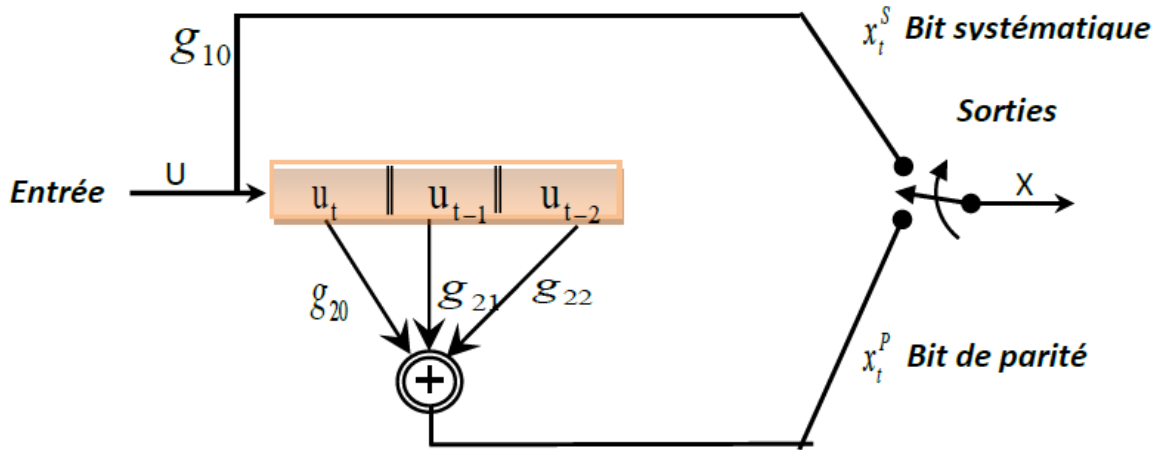


Figure I. 1: Codeur convolusionnel systématique non récursif.

- **Les Codes Systématiques Récursifs (CSR)**

Il existe un autre type de codeur, à savoir les codeurs récursifs et systématiques (RSC) est obtenu par les deux propriétés : La récursivité est caractérisée par le fait que la sortie de l'encodeur sont fonctions des entrées et des sorties précédentes ; et la propriété systématique, que nous avons déjà mentionné, qu'il s'agissait de transmettre les bits d'information dans les symboles codés. Le principe de fonctionnement du codeur CSR est illustré dans la figure (I.2). Le contenu de la première cellule de registre à décalage ne dépend pas seulement du bit d'information à l'entrée mais aussi du contenu des cellules de registre.[1]

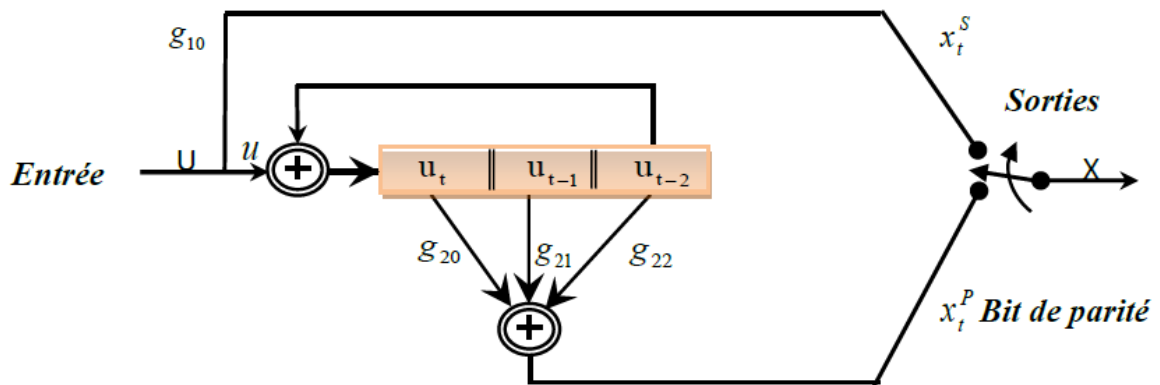


Figure I. 2: Codeur convolusionnel systématique récursif.

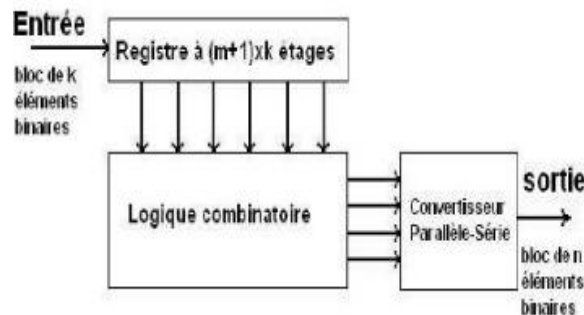
### I.2.2. Principe du codage convolutif [2]

Les codes convolutifs ou récurrents constituent une seconde famille de codes correcteurs d'erreurs au moins importante que les codes en blocs cyclique. Pour les codes

## Chapitre I : Codage et égalisation

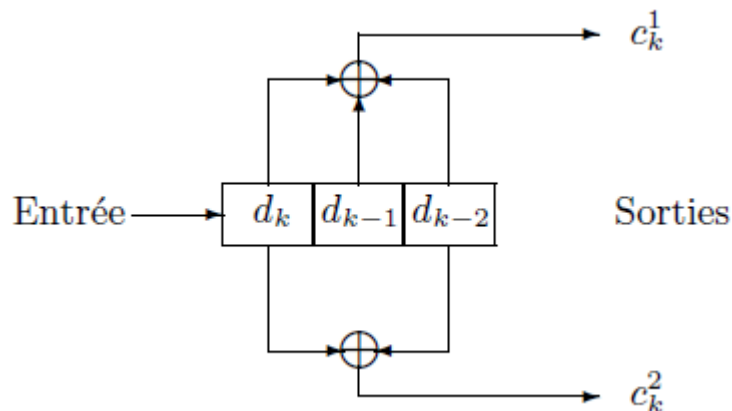
convolutifs, chaque bloc de  $n$  éléments binaires en sortie du codeur dépend non seulement du bloc de  $k$  éléments binaires présent à son entrée mais aussi des  $m$  blocs présents précédemment. Les codes convolutifs introduisent par conséquent un effet de mémoire d'ordre  $m$ . La quantité  $(m+1)$  s'appelle la longueur de contrainte du code. Le principe du codage convolutif est illustré par le schéma de la figure (I.3).

Un codeur est constitué d'un registre à  $(m+1)k$  étages qui mémorise les  $(m+1)$  blocs de  $k$  éléments binaires d'information, d'une logique combinatoire qui calcule les blocs de  $n$  éléments binaires fournis par le codeur et d'un convertisseur parallèle série. La quantité  $R=k/n$  est appelée, comme pour les codes en blocs, le rendement du code. Si les  $k$  éléments binaires d'information présents à l'entrée du codeur sont effectivement émis, c'est-à-dire se retrouvent explicitement dans le bloc de  $n$  éléments binaires en sortie du codeur, le code est systématique



**Figure I. 3:** principe d'un codeur convolutif.

Pour illustrer cette nouvelle famille de codes, nous avons présenté sur la figure (I.4) un exemple de codeur convolutif de rendement  $R=1/2$  et de longueur de contrainte  $(m+1)=3$ . Son entrée est constituée par les blocs de  $k=1$  élément binaire et sa sortie par des blocs de  $n=2$  éléments binaires.



**Figure I. 4 :** Exemple de codeur convolutifs non systématique ( $R=1/2$  ;  $m=2$ ).

## Chapitre I : Codage et égalisation

---

Le qualificatif « convolutif » appliqué à ces codes provient du fait que chaque sortie du codeur est égale au produit de convolution entre la suite binaire présente à l'entrée du codeur et la réponse du codeur, définie par ses séquences génératrices.

Les deux séquences génératrices sont  $g_i = [g_{i0}, g_{i1}, g_{i2}] ; i=1,2$ . En générale on exprime les séquences génératrices en octal et ainsi, pour le codeur de figure (I.4), nous avons :

$$g_1 = [1; 1; 1] = 7(\text{octal})$$

$$g_2 = [1; 0; 1] = 5(\text{octal})$$

Notons que, les sorties du codeur étant égales à une combinaison linéaire des éléments binaires d'information, le code est linéaire.

Les codes convolutifs sont aussi définis à partir de leurs polynomes générateurs exprimés en fonction de la variable  $D$  (delay) équivalente à la variable  $z^{-1}$  de la transformée en  $z$ .

En considérant toujours l'exemple de la figure (I.4), les polynomes générateurs de ce code ont pour expression :

$$G_1(D) = g_{10} + g_{11}D + g_{12}D^2$$

$$G_2(D) = g_{20} + g_{21}D + g_{22}D^2$$

Soit encore :

$$G_1(D) = 1 + D + D^2$$

$$G_2(D) = 1 + D^2$$

### I.2.3. Représentation des codes convolutifs [2]

L'introduction d'un effet de mémoire complique notablement la représentation des codes convolutifs à partir d'une matrice génératrice ou de polynomes générateurs. La représentation la plus usuelle et aussi la mieux adaptée l'élaboration d'algorithme de décodage est incontestablement la représentation graphique sous forme d'un arbre, d'un treillis ou encore d'un diagramme d'état. Pour introduire ces différentes représentations, nous allons utiliser le codeur convolutif non systématique représenté sur la figure (I.4).

- **Diagramme en arbre :**

Le diagramme en arbre associé au codeur convolutif de la figure (I.4) est représenté sur la figure (I.5). Sur ce diagramme nous avons adopté les conventions suivantes :

- Le temps s'écoule de la gauche vers la droite,



## Chapitre I : Codage et égalisation

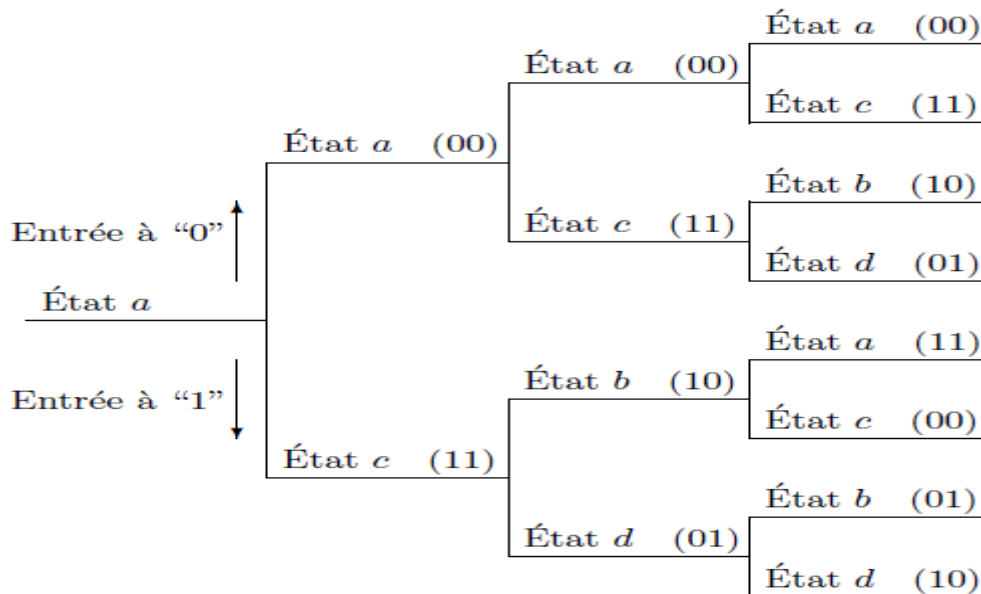
- Lorsque l'élément binaire à l'entrée du codeur est égal à 0 (respectivement à 1), le couple binaire en sortie du codeur (noté entre parenthèses) est porté par une branche montante (respectivement descendante) du diagramme en arbre. Les branches montantes et descendantes se séparent en un point appelé nœud. Plus généralement, en considérant des blocs de  $k$  éléments binaires d'information à l'entrée du codeur, chaque nœud donne naissance à  $2^k$  branches.

Pour une séquence binaire donnée à l'entrée du codeur, la séquence correspondante en sortie du codeur est représentée par un chemin dans l'arbre, constitué par une suite de branches.

Chaque bloc de  $n=2$  éléments binaires en sortie du codeur dépend du bloc de  $k=1$  élément binaire présent à son entrée mais aussi des  $m=2$  blocs de  $k$  éléments binaires contenus dans sa mémoire. Ces  $mk=2$  éléments binaires définissent l'état du codeur. Nous noterons les quatre états possible de codeur de la façon suivante :

$$a=00 ; b=01 ; c=10 ; d=11$$

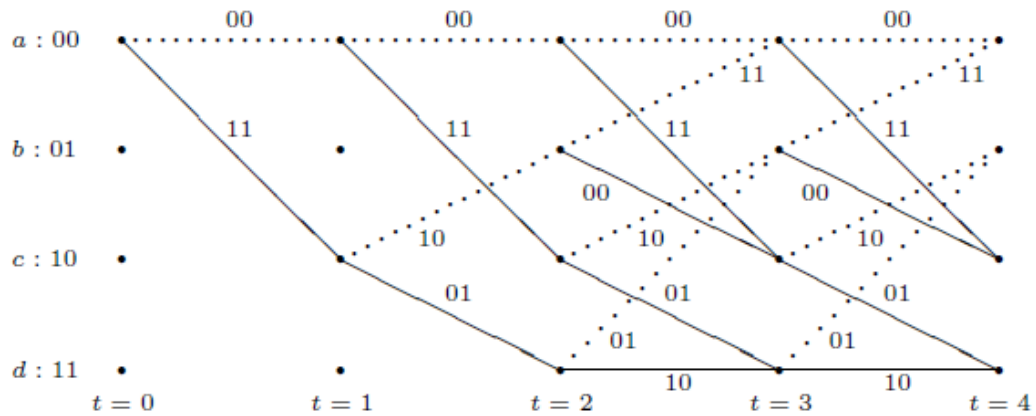
Quel que soit l'état initial du codeur, après  $(m+1)=3$  décalages à l'entrée du codeur, tous les états peuvent être atteints dans l'arbre.



