

3/98

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

D.E.R DE GENIE ELECTRIQUE & INFORMATIQUE

FILIERE : ELECTRONIQUE

Laboratoire de traitement de signal



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

*Pour l'obtention du diplôme d'ingénieur d'état
en Electronique*

Thème

**CONCEPTION DES CODEURS OPTIMAUX
POUR LA TCM A DEUX DIMENSIONS ET
EVALUATION DES PERFORMANCES DE
LA MODULATION ASYMETRIQUE
CODEE EN TREILLIS (ATCM)**

Proposé et Dirigé par :

M. Z. TERRA
M. B. DERRAS

Etudié par :

BOUNOUGHAZ A.E.KRIM
KAHLOUCHE MOHAMED

Promotion : Juin 1998

E.N.P. 10, AVENUE HASSEN BADI - EL-HARRACH - ALGER

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

D.E.R DE GENIE ELECTRIQUE & INFORMATIQUE

FILIERE : ELECTRONIQUE

Laboratoire de traitement de signal



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

*Pour l'obtention du diplôme d'ingénieur d'état
en Electronique*

Thème

**CONCEPTION DES CODEURS OPTIMAUX
POUR LA TCM A DEUX DIMENSIONS ET
EVALUATION DES PERFORMANCES DE
LA MODULATION ASYMETRIQUE
CODEE EN TREILLIS (ATCM)**

Proposé et Dirigé par :

M. Z. TERRA
M. B. DERRAS

Etudié par :

BOUNOUGHAZ A.E.KRIM
KAHLOUCHE MOHAMED

Promotion : Juin 1998

E.N.P. 10, AVENUE HASSEN BADI - EL-HARRACH - ALGER

ملخص :

هذا العمل يستعرض خوارزمية لإيجاد مشافر مثلى ، و ذلك بدمجها مع الموائم. تقدير محاسن الموائم الغير المتناظرة المدججة مع المشفر .

في هذا الإطار، قمنا بدراسة و إنجاز خوارزمية انجربوك لإيجاد مشافر مثلى على شكل برامج . بعد ذلك قمنا بتقدير محاسن الأنظمة التي تستعمل الموائم الغير متناظرة مع المشفر و ذلك بحساب احتمال حادثة خطأ و احتمال خطأ bit .

نتائج التطبيق هي أيضاً معروضة .

Résumé :

Ce travail expose l'algorithme de conception des codeurs optimaux combinés avec les modulations et l'évaluation des performances des modulations Asymétriques codées en treillis (MACT). Dans ce cadre, nous avons étudié et implémenté l'Algorithme de recherche des codeurs optimaux. Ensuite, nous avons évalué les performances des systèmes à (MACT) en estimant la probabilité d'événement d'erreur et probabilité d'erreur par bit.

Les résultats de la simulation sont aussi présentées.

Mots-clés : Modulation codée en treillis, codeurs optimaux, Asymétrique, probabilité d'événement d'erreur, probabilité d'erreur par bit.

Abstract :

This work expose a design procedure for more powerful codes combined with modulation and performance evaluation for trellis coded asymmetric modulation (TCAM). In this context, we have studied and implemented a search procedure for more powerful codes. Later, we have evaluated the performances of TCAM schemes which consist to estimate error event probability and bit error probability. The simulation results are presented.

Key words : trellis coded Modulation, Trellis, codes powerful, asymetrique.

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

REDACTIONS

Dédicaces

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Je dédié ce modeste travail

- *A* la mémoire de ma sœur HASSINA.
- *A* mes très chers parents pour leur soutien durant toutes mes années d'étude.
- *A* mon frère HACENE qui m'a beaucoup encouragé.
- *A* mes frères.
- *A* mes sœurs.
- *A* toute ma famille.
- *A* tous mes amis.

Abdelkrim

Je dédié ce modeste travail

- *A* mes chers parents.
- *A* mes frères et sœurs.
- *A* mes beaux frères.
- *A* toute ma famille.
- *A* tous mes amis.
- *A* toute la promotion d'Electronique de 1998.

Mohamed

المدسة الوطنية المتسدة التكنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

REMERCIEMENT

Remerciement



Ce travail à été effectué au sein du laboratoire de traitement de signal du département de Génie Electrique & Informatique sous la direction de Monsieur **Z.TERRA** Chargé de cour à l'ENP.

Nous tenons à lui exprimé notre profonde gratitude et nos sincères remerciements pour l'aide sans réserve qu'il nous a apportée par ces conseils éclairés tout au long de ce travail.

Nos remerciements s'adressent à tout les enseignants qui ont contribué à notre formation.

Nous tenons à remercier :

- Monsieur K.CHERCHALI pour sa disponibilité, son aide et ses précieux conseils.
- Monsieur Y.KHALEF pour son aide.
- Tous le personnel du périodique et de la bibliothèque.

A tous ceux qui ont participé de près ou de loin à l'élaboration de ce mémoire.

Nous finissons en saluant la promotion d'électronique de juin 1998.

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

SOMMAIRE

SOMMAIRE



Introduction générale	
Chapitre 1 : Généralités sur la modulation codée en treillis (TCM)	
1-1 Introduction	3
1-2 Besoin de la TCM	3
1-3 Transmission non codée	4
1-4 Principe de la TCM	5
1-4-1 Gain de codage des distances	5
1-4-2 Gain de codage asymptotique	5
1-5 Représentation de la TCM	6
1-5-1 Représentation d'Ungerboeck	6
1-5-1-1 Codes d'Ungerboeck	7
1-5-1-2 Partition d'ensemble	7
1-5-1-3 Règles de partitionnement	8
1-5-2 Représentation analytique	10
1-5-2-1 Règles de conception	12
1-5-2-2 Exemples à une dimension	12
1-5-2-3 Exemples à deux dimensions	16
1-5-3 Passage de la représentation analytique à la représentation d'Ungerboeck	17
1-6 Décodage de la TCM	18
1-6-1 Introduction	18
1-6-2 Décodage à maximum de vraisemblance	18
1-6-3 Algorithme de Viterbi	20
1-7 Evaluation des performances	23
1-7-1 Notion d'événement d'erreur	23
1-7-2 La borne supérieure de la probabilité d'erreur	23
1-7-2-1 La matrice d'erreur	24
1-7-2-2 La fonction de transfert et probabilité d'erreur	25
1-7-2-3 Interprétations	26
1-7-2-4 Amélioration de la borne supérieure de $P(e)$	26
1-7-3 Probabilité d'erreur par bit P_b	27
1-7-4 Calcul de la fonction de transfert	27
1-7-5 Calcul de la distance libre d_{free}	28
1-7-5-1 La technique de calcul utilisant la fonction de transfert	28
1-8 Conclusion	29
Chapitre 2 : Codeurs optimaux pour la TCM à deux dimensions	
2-1 Introduction	30
2-2 Codes convolutifs	30
2-2-1 Représentation	31
2-2-2 Matrice de parité	31
2-3 Définitions	32
2-3-1 Propriété d'un codeur minimal	33
2-3-2 Avantage de la structure d'un codeur systématique	33
2-3-3 Codes catastrophiques	33
2-4 Codes convolutifs pour la TCM	33
2-5 Passage d'un codeur convolutif à un codeur systématique avec feed-back	34

2-5-1 Description du codeur systématique à boucle de retour	36
2-6 Méthode de calcul de d_{free}	37
2-7 Recherche des codes optimaux	38
2-7-1 Position du problème	38
2-7-2 Données du problème	38
2-7-3 Paramètres à déterminer	38
2-7-4 Description de l'Algorithme	39
2-7-5 Règles de conception	39
2-7-6 L'organigramme de recherche des codes optimaux	40
2-7-6-1 Description des blocs essentiels de l'organigramme	40
2-8 Conclusion	42

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Chapitre 3 : La modulation Asymétrique codée en treillis (ATCM)

3-1 Introduction	43
3-2 Analyse et conception	43
3-2-1 Procédure de conception	45
3-3 Codeurs en treillis et la TCM Asymétrique (ATCM)	45
3-3-1 Codeur de rendement $\frac{1}{2}$ combiné avec la 4-PSK Asymétrique	45
3-3-2 Codeur de rendement $\frac{2}{3}$ combiné avec la 8-PSK Asymétrique	51
3-3-3 Codeur de rendement $\frac{3}{4}$ combiné avec la 16-PSK Asymétrique	55
3-3-4 La modulation 4-AM Asymétrique	57
3-4 Conclusion	57

Chapitre 4 : Implémentation de l'Algorithme de recherche des codeurs optimaux de la TCM à 2D

4-1 Introduction	60
4-2 Modèle de simulation	60
4-2-1 Source d'information	61
4-2-2 Bloc aiguillage 1	61
4-2-3 Bloc d'embrouillage (Scramble)	61
4-2-4 Bloc aiguillage 2	61
4-2-5 Codeurs systématique	61
4-2-6 Bloc de calcul	61
4-3 L'organigramme de recherche des codes optimaux	61
4-3-1 Langage utilisé	63
4-3-2 Programmation	63
4-3-3 Les Fonctions	63
4-3-4 Les subroutines	63
4-4 Les modulations traitées	64
4-5 Les résultats	64
4-5-1 La modulation 8-PSK	65
4-5-2 La modulation 16-PSK	69
4-5-3 La modulation 4-AM	74
4-5-4 La modulation 8-AM	78
4-5-5 La modulation 16-AM	82
4-5-6 La modulation 16QAM	86
4-6 Commentaires	91
4-7 Conclusion	92

Chapitre 5 : Evaluation des performances de la modulation Asymétrique codée en treillis	
5-1 Introduction	93
5-2 Modèle de simulation	93
5-2-1 Source d'information	93
5-2-2 Codeur en treillis	94
5-2-3 Source de bruit	94
5-2-4 Décodage	94
5-3 Modulations utilisées	94
5-4 Transcodage	95
5-5 Programmation	95
5-5-1 Langage utilisé	95
5-5-2 Calcul de la probabilité $P_e(e)$ et le nombre de bits erronés	95
5-5-3 L'organigramme	96
5-6 Résultats et commentaires	96
5-6-1 Probabilité d'événements d'erreur $P_e(e)$	102
5-6-2 Probabilité des bits erronés	102
5-7 Conclusion	102
Conclusion générale	103
Bibliographie	



المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

INTRODUCTION GENERALE

Introduction générale

Un système de transmission numérique véhicule de l'information entre une source et un destinataire en utilisant un support physique comme le câble, la fibre optique ou, encore, l'espace libre.

La qualité d'un système de transmission est évaluée, en général, en calculant la probabilité d'erreur par bit transmis, celle-ci est fonction de la technique de transmission utilisée, mais aussi du canal sur lequel le signal est transmis.

Le canal de téléphonie a subi beaucoup de changements car les techniques de modulation progressent pour un canal à bande limitée.

Ces techniques progressent pour deux raisons :

- commerciale.
- pour modéliser un canal LTI (linear time invariant).

La tâche de la conception d'un système de communication numérique est de réaliser un système efficace raisonnable pour la transmission d'information à partir d'une source à un débit et un niveau de fiabilité acceptables.

Les deux paramètres clés lors de la conception sont :

- la puissance du signal transmis
- la bande passante du canal

Pour un rapport signal sur bruit fixé, la seule façon pratique pour améliorer la qualité de transmission est d'utiliser le codage de canal.

La modulation codée en treillis (TCM) est la combinaison codage-modulation pour améliorer la fiabilité d'un système de transmission sans augmenter la puissance transmise ou la bande passante.

La TCM est utilisée en communication de l'information afin de minimiser le bruit par rapport à la transmission non codée sans augmenter le débit de l'information.

Jusqu'au milieu des années 70, les fonctions de codage et de modulation étaient traitées comme deux entités séparées et indépendantes. L'optimisation du codage au sens de la distance de Hamming ne donnait pas alors de bons résultats lorsque la modulation n'était pas une modulation à 2 ou à 4 états de phase.

C'est G.Ungerboeck qui est alors arrivé à une solution pratique qui permet de tirer pleinement profit du codage avec des modulations à grand nombre d'états. L'idée est de faire le codage directement dans l'espace Euclidien, les fonctions de codage et de modulation ne forment plus des opérations séparées mais une seule et unique entité.

La première configuration TCM, a été proposée en 1976 par G.Ungerboeck et I.Csajka. En 1984, un système TCM avec un gain de codage de 4 dB est adopté par la C.C.I.T.T pour être utilisé dans de nouveaux modems à haute vitesse dans la bande passante de la parole.

Depuis cette année, les modems de données sont apparus dans les marchés utilisant la TCM mais avec une grande performance : amélioration de l'égalisation, synchronisation, annulation d'écho, ... etc, pour transmettre des données à travers les canaux de la parole avec un débit de 14,4 k bits / s et plus. Une explosion de recherche et des implémentations actuelles de la TCM prend place à un point où aujourd'hui nous avons une bonne compréhension de la théorie et les capacités des méthodes TCM.

L'utilisation commune des techniques de TCM dans de telles applications, en satellite, en micro-onde terrestre et les communications mobiles pour augmenter le débit ou pour réaliser des opérations satisfaisantes en cas de faible rapport signal sur bruit.

Le but de notre travail dans ce PFE est fixé par le cahier de charge proposé et qui consiste à évaluer la TCM Asymétrique et la conception des codeurs optimaux utilisés dans la TCM à deux dimensions.

Notre plan de travail adopté est constitué de cinq chapitres :

Le premier chapitre présente des généralités sur la TCM où nous évoquerons les deux techniques de réalisations qui sont la réalisation graphique (Ungerboeck) et la réalisation analytique (Calderbank-Mazo) ainsi que des exemples d'implémentations.

Le second chapitre est consacré à l'étude et à la recherche des codeurs optimaux selon la technique de conception d'Ungerboeck.

L'étude des différentes configurations TCM Asymétriques est donnée au troisième chapitre.

Dans le quatrième chapitre, nous présentons les 24 codeurs conçu selon l'algorithme d'Ungerboeck pour les modulations 8-PSK, 16-PSK, 4-AM, 8-AM, 16-AM et 16-QAM combinées avec les codeurs 4, 8, 16 et 32 états.

Dans le dernier chapitre, nous présentons des mesures de probabilité d'événement d'erreur et probabilités par bit pour les modulations M-PSK ($M=4, 8, 16$) combinées avec des codeurs optimaux. Nous terminerons ce travail avec une conclusion générale.

CHAPTER

1

Chapitre 1

Généralités sur la TCM

1-1 Introduction :

La modulation codée en treillis est la combinaison de codage et la modulation pour améliorer la qualité d'un système de transmission numérique sans augmenter la puissance transmise ou la bande passante.

Le codeur en treillis combine la modulation avec le codage convolutif tandis que le récepteur combine le décodage et la démodulation en une seule opération.

Fondamentalement, il existe deux réalisations de la TCM :

- le codeur d'Ungerboeck qui comprend un codeur convolutif et des bits de contrôle suivi d'un bloc de transcodage.
- le codeur de Calderbank-Mazo qui peut être utilisé pour réaliser le codeur en treillis.

Nous verrons par la suite la procédure de passage de la forme de Calderbank-Mazo au codeur d'Ungerboeck ainsi que l'opération inverse, nous terminerons ce point avec des exemples d'illustrations.

Nous verrons par la suite la partie décodage où nous traitons l'algorithme de Viterbi, nous clôturons ce chapitre avec le concept de la probabilité d'erreur et le calcul de la distance d_{free} de la configuration TCM.

1-2 Besoin de la TCM : [1]

En général, la sélection de la modulation et les formes de codage pour la transmission dans un canal bruité par un bruit blanc gaussien additif AWGN est basée sur les paramètres suivants :

- Le débit d'information \mathfrak{R} c'est-à-dire le nombre de bits par seconde qu'on veut transmettre à travers le canal.
- La bande passante B disponible (en Hertz).

En général :
$$B = \frac{\alpha}{T} \quad \dots(1.1)$$

Où : T est la durée de la forme d'onde utilisée par le modulateur.

α est une valeur qui dépend de la définition de la bande passante.

- La probabilité d'erreur obtenue pour un rapport signal sur bruit donné.

Le rapport $r = \mathfrak{R} / B$ en bits/s/Hertz est le nombre de bits par seconde qu'on peut transmettre dans une bande de 1 Hz.

Le nombre de bits par signal est :

$$R = T \cdot \mathfrak{R} = T \cdot B \cdot r \quad \dots(1.2)$$

Quand $\alpha = 1$ nous avons $R = r$ qui est le nombre de bits par canal utilisé.

Par exemple 9600 bits/s peut être transmis à travers un canal de 2400 Hz en utilisant $9600/2400 = 4$ bits/canal utilisé (c'est-à-dire une modulation M-aire non codée pour $M=16$).

Considérons l'exemple de la conception d'un système de communication radio digitale dans lequel nous voulons transmettre un signal à enveloppe constante avec les paramètres suivants :

- $B = 4800$ Hz avec $\alpha = 2$.
- $\mathfrak{R} = 9600$ bits/s.

$$r = \frac{9,6}{4,8} = 2 \text{ bits/s/Hz.}$$

Si nous utilisons la modulation PSK, nous avons besoin d'une bande :

$$B = \frac{2}{T} \quad \dots(1.3)$$

alors :

$$T \cdot R = T \cdot B \cdot r = 4 \text{ bits/signal.}$$

Le système 16-PSK est une solution adéquate si elle satisfait une probabilité d'erreur exigée, pour l'instant si $p(e) \leq 10^{-5}$ est exigée, la modulation 16-PSK non codée est adéquate si le rapport signal sur bruit $\left(\frac{E_b}{N_0}\right)$ est supérieur à 18db, mais si ce rapport est égal à 15 dB, nous devons introduire la configuration TCM qui produit un gain de codage sans altéré la bande occupée par le signal transmis.