**Figure 1.5 :** Diagramme en arbre du codeur convolutif de la figure (I.4).

- **Diagramme en treillis :**

La sortie du codeur dépendant uniquement de son entrée et de son état, il est donc possible d'utiliser une représentation plus concise que l'arbre, appelé diagramme en treillis. Dans ce diagramme, sont pris en compte les états différents du codeur et la façon dont ils communiquent en fonction du temps. Sur la figure (I.6), nous avons représenté le treillis associé au codeur convolutif de la figure (I.4) en faisant l'hypothèse que l'état initial du codeur était  $\alpha=00$



**Figure I. 6 :** Diagramme en treillis du codeur convolutif de la figure (I.4)

Les branches en traits pointillés correspondent à la présence d'un élément binaire d'information égal à 0 à l'entrée du codeur et les branches en trait plein, à un élément binaire égal à 1. A chaque branche nous avons associé la valeur du couple binaire disponible en sortie du codeur.

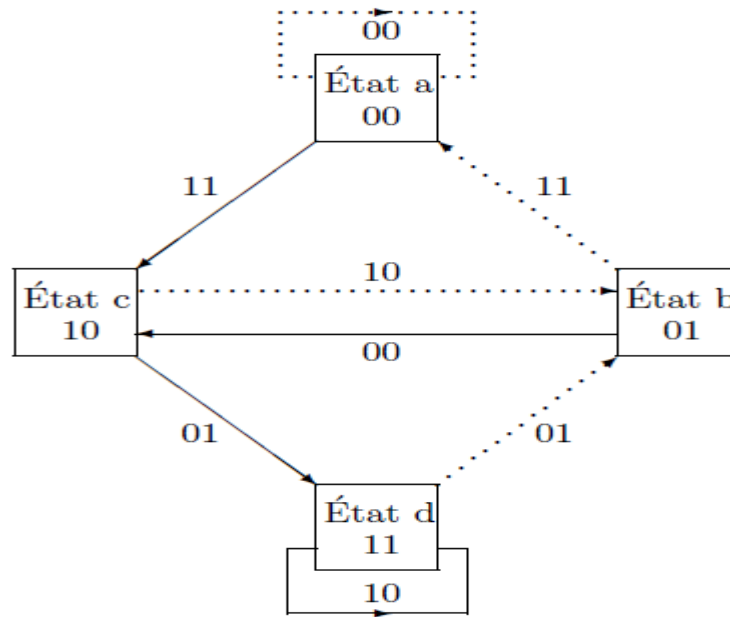
Après  $(m+1)$  décalage, quel que soit l'état initial du codeur, le motif du treillis se répète. De chaque nœud partent  $2^K$  branches (ici 2), et en chaque nœud convergent  $2^K$  branches.

Partant de l'état  $\alpha=00$  à l'instant  $t=0$  par exemple, nous voyons qu'il existe quatre chemins qui permettent d'atteindre l'état  $\alpha=00$  à l'instant  $t=4$  :

- 00 00 00 00    chemin1
- 11 10 11 00    chemin2
- 11 01 01 11    chemin3
- 00 11 10 11    chemin4

- **Diagramme d'état**

Le diagramme d'état est une autre représentation du fonctionnement d'un codeur convolutif, ne faisant pas apparaître explicitement le temps. Ce diagramme, qui peut se déduire du treillis, ne retient que les différents états du codeur et la façon dont ils communiquent. Sur la figure (I.6), nous avons représenté le diagramme d'état associé au codeur convolutif de la figure (I.4). Le diagramme d'état permet d'évaluer la fonction de transfert du codeur qui sera utilisée pour le calcul des performances du code.



**Figure I. 7 :** Diagramme d'état du codeur convolutif de la figure (I.5).

### I.3. Le décodage des codes convolutifs

Il existe plusieurs algorithmes de décodage des codes convolutifs. Le plus célèbre est probablement l'algorithme de Viterbi qui repose sur la représentation en treillis des codes. Il permet de trouver, à partir de la séquence des symboles reçus, la séquence d'états dans le treillis la plus probable. L'algorithme de Viterbi originel effectue un décodage à sortie ferme, c'est à- dire qu'il fournit une estimation binaire de chacun des symboles transmis. Il n'est donc pas directement adapté aux systèmes itératifs qui requièrent une information de confiance sur les décisions. Des adaptations de l'algorithme de Viterbi ont conduit aux versions à sortie pondérée dites SOVA (Soft-Output Viterbi Algorithm).

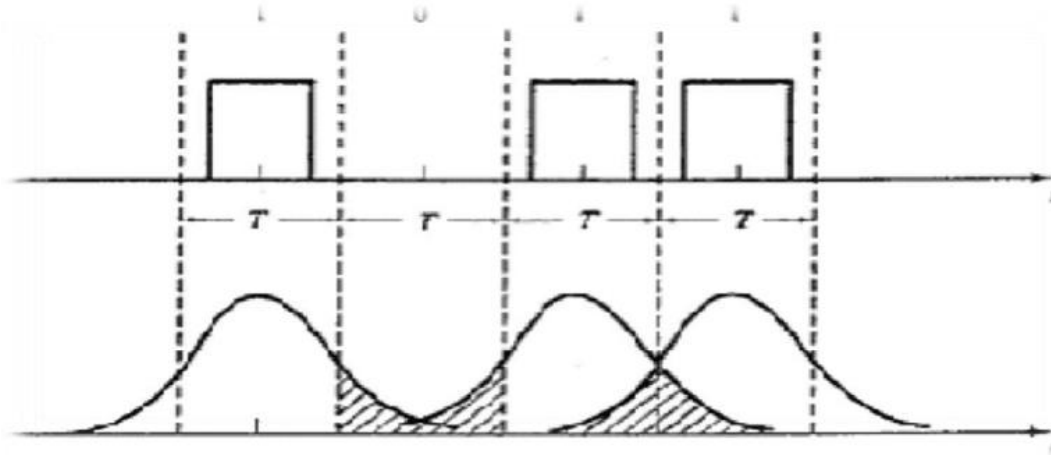
L'algorithme Maximum A Posteriori (MAP) permet le calcul de la valeur exacte de la probabilité a posteriori associée à chaque symbole transmis en utilisant la séquence reçue. [3]

### I.4. Intèrférence entre symboles (IES)

Dans un canal à bande passante infinie aussi appelé canal idéal (qui n'a pas de réalité physique), la bande passante est suffisamment large pour transmettre les signaux modulés sans distorsion. Un tel canal n'est plus utilisable lorsque le canal est partagé entre plusieurs utilisateurs. Pratiquement, les canaux de communication ont une bande passante limitée, et, par conséquent, les impulsions transmises ont tendance à s'étaler lors de leur propagation.[4]

## Chapitre I : Codage et égalisation

L'étalement des impulsions conduit aux chevauchements des impulsions adjacentes comme le montre la figure ci-dessous



**Figure I. 8 :** Interférence entre symboles.

Ce chevauchement entre les impulsions est appelé interférence entre les symboles (IES) peut causer une mauvaise interprétation des données reçues par le récepteur. La transmission multi-trajets produit aussi les IES. La figure (I.8) illustre un cas concret ; dans cette situation, l'onde émise peut emprunter deux chemins pour atteindre le récepteur : un trajet direct et un trajet réfléchi. Les temps de propagation étant différents sur ces deux trajets, le récepteur observe la superposition de deux versions décalées du signal émis. Nous pouvons donc décomposer l'échantillon  $y_k$  en une somme de trois termes :

$$y_k = a_k g_0 + \sum_{n \neq k} a_n g_{k-n} + W_k \quad (\text{I.1})$$

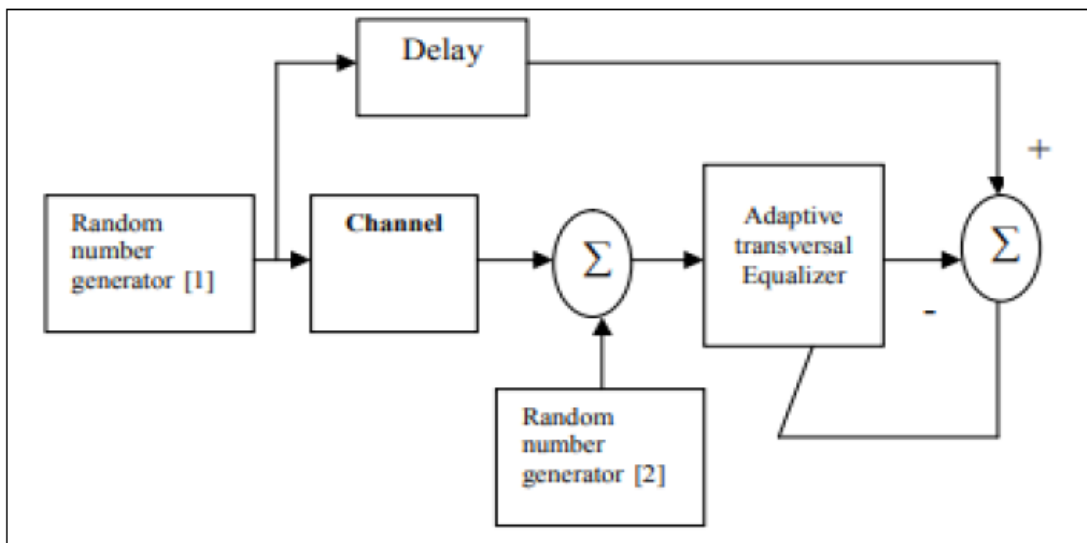
Le premier terme dépend du symbole  $a_k$ , le deuxième terme est appelé terme d'Interférence Entre Symboles (IES) et le troisième représente le bruit. Autrement dit  $y_k$  dépend de la donnée présente  $a_k$  et de des données voisins passées et futures.

La présence du terme d'IES, qui dépend de plusieurs symboles émis, ne permet pas d'effectuer une décision symbole par symbole qui soit en même temps une décision optimale. On peut alors considérer trois approches : la première consiste à faire en sorte que le terme d'IES soit toujours nul ; elle aboutit au critère de Nyquist.

Toutefois cette approche présente des faiblesses dans la mesure où elle ne permet pas de tirer au mieux profit de la bande disponible. La deuxième et la troisième approche sont l'égalisation linéaire (égaliseur sous optimale) et l'algorithme de Viterbi (égaliseur optimale) respectivement. L'objectif de l'égalisation est d'annuler les distorsions introduites par le canal de transmission et restituer le message émis à partir des observations reçues [4].

## I.5. Egalisation du canal

L'égalisation du canal est le processus de réduction des distorsions d'amplitude, de fréquence et de phase dans un canal dans le but d'améliorer les performances de transmission. L'opération de base de l'égalisation des canaux consiste à inverser l'effet du canal. L'égalisation des canaux est toujours effectuée après l'estimation du canal. L'égalisation adaptative est une technique qui s'adapte automatiquement aux propriétés variables du temps du canal de communication [4].



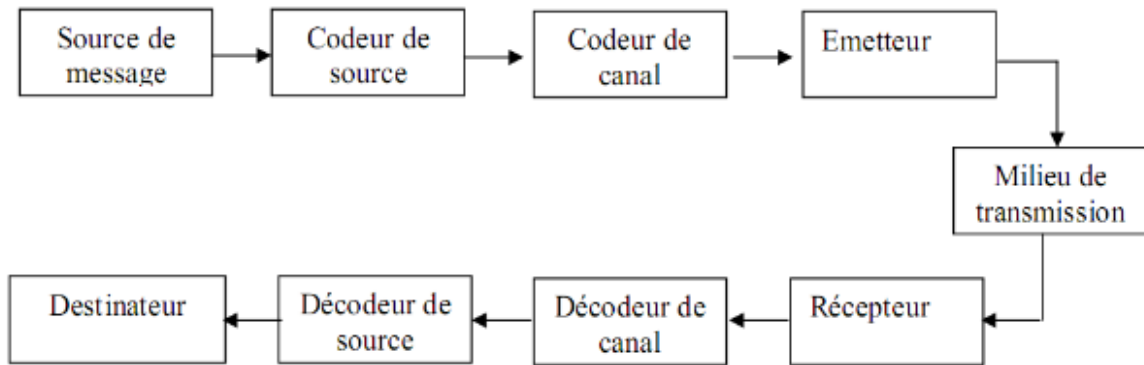
**Figure I. 9** : Technique d'égalisation.

L'égalisation des canaux peut également être effectuée dans le domaine fréquentiel.

## I.5. Le système de communication numérique [5]

Les systèmes de transmission numérique véhiculent de l'information entre une source et un destinataire en utilisant un support physique comme le câble, la fibre optique ou encore, la propagation sur un canal radioélectrique. Les signaux transportés peuvent être soit directement d'origine numérique, comme dans les réseaux de données, soit d'origine analogique (parole, image...) mais convertis sous une forme numérique. La tâche d'un système de transmission est d'acheminer l'information de la source vers le destinataire avec le plus de fiabilité possible. Le schéma synoptique d'un système de transmission numérique est donné à la figure (I.10). On se limite aux fonctions de base :





**Figure I.10** : Principe d'une chaîne de transmission numérique.

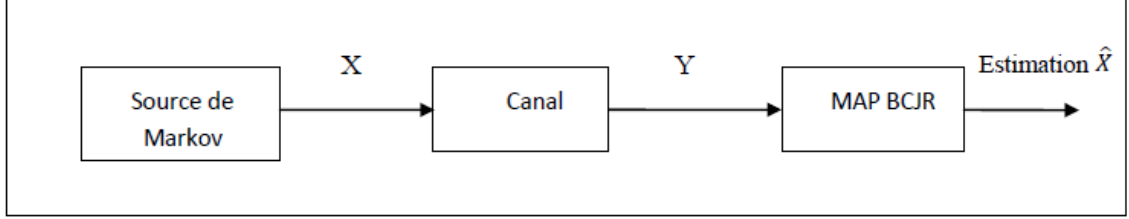
- La source émet un message numérique sous la forme d'une suite d'éléments binaires.
- Le codeur de source supprime les éléments binaires non significatifs (Compression de données ou codage de source).
- Le codeur de canal introduit de la redondance dans l'information en vue de la protéger contre le bruit présent sur le canal de transmission. Le codage de canal n'est possible que si le débit de la source est inférieur à la capacité du canal de transmission (La probabilité d'erreur  $P_e$  tend dans ce cas vers 0 d'après les travaux de Shannon).
- La modulation a pour rôle d'adapter le spectre du signal au canal (milieu physique) sur lequel il sera émis.
- Enfin, du côté récepteur, les fonctions de démodulation et de décodage sont les inverses respectifs des fonctions de modulation et de codage situés du côté émetteur.