1-3 Transmission non codée : [1]

Nous assumons ici le temps discret et amplitude continue et que la transmission de l'information est dans un canal bruité par un bruit blanc gaussien additif (AWGN).

Les messages délivrés au récepteur sont représentés par des points où des vecteurs dans un espace euclidien à N dimensions appelé espace des signaux.

Quand le vecteur X est transmis, le signal reçu est représenté par le vecteur $Z = X+W$ où W est le vecteur bruit, ces composantes sont des variables aléatoires indépendantes de moyennes nulles et de variances égales à $(N_0/2)$, le vecteur X est choisi d'un ensemble Ω' constitué de M' vecteurs.

L'énergie du signal moyen est donnée par :

$$E' = \frac{1}{M'} \cdot \sum_{x \in \Omega'} \|x\|^2 \quad \dots(1.4)$$

Considérons maintenant la transmission de la séquence $\{x_n\}_{n=0}^{K-1}$ de K signaux, L'indice n représente le temps discret.

Le récepteur qui minimise la probabilité d'erreur moyenne opère de la manière suivante :

Le récepteur observe premièrement la séquence y_0, y_1, \dots, y_{K-1} puis il décide en faveur de X_0

, X_1, \dots, X_{K-1} si le carré de la distance euclidienne $d^2 = \sum_{n=0}^{K-1} \|y_n - x_n\|^2$ est minimisé pour

$x_n = X_n$ avec $n=0, \dots, K-1$.

Un cas spécial important apparaît quand les signaux forment une séquence indépendante, dans ce cas la minimisation de d^2 est faite en minimisant séparément les termes individuels $\|y - x\|^2$ où $x \in \Omega'$ qui représente l'ensemble des signaux de la source émettrice.

Le carré de la distance minimum est $d_{\min}^2 = \min \|x - y\|^2$ où x et $y \in \Omega'$ et $x \neq y$.

En effet, la probabilité d'erreur $P(e)$ du symbole émis est bornée supérieurement en fonction de la distance minimum d_{\min} et la puissance moyenne du bruit $(N_0/2)$ selon la formule :

$$P(e) \leq \frac{M' - 1}{2} \operatorname{erfc} \left(\frac{d_{\min}}{2 \cdot \sqrt{N_0}} \right) \quad \dots(1.5)$$

Il est convenable de définir deux quantités utilisées dans la comparaison de différentes constellations.

- Le débit d'information ou l'efficacité spectrale de l'ensemble de signaux :

$$R = \frac{\log_2 M'}{N} \quad \dots(1.6)$$

où M' est le nombre de vecteurs dans l'ensemble Ω' et N est la dimension de l'espace des signaux.

- L'efficacité d'énergie :

$$\delta^2 = \frac{d_{\min}^2}{E'} \cdot \log_2 M' \quad \dots(1.7)$$

- L'énergie moyenne par bit est :

$$E_b = \frac{E'}{\log_2 M'} \quad \dots(1.8)$$

la probabilité d'erreur s'écrit de nouveau sous la forme :

$$P(e) \leq \frac{M' - 1}{2} \cdot \operatorname{erfc} \left(\frac{\delta}{2} \cdot \sqrt{\frac{E_b}{N_0}} \right) \quad \dots(1.9)$$

1-4 Principe de la TCM : [1]

Une façon d'améliorer la performance d'un système de communication est de rejeter la supposition que les signaux sont indépendants. Si nous augmentons l'ensemble Ω' à un ensemble Ω tel que $\Omega' \subset \Omega$ et M' à M tel que $M' < M$, nous sélectionnons les séquences M'^K comme un sous ensemble de Ω^K , nous pouvons avoir des séquences moins serrées et donc la croissance de la distance minimum entre eux, nous obtenons ainsi une distance minimum appelée d_{free} .

1-4-1 Gain de codage des distances :

En utilisant le principe de détection à maximum de vraisemblance, le gain de codage des distances est défini par :

$$G_1 = \frac{d_{free}^2}{d_{\min}^2} \quad \dots(1.10)$$

1-4-2 Gain de codage asymptotique :

Pour éviter une réduction du taux de transmission, la constellation est expansée de Ω' à Ω , ceci peut entraîner une expansion de l'énergie moyenne E' à E .

Le gain de codage asymptotique pour la configuration TCM est donné par :

$$G_2 = \frac{\left(\frac{d_{free}^2}{E} \right)}{\left(\frac{d_{\min}^2}{E'} \right)} \quad \dots(1.11)$$

E et E' sont les énergies moyennes dépensées pour la transmission avec codage et sans codage respectivement.

Le modèle général pour la TCM est illustré à la figure 1.1 :

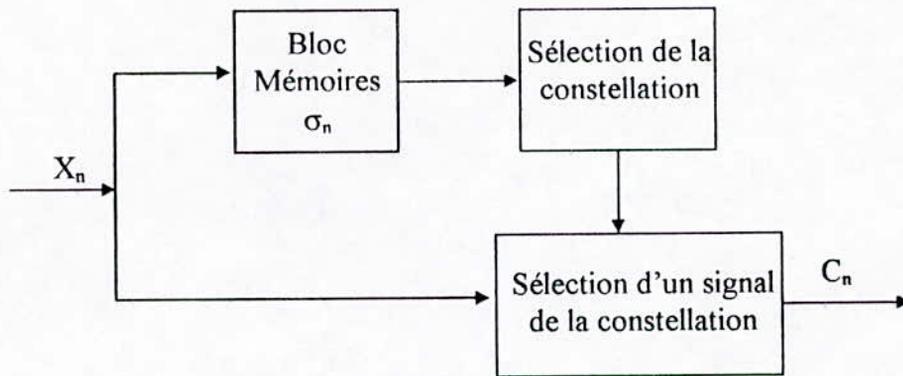


Figure 1.1 : Modèle général de la TCM

Le signal C_n transmis à l'instant n dépend du symbole émis à l'instant n et aussi d'un nombre fini L de symboles précédents X_{n-1}, \dots, X_{n-L} :

$$C_n = f(X_n, X_{n-1}, \dots, X_{n-L}) \quad \dots (1.12)$$

L'état du codeur à l'instant n est défini par :

$$\sigma_n = (X_{n-1}, \dots, X_{n-L}) \quad \dots (1.13)$$

Alors sous une forme compacte :

$$\begin{cases} C_n = f(X_n, \sigma_n) \\ \sigma_{n+1} = g(X_n, \sigma_n) \end{cases} \quad \dots (1.14)$$

La fonction f décrit à chaque instant, le symbole transmis choisi de la constellation qui est sélectionnée par la valeur de σ_n .

La fonction g , la partie mémoire du codeur, décrit l'évolution des états du modulateur.

Dans notre étude sur les codeurs TCM, nous allons considérer que les fonctions f et g sont invariantes dans le temps.

1-5 Représentation de la TCM :

Nous allons voir deux méthodes de réalisation de la modulation codée en treillis qui sont :

- La représentation d'Ungerboeck.
- La représentation analytique de Calderbank et Mazo.

1-5-1 Représentation d'Ungerboeck : [2], [1]

Cette méthode est due à Ungerboeck et consiste à modéliser la partie mémoire du codeur TCM par un codeur convolutif.

Si X_n est le symbole de source à l'instant n , qui peut prendre 2^k valeurs, nous pouvons le représenter comme une séquence de k digits binaires $X_n^{(1)}, \dots, X_n^{(k)}$ qui sont présentes simultanément au codeur.

En général, la sortie C_n du codeur à l'instant n s'écrit :

$$C_n = f(X_n^{(1)}, X_{n-1}^{(1)}, \dots, X_{n-v_1}^{(1)}, \dots, X_n^{(k)}, \dots, X_{n-v_k}^{(k)}) \quad \dots (1.15)$$

comme le montre la figure 1.2 :

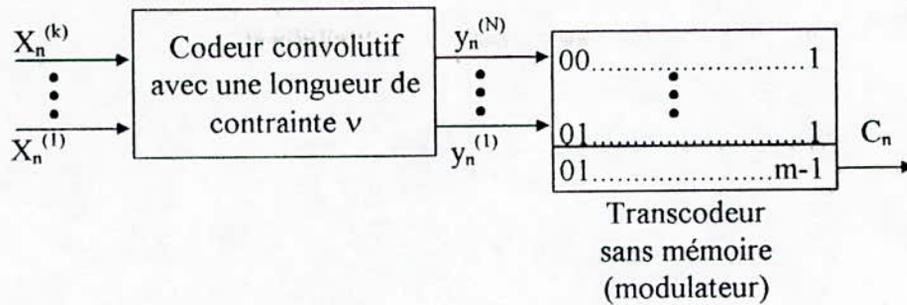


Figure 1.2 : Représentation d’Ungerboeck d’un codeur en treillis

De ce modèle nous avons :

- un codeur convolutif à k entrées binaires $X_n^{(1)}, \dots, X_n^{(k)}$ et N sorties binaires $y_n^{(1)}, \dots, y_n^{(N)}$; v représente le nombre d’éléments mémoires qui détermine 2^v états du codeur.
- un transcodeur (mapper) : est un bloc sans mémoires qui associe aux entrées $y_n^{(1)}, \dots, y_n^{(N)}$ un signal C_n appartenant à l’ensemble Ω .

1-5-1-1 Codes d’Ungerboeck : [1], [10]

Une famille importante des codes utilisés en TCM appelés les codes d’Ungerboeck. Ces codes satisfont les exigences suivantes :

- $M=2M'$ où M est le nombre de signaux utilisés pour transmettre $\log_2 M - 1$ bits par signal.
- le code convolutif utilisé est linéaire.

Il convient mieux de représenter le codeur d’Ungerboeck comme le montre la figure 1.3 :

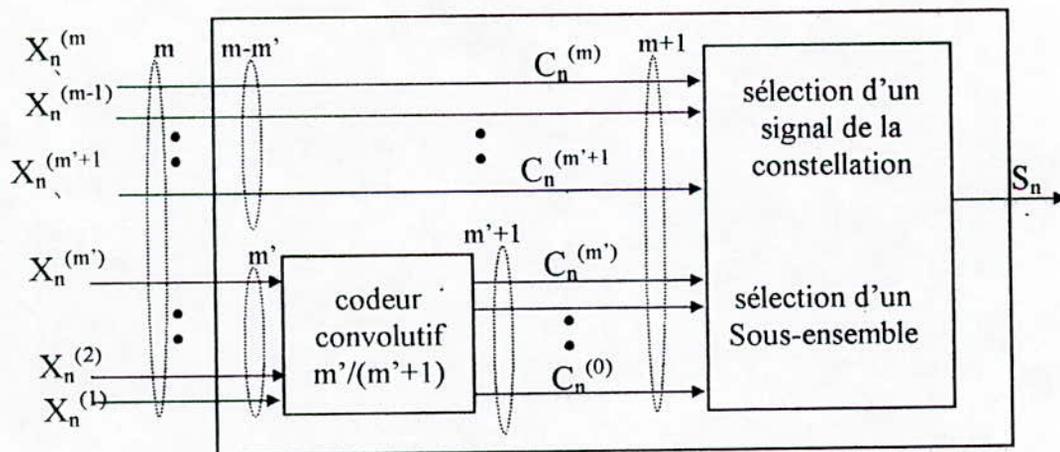


Figure 1.3 : schéma de principe de la TCM d’Ungerboeck

Parmi les m bits d’information à l’entrée du codeur, m’ bits entrent dans un codeur convolutif de rendement $\frac{m'}{m'+1}$ et les (m'+1) bits de sortie de celui-ci sélectionnent un sous-ensemble et les (m-m’) bits non codés sélectionnent un point dans ce sous-ensemble. la procédure d’Ungerboeck est basée sur un ensemble de règles heuristiques.

1-5-1-2 Partition d’ensemble :

Les modulations codées sont basées sur le principe de la partition d'alphabet qui a été introduit par Ungerboeck, la partition est faite de façon à augmenter la distance euclidienne minimale à l'intérieur des sous-ensembles obtenus.

La constellation M-aire est successivement partitionnée en 2,4,8,... sous-ensembles de tailles $\frac{M}{2}, \frac{M}{4}, \frac{M}{8}, \dots$ ayant progressivement une distance minimale croissante $d_{\min}^{(1)}, d_{\min}^{(2)}, d_{\min}^{(3)}, \dots$

1-5-1-3 Règles de partitionnement : [2],[3],[7]

La première étape de la conception est de choisir le nombre de bits m' afin d'établir une bonne connectivité entre les variables d'états du treillis, ce choix détermine la bonne valeur de la distance euclidienne de la configuration TCM.

Règle R1 : *Tous les signaux doivent être équiprobables.*

Règle R2 : *Les transitions parallèles doivent être assignées aux signaux d'un sous-ensemble avec une grande distance entre les sous-ensembles.*

Règle R3 : *Les 2^m transitions qui divergent d'un état commun ou convergent vers le même état doivent être assignées avec les signaux d'un sous-ensemble au premier niveau de la partition d'ensemble.*

Pour généraliser à des codes en treillis de débits $m/(m+1)$, le diagramme en treillis comporte 2^m branches issues de chaque noeud. Au delà des K premières branches, les chemins commencent à converger par groupe de 2^m . Si le codeur en treillis comporte v éléments mémoires et m' entrées du codeur convolutif alors :

- Le nombre d'états possible est 2^v
 - Le nombre de transitions de chaque état est $2^{m'}$
 - Le nombre de transitions parallèles partant de chaque état est $2^{m-m'}$
- Par conséquent, le diagramme d'état aura aussi 2^v noeuds où 2^m branches divergeante et convergeante de et vers chaque noeud.

Exemple 1.1 : [1], [3]

Soit le codeur en treillis de rendement $\frac{1}{2}$ à quatre états suivi d'un transcodeur (mapper) qui utilise la modulation 4-PSK schématisé par :

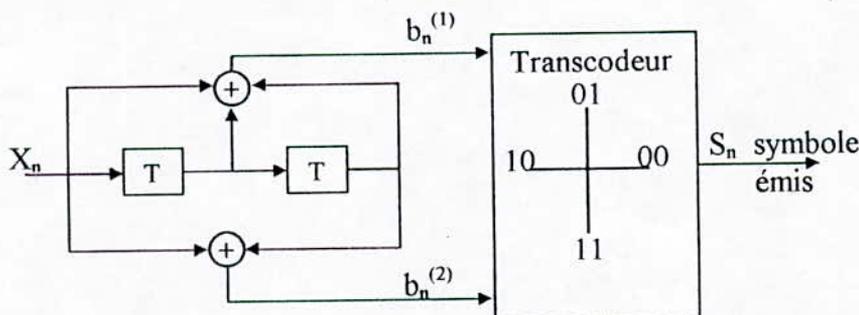


Figure 1.4 : Codeur en treillis à modulation 4-PSK

La partition d'ensemble de la modulation 4-PSK est illustrée dans la figure 1.5 :

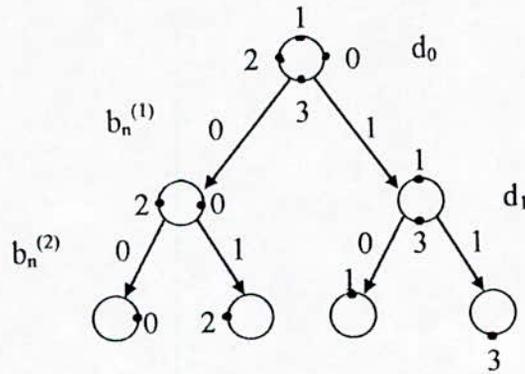


Figure 1.5 : Partition d'ensemble de la 4-PSK

où d_0 et d_1 sont les distances minimales entre les sous-ensembles avec $d_0=1,414$ et $d_1=2$. La portion de base du treillis du codeur de la figure 1.4 est illustrée dans la figure 1.6 :

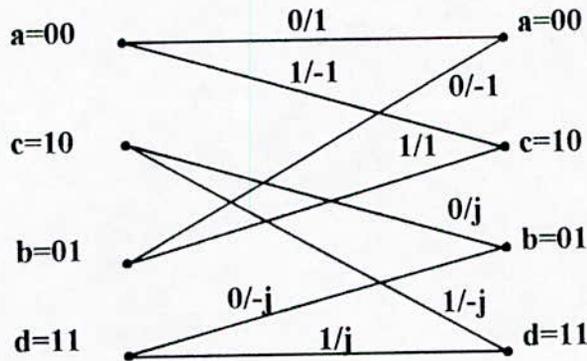


Figure 1.6 : Le treillis

Dans ce treillis, les transitions sont étiquetées par X_n/S_n où X_n est l'entrée du codeur et S_n est le symbole de sortie.

$b_n^{(1)}$ et $b_n^{(2)}$ sont les sorties du codeur convolutif, $b_n^{(1)}$ sélectionne le sous-ensemble dans la partition et $b_n^{(2)}$ sélectionne le point message dans ce sous-ensemble.