### I.6. Algorithme MAP-BCJR

L'algorithme MAP permet de calculer la probabilité a posteriori de chaque bit d'information ou de chaque symbole transmis et le décodeur correspondant sélectionne à chaque instant le bit ou le symbole le plus probable. Cet algorithme a été publié en 1974 par Bahl, Cocke, Jelinek et Raviv. La portée de cette méthode de décodage est restée confidentielle jusqu'à la découverte des turbocodes car elle n'apporte pas d'amélioration de performance notable par rapport à l'algorithme de Viterbi pour le décodage des codes convolutifs et s'avère plus complexe à mettre en oeuvre. En revanche la situation a changé en 1993 car le décodage des turbocodes fait appel à des décodeurs élémentaires à sorties pondérées ou souples et l'algorithme MAP, contrairement à l'algorithme de Viterbi, permet d'associer naturellement une pondération à chaque décision. [3]

## Chapitre I : Codage et égalisation

L'algorithme MAP estime la probabilité à posteriori APP de l'état et de la transition d'une source de Markov observée à travers un canal bruité qui peut être représenté par un treillis d'états (Figure I.11)



**Figure I. 11** : Source de Markov à travers un canal bruité.

Cette situation peut correspondre à plusieurs applications en communications numériques. A titre d'exemples, nous pouvons citer le codage convolutif, canal sélectif avec mémoire, ... Toutes ces applications sont équivalentes à des sources de Markov représentées par le treillis [6]

$$X_1^L = X_1 X_2 \dots X_L \quad (I.2)$$

L'observation à la sortie du canal est :

$$Y_1^L = Y_1 Y_2 \dots Y_L \quad (I.3)$$

L'objectif du décodeur est d'estimer l'APP des états

$$\text{Prob} \{s_t = m / y_1^L\} = \frac{\text{prob} \{s_t = m, y_1^L\}}{\text{prob} \{y_1^L\}} \quad (I.4)$$

Ou des transitions de l'état  $S_{t-1} = m'$  vers l'état  $S_t = m$

$$\text{Prob} \left\{ s_{t-1} = m', s_t = m / y_1^L \right\} = \frac{\text{prob} \{s_{t-1} = m', s_t = m, y_1^L\}}{\text{prob} \{y_1^L\}} \quad (I.5)$$

Il est simple de calculer les probabilités

$$\lambda_t(m) = \text{Prob} \{s_t = m, Y_1^L\} \quad (I.6)$$

Et

$$\sigma_t(m', m) = \text{Prob} \{S_{t-1} = m', S_t = m, Y_1^L\} \quad (I.7)$$

## Chapitre I : Codage et égalisation

---

Puisque  $Prob Y_1^L$  est constante, alors, il est facile de calculer l'équation (I.4) par :

$$Prob \left\{ S_t = m / Y_1^L \right\} = \frac{\lambda_t(m)}{prob \{Y_1^L\}} \quad (I.8)$$

Et l'équation (I.5) par :

$$Prob \left\{ S_{t-1} = m', S_t = m / Y_1^L \right\} = \frac{\sigma_t(m',m)}{prob \{y_1^L\}} \quad (I.9)$$

Soit les fonctions de probabilités :

$$\alpha_t(m) = Pr(S_t = m, Y_1^t) \quad (I.10)$$

$$\beta_t(m) = prob(y_{t+1}^L / S_t = m) \quad (I.11)$$

$$\gamma_t(m, m') = Prob \{ S_t = m, Y_t / S_{t-1} = m' \} \quad (I.10)$$

*Bahl et al* montrent que

$$\lambda_t(m) = \alpha_t(m) \beta_t(m) \quad (I.13)$$

Et

$$\sigma_t(m',m) = \alpha_{t-1}(m') \gamma_t(m',m) \beta_t(m) \quad (I.14)$$

Et pour  $t=1$  à  $L$

$$\alpha_t(m) = \sum_{m'=0}^{M_{tr}-1} \alpha_{t-1}(m') \gamma_t(m',m) \quad (I.15)$$

$M_{tr}$  Représente le nombre d'états de la chaîne de Markov

Les conditions aux limites (initiales) sont :

$$\alpha_0(0) = 1 \text{ et } \alpha_0(m) = 0 \text{ pour } m \neq 0 \quad (I.16)$$

Et aussi, pour  $t=1$  à  $L$ , *Bahl et al.* [8] montrent que les  $\beta$  sont donnés par:

$$\beta_0(m) = \sum_{m'=0}^{M_{tr}-1} \beta_{t+1}(m) \gamma_{t+1}(m',m) \quad (I.17)$$

## Chapitre I : Codage et égalisation

---

La terminaison de la source est forcée à 0, alors :

$$\beta_L(0) = 1 \text{ et } \beta_L(m) = 0 \text{ pour } m \neq 0 \quad (\text{I.18})$$

$(m', m)$  dépend du canal:

$$\begin{aligned} \gamma_t(m', m) &= \text{prob}\{S_t = m, Y_t/S_{t-1}=m'\} \\ &= \sum_X \text{prob}\{S_t = m/S_{t-1} = m'\} \cdot \text{prob}\{X_t = X/S_{t-1} = m', S_t m\} \cdot \text{prob}\left\{\frac{Y_t}{X}\right\} \end{aligned}$$

Ou

$$= \sum_X p_t(m / m') q(X / m', m) R(Y_t, X) \quad (\text{I.19})$$

$$P_t(m / m') = \text{prob}\{S_t = m/S_{t-1} = m'\} \quad (\text{I.20})$$

Généralement, les symboles sont équiprobables, par conséquence les probabilités de transition sont les mêmes. Alors  $P_t(m/m')$  est constante [6]

De plus :

$$P_t(x/m', m) = \begin{cases} \mathbf{1} & \text{Si } m' \Rightarrow m \text{ donne } X \\ \mathbf{0} & \text{Ailleur} \end{cases} \quad (\text{I.21})$$

Et finalement,  $R(Y_t, X)$  est gaussien pour un canal gaussien de moyenne  $X$ , où  $X$  est le symbole transmis.

En se basant sur la définition de ces métriques, la probabilité *a posteriori* à l'instant  $t$  que le symbole soit égal à 0 est donnée par la relation :

$$\text{Prob}(x_t = 1) = \sum_{T_r=1} \alpha_{i-1}(m') \gamma_i(m', m) \beta_i(m) \quad (\text{I.22})$$

Et de la même façon, on exprime la probabilité que le symbole ci est égale à 1 par :

$$\text{Prob}(x_t = 0) = \sum_{T_r=0} \alpha_{i-1}(m') \gamma_i(m', m) \beta_i(m) \quad (\text{I.23})$$

Finalement, on peut évaluer le log rapport de ressemblance LRV (ou LLR log likelihood ratio) *a posteriori* sur le symbole  $x_t$  à l'instant  $i$  par :

$$\text{LLR}(x_t) = \log \frac{p_1(x_t=1)}{p_0(x_t=0)} \begin{cases} \geq 1 \\ \leq 0 \end{cases} \quad (\text{I.24})$$

**I.7. Déroulement de L'algorithme MAP**

1) Initialisation

$$\alpha_0(0) = 1 \text{ et } \alpha_0(m) = 0 \text{ pour } m \neq 0 \quad (\text{I.25})$$

$$\beta_0(0) = 1 \text{ e } \beta_0(m) = 0 \text{ pour } m \neq 0 \quad (\text{I.26})$$

2) L'échantillon  $Y_t$  est reçu, on calcule :

$$\gamma_t(m', m) = R(Y_t, X) \quad (\text{I.27})$$

$$\alpha_t(m) = \alpha_{t-1}(m') \gamma_t(m', m) \text{ de } t = 1 \text{ à } L \quad (\text{I.28})$$

3) Fin de la réception de la séquence  $Y_1^L$ , on calcule

$$\beta_t(m) = \sum_{m=0}^{Mtr-1} \beta_{t+1}(m') \gamma_{t+1}(m', m) \text{ de } t = L - 1 \quad (\text{I.29})$$

4) Calcul du rapport de vraisemblance LR (Likelihood Ratio) :

$$LR(X_t) = \frac{\text{prob}(X_t=1/Y_t^L)}{\text{prob}(X_t=0/Y_t^L)} \quad (\text{I.30})$$

$$= \frac{\sum_{X_t=1} \text{prob}\{m', m | Y_1^L\} \frac{1}{\text{prob}(Y_1^L)}}{\sum_{X_t=0} \text{prob}\{m', m | Y_1^L\} \frac{1}{\text{prob}(Y_1^L)}}$$

$$= \frac{\sum_{X_t=1} \sigma_t(m', m)}{\sum_{X_t=0} \sigma_t(m', m)}$$

$$= \frac{\sum_{X_t=1} \alpha_{t-1}(m') \gamma_t(m', m) \beta_t(m)}{\sum_{X_t=0} \alpha_{t-1}(m') \gamma_t(m', m) \beta_t(m)}$$

Pour  $N=i : L$

$$5) \text{ Si le } LR(X_t) > 1 \text{ alors } X_t = 1 \quad (\text{I.31})$$

$$\text{Sinon } LR < 1 \text{ alors } X_t = 0 \quad (\text{I.32})$$

On peut aussi utiliser le logarithme du rapport de vraisemblance (Logarithme Likelihood Ratio) LLR. Dans ce cas, le seuil de décision devient 0 (log(1)), alors :

$$\text{Si le } LLR > 0 \text{ alors } X_t = 1 \quad (\text{I.33})$$

$$\text{Si non } LLR < 0 \text{ alors } X_t = 0 \quad (\text{I.34})$$

### I.8. Décodage avec l'algorithme MAP [7]

Un décodeur MAP fournit, pour chaque symbole codé  $d_i$  la probabilité a posteriori (avec l'entière connaissance de la séquence  $y$  reçue par le décodeur).. La décision dure correspondante est la valeur  $j$  qui maximise la probabilité a posteriori. Ces probabilités peuvent s'exprimer en fonction des vraisemblances conjointes  $p(d_i=j, y)$  :

$$\Pr(d_i \equiv j|y) = \frac{p(d_i \equiv j, y)}{\sum_{k=0}^{2^m-1} p(d_i \equiv k, y)} \quad (I.35)$$

Or la structure en treillis du code permet de décomposer le calcul des vraisemblances conjointes en séparant les observations sur le passé des observations sur le futur. Cette décomposition utilise ainsi une métrique récurrente aller  $\alpha_i(s)$  (forward) pour déterminer la vraisemblance d'un état du treillis à l'instant  $i$  selon son passé, une métrique récurrente retour  $\beta_i(s)$  (backward) pour déterminer la vraisemblance d'un état du treillis à l'instant  $i$  selon son futur, et une métrique mesurant la vraisemblance d'une branche entre deux états du treillis  $\gamma_i(s', s)$ .

Les métriques récurrentes allées et retour sont calculés de la manière suivante :

$$p(d_i \equiv j, y) = \sum_{(s', s)/d_i \equiv j} \beta_{i+1}(s) \alpha_i(s') \gamma_i(s', s) \quad (I.36)$$

$$\alpha_{i+1}(s) = \sum_{s'=0}^{2^v-1} \alpha_i(s') \gamma_i(s', s) \quad \text{Pour } i = 0 \dots N - 1 \quad (I.37)$$

L'initialisation de ces métriques dépend respectivement de la connaissance de l'état de départ et de l'état final du treillis. Par exemple, la connaissance de l'état de départ  $S_0$  du codeur permet d'initialiser  $\alpha_0(S_0)$  à 1 et les autres  $\alpha_0(s)$  à 0. Si l'état est inconnu, on initialise toutes les métriques à la même valeur. Par ailleurs, la métrique de vraisemblance d'une branche s'exprime :

$$\beta_i(s) = \sum_{s'=0}^{2^v-1} \beta_{i+1}(s') \gamma_i(s', s) \quad \text{Pour } i = 0 \dots N - 1 \quad (I.38)$$

Avec  $p(y_i|x_i)$  la probabilité de transition du canal, qui s'exprime dans le cas d'un canal gaussien comme :

$$\gamma_i(s', s) = p(y_i|x_i) \cdot Pr^\alpha(d_i = d_i(s', s)) \quad (I.39)$$

Où  $x_i$  est le vecteur de symboles modulés.

La probabilité a priori d'émettre le m-uplet d'information correspondant à la transition de  $s'$  vers  $s$   $Pr^\alpha(d_i = d_i(s', s))$  vaut 0 si la transition n'existe pas dans le treillis. Sinon sa valeur est dépendante de la statistique de la source. Pour une source équiprobable, on a,  $Pr^\alpha(d_i = j) = \frac{1}{2^m}$ . Dans le cadre d'un décodage itératif, l'information a priori tient

également compte de l'information extrinsèque entrante. L'information extrinsèque générée par le décodeur est simplement l'information a posteriori avec une métrique de branche modifiée :

$$p(y_i|x_i) = \prod_{k=1}^n \left( \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(y_{i,k}-x_{i,k})^2}{2\sigma^2}} \right) = K \cdot e^{\frac{\sum_{k=1}^n y_{i,k} \cdot x_{i,k}}{\sigma^2}} \quad (I.40)$$

La métrique de branche ne tient alors plus compte des informations déjà disponibles dans le décodeur auquel l'information extrinsèque est destinée. Pour un turbocode convolutif concaténé en parallèle, il s'agit de retirer la partie systématique et la nouvelle métrique de branche s'exprime :

$$\text{Pr}^{\text{ex}}(d_i \equiv j|y) = \frac{\sum_{(s',s)/d_i \equiv j} \beta_{i+1}(s) \alpha_i(s') \gamma_i^{\text{ex}}(s',s)}{\sum_{(s',s)} \beta_{i+1}(s) \alpha_i(s') \gamma_i^{\text{ex}}(s',s)} \quad (I.41)$$

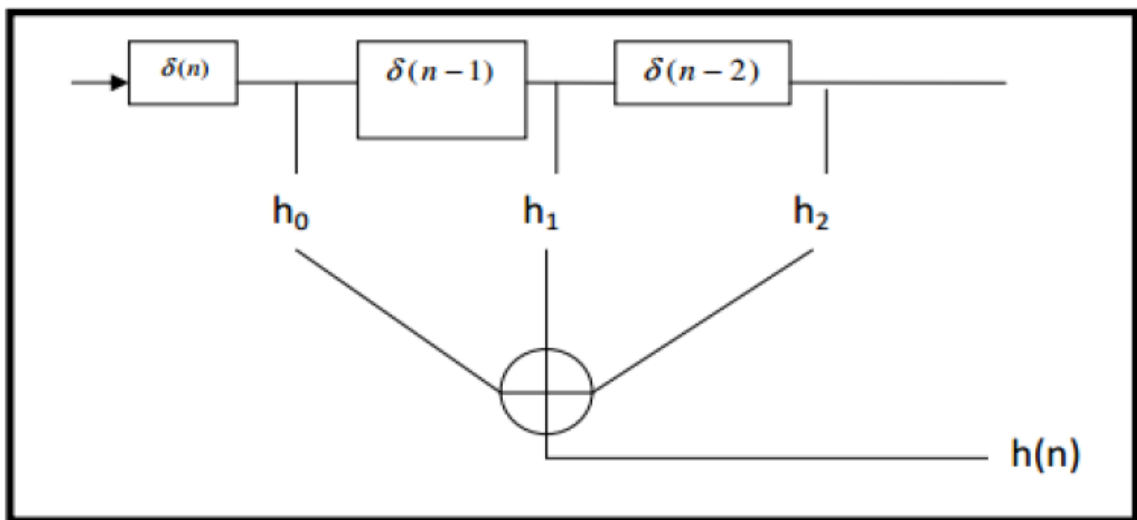
### I.9 L'algorithme MAP-BCJR appliqué à l'égalisation [8]

Soit la réponse impulsionnelle  $h(n)$  du canal discret équivalent :

$$H(n) = h_0(n) + h_1 \delta(n-1) + h_2(n-2) \quad (I.42)$$

Puisque la longueur du canal est 3, sa mémoire est égale à 2. Par conséquent, le nombre d'états (possibilités) du canal est égal à  $M^2$ . Supposons qu'une modulation BPSK est utilisée ( $M=2$  est le nombre d'états). Donc, le nombre d'états du canal est  $M^2 = 4$  états. Ces états sont  $(-1, -1)$ ,  $(-1, +1)$ ,  $(+1, -1)$  et  $(+1, +1)$ .

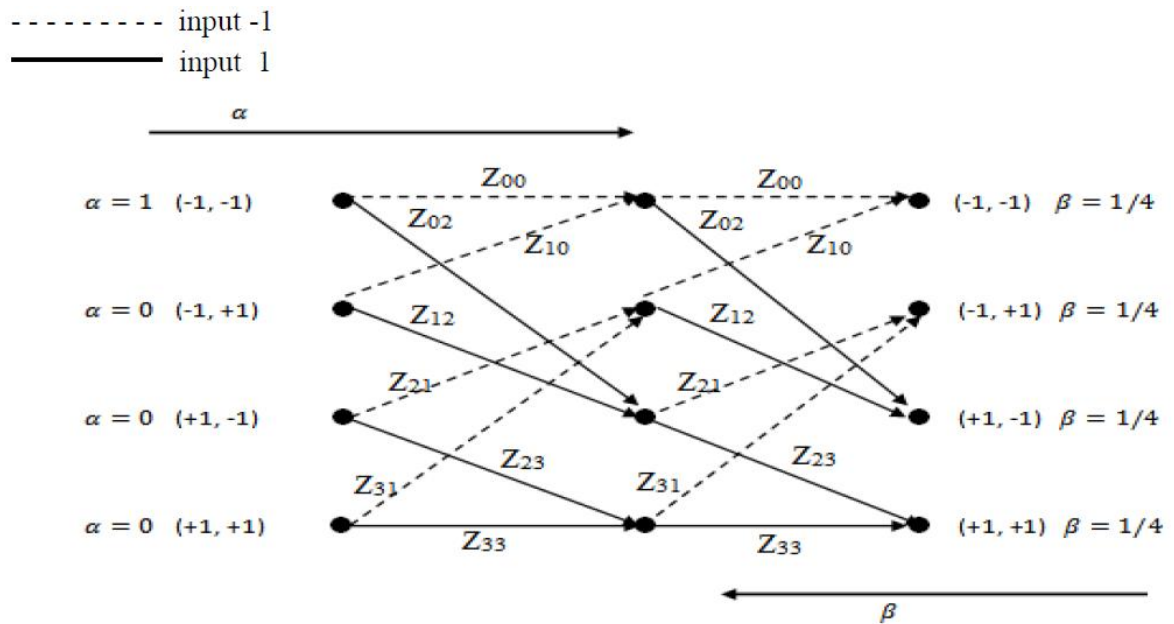
L'émission d'un symbole provoque une transition de la mémoire du canal d'un état à un autre. Les états et les transitions constituent ainsi un treillis du canal. La (Figure I.12) montre le treillis à 4 états.



**Figure I. 12:** Structure transversale du filtre RIF.

**I.9.1. Diagramme en treillis**

Il montre toutes les transitions possibles entre les états du treillis à chaque inslaut



**Figure I. 13 :** Diagramme en treillis d'un canal de longueur3

**I.9.2. Calcul des Poids des Branches**

Le canal est modélisé par un filtre linéaire, par conséquent, les différents Poids  $Z_m$ , des branches sont donnés par :

$$Z_{00} = (-1) h_0 + (-1) h_1 + (-1) h_2 \quad (I.43)$$

$$Z_{02} = (1) h_0 + (-1) h_1 + (-1) h_2 \quad (I.44)$$

$$Z_{10} = (-1) h_0 + (-1) h_1 + (1) h_2 \quad (I.45)$$

$$Z_{12} = (1) h_0 + (-1) h_1 + (1) h_2 \quad (I.46)$$

$$Z_{21} = (-1) h_0 + (1) h_1 + (-1) h_2 \quad (I.47)$$

$$Z_{23} = (1) h_0 + (1) h_1 + (-1) h_2 \quad (I.48)$$

$$Z_{31} = (-1) h_0 + (1) h_1 + (1) h_2 \quad (I.49)$$

$$Z_{33} = (1) h_0 + (1) h_1 + (1) h_2 \quad (I.50)$$



### I.9.3. Calcul des $\gamma$

Pour calculer le rapport de l'équation (I.32), il faut tout d'abord calculer les probabilités de transition  $\gamma$ , puis les probabilités  $\alpha$  et  $\beta$ . Puisque le treillis de la (figure I.12) est formé de huit branches pour chaque instant, donc, il faut calculer huit probabilités de transition  $\gamma$ . Chaque probabilité dépend de l'état de départ et de l'état d'arrivée. A chaque instant 'i', un échantillon 'yi' est reçu. On calcule les huit '8' probabilités de transition  $\gamma$ . On obtient :

Pour i variant de 1 à L :

$$\gamma_{00}(i) = \exp\left(-\frac{(y(i)-z_{00})^2}{2\sigma^2}\right) \quad (I.51)$$

$$\gamma_{02}(i) = \exp\left(-\frac{(y(i)-z_{02})^2}{2\sigma^2}\right) \quad (I.52)$$

$$\gamma_{10}(i) = \exp\left(-\frac{(y(i)-z_{10})^2}{2\sigma^2}\right) \quad (I.53)$$

$$\gamma_{12}(i) = \exp\left(-\frac{(y(i)-z_{12})^2}{2\sigma^2}\right) \quad (I.54)$$

$$\gamma_{21}(i) = \exp\left(-\frac{(y(i)-z_{21})^2}{2\sigma^2}\right) \quad (I.55)$$

$$\gamma_{23}(i) = \exp\left(-\frac{(y(i)-z_{23})^2}{2\sigma^2}\right) \quad (I.56)$$

$$\gamma_{31}(i) = \exp\left(-\frac{(y(i)-z_{31})^2}{2\sigma^2}\right) \quad (I.57)$$

$$\gamma_{33}(i) = \exp\left(-\frac{(y(i)-z_{33})^2}{2\sigma^2}\right) \quad (I.55)$$

Fin pour.

### I.9.4. Calcul des $\alpha$

D'après le treillis de la (figure I.13), et l'équation (I.33), les probabilités  $\alpha$  sont calculées.

Pour i variant de 1 à L :

$$\alpha_0(i) = \alpha_0(i-1) \gamma_{00}(i) + \alpha_1(i-1) \gamma_{10}(i) \quad (I.56)$$

$$\alpha_1(i) = \alpha_2(i-1) \gamma_{21}(i) + \alpha_3(i-1) \gamma_{31}(i) \quad (I.57)$$

$$\alpha_2(i) = \alpha_0(i-1) \gamma_{02}(i) + \alpha_1(i-1) \gamma_{12}(i) \quad (\text{I.58})$$

$$\alpha_3(i) = \alpha_2(i-1) \gamma_{23}(i) + \alpha_3(i-1) \gamma_{33}(i) \quad (\text{I.59})$$

Fin pour.

### I.9.5. Calcul des $\beta$

De la même façon, d'après le treillis de la figure (I.13), et l'équation (I.34), les probabilités  $\beta$  sont calculées par :

Pour  $i$  variant de 1 à L :

$$\beta_0(i) = \beta_0(i+1) \gamma_{00}(i+1) + \beta_2(i+1) \gamma_{02}(i+1) \quad (\text{I.60})$$

$$\beta_1(i) = \beta_0(i+1) \gamma_{10}(i+1) + \beta_2(i+1) \gamma_{12}(i+1) \quad (\text{I.61})$$

$$\beta_2(i) = \beta_1(i+1) \gamma_{21}(i+1) + \beta_3(i+1) \gamma_{23}(i+1) \quad (\text{I.62})$$

$$\beta_3(i) = \beta_1(i+1) \gamma_{31}(i+1) + \beta_3(i+1) \gamma_{33}(i+1) \quad (\text{I.63})$$

Fin pour

### I.9.6. Calcul des Probabilités des Symboles

#### I.9.6.1. Calcul de la Probabilité Du Symbole +1 à l'instant 'i'

A chaque instant, la probabilité du symbole +1 est donnée par la somme des probabilités des transitions provoquées par une entrée +1 au canal. (Les 4 branches en trait continu du treillis de la figure (I.13). C'est-à dire:

$$P_1(i) = \alpha_0(i-1) \gamma_{02}(i) \beta_2(i) + \alpha_1(i-1) \gamma_{12}(i) \beta_2(i) + \alpha_2(i-1) \gamma_{23}(i) \beta_3(i) + \alpha_3(i-1) \gamma_{33}(i) \beta_3(i) \quad (\text{I.64})$$

#### I.9.6.2. Calcul de la probabilité du symbole -1 à l'instant 'i'

De la même façon, la probabilité du symbole -1 est donnée par:

$$P_0(i) = \alpha_0(i-1) \gamma_{00}(i) \beta_0(i) + \alpha_1(i-1) \gamma_{10}(i) \beta_0(i) + \alpha_2(i-1) \gamma_{21}(i) \beta_1(i) + \alpha_3(i-1) \gamma_{31}(i) \beta_1(i) \quad (\text{I.65})$$

### I.9.7. Calcul du Log Rapport de Vraisemblance LLR

Il est obtenu en calculant le rapport:

Pour  $i$  variant de 1 à  $L$  :

$$LLR(i) = \log\left(\frac{P_i(i)}{P_0(i)}\right) \quad (\text{I.66})$$

Fin pour.

### I.10. Détection

Après l'obtention des LLR des symboles émis, on peut prendre une décision. D'après les équations (I.36 et I.37), le seuil de détection doit être 0. C'est-à-dire:

Pour  $i$  variant de 1 à  $L$  :

$$\text{Si } LLR(i) > 0 \quad \hat{x}(i) = 1 \quad (\text{I.67})$$

$$\text{Sinon} \quad \hat{x}(i) = 0 \quad (\text{I.68})$$

### I.11. Conclusion

Dans ce chapitre, nous sommes intéressés au codage et égalisation, nous avons présenté les codes convolutifs, problème des IES et égalisation. Ensuite nous avons étudiés l'algorithme MAP (appliqué au décodage et à l'égalisation). Il sera question dans le chapitre suivant de voir les Turbo récepteurs.



# Chapitre II : Les turbo récepteurs

### II.1. Introduction

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. Depuis leur invention, en 1990, les turbo codes ont fait l'objet de brevets et de nombreuses publications. Ils ont suscité l'intérêt de la communauté scientifique, notamment pour les applications de diffusion numérique de télévision, par voie hertzienne et satellite, et plus généralement pour les futures autoroutes de l'information [9].