Exemple 1.2 : [2], [3], [1]

Soit le codeur en treillis de rendement 2/3 à quatre états suivi d'un transcodeur qui utilise la modulation 8-PSK :

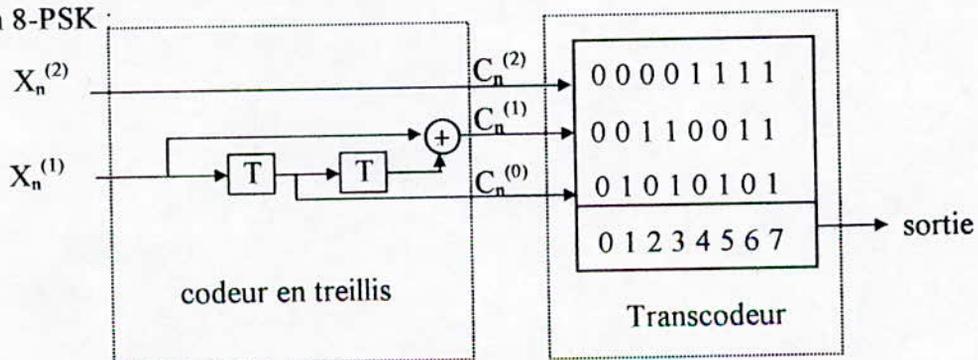


Figure 1.7 : Codeur en treillis de rendement 2/3 et modulation 8-PSK

Ce schéma représente un codeur d'Ungerboeck à modulation 8-PSK pour une transmission de 2 bits/symbole, la constellation de ce codeur est de taille égale à $2^3 = 8$ points messages.

Dans le bloc de transcodage, le codeur TCM réalise la correspondance entre les sorties binaires du codeur en treillis et les numéros des points messages.

La partition d'alphabet de la modulation 8-PSK est représentée par la figure 1.8 :

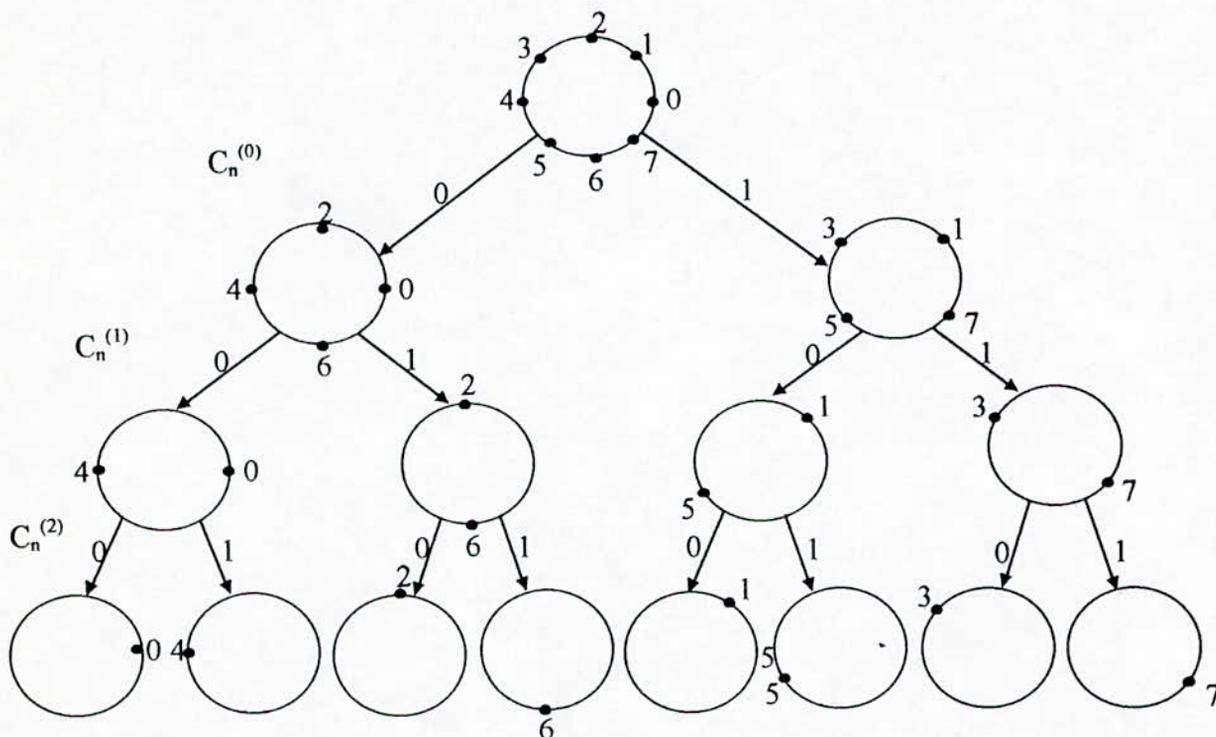


Figure 1.8 : La partition d'ensemble de la 8-PSK

L'entrée $X_n^{(2)}$ non codée produit les transitions parallèles dans le treillis du codeur, les deux valeurs possibles de $X_n^{(2)}$ produisent deux transitions parallèles, tandis que l'entrée $X_n^{(1)}$ contrôle entièrement les couples de transitions qui lient les états entre eux.

L'état du codeur est défini par le contenu des éléments mémoires $X_{n-1}^{(1)}, X_{n-2}^{(1)}$.

Les bits $C_n^{(0)}, C_n^{(1)}$ sélectionnent les sous-ensembles dans la partition et le bit $C_n^{(2)}$ sélectionne le point message dans la partition.

1-5-2 Représentation analytique : [1],[8]

Cette méthode est due à Calderbank et Mazo et est basée sur une expression analytique du transcodage des symboles de la source.

On considère une source d'information qui génère une séquence (a_n) de symboles pris à partir d'un alphabet de 2^m éléments.

La structure générale de cette méthode est schématisée par la figure 1.9 :

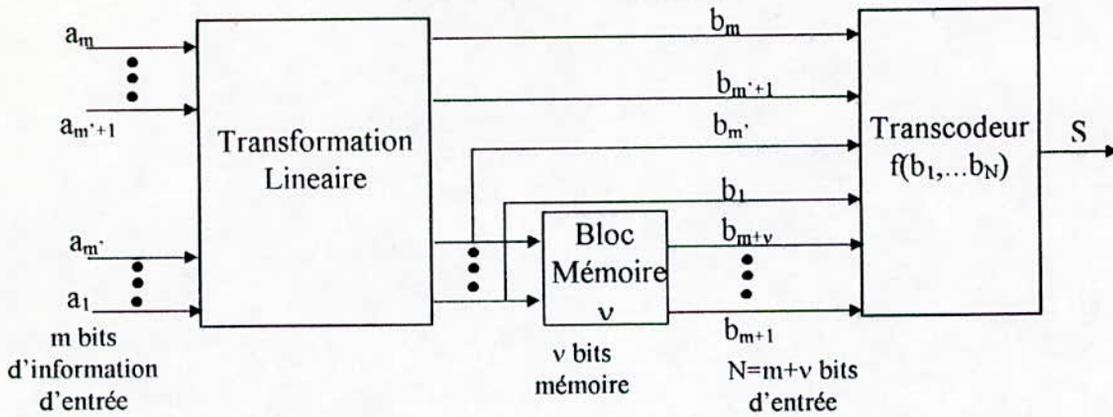


Figure 1.9 : Structure générale de la représentation analytique

Ce schéma comporte les blocs suivants :

- **Bloc de la transformation Linéaire :**

Ce bloc convertit la séquence d'entrée (a_n) de valeurs $\{0, 1\}$ respectivement en une nouvelle séquence (b_n) de valeurs $\{1, -1\}$ selon la formule suivante :

$$b_n = 1 - 2a_n \quad \dots(1.16)$$

- **blocs mémoire et transcodage :**

L'effet du codeur en treillis est de transformer la séquence de symboles à une séquence de signaux exprimée par la relation :

$$S = f(a_1, \dots, a_N) \quad \dots(1.17)$$

La fonction f doit être non linéaire pour avoir un gain de codage.

De manière générale le signal S s'écrit :

$$S = c_0 + \sum_{i=1}^N c_i \cdot a_i + \sum_{\substack{i,j=1 \\ j>i}}^N c_{ij} \cdot a_i \cdot a_j + \sum_{\substack{i,j,l=1 \\ l>j>i}}^N c_{ijl} \cdot a_i \cdot a_j \cdot a_l + \dots + c_{12\dots N} \cdot a_1 \cdot a_2 \dots a_N \quad \dots(1.18)$$

le signal S du canal dépend des m bits d'entrées (a_1, \dots, a_m) et des v bits de sortie du bloc mémoire (b_{m+1}, \dots, b_{m+v}) ainsi en faisant la transformation linéaire de l'équation 1.16 le signal S est une fonction des N bits d'entrées du transcodeur tel que $N=m+v$ donné par la formule :

$$S = d_0 + \sum_{i=1}^N d_i \cdot b_i + \sum_{\substack{i,j=1 \\ j>i}}^N d_{ij} \cdot b_i \cdot b_j + \sum_{\substack{i,j,l=1 \\ l>j>i}}^N d_{ijl} \cdot b_i \cdot b_j \cdot b_l + \dots + d_{12\dots N} \cdot b_1 \cdot b_2 \dots b_N \quad \dots(1.19)$$

Cette équation écrite sous forme matricielle :

$$X = B \cdot D \quad \dots(1.20)$$

avec :

$$X = \begin{bmatrix} f(+1,+1,\dots,+1) \\ f(-1,+1,\dots,+1) \\ \dots\dots\dots \\ \dots\dots\dots \\ f(-1,-1,\dots,-1) \end{bmatrix} \quad \dots(1.21)$$

$$B_i = [1 \ b_1 \ b_2 \ \dots \ b_N \ b_1.b_2 \ b_2.b_3 \ \dots \ b_1.b_2\dots.b_N] \quad \dots(1.22)$$

B_i est la i ème ligne de la matrice B
et

$$D^T = [d_0, d_1, \dots, d_{12\dots N}] \quad \dots(1.23)$$

Pour une modulation de dimension K , X et D sont de dimensions $2^N \times K$, B est de dimension $2^N \times 2^N$.

Le vecteur colonne D est donné par :

$$D = \frac{1}{2^N} B^T . X \quad \dots(1.24)$$

1-5-2-1 Règles de conception : [1]

Pour obtenir des descriptions analytiques du codeur avec complexité minimale et performance optimale, les règles posées par Ungerboeck et Turgeon doivent être appliquées. Un codeur analytique est de complexité minimale si l'équation (1.19) contient un nombre minimal de coefficients d non nuls et chaque bit d'entrée apparaît une fois dans la formule.

La première étape de la procédure est de respecter les règles R1, R2, R3 d'Ungerboeck citées ci-dessus.

La deuxième étape est de déterminer l'état du codeur en utilisant les règles T1, T2 proposées par Turgeon.

Règle T1 : *Pour établir la séquence du signal pour l'état 1, nous déterminons la valeur maximum du signal (MSV) au niveau m de la partition d'ensemble ; l'assignement commence au niveau m avec les deux signaux du sous-ensembles contenant le MSV.*

Règle T2 : *Chaque bits d'entrée doit être associé à un signal différence unique. Le signal différence représente simplement la différence de deux signaux du canal quand un seul bit change dans le bloc d'entrée de la source des bits.*

Pour un bloc d'entrée $(b_1, b_2, \dots, b_k, b_{k+1}, \dots, b_N)$ constitué par les symboles $\{1, -1\}$ le i -ème signal différence est donné par :

$$\delta_i = f(b_1, b_2, \dots, b_i, \dots, b_k, b_{k+1}, \dots, b_N) - f(b_1, b_2, \dots, -b_i, \dots, b_k, b_{k+1}, \dots, b_N) \quad \dots(1.25)$$

le i -ème bit a changé de signe.

Dans ces règles la valeur maximale du signal est la valeur numérique associée avec le symbole de sortie du transcodeur.

Dans ce qui suit nous traitons des exemples de configurations TCM à une et à deux dimensions ainsi que leurs réalisations avec la méthode analytique (Calderbank et Mazzo).

1-5-2-2 Exemples à une dimension :

- **La modulation 4-AM : [1]**

La partition d'ensemble de cette modulation est donnée par la figure 1.10 :

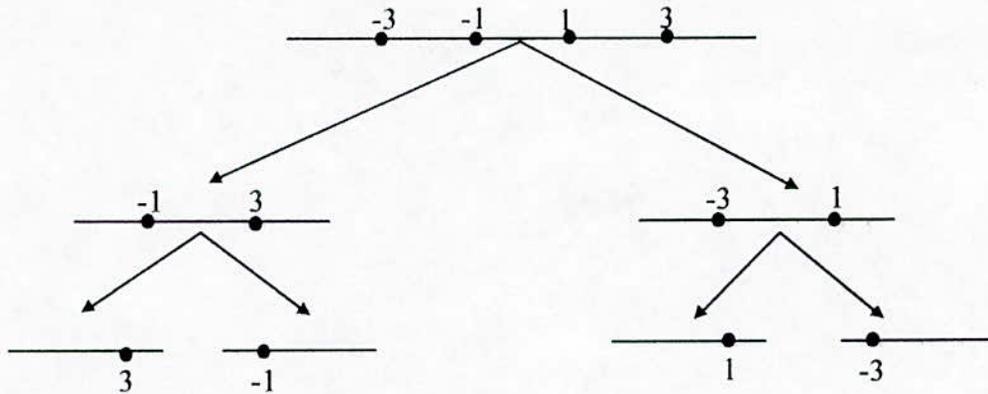


Figure 1.10 : Partition de la 4-AM

Dans la modulation 4-AM le codeur à quatre états selon la méthode analytique est donné par la figure 1.11 :

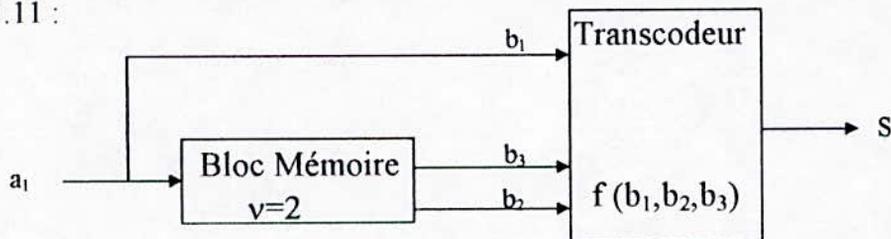


Figure 1.11 : Structure du codeur analytique 4-AM à 4 états

b_2, b_3 définissent l'état du codeur analytique.

La structure en treillis est illustrée dans la figure 1.12 :

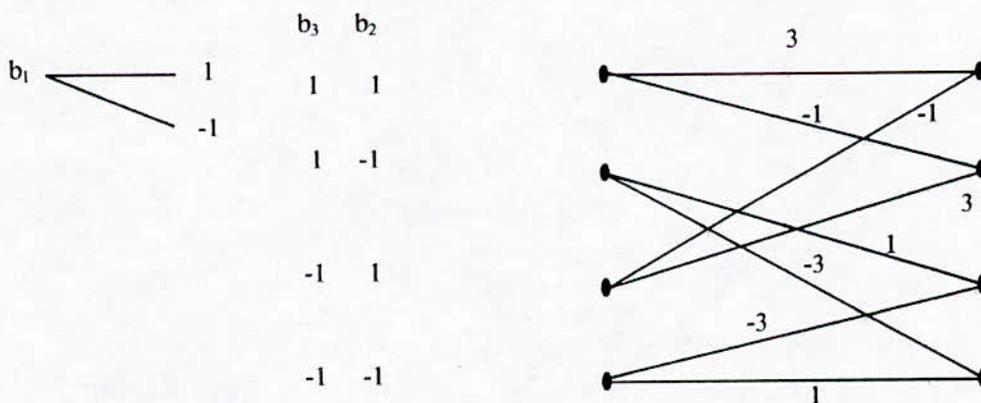


Figure 1.12 : La structure en treillis

Du treillis nous avons :

$$\begin{aligned}
 f(+1,+1,+1) &= +3 & f(-1,+1,+1) &= -1 \\
 f(+1,-1,+1) &= +1 & f(-1,-1,+1) &= -3 \\
 f(+1,+1,-1) &= -1 & f(-1,+1,-1) &= +3 \\
 f(+1,-1,-1) &= -3 & f(-1,-1,-1) &= +3
 \end{aligned}
 \tag{1.26}$$

Les signaux de différence δ_1, δ_2 et δ_3 sont donnés par :

$$\begin{aligned}
 \delta_1 &= f(b_1, b_2, b_3) - f(-b_1, b_2, b_3) = 4 \\
 \delta_2 &= f(b_1, b_2, b_3) - f(b_1, -b_2, b_3) = 2 \\
 \delta_3 &= f(b_1, b_2, b_3) - f(b_1, b_2, -b_3) = 4
 \end{aligned}
 \tag{1.27}$$

La relation matricielle $X = B.D$ est :

$$\begin{bmatrix} f(+1,+1,+1) \\ f(-1,+1,+1) \\ f(+1,-1,+1) \\ f(-1,-1,+1) \\ f(+1,+1,-1) \\ f(-1,+1,-1) \\ f(+1,-1,-1) \\ f(-1,-1,-1) \end{bmatrix} = \begin{bmatrix} +3 \\ -1 \\ +1 \\ -3 \\ -1 \\ +3 \\ -3 \\ +1 \end{bmatrix} = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & +1 & -1 & -1 & +1 & -1 \\ +1 & +1 & -1 & +1 & -1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & -1 & +1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & -1 & +1 & +1 & +1 & -1 \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_{12} \\ d_{13} \\ d_{23} \\ d_{123} \end{bmatrix}
 \tag{1.28}$$

$$\begin{aligned}
 S &= f(b_1, b_2, b_3) \\
 &= d_0 + d_1 \cdot b_1 + d_2 \cdot b_2 + d_3 \cdot b_3 + d_{12} \cdot b_1 \cdot b_2 + d_{13} \cdot b_1 \cdot b_3 + d_{23} \cdot b_2 \cdot b_3 + d_{123} \cdot b_1 \cdot b_2 \cdot b_3
 \end{aligned}
 \tag{1.29}$$

La recherche du vecteur colonne D en résolvant l'équation (1.24) donne :

$$d_0 = d_1 = d_3 = d_{12} = d_{23} = d_{123} = 0 \quad d_2 = 1 \quad d_{13} = 2$$

La description analytique du codeur est :

$$S = b_2 + 2 \cdot b_1 \cdot b_3 \tag{1.30}$$

La réalisation de cette description analytique est illustrée par la figure 1.13 :

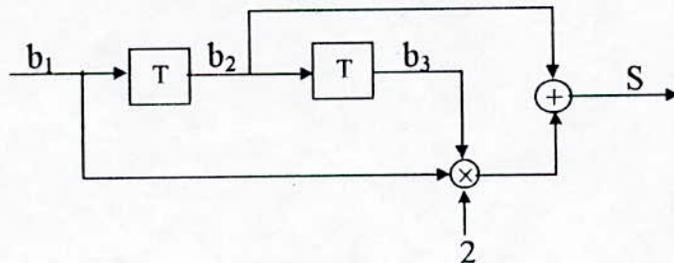


Figure 1.13 : Implémentation du transmetteur de la description analytique

• La modulation 16-PSK : [1]

La constellation de la modulation 16-PSK est donnée par la figure 1.14 :

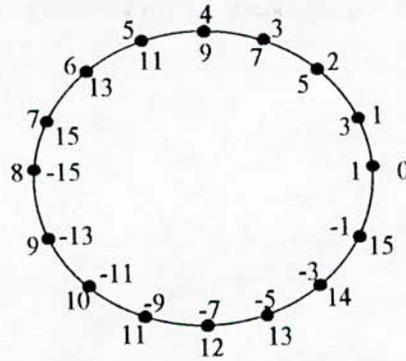


Figure 1.14 : La constellation de la modulation 16-PSK

Nous allons voir que la modulation 16-PSK à deux dimensions peut être générée d'une constellation à une dimension.

La modulation 16-PSK codée en treillis sera générée en utilisant $e^{j(s-1)\pi/16}$ où S est généré selon la modulation AM.

La structure du codeur selon la représentation analytique est donnée par la figure 1.15 :

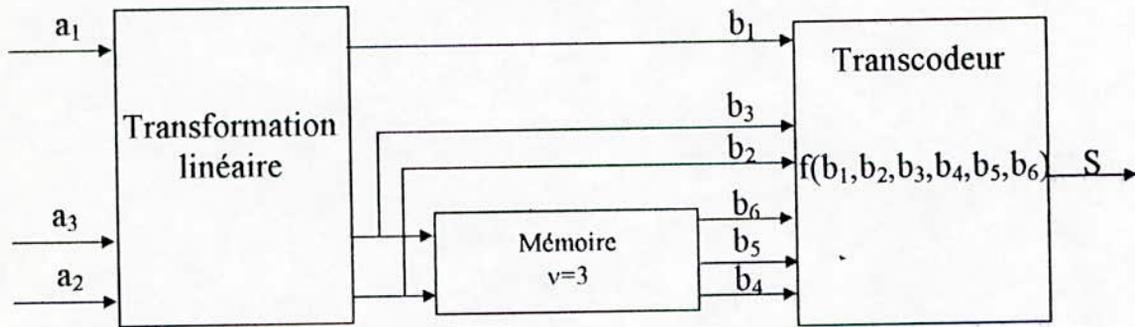


Figure 1.15 : La structure du codeur selon la méthode analytique

les bits b_4, b_5, b_6 définissent l'état du treillis.

La structure en treillis est illustrée dans la figure 1.16 :

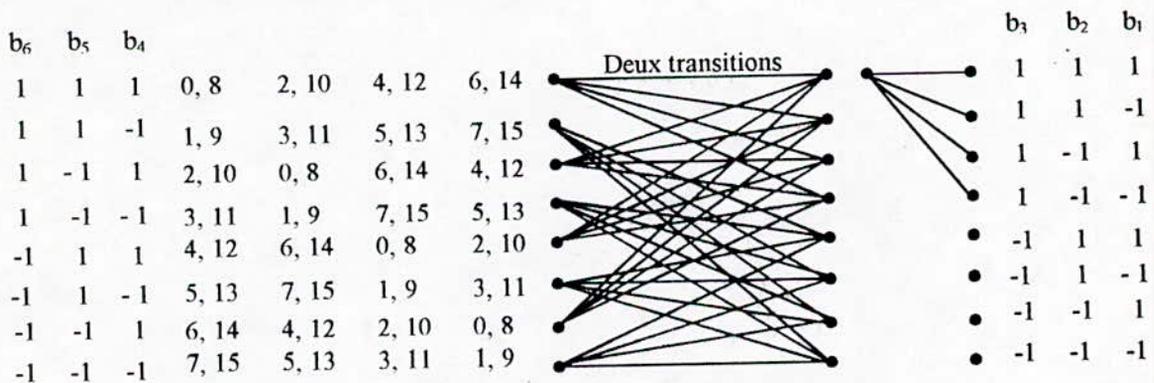


Figure 1.16 : La structure du treillis

La résolution de l'équation $D = \frac{1}{2^6} B^T \cdot X$ donne :

$d_1 = 8$ $d_4 = -1$ $d_{25} = -2$ $d_{36} = -4$ et les autres coefficients du vecteur D sont nuls.

La description analytique du codeur en treillis est donc :

$$S = 8.b_1 - 4.b_3.b_6 - 2.b_2.b_5 - b_4 \quad \dots(1.31)$$

L'implémentation du transmetteur est représentée sur la figure 1.17 :

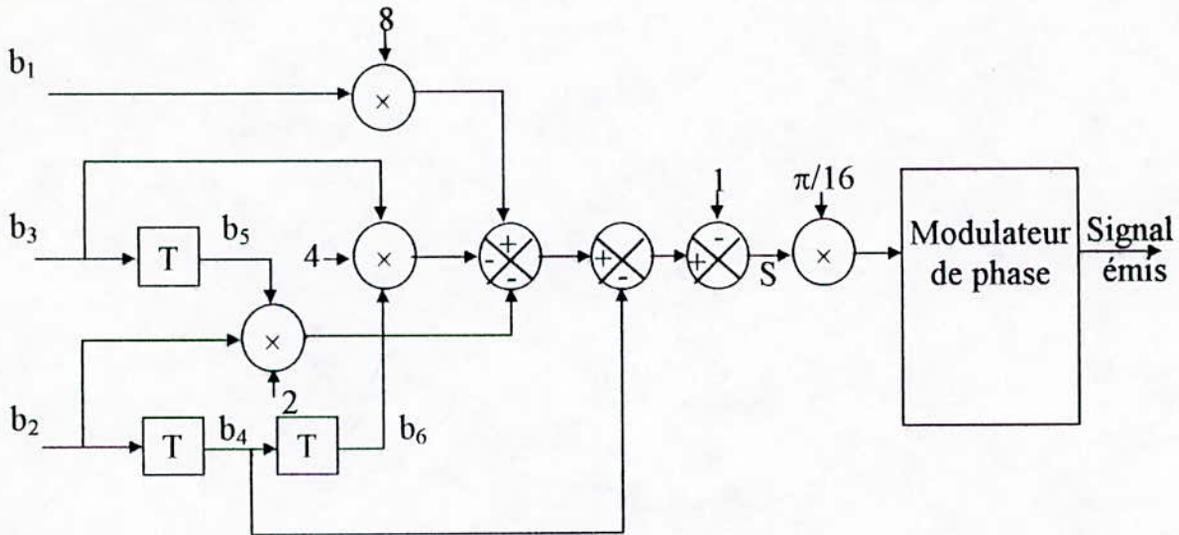


Figure 1.17 : Transmetteur de la description analytique de la modulation 16-PSK à 8 états

1-5-2-3 Exemple à deux dimensions :

- La modulation 16-QAM à 8 états : [1], [2], [5]

La partition d'ensemble de la modulation 16-QAM est illustrée dans la figure 1.18 :

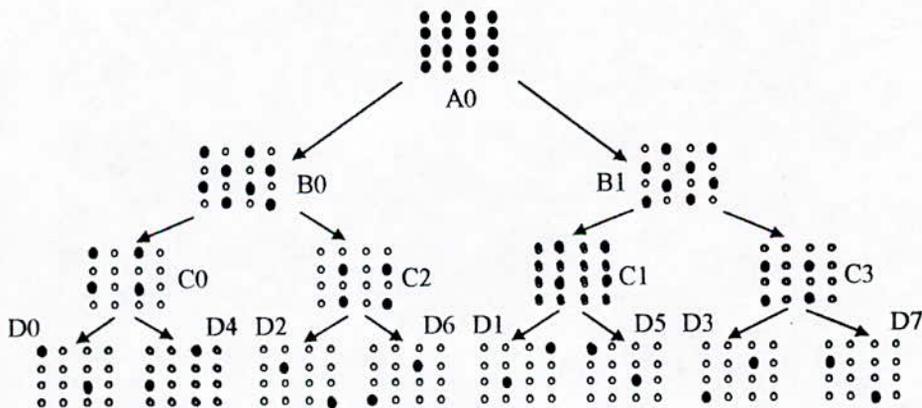


Figure 1.18 : la constellation de la modulation 16-QAM

La constellation 16-QAM que nous notons A0 est partitionnée dans une première étape en deux sous-ensembles B0 et B1 dont la distance minimale d_1 est $d_1 = \sqrt{2}.d_0$ où d_0 est la distance minimale dans A0.

Ensuite B0 est partitionné en C0 et C2 et B1 est partitionné en C1 et C3 tel que la distance minimale entre les sous-ensembles C_i ($i = 0,1,2,3$) est $d_2 = \sqrt{2}.d_1 = 2.d_0$.

Enfin, chaque sous-ensemble C_i est partitionné en deux sous-ensembles D ayant chacun deux points.

Chaque étape de partitionnement d'alphabet augmente la distance minimale dans les sous-ensembles de 3 dB.

Dans cette modulation nous avons $m = 3$, nous choisissons $m' = 2$ ce qui donne deux transitions parallèles partant de chaque état, nous considérons un codeur à 8 états c'est à dire le nombre d'éléments mémoires égale à 3.

Le treillis de cette modulation est illustré dans la figure 1.19 :

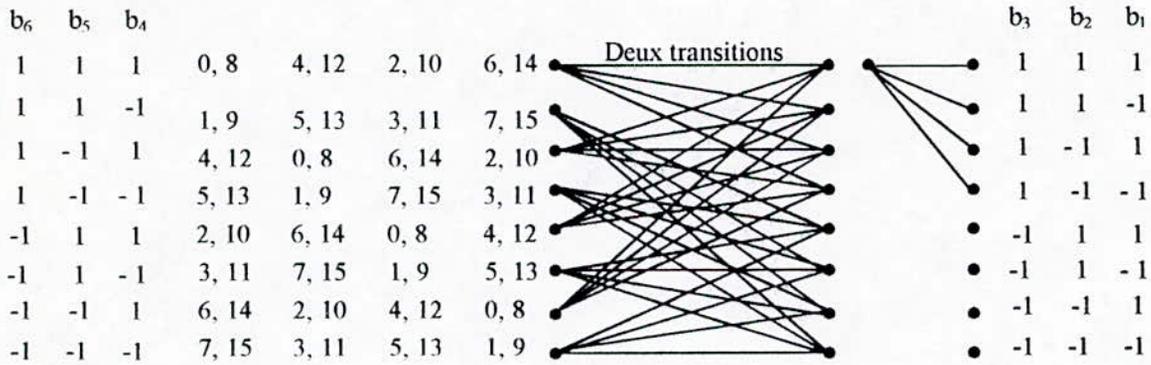


Figure 1.19 : Diagramme en treillis pour la modulation 16 QAM à 8 états

La résolution de l'équation (1.24) donne :

$$S = S_1 + j.S_2 \quad \dots(1.32)$$

où S_1 est la composante en phase et S_2 est la composante en quadrature.

$$\begin{aligned} S_1 &= 2.b_1.b_2.b_4.b_5 - b_3.b_4.b_6 \\ S_2 &= b_3.b_6 - 2.b_1 \end{aligned} \quad \dots(1.33)$$

1-5-3 Passage de la représentation analytique à la représentation d'Ungerboeck : [1]

Pour passer du codeur de Calderbank et Mazo au codeur d'Ungerboeck nous suivons la procédure de Turgeon :

Règle 1 : La première étape est de dresser une table qui assigne les bits de sorties Z_0, Z_1, \dots, Z_m aux signaux de la constellation.

Règle 2 : La deuxième étape est d'exprimer le signal de sortie S en fonction de Z_0, Z_1, \dots, Z_m ceci est accomplis en trouvant successivement pour chaque $Z_i (i=0, 1, \dots, m)$ le signal de différence dans chaque dimension entre $Z_i=1$ et $Z_i=-1$.

Règle 3 : Les $\delta_i/2$ sont assignés aux Z_i .

Finalement, le signal de sortie est exprimé en deux expression, une expression en fonction de Z_i et une autre en fonction des bits b_1, b_2, \dots, b_N . Nous trouvons ensuite la relation existante entre les bits de sorties Z_0, Z_1, \dots, Z_m et les bits b_1, b_2, \dots, b_N . Ces relations seront exploitées de la manière suivante :

- Une multiplication dans la convention ± 1 est équivalente à un XOR dans la convention $0, 1$.
- Les bits de sorties du codeurs d'Ungerboeck seront exprimés en fonction des bits de la séquence binaires a_0, a_1, \dots, a_N .

Dans ce qui suit nous traitons un exemple de passage de la méthode analytique (Calderbank et Mazo) à l'implémentation d'Ungerboeck.

Exemple 1.3 :

Soit l'exemple du codeur quatre états utilisant la modulation 4-AM, l'assignement des signaux de la constellation 4-AM en des bits Z_0, Z_1 est donné selon le tableau I-1 :

MSB	Z_1	1	1	-1	-1
LSB	Z_0	1	-1	1	-1
Signal de sortie	S	3	1	-1	-3

Tableau 1.1

$$S = 2 \cdot Z_1 + Z_0 \quad \dots(1.34)$$

des relations (1.30) et (1.34) :

$$\begin{aligned} Z_1 &= b_1 \cdot b_3 \\ Z_0 &= b_2 \end{aligned} \quad \dots(1.35)$$

Les bits de sorties $C_n^{(0)}, C_n^{(1)}$ du codeur d'Ungerboeck s'écrivent en fonction des bits a_1, a_2, a_3 :

$$\begin{aligned} y_1 &= a_1 \oplus a_3 \\ y_0 &= a_2 \end{aligned} \quad \dots(1.36)$$

- La réalisation d'Ungerboeck est :

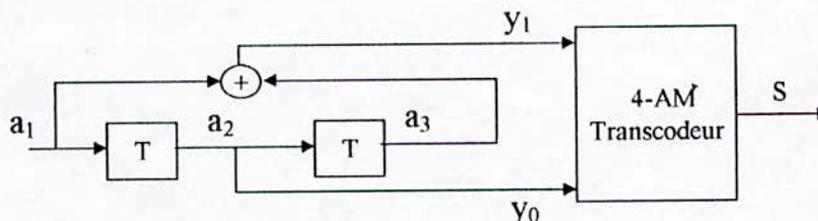


Figure 1.20 : Codeur d'Ungerboeck

1-6 Décodage de la TCM :

1-6-1 Introduction :

Il existe plusieurs algorithmes de décodage (détection) de la TCM, nous nous limiterons à présenter le plus utilisé : L'algorithme de Viterbi. Cet algorithme suit la règle de décision du maximum de vraisemblance, et recherche parmi toutes les séquences possibles de symboles que le codeur peut émettre, celle qui a la distance minimale de la séquence fournie par le démodulateur.

Pour présenter cet algorithme, supposons que le canal de transmission soit de type stationnaire gaussien et sans mémoire.

1-6-2 Décodage à maximum de vraisemblance : [3], [4]

Avant d'exposer le principe de l'algorithme de Viterbi, nous allons montrer que dans le cas d'un canal gaussien, le décodeur à vraisemblance maximale se réduit au décodage à distance euclidienne minimale.