Ce chapitre est consacré aux Turbo récepteurs. Nous nous sommes intéressés aux principes turbo, turbo codage MAP et la turbo égalisation MAP.

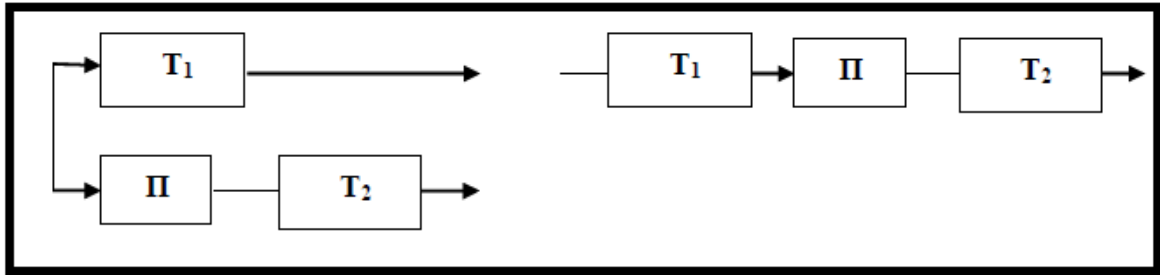
### II.2. Principe turbo

Le principe « Turbo » a été tout d'abord proposé dans le cas du décodage itératif pour des codes concaténés en parallèle (Turbo codes) puis pour des codes concaténés en série (Turbo Codes produit). Nous pouvons proposer une définition plus générale du principe Turbo à partir de la définition des Turbo codes de Berrou [10] :

#### II.2.1. Définition

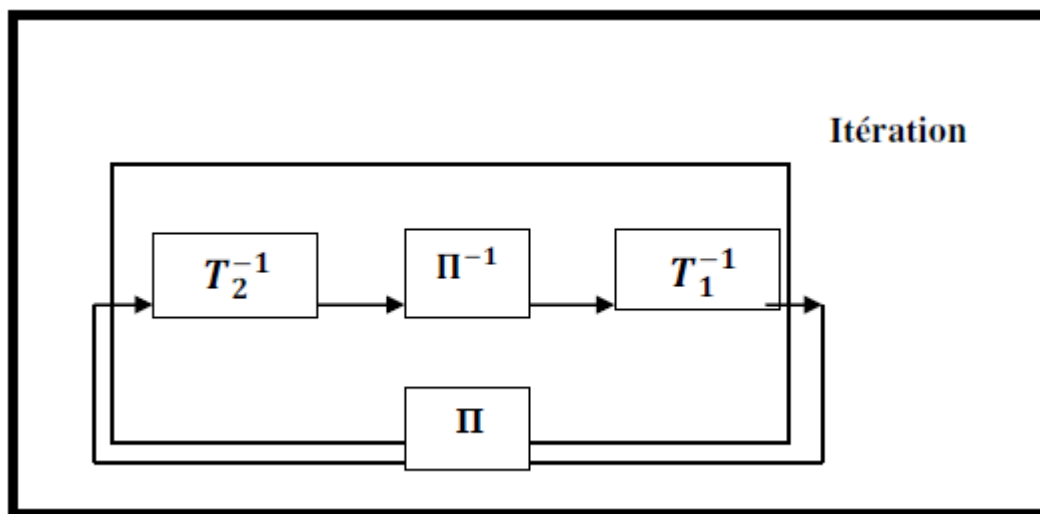
Une technique de réception « Turbo » est une structure itérative, où les données sont traitées de manière pondérée, par au moins deux algorithmes qui agissent « différemment » [10].

Précisons le sens d'agir « différemment », la différence peut provenir du fait que les algorithmes fournissent des données de natures différentes, comme dans le cas de l'estimation et de l'égalisation conjointe [1]. Si en revanche, les algorithmes traitent les mêmes données et fournissent des estimations sur les mêmes données, il est important de décorréler ces données d'un traitement à l'autre (cas des turbo codes ou de l'égalisation et du décodage conjoints), la différence provient dans ce cas de l'utilisation d'un entrelaceur (**II**) entre les deux traitements **T1** et **T2**, comme le montrent les figures II.1 et II.2, Ainsi, l'émetteur peut être parallèle ou série (figure II.1).



**Figure II. 1** : principe turbo et émetteur parallèle ou série.

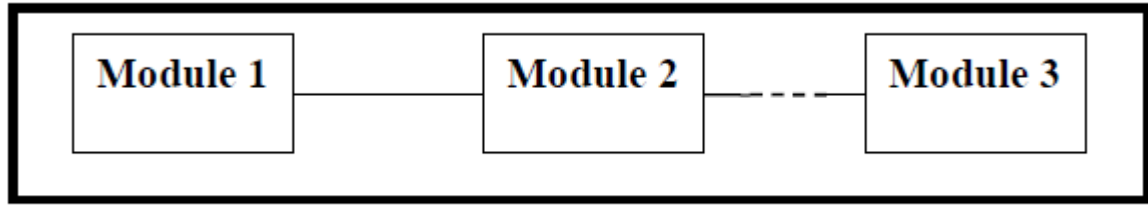
Quant au récepteur, il devient :



**Figure II. 2** : principe Turbo récepteur [8].

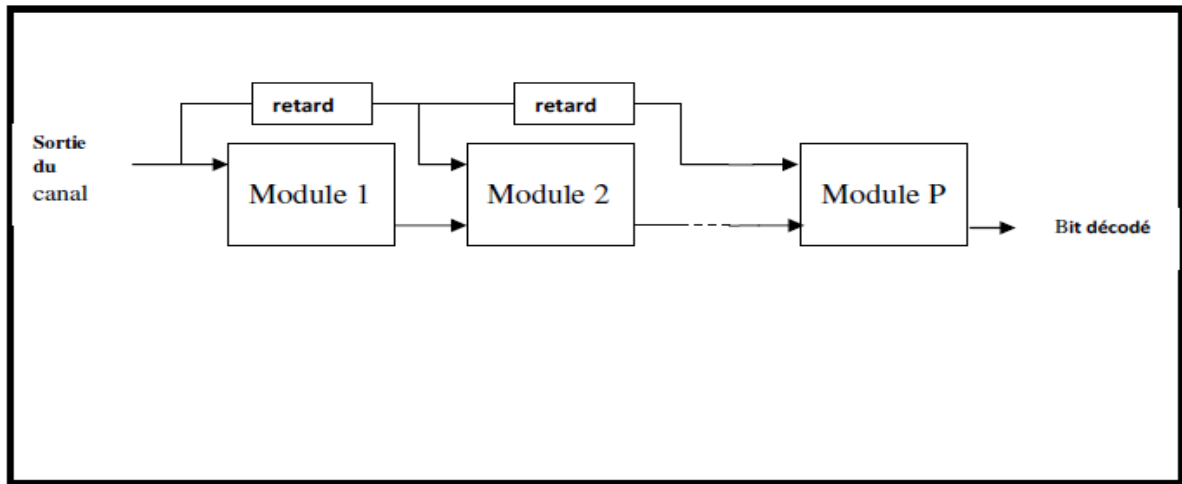
$T_1^{-1}$  et  $T_2^{-1}$  sont les traitements inverses des traitements réalisés en émission.  $\Pi$  et  $\Pi^{-1}$  représentent respectivement l'entrelaceur et le désentrelaceur associé. Les données sont supposées entrelacées à l'émission. Chaque couple de traitement  $(T_1^{-1}, T_2^{-1})$  est appelé itération. Le turbo récepteur est donc un traitement itératif où le résultat de l'itération  $P^{-1}$  est utilisé par l'itération 'p' suivante [8].

Le Turbo récepteur est donc un traitement itératif, où le résultat de l'itération  $P^{-1}$  est utilisé à l'itération p suivante, à ce récepteur correspond donc un schéma modulaire présenté (figureII.3).



**Figure II. 3 :** *principe itératif du Turbo égaliseur.*

Les deux traitements sont réalisés de manière conjointe. Ce schéma itératif permet de régler le problème des retards dus à l'entrelaceur, et aux traitements  $T_1^{-1}, T_2^{-1}$ . La figure (II.4) montre les schémas d'un récepteur itératif. Chaque module constitue une itération [11].



**Figure II. 4 :** Traitement itératif.

### II.3. Traitement Turbo et information extrinsèque

Nous présentons le principe Turbo dans le cas général. La probabilité à posteriori d'un événement '**A**' sachant la réalisation de '**B**' est donnée par la vraisemblance [12] :

$$p\left(\frac{A}{B}\right) = \frac{p(B/A)p(A)}{p(B)} \quad (\text{II.1})$$

Dans les applications d'ingénierie des systèmes numériques, on s'intéresse à la Probabilité A Posteriori l'APP :

$$P\left(d = i/r\right) = \frac{p(r/d=i)p(d=1)}{p(r)} \quad (\text{II.2})$$

## Chapitre II : Les turbo récepteurs

---

Où ' $d$ ' désigne le symbole émis, ' $r$ ' le symbole reçu, et ' $i$ ' représente le  $i^{\text{ème}}$  signal ( $i=1$  à  $M$ ) appartenant à la classe des signaux de la modulation choisie.

L'APP maximale donne le signal le plus probable transmis. Ce critère minimise la probabilité d'erreur qui est l'objectif principal pris en considération dans la conception des systèmes de transmission numérique. Par exemple, dans le cas binaire (deux signaux), nous comparons les deux APP, puis nous prenons une décision :

$$P(d = +1/r) \geq p(d = -1/r) \quad (II.3)$$

Cette décision est appelée « décision basée sur le critère MAP ». En utilisant (II.3), on obtient :

$$\frac{p(d=+1/r)}{p(d=-1/r)} \geq 1 \quad (II.4)$$

En développant le premier terme et en introduisant le logarithme népérien, on obtient :

$$\begin{aligned} L\left(\frac{d}{r}\right) &= \log\left(\frac{p(d=+1/r)}{p(d=-1/r)}\right) \quad (II.5) \\ &= \log\left(\frac{p(r/d = +1) \cdot p(d = +1)}{p(r/d = -1) \cdot p(d = -1)}\right) \\ &= \log\left(\frac{p(r/d = +1)}{p(r/d = -1)}\right) + \log\left(\frac{p(d = +1)}{p(d = -1)}\right) \end{aligned}$$

On désigne ces quantités par [13]:

$$L\left(\frac{d}{r}\right) = L_c\left(\frac{r}{d}\right) + L(d) \quad (II.6)$$

Où  $L(r/d)$  est le logarithme du rapport de vraisemblance 'LLR' (Logarithme of Likelihood Ratio) des observations ' $r$ ' en sortie du canal sachant les réalisations de  $d = +1$  ou  $d = -1$  supposés transmis.  $L(d)$  est le LLR à priori des symboles d'information. L'équation (I.6) peut s'écrire sous la forme :

$$L'(\hat{d}) = L_c(r) + L(d) \quad (II.7)$$

L'utilisation d'un codeur est obligatoire dans un système Turbo. Berrou montre que le LLR à la sortie du décodeur vaut :



## Chapitre II : Les turbo récepteurs

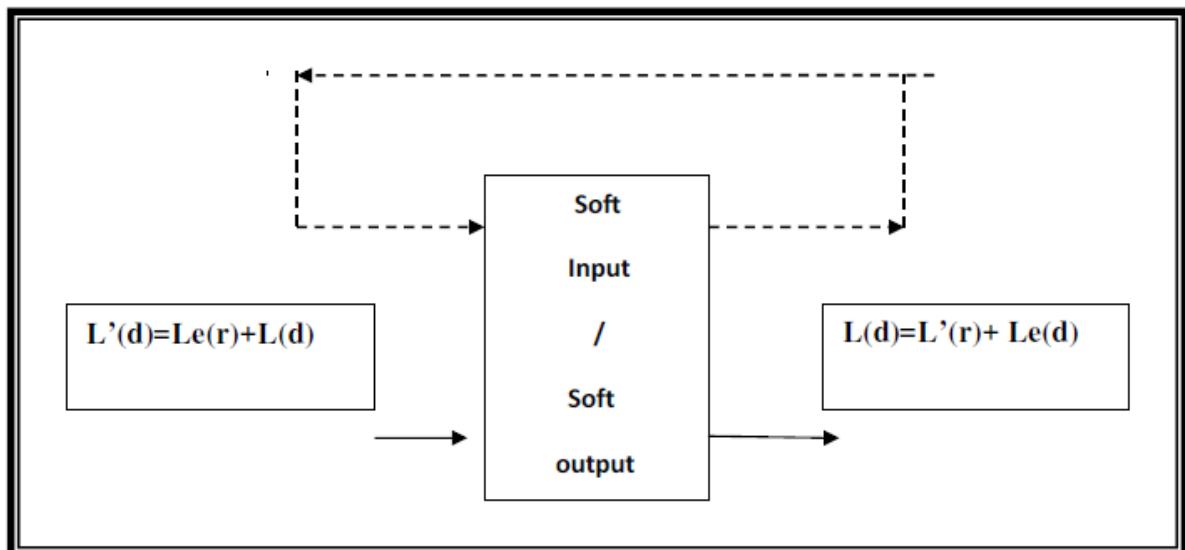
$$L(\hat{d}) = L'(\hat{d}) + L_e(\hat{d}) \quad (II.8)$$

Où  $L'(\hat{d})$  est le LLR d'un symbole de données à l'entrée du décodeur, et  $L_e(\hat{d})$  est appelée l'information extrinsèque. Elle représente la contribution du décodeur. (II.8) partitionne le LLR du décodeur en une portion de données représentées par les mesures à l'entrée et une partie extrinsèque représentée par la contribution du décodeur due à la redondance. En remplaçant l'équation (II.7) dans l'équation (II.8), on obtient :

$$L(\hat{d}) = L_c(r) + L_a(d) + L_e(\hat{d}) \quad (II.9)$$

Le signe de  $L(\hat{d})$  donne une décision sur les symboles et le module donne la fiabilité de cette décision [11].