Soit le modèle vectoriel d'un système de communication numérique illustré dans la figure 1.21 :

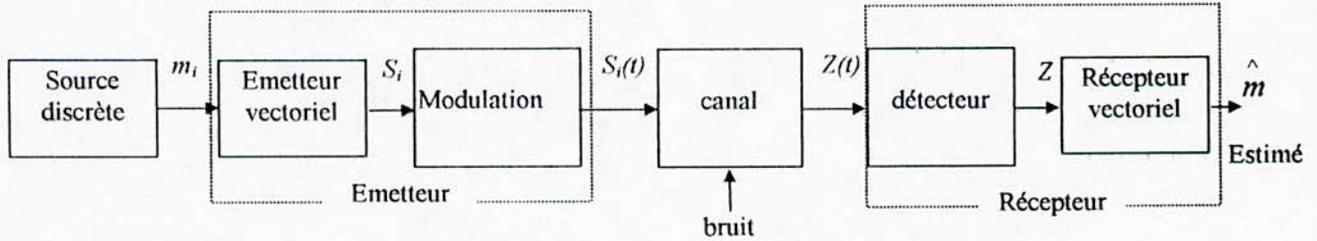


Figure 1.21 : **Modèle vectoriel d'un système de communication numérique**

Lorsque le vecteur Z est observé, on décide en faveur de m_i , c'est-à-dire $\hat{m} = m_i$, la probabilité moyenne de l'erreur pour cette décision est

$$P_e(m_i, Z) = P(m_i \text{ n'est pas transmis} / Z) = 1 - P(m_i \text{ transmis} / Z) \quad \dots(1.37)$$

Puisqu'on doit minimiser $P_e(m_i, Z)$, il est équivalent à maximiser $P_e(m_i \text{ transmis} / Z)$. Donc la règle de la décision optimale est

On décide pour $\hat{m} = m_i$, si

$$P(m_i \text{ émis} / Z) \geq P(m_k \text{ émis} / Z), \quad \forall k \neq i, \quad k = 1, \dots, M$$

Cette règle de décision est appelée décision au sens des maximum à posteriori ou (MAP). En utilisant la loi de Bays, on peut écrire la règle précédente comme suit

On décide pour $\hat{m} = m_i$, si

$$\frac{P_k f_z(Z/m_k)}{f_z(Z)} \text{ est maximum pour } k = i.$$

mais pour des symboles équiprobables $P_k = \frac{1}{M}$.

D'autre part $f_z(Z)$ est indépendant du signal émis. Donc la règle de décision devient

On décide pour $\hat{m} = m_i$, si

$$f_z(Z/m_k) \text{ est maximum pour } k = i,$$

où $f_z(Z/m_k)$ est appelée fonction de vraisemblance qu'on remplace souvent par son logarithme

qui donne : On décide pour $\hat{m} = m_i$, si

$$\ln[f_z(Z/m_k)] \text{ est maximum pour } k = i.$$

Une telle décision est appelée « décision au sens du maximum de vraisemblance ». Donc cette règle peut être interprétée de la manière suivante :

On calcule $\ln[f_z(Z/m_k)]$, $i = 1, 2, \dots, M$, ensuite on compare et on décide en faveur du maximum donnée par $i = k$.

Il est préférable de donner une interprétation graphique de la décision de maximum de vraisemblance. Soit R_1, R_2, \dots, R_M , les zones correspondantes aux points message $S_i = (\alpha_{i1} \alpha_{i2} \dots \alpha_{in})^T$ représentées par les signaux $S_i(t)$, $i = 1, 2, \dots, M$, et comme indiqué à la figure 1.22 :

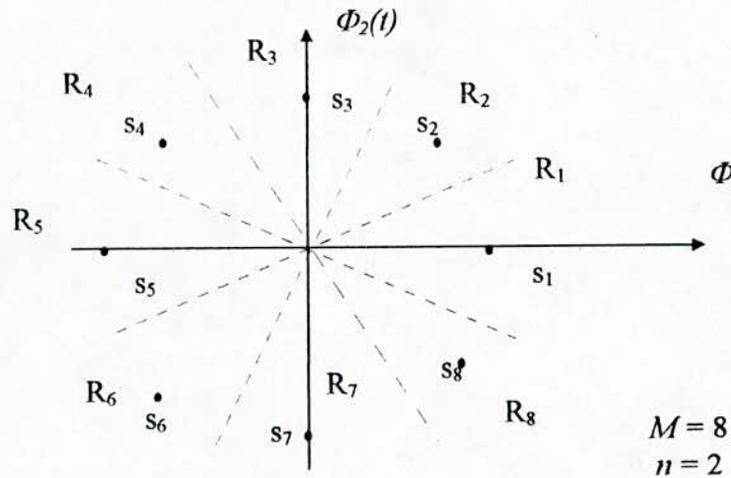


Figure 1.22 : Huit zones de décision (R_1, R_2, \dots, R_8)

La décision MV peut être formulée de la manière suivante :

$$\left\{ \begin{array}{l} \text{Le point d'observation représenté par le vecteur } Z \text{ se} \\ \text{trouve dans la région } R_i \text{ (c'est-à-dire on décide en faveur} \\ \text{de } m_i) \text{ si} \\ \ln[f_z(Z / m_k)] \text{ est maximum pour } k = i. \end{array} \right.$$

Pour un canal gaussien

$$f_z(Z / m_k) = (\pi N_0)^{-\frac{n}{2}} \cdot e^{-\frac{1}{N_0} \sum_{j=1}^n (z_j - \alpha_{kj})^2}, \quad k = 1, \dots, M \quad \dots(1.38)$$

$$\ln f_z(Z / m_k) = -\frac{n}{2} \ln(\pi N_0) - \frac{1}{N_0} \sum_{j=1}^n (z_j - \alpha_{kj})^2, \quad i = 1, \dots, M \quad \dots(1.39)$$

La fonction $\ln[f_z(Z / m_k)]$ est maximum lorsque $\sum_{j=1}^n (z_j - \alpha_{kj})^2$ est minimum.

Donc la règle de décision, donnée ci-dessus devient :

On décide en fonction du symbole m_i , (c'est-à-dire Z se trouve dans la région R_i) si

$$\|Z - S_k\|^2 = \sum_{j=1}^n (z_j - \alpha_{kj})^2 \text{ est minimum pour } k = i.$$

On remarque $\|Z - S_k\|^2$ n'est que la distance euclidienne entre les valeurs z_i et α_{ki} qui représente la distance entre le point d'observation et le point message. Dans la dernière règle donnée ci-dessus, on cherche le point message le plus proche du point observé et ce au sens d'Euclid.

1-6-3 Algorithme de Viterbi : [3], [5], [11]

Pour la présentation de l'algorithme de Viterbi, nous considérons la modulation 4-PSK avec un codeur à 4 états où la représentation en treillis est à la figure 1.4.

Le décodage à maximum de vraisemblance consiste à choisir le chemin correspondant à la séquence des symboles qui diffère de la séquence reçue en une distance euclidienne minimale.

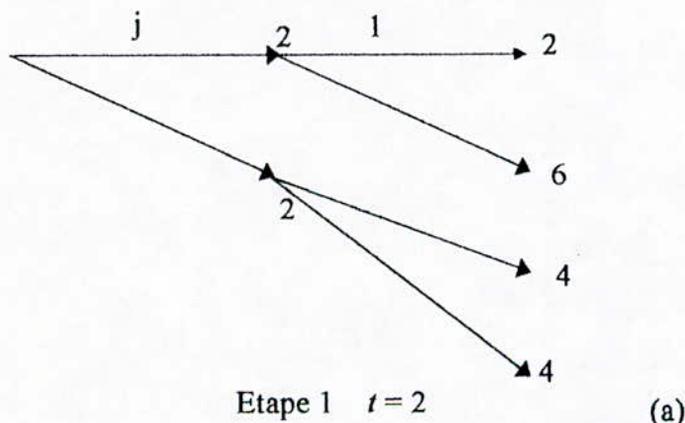
L'algorithme peut être divisé en deux parties :

- la première s'étend du début du treillis jusqu'au niveau $t = K - 1$, où K est la longueur de contrainte. Le décodage consiste à attribuer une distance nulle à l'état initial, et à chaque autre état σ , atteint par la transition d'un état σ' , un carré de distance, qui est la somme des carrés de la distance à l'état σ' et la distance de transition. La distance de transition à un niveau t du treillis entre deux états σ et σ' , est la distance euclidienne entre le symbole correspondant à cette transition du codeur et le symbole de la séquence reçue.
- La deuxième partie s'étend du niveau $t = K$ jusqu'au niveau $t = [L/m] + K - 1$ où L est la largeur du message et de m le nombre des entrées du codeur.

Le décodage consiste à choisir la branche pour laquelle la distance au noeud σ sera minimale et éliminer toutes les autres. La branche choisie est appelée *survivant*. Si le choix de plusieurs branches donnera la même distance minimale au noeud σ , alors peu importe, on choisira une de ces branches. Cette opération est répétée avec tous les 2^m états, et cela pour tous les niveaux de cette partie. A la fin de cette partie nous obtiendrons 2^m chemins, desquels le décodeur choisira celui qui a la distance la plus proche, au sens d'Euclid de la séquence émise.

En résumé, l'algorithme de Viterbi peut être mis en œuvre de façon commode à partir de diagramme en treillis en appliquant les règles suivantes :

1. A partir du $K^{\text{ème}}$ étape (niveau) du diagramme où 2^m branches arrivent à chaque état, calculer pour chacun des états, la distance euclidienne de chacun des 2^m chemins qui arrivent à l'état. Pour chaque état, conserver le chemin dont la distance euclidienne est la plus faible, (le *survivant*) et éliminer les autres.
2. Répéter l'opération pour chaque étape t du treillis tant que $t \leq [L/m] + K - 1$.
3. Choisir le chemin dont la distance euclidienne est la plus petite et éliminer tous les autres.



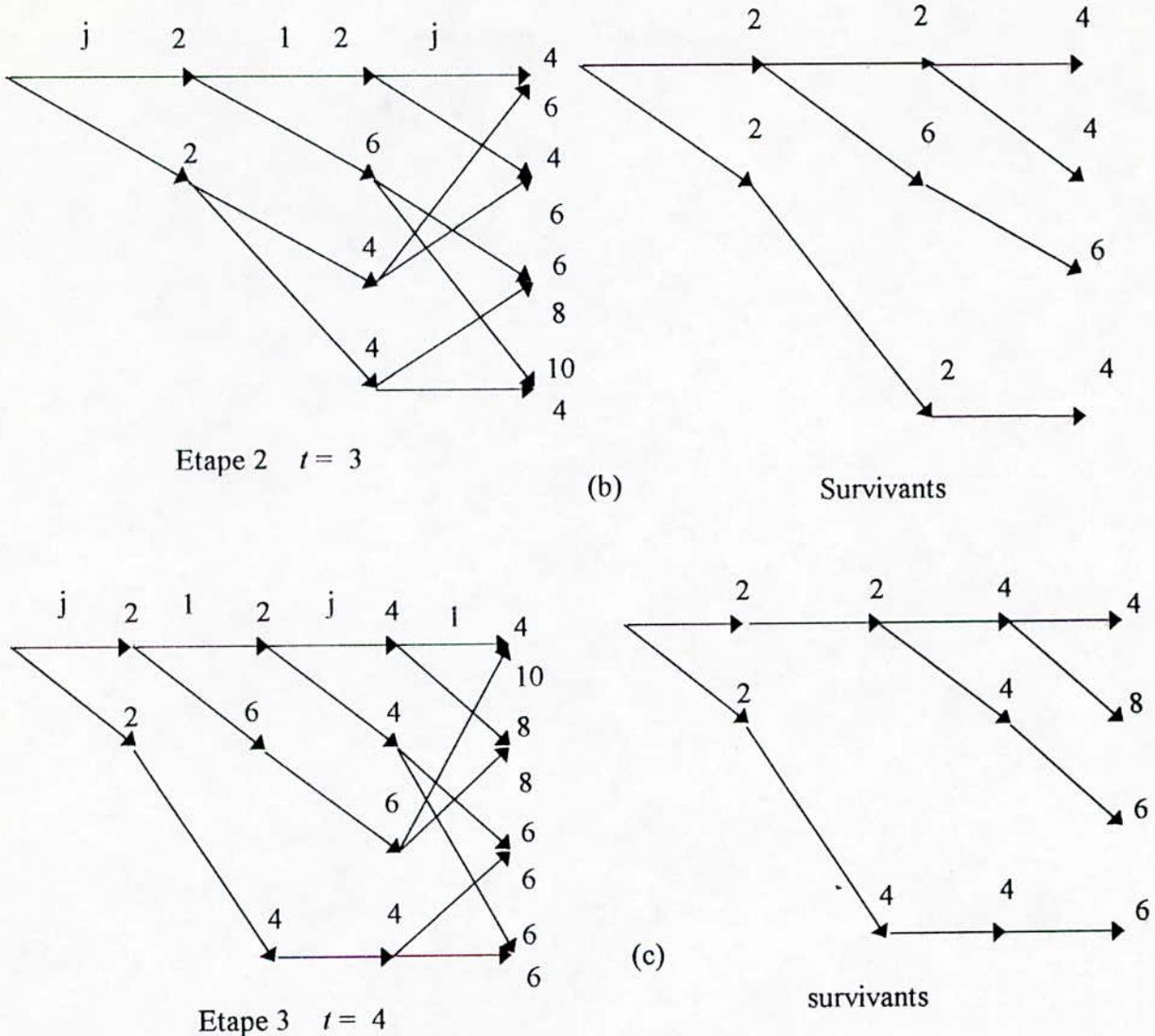


Figure 1.23 : Exemple de décodage utilisant l'algorithme de Viterbi

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

Toutefois, lorsqu'on examine les différents survivants à une étape t , on remarque qu'avec une grande probabilité, ils ont tous le même passé entre les étapes 0 et $t-dp$ où dp est la profondeur de décision.

On peut par conséquent décider définitivement à l'étape t toutes les données émises jusqu'à l'étape $t-dp$. Comme règle empirique, on peut fixer le paramètre dp égal à 5 ou 6 fois le nombre d'états du décodeur. Avec ces valeurs les performances du décodeur sont quasiment les mêmes que lorsque la décision se fait en fin de la séquence.

Cette stratégie permet de réduire la complexité du décodeur et le retard de décision. La complexité d'un décodeur de Viterbi est proportionnelle au nombre d'états de treillis.

Celui-ci croît exponentiellement avec la longueur de contrainte. Pour cette raison les codes utilisés en pratique ont une longueur de contrainte inférieure à 10.

Exemple 1.4 :

Soit le codeur de la figure (1.4), ce codeur génère une séquence de n symboles égaux à 1, laquelle après transmission à travers le canal donne la séquence reçue (j 1 j 1) comprenant deux erreurs. La figure 1.23 a montre la première partie de décodage. La distance euclidienne à chaque état y est indiquée.

Les figures 1.23 b et 1.23 c montrent la deuxième partie, chacune de ces figures compte deux treillis, dans celui de gauche où à conserver tous les chemins aboutissant à chaque noeud , tandis que dans celui de droite on a gardé que les survivants. Le chemin conservé après la décision finale est celui dont l'erreur cumulée est la plus petite , c'est à dire le chemin (a a a a) (a : c'est l'état 00) correspondant à la séquence des 1.

1-7 Evaluation des performances :

Dans ce paragraphe nous considérons la probabilité d'erreur et la distance libre (dfree) des configuration TCM. le fameux paramètre commun utilisé pour l'évaluation de la qualité d'un système de communication numérique est la probabilité d'erreur.

Nous découvrirons que la probabilité d'erreur est asymptotiquement bornée par une fonction monotone décroissante quand la distance dfree croit.

Effectivement ceci montre que la distance dfree est le seul paramètre significatif pour la comparaison des systèmes TCM utilisés pour la transmission dans canal bruité par un bruit blanc gaussien additif (AWGN).

1-7-1 Notion d'événement d'erreur : [6], [12]

Un événement d'erreur est formé par un couple de séquences distinguées, ayant le même état de départ et le même état d'arrivée dans le treillis après un certain nombre de pas.

Un événement d'erreur de longueur L est donné par deux séquences :

$$(S_t) = (\sigma_t, \dots, \sigma_{t+L-1})$$

$$(S'_t) = (\sigma'_t, \dots, \sigma'_{t+L-1})$$

telles que : $\sigma_t = \sigma'_t$ et $\sigma_{t+L} = \sigma'_{t+L}$

$$\sigma_i \neq \sigma'_j \quad i = t+1, \dots, t+L-1$$

où σ_t représente l'état du codeur à l'instant t.

Le schéma de la figure 1.24 représente deux chemins divergent du même état σ_t et convergent vers le même état σ_{t+L} .

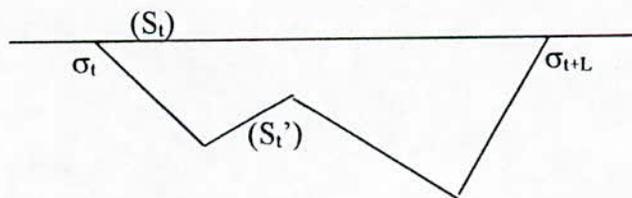


Figure 1.24 : Exemple d'un événement d'erreur

1-7-2 La borne supérieure de la probabilité d'erreur : [1]

Considérons le modèle d'Ungerboeck de rendement $m/(m+1)$ selon la figure 1.25 :

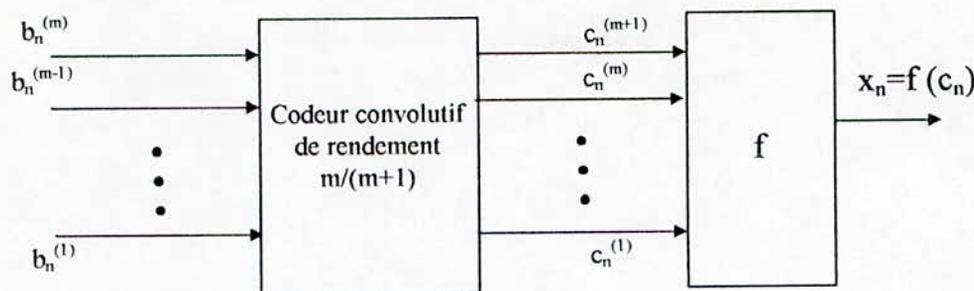


Figure 1.25 : Modèle d'Ungerboeck

Un événement d'erreur défini ci-dessus est décrit par deux séquences de longueur L de symboles codés qui sont $[c_K, c_{K+1}, \dots, c_{K+L-1}]$ et $[c'_K, c'_{K+1}, \dots, c'_{K+L-1}]$ tels que :

$$c'_K = c_K \oplus e_K, \dots, c'_{K+L-1} = c_{K+L-1} \oplus e_{K+L-1}$$

où $e_i, i = K, \dots, K+L-1$, forment une séquence d'erreur binaire.

Soient X'_L et X_L deux séquences de symboles de longueur L, quand le démodulateur à maximum de vraisemblance décide en faveur de la séquence erronée X'_L au lieu de la séquence correcte X_L dans ce cas un événement d'erreur apparaît.

La probabilité d'erreur est obtenue en sommant sur tout L, $L=1, 2, \dots$, les probabilités d'événements d'erreurs, le principe d'union annonce que :

$$P(e) \leq \sum_{L=1}^{\infty} \sum_{X_L} \sum_{X'_L \neq X_L} P[X_L] \cdot P[X_L \rightarrow X'_L] \quad \dots(1.40)$$

Puisqu'il ya une correspondance entre x_n et c_n la formule de la probabilité d'erreur devient :

$$P(e) \leq \sum_{L=1}^{\infty} \sum_{C_L} P[C_L] \cdot \sum_{C'_L \neq C_L} P[C_L \neq C'_L] \quad \dots(1.41)$$

La transmission est faite dans un canal (AWGN) où le bruit possède une densité spectrale ($N_0/2$) et la détection est à maximum de vraisemblance, $P(C_L \rightarrow C'_L)$ peut être borner selon la limite de Bhattacharia et l'expression de la P(e) devient :

$$P(e) \leq \sum_{L=1}^{\infty} \sum_{E_L \neq 0} W[E_L] \quad \dots(1.42)$$

avec $W(E_L) = \sum_{C_L} P(C_L) \cdot Z^{|f(C_L) - f(C_L \oplus E_L)|^2}$ et $Z = \exp\left(\frac{-1}{4 \cdot N_0}\right)$

L'équation (1.42) montre que P(e) est bornée par une somme sur les événements d'erreurs possibles en fonction du vecteur E_L qui cause les événements d'erreurs. Pour évaluer P(e) il suffit d'énumérer ces vecteurs d'erreurs ceci est fait en utilisant la fonction de transfert du diagramme d'état d'erreur .

1-7-2-1 La matrice d'erreur : [1]

La matrice d'erreur $G(e_i)$ est défini comme suit :
L'entrée p, q de $G(e_i)$ est zéro si aucune transition de l'état p vers l'état q n'est possible

$$[G(e_i)]_{p,q} = \frac{1}{M} \cdot \sum_{C_{p \rightarrow q}} D^{|f[C_{p \rightarrow q}] - f[C_{p \rightarrow q} \oplus e_i]|^2} \quad \dots(1.43)$$

$C_{p \rightarrow q}$ est le vecteur généré par la transition de l'état p vers l'état q.

$$W(E_{i,l}) = \frac{1}{N} \cdot I^T \cdot \prod_{n=1}^L G(e_n) \cdot I \quad \dots (1.44)$$

I est un vecteur de dimension (N×1) contenant des 1, les éléments p, q de la matrice

$\prod_{n=1}^L G(e_n)$ énumère les distances euclidiennes entraînées dans la transition de l'état p vers l'état q en L étapes.

Nous observons que les vecteurs erreurs e_1, e_2, \dots, e_L dans la séquence ne sont pas indépendants.

Pour le calcul de P(e) nous avons besoin d'un modèle prenant en compte leur dépendance. Cette tâche sera accomplie par la description des connections entre les vecteurs le long du diagramme d'état.

Le diagramme d'état d'erreur a une structure déterminée seulement par le code linéaire convolutif et diffère du diagramme d'état de code seulement dans la dénomination de ces états et l'étiquetage des branches.

1-7-2-2 La fonction de transfert et probabilité d'erreur : [1]

Nous avons :

$P(e) \leq T(D) \Big|_{D=e^{-\frac{1}{4N_0}}}$ où T(D) est la fonction de transfert du diagramme d'état d'erreur définie par la formule :

$$T(D) = \frac{1}{N} \cdot I^T \cdot G \cdot I \quad \dots (1.45)$$

où $G = \sum_{L=1}^{\infty} \sum_{E_L \neq 0} \prod_{n=1}^L G(e_n)$ est la matrice fonction de transfert du diagramme d'état d'erreur.

Exemple 1.5 :

Considérons la configuration TCM où son diagramme en treillis est donné par la figure 1.26 :

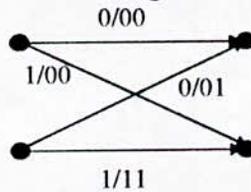


Figure 1.26 : Diagramme en treillis à 4 états

Le diagramme d'état d'erreur est donné par la figure 1.27 :

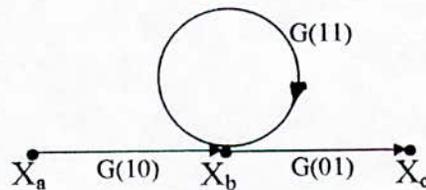


Figure 1.27 : diagramme d'état d'erreur

$$\begin{aligned} X_a &= G(10) \cdot X_a + G(11) \cdot X_b \\ X_c &= X_b \cdot G(01) \end{aligned} \quad \dots (1.46)$$

$$G(e_2 e_1) = \frac{1}{2} \cdot \left[\begin{array}{l} D^{\|f(00)-f(e_2 e_1)\|^2} D^{\|f(10)-f(\bar{e}_2 \bar{e}_1)\|^2} \\ D^{\|f(01)-f(e_2 \bar{e}_1)\|^2} D^{\|f(11)-f(\bar{e}_2 e_1)\|^2} \end{array} \right] \dots(1.48)$$

Si on considère la modulation 4-PSK :

$f(00) = 1$, $f(01) = j$, $f(10) = -1$, $f(11) = -j$

Le calcul de la fonction transfert T(D) donne :

$$T(D) = \frac{D^6}{1-D^2}$$

La probabilité d'erreur P(e) est bornée selon la formule :

$$P(e) \leq \frac{e^{\frac{-3}{2.N_0}}}{1 - e^{\frac{-1}{2.N_0}}} \dots(1.49)$$

1-7-2-3 Interprétations : [1]

L'inspection de la matrice G nous amène à considérer différents degrés de symétrie dans la configuration TCM . Pour la modulation on remarque que l'exposant de D pour G(01) est le même pour G(11) et G(10).

Il y a une symétrie et ceci peut être interprété comme étant tout les trajets dans le treillis contribuent d'une façon égale à la probabilité d'erreur.

Donc dans l'analyse des configurations TCM, nous prenons n'importe quel trajet comme référence et nous calculons la probabilité d'erreur.

1-7-2-4 Amélioration de la borne supérieure de P(e) : [1]

Une borne supérieure serrée peut être obtenue en substituant à la borne de Bhattacharyya l'expression :

$$P(C_L \rightarrow C_L') = \frac{1}{2} \operatorname{erfc} \left(\frac{\|f(C_L) - f(C_L')\|}{2\sqrt{N_0}} \right) \dots(1.50)$$

où :
$$\operatorname{erfc}(X) = \frac{2}{\sqrt{\pi}} \int_X^\infty \exp(-u^2) . du$$

$$P(e) \leq \frac{1}{2} \operatorname{erfc} \left(\frac{d_{free}}{2\sqrt{N_0}} \right) e^{\left(\frac{d_{free}}{4.N_0} \right)} . T(D) \text{ pour } D = e^{\left(\frac{-1}{4.N_0} \right)} \dots(1.51)$$

pour un rapport signal sur bruit suffisamment élevé :

$$p(e) \approx N(d_{free}) . \frac{1}{2} . \operatorname{erfc} \left(\frac{d_{free}}{2\sqrt{N_0}} \right) \dots(1.52)$$

où N(dfree) dénote la multiplicité moyenne de l'erreur avec une distance dfree (c'est le nombre moyen de voisins à une distance dfree).

1-7-3 Probabilité d'erreur par bit P_b(e) : [3], [6], [1]

Ce paramètre est considéré comme une mesure importante de la qualité du système. Nous pouvons montré que :

$$p(e) \approx N(df_{free}) \cdot \frac{1}{2} \cdot \operatorname{erfc}\left(\frac{df_{free}}{2\sqrt{N_0}}\right) \quad \dots(1.52)$$

où $N(df_{free})$ dénote la multiplicité moyenne de l'erreur avec une distance df_{free} (c'est le nombre moyen de voisins à une distance df_{free}).

1-7-3 Probabilité d'erreur par bit $P_b(e)$: [3], [6], [1]

Ce paramètre est considéré comme une mesure importante de la qualité du système. Nous pouvons montrer que :

$$P_b(e) \leq \frac{1}{m} \sum_i \sum_j N(d_i) b_j \cdot P(d_i) \quad \dots(1.53)$$

où m est le nombre d'entrées du codeur et b_j est le nombre de bits erronés dans un événement d'erreur ayant une distance d_i .

Une borne supérieure pour $P_b(e)$ peut aussi être obtenue :

$$P_b(e) \leq \frac{1}{m} \cdot \frac{\partial}{\partial I} T(D, I) \Big|_{I=1, D=e^{\frac{-1}{4 \cdot N_0}}} \quad \dots(1.54)$$

Une autre borne supérieure est obtenue :

$$P_b(e) \leq \frac{1}{2 \cdot m} \cdot \operatorname{erfc}\left(\frac{df_{free}}{2\sqrt{N_0}}\right) \cdot \exp\left(\frac{df_{free}^2}{4 \cdot N_0}\right) \cdot \frac{\partial}{\partial I} T(D, I) \Big|_{I=1, D=e^{\frac{-1}{4 \cdot N_0}}} \quad \dots(1.55)$$

1-7-4 Calcul de la fonction de transfert : [1]

Le calcul de la fonction de transfert $T(D, I)$ est accomplis en deux étapes :

- L'évaluation pour chaque vecteur erreur, le poids d'Euclide $W(e)$.
- ensuite chaque $W(e)$ est multiplié par I^ε où ε est le nombre de bits erronés causés par e . Les branches du diagramme d'état d'erreur sont étiquetées par les poids $W(e)$ calculés.

Exemple 1.6 :

$m = 2$ (le nombre d'entrées du codeur en treillis).

Le poids d'erreur est donné par :

$$W(e_3 e_2 e_1) = \frac{1}{4} \cdot \sum D^{|f(C) - f(C \oplus e)|^2} \quad \dots(1.56)$$

Considérons typiquement le calcul de $W(011)$. La procédure est illustrée dans le tableau 1.2 :

C(1)	C(1)⊕e	d[f(C(1)), f(C(1)⊕e)]
000	011	3,414
010	001	0,586
100	111	3,414
110	101	0,586

Tableau 1.2

$$W(011) = \frac{D^{0,586} + D^{3,414}}{2} \quad \dots(1.57)$$

Pour un codeur en treillis à deux états : le diagramme d'état d'erreur est illustré par la figure 1.29 :

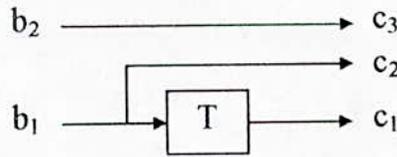
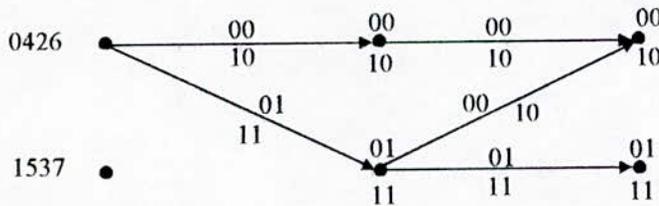
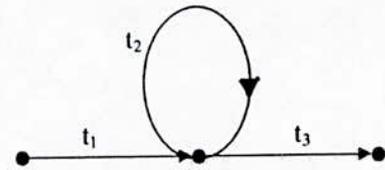


Figure 1.28 : Codeur à deux états de rendement 2/3



(a): Diagramme en treillis



(b): Diagramme d'état d'erreur

Figure 1.29

$$\begin{aligned}
 t_1 &= W(010).I + W(110).I^2 \\
 t_2 &= W(011).I + W(111).I^2 \quad \dots(1.58) \\
 t_3 &= W(001).I + W(101).I^2
 \end{aligned}$$

La fonction de transfert $T(D,I)$ est :

$$T(D,I) = \frac{t_1 \cdot t_3}{1 - t_2} \quad \dots(1.59)$$

1-7-5 Calcul de la distance libre d_{free} :

Le calcul de d_{free} est une mesure de la qualité de performance d'un système de transmission numérique utilisant la TCM. Il existe plusieurs méthode de calcul de d_{free} , nous citons la technique de calcul utilisant la fonction de transfert.

1-7-5-1 La technique de calcul utilisant la fonction de transfert : [1]

Cette technique est basée sur le diagramme d'état d'erreur, la fonction de transfert contient l'information sur d_{free} , le développement de $T(D)$ en série et le plus petit exposant de D est d_{free}^2 , nous obtenons cet exposant en utilisant des méthodes numériques.

$$T(D) = N(d_{free}).D^{d_{free}^2} + N(d_{next}).D^{d_{next}^2} + \dots \dots \dots \quad \dots(1.60)$$

d_{next} est la deuxième petite distance euclidienne.

$$d_{free}^2 = \frac{LnT(D)}{LnD} + \varepsilon(D) \quad \dots(1.61)$$

$\varepsilon(D)$ est une fonction positive tendante vers zéro lorsque D tend vers zéro.

1-8 Conclusion :

Dans ce chapitre, nous avons présenté des généralités et des principes de base de la TCM, nous avons donné deux méthodes de réalisation de la TCM qui sont la représentation d'Ungerboeck et la méthode analytique. Cette dernière est plus simple à implémenter par rapport à la première.

Nous avons traité quelques exemples montrant les deux représentations ainsi que le passage de l'une vers l'autre tout en donnant les partitions d'ensembles et les schémas des codeurs.

Nous avons démontré qu'une mesure asymptotique de la performance du gain d'un système TCM est la comparaison de la distance euclidienne d_{free} du codeur en treillis relative à la distance minimum de la modulation non codée. Cette performance de mesure est une indication de la réduction maximum dans l'énergie du bit sur bruit $\frac{E_b}{N_0}$ requis.

Le but du prochain chapitre sera l'étude et la conception des codeurs optimaux utilisés dans cette TCM.

CHAPTER

2

Chapitre 2

Codeurs optimaux pour la TCM à deux dimensions

2-1 Introduction :

Nous avons vu dans le chapitre précédent que la probabilité d'erreur dans un système de transmission est une fonction du rapport signal sur bruit de la liaison. Pour augmenter la qualité de la transmission, on est donc amené à augmenter ce rapport, soit en augmentant la puissance du signal émis, soit en réduisant le facteur du bruit du récepteur. Malheureusement, cela n'est pas toujours possible et on se heurte très vite à une limitation d'ordre technologique ou économique, c'est pour cela qu'on fait appel au codage de canal.

Le codage de canal consiste à ajouter aux symboles messages à transmettre des symboles de contrôle. La nécessité d'introduire de la redondance est pour protéger l'information contre les erreurs de transmission.

Les codes correcteurs d'erreurs peuvent être classés en deux grandes catégories :

- Les codes en bloc.
- Les codes convolutifs.

Les codes convolutifs sont très utilisés par rapport aux codes en bloc parce qu'ils sont simples à concevoir et sont plus performants.

Le but principal de ce chapitre est l'établissement d'un algorithme de recherche des codes optimaux utilisés en TCM.

Nous débuterons ce chapitre par la structure algébrique des codeurs convolutifs, nous donnerons quelques définitions sur les codeurs (équivalents, optimaux, canoniques,... etc.), ensuite nous évoquerons le concept d'un codeur systématique à boucle de retour ainsi que la passage d'un codeur convolutif à un codeur systématique, nous verrons le concept de la matrice de parité, les règles de rejection d'Ungerboeck utilisées dans la recherche des codeurs optimaux. Nous terminerons le chapitre en donnant l'organigramme de cette recherche.

2-2 Codes convolutifs : [9], [5], [4]

Un codeur convolutif est défini comme étant un circuit linéaire séquentiellement constant. Un codeur convolutif binaire est un système à mémoire finie qui génère n bits binaires chaque fois que l'on présente m bits d'informations à son entrée. Contrairement aux codes en bloc, les n bits de sorties ne dépendent pas seulement du bloc de m bits d'informations à l'entrée du codeur mais aussi des D blocs précédents.