Berrou et al. Affirment que  $L_e(\hat{d})$  a le même signe de  $\hat{d}$ . Elle peut être donc utilisée pour améliorer les LLR associés de chaque bit décodé. La sortie  $L(d')$  du décodeur est composée du LLR à l'entrée du décodeur  $L_e(d')$ , et de l'information extrinsèque  $L_e(\hat{d})$  qui constitue la contribution du décodeur. Dans un traitement itératif, l'information extrinsèque est réinjectée à l'entrée du traitement suivant pour servir comme un affinement des connaissances à priori pour la prochaine itération Figure (II.5). Pour la première itération du traitement Turbo, on n'a aucune information à priori sur les symboles, par conséquent, les données sont supposées équiprobables et LLR à priori est nul, c'est-à-dire  $L(d)=0$ .



**Figure II. 5:** Différentes informations générées par un traitement SISO [13].

Si on se place à la sortie du canal, LLR du symbole reçu ' $r_n$ ' à l'instant ' $n$ ',  $L_c(r_n)$  est donnée par :

$$\begin{aligned}
 L_c(r_n) &= \log_c \left[ \frac{p\left(\frac{r_n}{d_n}=+1\right)}{p\left(\frac{r_n}{d_n}=-1\right)} \right] && \text{(II.10)} \\
 &= \log_c \left[ \frac{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(r_b-1)^2}{2\sigma^2}\right)}{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(r_b+1)^2}{2\sigma^2}\right)} \right] \\
 &= \left(-\frac{(r_b-1)^2}{2\sigma^2}\right) - \left(-\frac{(r_b+1)^2}{2\sigma^2}\right) \\
 &= \frac{2}{\sigma^2} r_n
 \end{aligned}$$

$\sigma^2$  Est la variance du bruit à l'entrée du turbo récepteur.

Dans le cas de la concaténation de deux traitements, un récepteur classique n'est pas optimal. Par exemple, pour la concaténation de deux codeurs, le premier décodeur ne bénéficie pas de la redondance totale. Pour cela, on utilise une boucle de retour entre les deux traitements. Elle permet l'échange de ces informations par le mécanisme suivant

### II.4. Mécanisme Turbo

- 1- Les informations a priori  $L(d)=0$ .
- 2- Effectuer le traitement 1 (obtention de  $L(\hat{d})$ ) et obtenir l'information extrinsèque  $L_{e1}(\hat{d})$
- 3- Mettre  $L(d) = L_{e1}(\hat{d})$ . L'information extrinsèque devient à priori.
- 4- Effectuer le traitement 2 et obtenir  $L_{e2}(\hat{d})$ .
- 5- Mettre  $L(d) = L_{e2}(\hat{d})$ . L'information extrinsèque devient à priori. Revenir à 2.

### II.5. Turbo codes

Les Turbo-Codes, grâce à leurs excellentes capacités de correction d'erreurs, sont considérés comme une partie fondamentale des normes de télécommunication actuelles telles que **LTE**, **HSPA**, **DVB-S**. Le turbo-codeur est la concaténation parallèle de deux codeurs convolutifs séparés par un entrelaceur qui permute les séquences de bits afin de casser les relations de voisinage entre eux, ce qui permet d'accroître les capacités de correction d'erreurs.

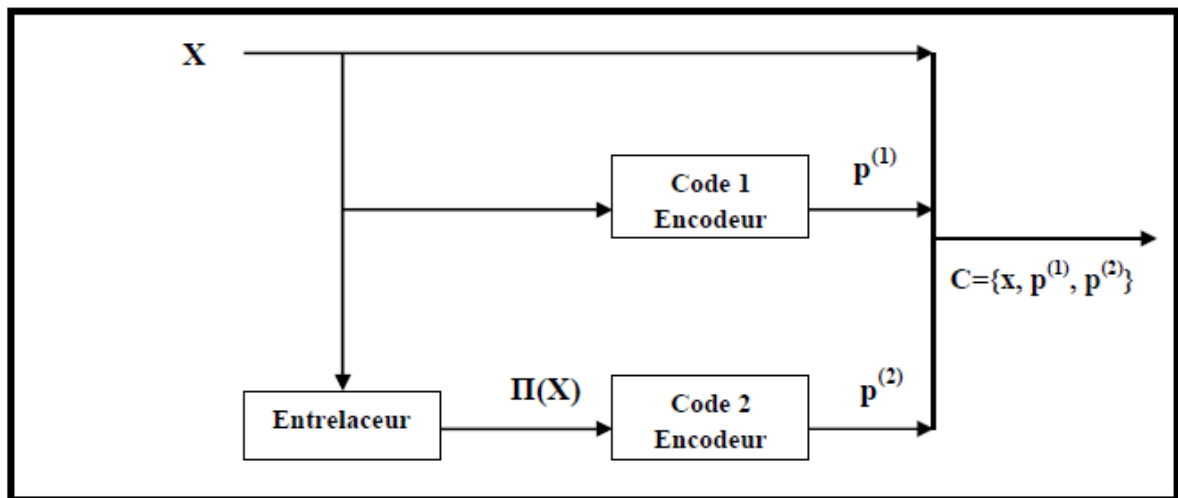
Le décodeur est constitué par la concaténation en série de deux décodeurs convolutifs qui sont séparés par deux entrelaceurs. Ces deux décodeurs partagent leurs informations de manière itérative afin de décoder le message reçu. Ce procédé itératif, introduit au niveau du décodage, permet d'obtenir des gains de performances considérables

## Chapitre II : Les turbo récepteurs

et de se rapprocher à la limite de Shannon. En outre, grâce à la faible complexité des algorithmes du décodage itératif [10].

### II.5.1. Turbo codeur

Un turbo-codeur est la concaténation de deux codeurs convolutifs : le premier encode les bits d'informations  $\mathbf{X}$  dans l'ordre naturel pour générer les bits de parité  $\mathbf{P}^{(1)}$ , tandis que le second encode les bits d'informations ( $\mathbf{X}$ ) dans l'ordre entrelacé (le message original est introduit dans l'entrelaceur), pour générer les bits de parités  $\mathbf{P}^{(2)}$ , comme c'est indiqué dans la Figure (II.6) [10].



**Figure II. 6 : Turbo-Codeur.**

Les Turbo-Codes étant des codes systématiques, le mot de code à la sortie du codeur est construit par la concaténation parallèle des bits d'information  $\mathbf{X}$  et des bits de parités générés par les deux codeurs ( $\mathbf{p}^{(1)}$  et  $\mathbf{p}^{(2)}$ ), Figure (II.6). Les codeurs utilisés dans les Turbo-Codes sont généralement identiques, mais il est également possible d'utiliser des codeurs différents. Toutefois, du fait de la présence d'un entrelaceur, les bits de parité générés par les deux codeurs sont toujours différents même si les codeurs sont identiques. (Figure II.6)[12].

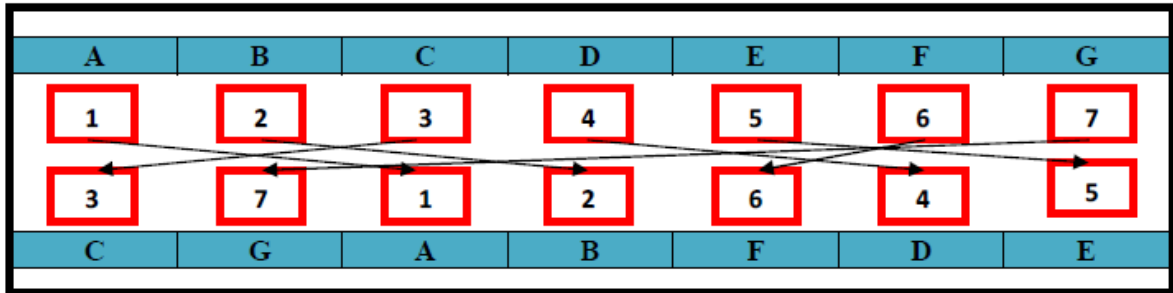
### II.5.2 Entrelaceur

L'entrelacement est exprimé à partir d'une séquence de permutation  $\Pi = \{\pi_1, \pi_2, \pi_3, \dots, \pi_n\}$  dont la séquence  $\{\pi_1, \pi_2, \pi_3, \dots, \pi_n\}$  représente la permutation des entiers de 1 jusqu'à  $n$ .

La fonction d'entrelacement consiste à générer deux bits de parités complètement différents une fois qu'ils sont introduits dans deux codeurs. La meilleure performance des

## Chapitre II : Les turbo récepteurs

Turbo-Codes est normalement réalisée par l'utilisation d'un entrelaceur qui effectue des permutations aléatoires pour des milliers de bits.



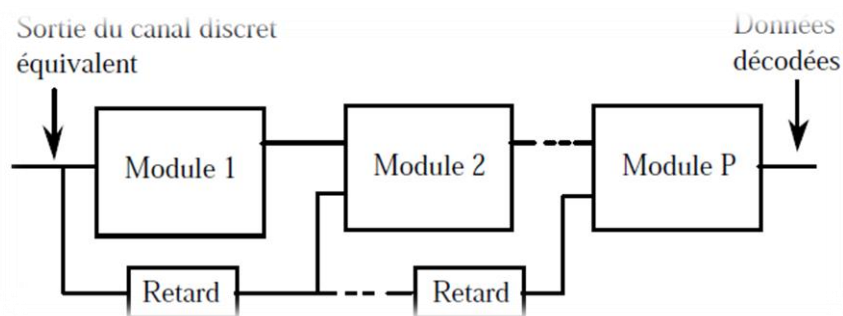
**Figure II.7 :** Fonction d'entrelacement.

La Figure (II.6) montre un exemple de règle d'entrelacement représenté sous la forme d'une séquence de données **A, B, C, D, E, F, G** qui sera entrelacée selon la règle de permutation représentée par les séquences numériques. Le résultat de cette permutation sera le suivant : la séquence de données après permutation est la suivante **C, G, A, B, F, D, E** [12].

### II.6. Structure de l'égaliseur

Le turbo-égaliseur égalise et décode à plusieurs reprises un même bloc d'information en utilisant les informations fournies par le traitement précédent.

On appelle module un traitement élémentaire d'un bloc d'information [14]



**Figure II. 8 :** Principe du turbo égaliseur.

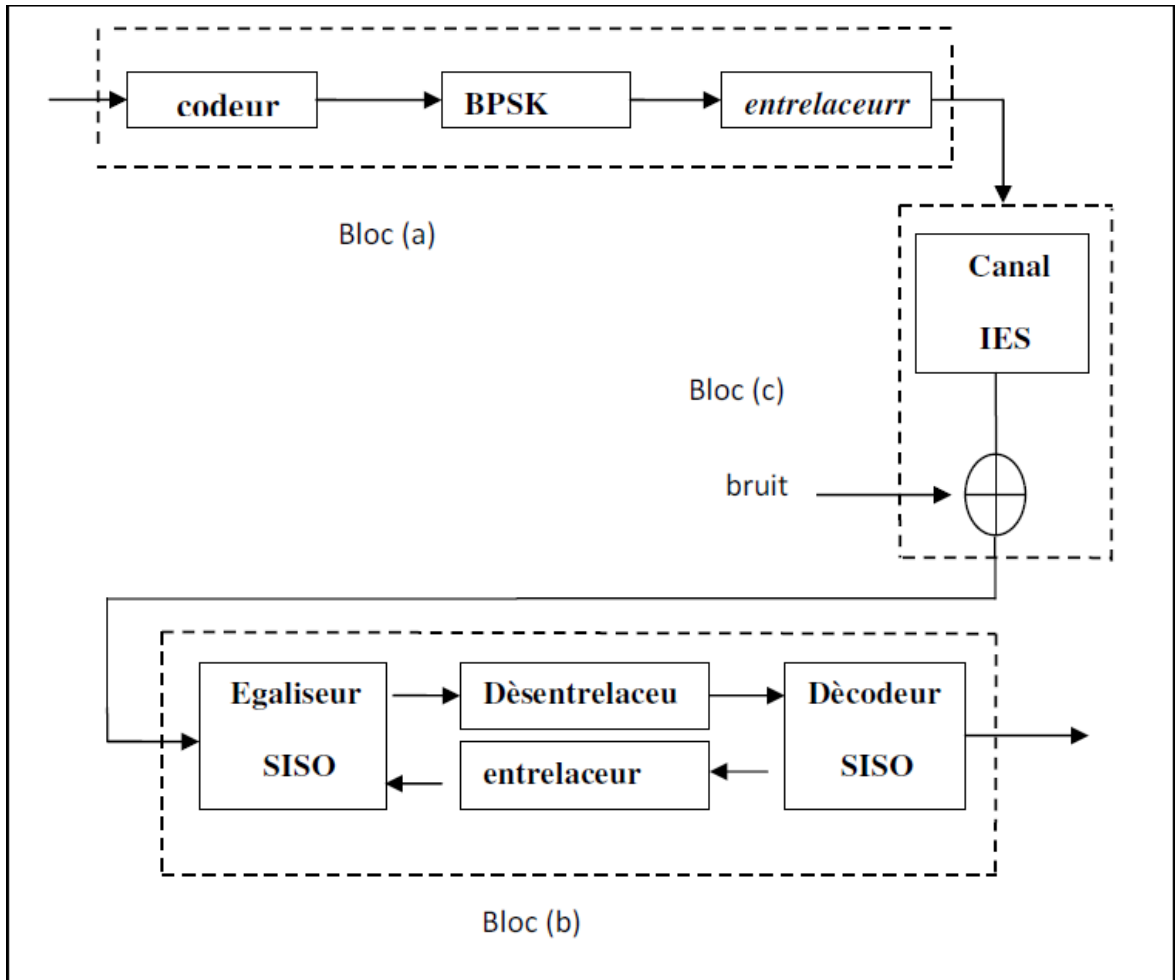
Chaque module  $p$  ( $p=1 \dots\dots\dots P$ ) comprend une fonction d'égalisation, un désentrelaceur, un convertisseur M-aire binaire, un décodeur de canal, un convertisseur binaire M-aire et un entrelaceur.