Les codes convolutifs introduisent ainsi un effet mémoire d'ordre D , ($D+1$) étant la longueur de contrainte. le schéma général du codeur convolutif est illustré à la figure 2.1 :

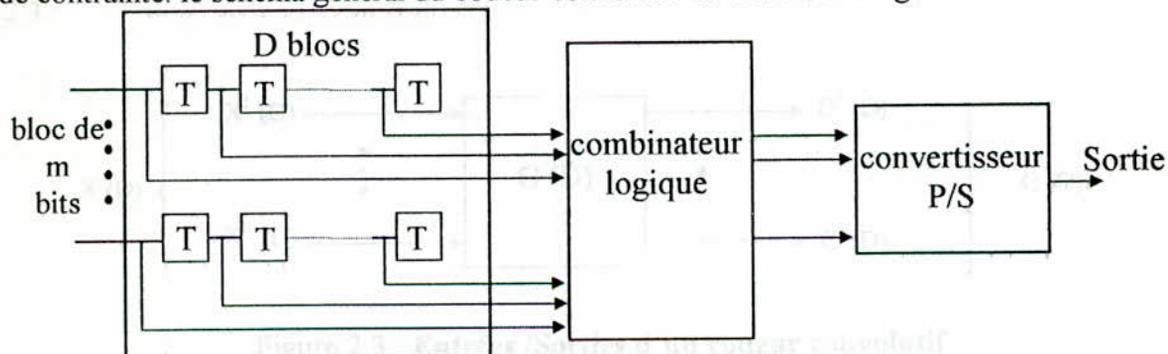


Figure 2.1 : Schéma général d'un codeur convolutif

$G(D)$: représente la fonction de transfert.

$C(D)$: représente la séquence de sortie sous forme polynomiale.

Où D est le retard unité.

La relation entrée / sortie est donnée par la relation :

$$C(D) = G(D). X(D) \quad \dots(2.3)$$

La matrice de parité $H(D)$ vérifie les conditions :

$$\begin{aligned} C(D).H^T(D) &= 0 \\ G(D).H^T(D) &= 0 \end{aligned} \quad \dots(2.4)$$

T : représente la transposée.

Si le code est systématique, la matrice génératrice est simplement déduite de la matrice de parité et vice versa. La longueur de contrainte d'un codeur convolutif est définie comme étant le degré maximum du polynôme générateur $G(D)$.

Pour un codeur à 4 états illustré dans la figure 2.4 :

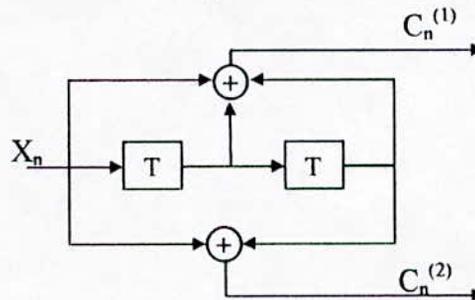


Figure 2.4 : Codeur à 4 états

Les sorties de ce codeur à l'instant n sont données par :

$$\begin{aligned} C_n^{(1)} &= X_{n-2} \oplus X_{n-1} \oplus X_n \\ C_n^{(2)} &= X_n \oplus X_{n-2} \end{aligned}$$

En utilisant le retard unité D :

$$\begin{aligned} C^{(1)}(D) &= D^2.X(D) \oplus D.X(D) \oplus X(D) \\ C^{(2)}(D) &= X(D) \oplus D^2.X(D) \end{aligned}$$

Les matrices génératrice et parité sont :

$$\begin{aligned} G(D) &= [1 \oplus D^2, 1 \oplus D \oplus D^2] \\ H(D) &= [1 \oplus D \oplus D^2, 1 \oplus D^2] \end{aligned} \quad \dots(2.5)$$

2-3 Définitions : [9]

- Deux codeurs sont dits équivalents s'ils génèrent le même code.
- Soit un codeur G réalisé avec v éléments mémoires, G_m est dit codeur minimal (optimal) s'il est réalisé avec μ éléments mémoires tel que : $\mu \leq v$
- Un codeur est dit canonique s'il est réalisé avec μ éléments mémoires où μ est le nombre minimal d'éléments mémoires dans tous les codeurs équivalents, nous avons ainsi un codeur équivalent minimal.

- Les codeurs optimaux sont considérés comme une classe des codeurs canoniques.
- Chaque codeur convolutif est équivalent à un codeur systématique et chaque codeur systématique est canonique.

2-3-1 Propriété d'un codeur minimal : [9]

Le codeur minimal a une propriété appelée le degré de prédictibilité utilisé dans l'analyse des événements d'erreurs.

2-3-2 Avantage de la structure d'un codeur systématique : [2]

L'avantage de la structure d'un codeur systématique est qu'aucune supposition au préalable n'est nécessaire sur les degrés individuels v_j des polynômes générateurs. Le codeur systématique minimal doit vérifier :

$$v = \sum_{j=1}^{m'} v_j \quad \dots (2.6)$$

où v est le nombre d'éléments mémoires du codeur systématique.

2-3-3 Codes catastrophique : [7]

Un code est dit catastrophique lorsqu'il ya un nombre infini d'événements d'erreurs ayant même distance minimum.

Soit la partition de la constellation de la modulation 4-PSK illustrée dans la figure 1.5, considérons la transmission de 2 bits/symbole utilisant un bit pour sélectionner un des deux sous-ensembles et l'autre bit sélectionne un point dans ce sous-ensemble, ceci peut être représenté par un diagramme en treillis à deux états comme le montre la figure 2.5 :

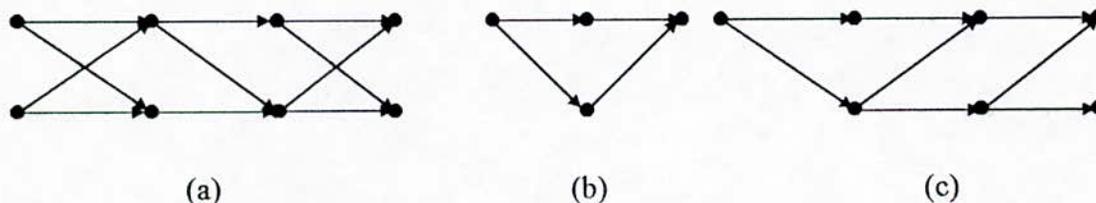


Figure 2.5 : (a) : Le treillis pour un codeur à 2 états

(b) : Événement d'erreur de longueur 2 avec une distance minimum

(c) : Un nombre infini d'événement d'erreur avec même distance minimum

De la figure 2.5 b nous assumons un seul événement d'erreur probable de longueur 2 ayant une distance minimum égale à 2, ce qui donne un gain de 3 dB par rapport au système non codé, de la figure 2.5 c nous avons un nombre infini d'événements d'erreur ayant même distance d'où découle le concept d'un codeur catastrophique.

2-4 Codes convolutifs pour la TCM : [10], [2]

Les modulations codées sont basées sur le principe de la partition d'alphabet qui à été introduit par Ungerboeck.

Les modulations codées conventionnelles utilisent une constellation à 2^{m+1} points, pour transmettre m bits d'information par symbole émis. Le gain de codage par rapport à une modulation non codée équivalente est déterminé par le niveau de partition de l'alphabet et le

codage utilisé. Parmi les m bits d'information à l'entrée du codeur, m' bits entrent dans un codeur convolutif de rendement $\frac{m'}{m'+1}$ et les $(m'+1)$ bits en sortie de celui-ci sélectionnent un sous-ensemble de la constellation.

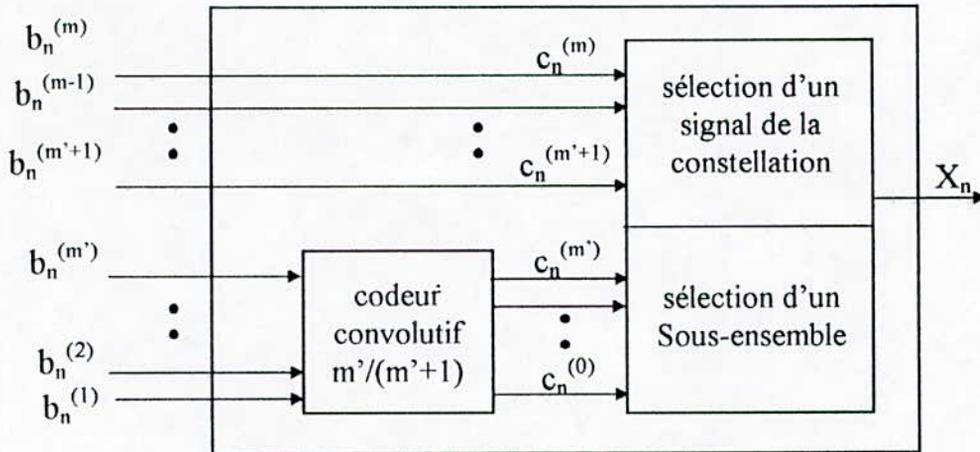


Figure 2.6 : Structure générale d'un codeur/modulateur

A chaque instant n , le codeur convolutif reçoit m' bits d'informations et génère $(m'+1)$ bits codés qui servent pour la sélection d'un sous-ensemble.

Un codeur convolutif de rendement $\frac{m'}{m'+1}$ est défini par l'équation de parité qui pose une contrainte sur les bits codés en une fenêtre temporelle mobile de largeur $(v+1)$ où v est la longueur de contrainte, selon l'équation de parité :

$$\sum_{i=0}^m (h_v^i \cdot C_{n-v}^i \oplus h_{v-1}^i \cdot C_{n-v+1}^i \oplus \dots \oplus h_0^i \cdot C_n^i) = 0 \quad \dots (2.7)$$

Les quantités h_l^i où $0 \leq l \leq v$ et $0 \leq i \leq m'$ sont les coefficients binaires de parité pour le code. L'équation définit seulement les séquences de code et non la relation entrée/sortie pour un codeur.

2-5 Passage d'un codeur convolutif à un codeur systématique avec feed-back : [2]

En utilisant la notation polynomiale la séquence binaire $C(D)$ doit satisfaire l'équation de parité :

$$[C^m(D), \dots, C^1(D), C^0(D)] \cdot [H^m(D), \dots, H^1(D), H^0(D)]^T = 0 \quad \dots (2.8)$$

T : représente la transposée.

Notons qu'un codeur est minimal si sa longueur de contrainte v est le degré maximum de $H^j(D)$ avec $0 \leq j \leq m$.

Pour $v \geq 2$ les polynômes de parités s'écrivent selon la forme suivante :

$$H^j(D) = 0 \quad m' < j \leq m, \quad \dots (2.9)$$

$$H^j(D) = 0 + h_{v-1}^j \cdot D^{v-1} + h_{v-2}^j \cdot D^{v-2} + \dots + h_1^j \cdot D + 0, \quad 1 \leq j \leq m', \quad \dots (2.10)$$

$$H^0(D) = D^v + h_{v-1}^0 \cdot D^{v-1} + h_{v-2}^0 \cdot D^{v-2} + \dots + h_1^0 \cdot D + 1 \quad \dots (2.11)$$

$$h_v^j = h_v^j = \begin{cases} 0 & J \neq 0 \\ 1 & J = 0 \end{cases} \quad (v > 2) \quad \dots (2.12)$$

L'équation (2.9) signifie qu'il ya (m-m') bits non codés à l'entrée du codeur, ceci entraîne $2^{(m-m')}$ transitions parallèles entre états.

Au lieu de générer des séquences codées C(D) par des codeurs minimaux, les codeurs systématiques équivalents avec feed-back peuvent être aussi utilisés pour générer des séquences codées.

Le modèle mathématique d'un codeur systématique avec feed-back est le suivant :

$$[C^m(D) \dots C^0(D)] = [X'^m(D) \dots X'^1(D)] \cdot \begin{bmatrix} | 0 \\ | \dots \\ | H^m(D)/H^0(D) \\ | \dots \\ | H^1(D)/H^0(D) \end{bmatrix} \quad \dots (2.13)$$

$X'(D)$ représente la séquence embrouillée (scramble) d'entrée du codeur obtenue à partir de la séquence X(D).

I_m : matrice identité (m×m).

En écrivant les sorties C(D) en fonction de X'(D) et H(D) on montre que la structure générale du codeur convolutif systématique avec feed-back est donnée par la figure 2.7 :

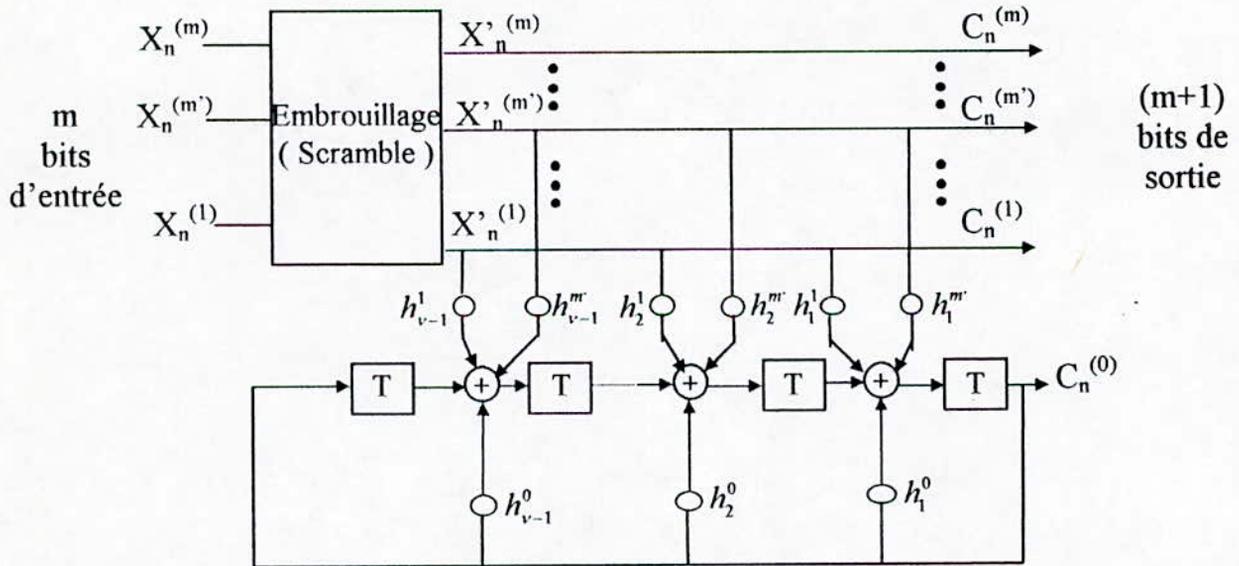


Figure 2.7 : Codeur convolutif systématique avec feed-back

$$C_n^{(m)} = X_n'^{(m)}$$

$$C_n^{(m-1)} = X_n'^{(m-1)}$$

$$C_n^{(m)} = X_n^{(m)} \dots (2.14)$$

$$C_n^{(1)} = X_n^{(1)}$$

$$C_n^{(0)} = C_{n-v}^{(0)} \oplus \sum_{i=1}^{v-1} h_i^0 \cdot C_{n-i}^{(0)} \oplus \sum_{i=1}^{v-1} \sum_{j=1}^m h_i^j \cdot X_{n-i}^{(j)}$$

2-5-1 Description du codeur systématique à boucle de retour : [7], [2]

Le codeur systématique à boucle de retour de la figure 2.7 comporte deux blocs :

- **Embrouilleur (Scramble) :**

L'embrouilleur est un bloc à registres à décalages utilisé pour rendre la séquence d'entrée plus aléatoire.

- **Bloc mémoire avec boucle de retour :**

Ce bloc est constitué d'un ensemble d'éléments mémoires ayant comme sortie le bit $C_n^{(0)}$ qui sélectionne le point message.

Exemple 2.1 : [2], [1]

Dans cet exemple, nous allons voir comment passer du codeur à feed-back vers la représentation d'Ungerboeck. Selon les tableaux donnés par Ungerboeck où les coefficients de la matrice de parité sont donnés dans la forme octale.

Pour le codeur à quatre états, modulation 4-AM on a :

$$H(D) = [H_0(D), H_1(D)] = [5, 2]_8 \dots (2.15)$$

sous la forme binaire :

$$H(D) = [101, 010] \dots (2.16)$$

sous la forme polynomiale :

$$H(D) = [1+D^2, D] \dots (2.17)$$

On a deux sorties $C^1(D)$, $C^0(D)$ attaquant le transcodeur. La matrice de parité vérifiée la condition :

$$(C^0(D), C^1(D)) \cdot H^T(D) = 0 \dots (2.18)$$

alors à l'instant n :

$$C_n^{(0)} = C_{n-2}^{(0)} \oplus C_{n-1}^{(1)} \dots (2.19)$$

- **La réalisation de l'équation (2.19) :**

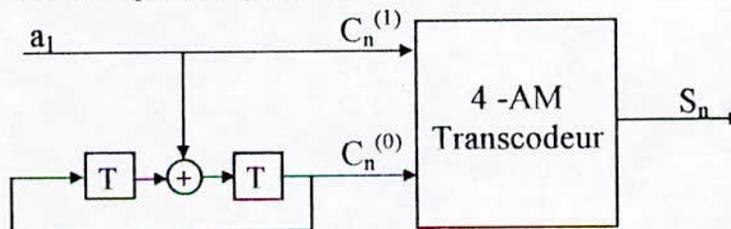


Figure 2.8 : Réalisation en forme de boucle de retour

Nous allons voir maintenant le passage du codeur systématique à la représentation d'Ungerboeck. Selon la condition (2.20) réalisée par le polynôme générateur $G(D)$ et la matrice de parité $H(D)$:

$$G(D).H^T(D) = 0 \quad \dots(2.20)$$

$$G(D) = [G_0(D), G_1(D)] \quad \dots(2.21)$$

La condition (2.20) donne :

$$G_0(D).(1+D^2) + G_1(D).D = 0 \quad \dots(2.22)$$

La solution est :

$$G(D) = [D, 1+D^2] \quad \dots(2.23)$$

$$G(D) = [010, 101] = [2, 5]_8 \quad \dots(2.24)$$

d'où découle la représentation d'Ungerboeck.