### II.7. Principe de la turbo-égalisation

Le concept de turbo-égalisation a vu le jour au sein des laboratoires de l'ENST Bretagne au début des années 90, sous l'impulsion des résultats spectaculaires obtenus avec les turbo-codes.

Il est né d'un constat très simple : le schéma de transmission de la figure (II .1) peut être vu comme la concaténation en série de deux codes, séparés par un entrelaceur, le second code étant formé par la mise en cascade de l'opération de mapping avec le canal. Vu sous cet angle, il semblait alors naturel d'appliquer une stratégie de décodage de type <<turbo>> en réception, c'est-à-dire un échange réciproque et itératif d'information probabiliste (information extrinsèque) entre l'égaliseur et le décodeur. [14]

Le premier schéma de turbo-égalisation a été proposé en 1995 par Douillard et al. Ce schéma met en œuvre un égaliseur de Viterbi à entrée et sortie pondérées (Soft Input Soft Output, ou SISO) selon l'algorithme SOVA (Soft Output Viterbi Algorithm) . Le principe a ensuite été repris en 1997 par Bauch et al, en substituant à l'égaliseur SOVA un égaliseur SISO optimal au sens du critère MAP, utilisant l'algorithme développé par et al. (Algorithme BCJK).



**Figure II .9 :** Architecture de base du récepteur itératif utilisant le principe turbo. [14]

La figure(II.9), montre respectivement la structure d'un émetteur en présence du décodeur de canal la figure (II.9(a)), la structure de base du récepteur itératif la figure (II.9(b)) et le modèle de canal la figure (II.9(c)) en présence de l'interférence entre symbole pour une modulation BPSK. La grande majorité des architectures de la turbo-égalisation décrites dans la littérature utilisent la même architecture de base, sauf pour le module de l'égaliseur SISO (Soft Input Soft Output), qui change d'une structure à l'autre.

Pour le décodeur, on utilise dans la plupart des cas un algorithme basé sur le critère de probabilité a posteriori maximale (MAP).

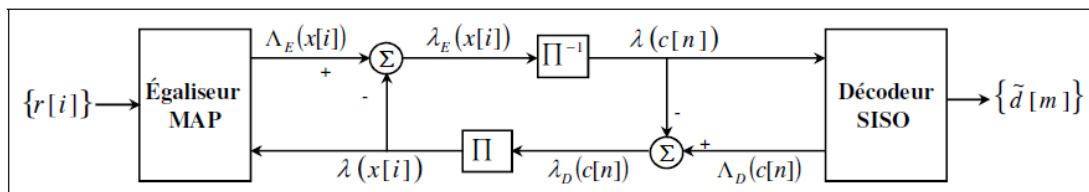
A la réception, une turbo-égaliseur figure (II.9b) est employée. Ce dernier consiste en trois fonctions : [14]

- **Egaliseur SISO:** suppression ou réduction de l'effet de l'interférence entre symboles (IES) introduite par le canal à trajets multiples.

- **Entrelacement /Désentrelacement des données :** dont le rôle consiste ici à casser les paquets d'erreurs en sortie de l'égaliseur (pour éviter un phénomène de propagation d'erreur), et à décorrélérer au maximum les données probabilistes échangées entre l'égaliseur et le décodeur de canal. Le turbo-égaliseur est alors capable de compenser entièrement la gégradation causée par l'IES, sous réserve que l'entrelaceur soit suffisamment grand et judicieusement construit.
- **Décodeur SISO :** La mise en œuvre d'un décodage itératif en réception, instaure un échange d'information probabiliste entre les traitements, dont on sait aujourd'hui qu'à partir d'un rapport signal sur bruit suffisant (appelé (« seuil de convergence »)), il converge vers les performances du récepteur optimal conjoint un certain nombre d'itérations.

### II.9. Turbo égalisation MAP

Dans cette architecture, l'annulation de l'IES est assurée par un détecteur à maximum a posteriori (MAP). Le principe de cette architecture est que le canal multi traject s'apparente à un code convolutif de rendement 1. La détection optimale, selon le critère du maximum de vraisemblance, des symboles générés par le canal peut être réalisée en appliquant l'algorithme MAP sur ce treillis. Le schéma de principe de cette architecture est donné à la figure (II.10). [14]



**Figure II. 10 :** Turbo égaliseur utilisant le critère MAP. [15]

D'une manière générale, le turbo-égaliseur correspondant au scénario de transmission de la figure (II.9) prend la forme représentée en figure (II.10).

Il se compose en premier lieu d'un égaliseur SISO, qui prend en entrée à la fois le vecteur  $r[i]$  des données observées à la sortie du canal, et une information probabiliste a priori sur l'ensemble des bits codés et entrelacés

### **II.11. Conclusion**

Dans ce chapitre nous sommes intéressés aux turbo récepteurs, leur principe, turbo codage MAP, et turbo égalisation MAP.

Après avoir présenté les turbo récepteurs, nous présentons dans le chapitre suivant le couplage turbo égaliseur – turbo décodeur.





Chapitre III :  
Couplage turbo égaliseur - turbo décodeur

### III.1. Introduction

Dans le domaine de la théorie du codage et de la communication numérique, les codes turbo ont fait le développement le plus excitant des dernières années. Le turbo décodeur fonctionne bien lorsque les interférences entre symboles IES sont absentes, c'est-à-dire, lorsque canal est à bande infinie. Cependant, les performances se dégradent rapidement en présence des IES. Un égaliseur est donc nécessaire.

Dans cette chapite nous présentons une qui combine un turbo égaliseur avec turbo décodeur.

### III.2. Modèle du système

Les bits d'information binaires  $\{d_i\}$  sont codés par un turbo codeur convolutif. Il est constitué de deux codeurs récursifs systématiques identiques, séparés par un entrelaceur ( $\pi$ ) pseudo aléatoire. Les bits codés  $\{c_i\}$  sont des blocs entrelacés, modulés en BPSK et transmis sur une canal sélectif en fréquence (avec des nterérences entre symboles IES). Les symboles BPSK seront appelés  $\{c_k\}$ . [16]

### III.3. Décodeur

Avant de présenter la structure du décodeur, nous définissons d'abord le bloc MAP illustré sur la figure (III.1). C'est un décodeur SISO (Soft Input Soft Output) MAP. Le treillis considéré peut être celui du code convolutif ou celui du canal avec IES. Pour un codeur systématique ayant des données systématiques et des données codées, le bloc MAP a les entrées et sorties de la figure (3.1). L'information a priori ( $L_a$ ) sur les données bits d'information est donné par [16]

$$L_a(\alpha_i) = \log \frac{\text{Prob}(\alpha_i=+1)}{\text{Prob}(\alpha_i=-1)} \quad (\text{III.1})$$

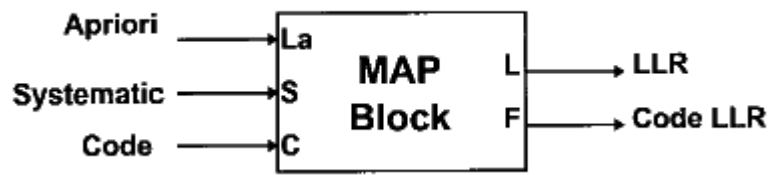


Figure III. 1: Un bloc MAP.

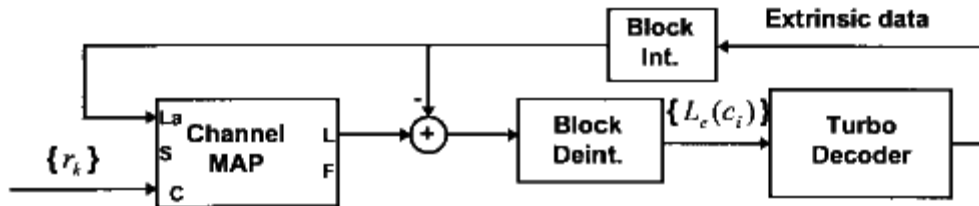


Figure III. 2: Structure générale du décodeur combiné.

Les informations extrinsèques sont extraites après chaque itération. Les LLR des bits codés seront utilisés par l'égaliseur MAP comme informations a priori dans l'itération suivante. (Figure III.3)

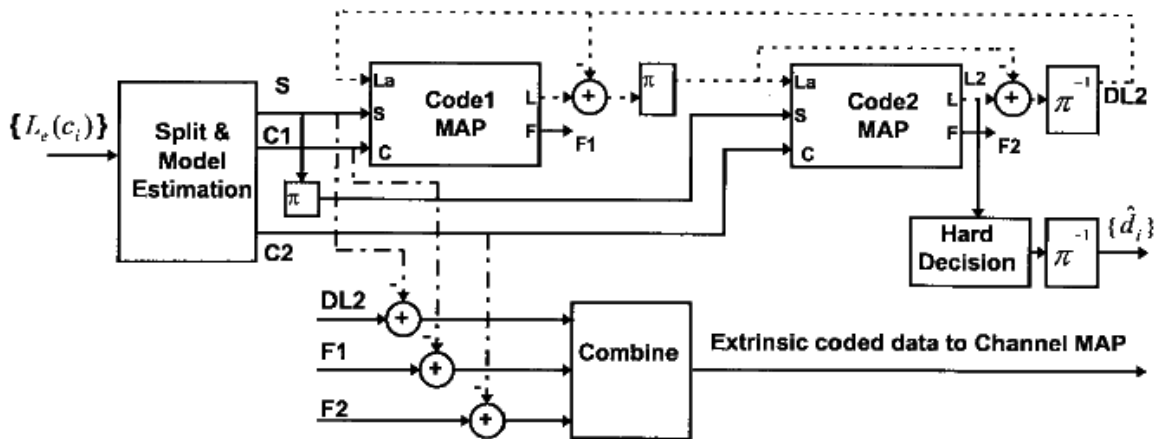


Figure III. 3: Structure du décodeur turbo.

S : LLR des données systématiques (estimées) ;

C1 : LLR des bits codés du premier codeur ;

C2 : LLR des bits codés du deuxième codeur ;

DL2 ; LLR extrinsèque désentrelacé (Sortie du 2<sup>ème</sup> codeur);

F1 : LLR des bits codés à la sortie du décodeur 1;

F2 : LLR des bits codés à la sortie du décodeur 2 [16].

### III.4. Modèle du canal équivalent (canal + égaliseur)

Pour utiliser la sortie LLR de l'égaliseur MAP par le turbo décodeur (en tant que données systématiques et codées), une conversion des LLR en un format Soft équivalent est nécessaire. C'est fait par le modèle de canal équivalent suivant [16] :

Nous supposons que la sortie de l'égaliseur MAP représente le rapport log-vraisemblance LLR d'un bruit blanc additif équivalent. [16] montre que pour estimer les données Soft à l'entrée du turbo décodeur, il faut estimer dans un premier temps la variance suivante [16]:

$$\sigma_q^2 = \frac{1}{\alpha^2} E\{n_k^2\} = \frac{E\{L_e(c_k)^2\}}{[E\{L_e(c_k)\check{c}_k\}]^2} - 1 \quad (\text{III.2})$$

Avec

$$c_k = \text{Sign}\{L(c_k)\} \quad (\text{III.3})$$

Les données soft à l'entrée du turbo décodeur sont estimées par :

$$c_k = \frac{\sigma_q^2}{2} L_e(c_k) \quad (\text{III.4})$$

### **III.5. Conclusion**

Dans ce chapitre nous sommes intéressés au principe de couplage turbo égaliseur – turbo décodeur et leur structure.

Après avoir présenté le couplage, nous présentons dans le chapitre suivant, les résultats de simulation.



# Chapitre IV : Résultats de simulation

## **VI.1.Introduction**

La simulation joue un rôle important dans la conception des systèmes de communications. Elle est utilisée pour investiger des questions du type "What If" ("Quoi Si") à propos d'un système réel. Les changements majeurs d'un système peuvent être tout d'abord simulés pour prédire leur impact sur les performances de celui-ci. Elle est aussi utilisée pour étudier un système durant le stade de conception, et avant qu'il soit construit.

La simulation peut être seulement utilisée comme outil d'analyse pour prédire l'effet des changements sur les performances d'un système existant. De plus, elle permet d'effectuer des comparaisons de performances entre des systèmes de transmissions existants ou proposés.

Dans ce chapitre, nous évaluons par simulation les performances de turbo décodage et couplage turbo égaliseur – turbo décodeur en termes de taux d'erreurs binaire (estimation de la probabilité d'erreur) implémenté sous Matlab.

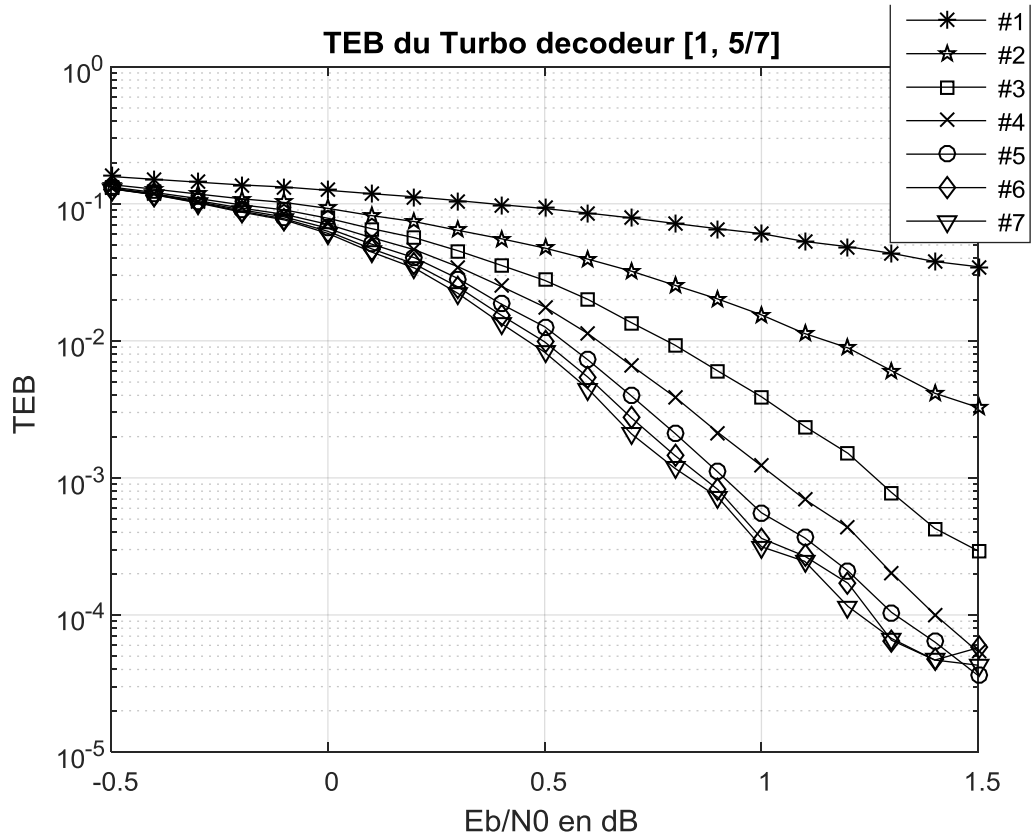
## **VI.2. Conditions de simulation**

Nous avons utilisés les paramètres suivants :

1. Longueur des paquets des données  $L=1024$ .
2. Codeur convolutif  $[1\ 5/7]$  de rendement  $R=1/2$ .
3. Modulation BPSK.
4. Canal sélectif : nous avons choisi le canal de PROAKIS B de réponse impulsionnelle  $C(n)= 0.7358\ \delta(n) + 0.0414\ \delta(n-1) + 0.6733\ \delta(n-2)$ .
5. Canal gaussien (Bruit blanc gaussien additif).
6. Rapport signal sur bruit : RSB de -0.5 à 1.5 dB.
7. Nombre de paquets transmis=1500paquets.
8. Nombre d'itération 7.
9. Entrelaceur pseudo-aléatoire de Matlab (Fonction (RANDINTRLV)).

### VI.3. Performance de turbo décodeur [1,5/7]

La figure (IV.1) montre les performances en termes de TEB du turbo décodeur MAP [1, 5/7]



**Figure IV. 1:** Performance du Turbo décodage [1 5/7]

Nous remarquons que le taux d'erreur binaire TEB se diminue après chaque itération. il atteint  $3 \times 10^{-5}$  au RSB = 1.5 dB.

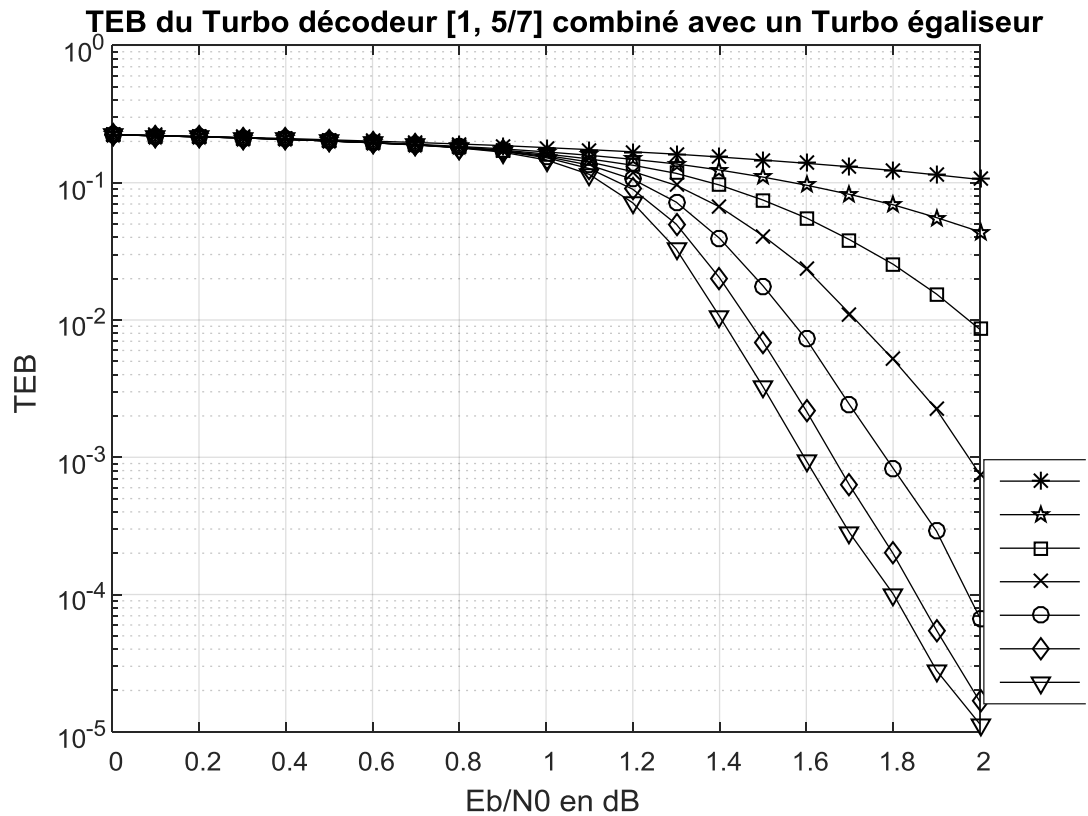


## VI.4. Performance du turbo égaliseur - turbo décodeur

Nous avons utilisés les paramètres suivants :

1. Longueur des paquets des données  $L=1024$ .
2. Codeur convolutif  $[1\ 5/7]$  de rendement  $R=1/2$ .
3. Modulation BPSK.
4. Canal sélectif : nous avons choisi le canal
$$c(n)=0.7358\delta(n) + 0.0414\delta(n-1) + 0.6733\delta(n-2).$$
5. Canal gaussien (Bruit blanc gaussien additif).
6. Rapport signal sur bruit : RSB de 0 à 2dB.
7. Nombre de paquets transmis=3000 paquets.
8. Nombre d'itération 7.
9. Entrelaceur pseudo-aléatoire de Matlab (Fonction (RANDINTRLV)).

La figure (IV.2) montre les performances en termes de TEB du turbo égaliseur MAP couplé au turbo décodeur  $[1, 5/7]$ . Le canal sélectif utilisé a la réponse impulsionnelle suivante  $C(n)=0.7358\delta(n) + 0.0414\delta(n-1) + 0.6733\delta(n-2)$ .




**Figure IV. 2 :** Performance du turbo égaliseur - turbo décodeur.

Les résultats de simulation montrent une réduction du taux d'erreur TEB après chaque itération. Il atteint presque le TEB d'un canal idéal codé sans interférence entre symboles 'IES'.

## Conclusion

Dans ce chapitre nous avons évalué les performances du Turbo décodeur MAP couplé à un turbo égaliseur d'un canal sélectif.



# Conclusion Général

## Conclusion Générale

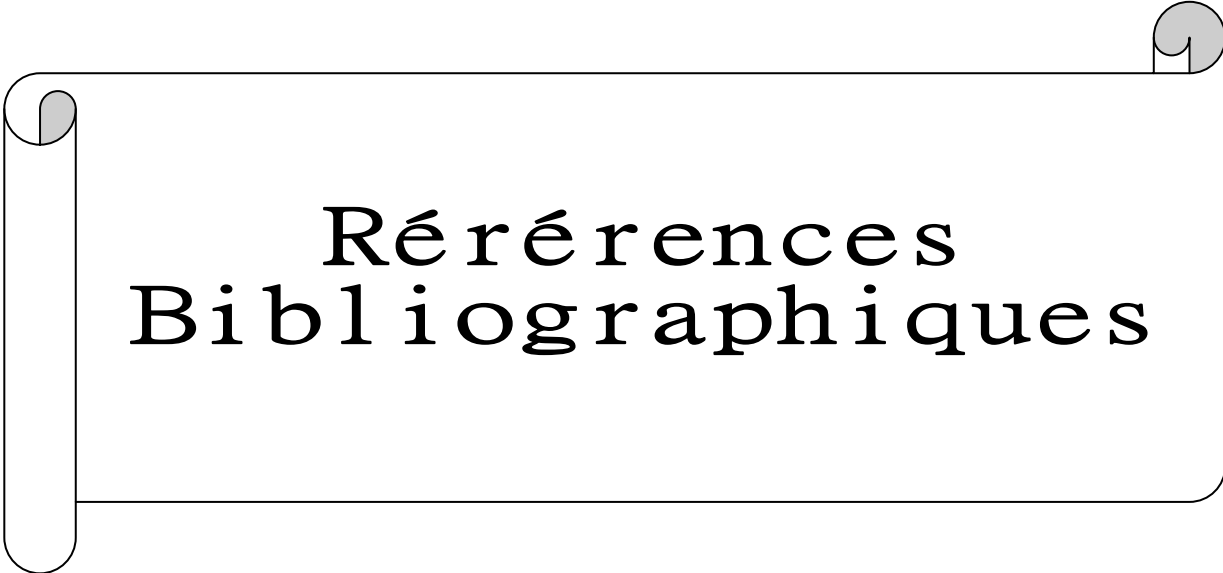
---

### Conclusion générale

Dans ce mémoire de fin d'études, nous nous sommes intéressés au turbo égalisation MAP, utilisée pour annuler les interférences entre symboles et son couplage à un turbo décodeur MAP. Les interférences dites IES sont créées par les canaux sélectifs en fréquence et peuvent dégrader les performances du turbo décodeur si aucune technique d'égalisation n'est utilisée à la réception.

Nous nous sommes intéressés aussi au couplage turbo égaliseur –turbo décodeur. L'étape principale pour réaliser ce regroupement est l'adaptation des LLR à la sortie de l'égaliseur avant son utilisation par le turbo décodeur. C'est-à-dire, il faut appliquer une conversion Soft-Channel des informations extrinsèques fournies par l'égaliseur MAP.

Les résultats de simulation ont montré le bon fonctionnement du couple turbo égaliseur – turbo décodeur mais avec une certaine perte en rapport signal sur bruit RSB. Cette perte dépend bien sûr du degré de sélectivité du canal. C'est-à-dire, elle dépend du canal sélectif lui-même.



Références  
Bibliographiques

## Référence Bibliographique

---

- [1] B.Bilal et Y.Amina. « Estimation des performances du TEB des turbo-codes dans un canal AWGN », Mémoire de Master L'Université SAAD DAHLAB de BLIDA, 2017/2018.
- [2] A.Glavieux, M. Joindot. « Communication numériques, Introduction », MASSON, 1996.
- [3] C.Berrou. « Codes et turbocodes », École Nationale Supérieure des Télécommunications de Bretagne CS 83818, Le 2007.
- [4] S.Hassen, S.Youcef. « Compensation des distorsions dans un système OFDM sous différents types de modulations numériques », Mémoire Présenté pour l'obtention du diplôme de master L'Université Mohamed Boudyaf Msila, 2017/2018.
- [5] T.Sartenaer. « Transmission numérique en bande de base », Cours de Télécommunications, Novembre 2006.
- [6] H. Zahoum. « Modulation numérique », Mémoire de Master L'Université Dr Moulay Taher de Saida, 2012/2013.
- [7] C. Abdelfattah, L. Abdelkrim, « Turbo égalisation MMSE », *Mémoire de fin d'étude* Pour l'obtention du diplôme de Master en Télécommunications L'université Dr Moulay Taher de Saida, 2019.
- [8] O.Muller, « Architectures multiprocesseurs monoprocesseurs génériques pour turbo communications haut-débit », Thèse de Doctorat L'université De Bretagne-Sud en habilitation conjointe avec l'École Nationale Supérieure des Télécommunications de Bretagne, Le 13 Décembre 2007.
- [9] F. ABDERREZZAQ, O. FAIÇEL. « Entrelaceurs S-Random appliqués à la turbo Égalisation MAP », *Mémoire de fin d'étude* Pour l'obtention du diplôme de Master en Télécommunications L'université Dr Moulay Taher de Saida, 2017.
- [10] P. Adde, R. Pyndiah, Jean.R. Inisan et Y. Sichez. « Conception d'un turbo décodeur de code produit », Ecole Nationale Supérieure des Télécommunications de Bretagne BP 832, Le 15-19 Septembre 1997.
- [11] T.Dahou, G.Khedidja. « Entrelaceurs Turbo à base de permutations cryptographiques », *Mémoire de fin d'étude* Pour l'obtention du diplôme de Master en Télécommunications, Le 2/06/2016.
- [12] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier and A. Glavieux. « Iterative correction of intersymbol interference: Turbo equalization », *European Transactions on Telecommunications*, vol 6, N°5, pp. 507-511, Le 07 Oct 1995.

## Référence Bibliographique

---

[13] B. Sklar. « Digital communication: Fundamentals and Applications », second Edition, communications Engineering services Californie and université de Californie, los Angel.

[14] B. Aicha, R. Khadidja. « Performances d'un turbo égaliseur M-BCJR », Mémoire de fin d'étude pour l'option du diplôme Master en Télécommunications, L'université Dr. Moulay Taher Saida, Le 16/06/2015.

[15] A. Berdai. « Turbo - égalisation de faible complexité avec estimation des canaux multi trajets à évanouissements rapides », thèse présentée à la Faculté des études supérieures de l'Université Laval dans le cadre du programme de doctorat génie électrique pour l'obtention du grade de Philosophé Doctor, Le 2011.

[16] Y. Zarai. « Combined Turbo Equalization and Turbo Decoding », Article in IEEE Communications Letters, May 1998.