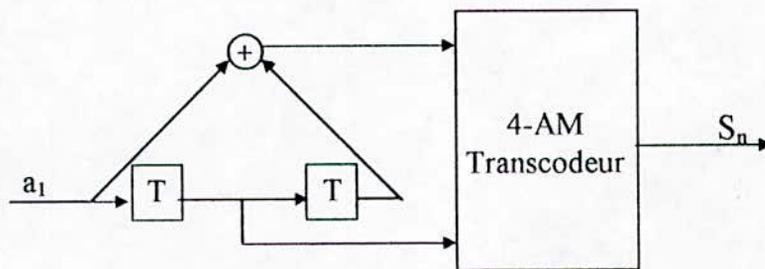


Figure 2.9 : Représentation d'Ungerboeck

2.6 Méthode de calcul de d_{free} : [10], [2]

Considérons en deuxième étape le modèle d'Ungerboeck de rendement $\frac{m}{m+1}$ proposé en figure 2.6, la distance euclidienne libre est exprimée par :

$$d_{free} = \min(\Delta_{m+1}, d_{free}(m')) \quad \dots(2.25)$$

Où : $d_{free}(m')$: la distance minimum du treillis lui-même.

Δ_{m+1} : la distance minimum entre transitions parallèles.

Selon l'équation (2.7), nous observons que la séquence de sortie $[C_n^{m'}, \dots, C_n^0]$ peut avoir des valeurs arbitraires pour chaque m' -tuples $[C_n^{m'}, \dots, C_n^1]$ avec un choix approprié de C_n^0 afin que cette équation soit vérifiée.

Soient $\{\underline{C}'_n\}$ la séquence de sortie erronée et $\{\underline{E}_n\}$ la séquence d'erreurs :

$$\{\underline{C}'_n\} = \{\underline{C}_n \oplus \underline{E}_n\} \quad \dots(2.26)$$

Puisqu'on a affaire à des codes linéaires alors :

$$e(D) = e_K.D^K + e_{K+1}.D^{K+1} + \dots + e_{K+L}.D^{K+L} \quad \dots(2.27)$$

avec $e_K, e_{K+L} \neq 0 ; L \geq 0$.

$$C'(D) = C(D) \oplus e(D) \quad \dots(2.28)$$

$C'_K = C_K \oplus e_K$, $C'_{K+1} = C_{K+1} \oplus e_{K+1}$, , $C'_{K+L} = C_{K+L} \oplus e_{K+L}$
 où L est la longueur de l'événement d'erreur.

Dans le but de borner inférieurement la distance euclidienne entre les séquences du canal a(D) et a'(D) obtenues à partir de C(D) et C'(D), le poids d'Euclide est défini par :

$$W(e_n) = \min \left\{ d[a(C_n), a(C_n \oplus e_n)] \right\} \quad \dots (2.29)$$

où d[...] est la distance euclidienne entre les signaux spécifiés du canal.

Pour le carré de la distance entre a(D) et a'(D) nous obtenons alors :

$$\sum_{i=K}^{K+L} d^2[a(C_i), a(C_i \oplus e_i)] \geq \sum_{i=K}^{K+L} W^2(e_i) = W^2(e(D)) \quad \dots (2.30)$$

La distance libre d_{free} est déterminée d'une manière analogue à celle de Hamming seulement le poids de Hamming du vecteur erreur sera remplacé par le carré du poids d'Euclide :

$$W^2(e_n) = \left\{ \min d[a(C_n), a(C_n \oplus e_n)] \right\}^2 \quad \dots (2.31)$$

alors :

$d_{free}^2 = \min [W^2(e(D))]$ sur toutes les séquences codées non nulles. C'est-à-dire la minimisation de d_{free} s'étale sur toutes les séquences codées non nulles.

$$d_{free}^2 = \min \left\{ \sum_{i=K}^{K+L} \min^2 \{ d(a(C_i), a(C_i \oplus e_i)) \} \right\} \quad \dots (2.32)$$

2-7 Recherche des codes optimaux :

2-7-1 Position du problème :

Le problème de base à résoudre est de trouver un codeur systématique avec feed-back qui maximise la distance libre d_{free} pour une longueur de contrainte v donnée et pour une modulation bien spécifiée.

Notre objectif est donc de trouver pour chaque modulation (M-AM, M-PSK, QAM...) , pour un ensemble d'éléments mémoires donné, les paramètres de la matrice de parité, c'est-à-dire les meilleurs paramètres qui maximisent la distance libre d_{free} et minimisent la probabilité d'erreur moyenne par bit.

C'est alors Ungerboeck qui a proposé en 1982 un algorithme de recherche systématique des codeurs optimaux à feed-back. Cette conception est basée sur des règles heuristiques.

2-7-2 Données du problème :

On donne :

- le type de modulation (8-PSK, 16-PSK, 16-QAM, ...etc)
- les différentes distances entre les sous-ensembles de la partition ($\Delta_0, \Delta_1, \dots, \Delta_m$).
- l'ensemble d'éléments mémoires v .

2-7-3 Paramètres à déterminer :

Les paramètres à déterminer sont les éléments de la matrice de parité H : cette matrice de parité H est donnée par :

$$H = \begin{bmatrix} h_0^0 & h_1^0 & \dots & h_v^0 \\ h_0^1 & h_1^1 & \dots & h_v^1 \\ \dots & \dots & \dots & \dots \\ h_0^{m-1} & \dots & \dots & h_v^{m-1} \\ h_0^m & \dots & \dots & h_v^m \end{bmatrix} \dots (2.33)$$

donc il ya $(m^2+1).(v+1)$ paramètres à déterminer, mais dans notre recherche nous allons exclure les codes catastrophiques, alors d'après la relation (2.12) la matrice H devient :

$$H = \begin{bmatrix} 1 & h_1^0 & \dots & h_{v-1}^0 & 1 \\ 0 & h_1^1 & \dots & h_{v-1}^1 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & h_1^{m-1} & \dots & h_{v-1}^{m-1} & 0 \\ 0 & h_1^m & \dots & h_{v-1}^m & 0 \end{bmatrix} \dots (2.34)$$

donc il nous reste $(m^2+1). (v-1)$ paramètres à déterminer.

2-7-4 Description de l'algorithme : [2]

La procédure de recherche est exhaustive avec un nombre de règles de rejections.

• Notation :

Les coefficients de la ligne j de la matrice de parité sont représentés dans la base 2 par ih^j qui s'écrit sous la forme :

$$ih^j = [h_{v-1}^j h_{v-2}^j \dots h_1^j] \dots (2.35)$$

où h_1^j est le bit de poids faible et h_{v-1}^j est le bit de poids fort.

La représentation de ih^j dans la base décimale est :

$$0 \leq ih^j = (h_{v-1}^j . 2^{v-1} + h_{v-2}^j . 2^{v-2} + \dots + h_1^j . 2^0) \leq 2^{v-1} - 1 \quad ; \quad 0 \leq j \leq m \quad \dots (2.36)$$

Définition :

Un code incomplet au niveau l, tel que $0 \leq l \leq m$, est spécifié par :

$$\underline{ih}^l = [ih^l, ih^{l-1}, \dots, ih^0] \dots (2.37)$$

Ceci est équivalent à mettre les entrées $X^{l+1}(D), \dots, X^m(D)$ à zéro pour le code complet spécifié par \underline{ih}^m .

2-7-5 Règles de conception : [2]

• Règle 1 : Un code peut être rejeté au niveau l si :

$$ih_{rev}^l < ih^l \text{ et } ih_{rev}^j = ih^j \text{ pour } 0 \leq j < l.$$

Où ih_{rev}^l est le réversible de ih^l .

- **Règle 2** : Soit $H'(D)$ dérivée de $H(D)$ telle que :

$$H'(D)=[H^m(D),\dots,H^t(D),\dots,H^s(D)\oplus H^s(D),\dots,H^0(D)] \quad ; \quad 0 \leq s < t \leq m \quad \dots(2.38)$$

Si :

$$e(D)=[e^m(D),\dots,e^t(D),\dots,e^s(D),\dots,e^0(D)] \quad \dots(2.39)$$

satisfait :

$$e(D).H^T(D)=0 \quad \dots(2.40)$$

alors :

$$e'(D)=[e^m(D),\dots,e^t(D)\oplus e^s(D),\dots,e^s(D),\dots,e^0(D)] \quad \dots(2.41)$$

satisfait :

$$e'(D).H^{T'}(D)=0 \quad \dots(2.42)$$

avec T dénote la transposée.

Un code est rejeté au niveau l si l'une des conditions suivantes est satisfaite :

- $h \geq 1$: $ih^l \oplus ih^{l-1} < ih^{l-1}$ (2.43)

- $h \geq 2$: $\begin{cases} ih^l \oplus ih^{l-2} < ih^{l-2} \\ ih^l \oplus ih^{l-1} \oplus ih^{l-2} < ih^{l-2} \end{cases}$ (2.44)

- $h \geq 3$: $\begin{cases} ih^l \oplus ih^{l-3} < ih^{l-3} \\ ih^l \oplus ih^{l-2} \oplus ih^{l-3} < ih^{l-3} \\ ih^l \oplus ih^{l-1} \oplus ih^{l-3} < ih^{l-3} \\ ih^l \oplus ih^{l-1} \oplus ih^{l-2} \oplus ih^{l-3} < ih^{l-3} \end{cases}$ (2.45)

- **Règle 3** : Un code est rejeté au niveau l si sa distance euclidienne $dfree^{(l)}$ ne dépasse pas $dfree^{(m)} = dlim$ trouvé parmi les codes complets inspectés précédemment. Initialement $dlim$ est mise à zéro.

- **Règle 4** : Un code est rejeté si :

$$GCD(H^m(D),\dots,H^0(D)) \neq 1 \quad \dots(2.46)$$

où GCD représente le plus grand diviseur commun.

C'est à dire $(H^m(D),\dots,H^0(D))$ doivent être primaires entre eux.

2-7-6 L'organigramme de recherche des codes optimaux : [2]

L'organigramme est donné a la figure 2.10, il est basé sur un ensemble de règles empiriques, et à pour but de chercher et de trouver les coefficients de la matrice de parité représentés dans le codeur systématique de la figure 2.7 et la distance libre $dfree$.

2-7-6-1 Description des blocs essentiels de l'organigramme :

Dans cet organigramme de recherche, nous incrémentons dans la boucle extérieure ih^0 de 0 à $2^{v-1}-1$ et dans la boucle intérieure qui suit ih^1 de 0 à $2^{v-1}-1$,...etc. Un code est rejeté au niveau l s'il ne satisfait pas une des règles décrites ci-dessus.

Dans l'organigramme nous avons :

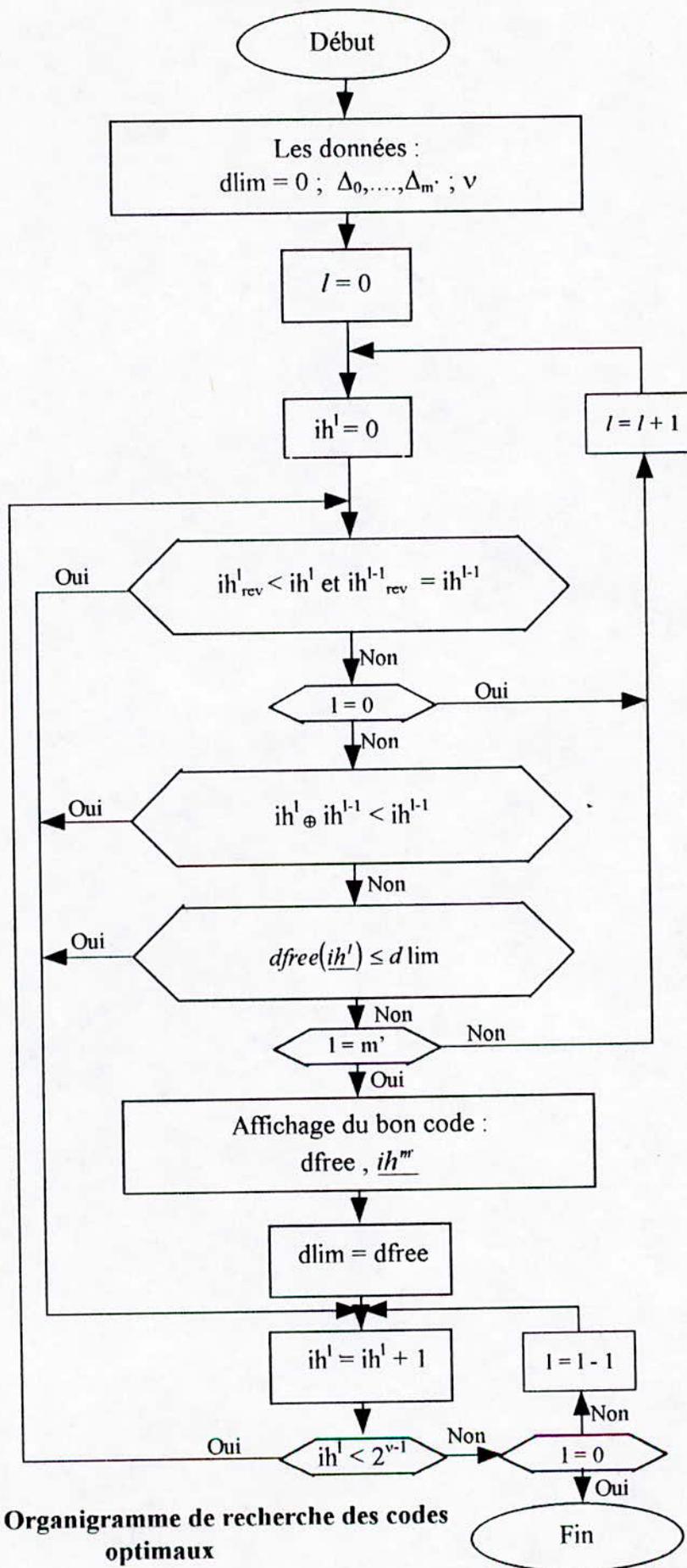


Figure 2.10 : Organigramme de recherche des codes optimaux

- Le bloc calculant le réversible de ih^l et ih^{l-1} pour réaliser la règle 2, par exemple si $ih^l = 0111$
 $\Rightarrow ih_{rev}^l = 1110$.
- Le XOR de ih^l et ih^{l-1} se fait bit par bit.
- Le bloc calculant $dfree$: le calcul de $dfree$ est basé sur la connaissance des $(m+1)$ bits de sorties du codeur systématique à boucle de retour. En particulier le bit de sortie $C_n^{(o)}$ dépend des bits d'entrées $X_n^{(i)}$ ($i=1, \dots, m'$) de la séquence embrouillée, de ces valeurs retardées ainsi sur les valeurs de la matrice de parité.
- Le bloc affichage donne la distance libre $dfree$ et les polynômes de la matrice de parité.

2-8 Conclusion :

Nous avons vu dans ce chapitre l'importance de la matrice de parité, l'avantage des codeurs systématiques à boucle de retour qui a poussé Ungerboeck à les utiliser, la méthode de calcul de $dfree$ qui est basée sur le poids d'Euclide et la complexité de l'organigramme de recherche des codes optimaux.

Une mise en application de cet organigramme permettant de concevoir des codes optimaux, est prévue au chapitre 4.

Dans le chapitre suivant, nous évoquerons la modulation asymétrique codée en treillis qui consiste à améliorer la distance libre $dfree$ en augmentant les distances minimales entre les sous-ensembles.

CHAPTER

3

Chapitre 3

Modulation Asymétrique codée en treillis (ATCM)

3-1 Introduction :

Dans ce chapitre nous verrons que l'ensemble des signaux symétriques (uniformément espacés dans la constellation) qui sont optimaux dans un environnement AWGN non codé ne sont pas nécessairement optimaux pour systèmes codés. Nous entendons par optimum une probabilité par bit minimum et une distance euclidienne d_{free} maximale.

Traditionnellement, les constellations symétriques sont utilisées pour les systèmes codés et non codés. En concevant des constellations asymétriques où les points de messages dans la constellation ne sont pas uniformément espacés, ceci permet d'améliorer le gain de codage vis à vis des configurations TCM basées sur des constellations symétriques sans avoir recours à augmenter la puissance émise ou changer la bande passante imposée par le système, c'est Divsalar et Yuen qui ont apporté l'idée d'utiliser la TCM avec des modulations asymétriques.

Dans ce chapitre, nous nous limiterons aux TCM à deux dimensions, l'approche de conception peut être étendue aux configurations TCM multidimensionnelles.

3-2 Analyse et conception : [1]

Dans cette section nous utiliserons des codeurs en treillis ayant m entrées et $(m+1)$ sorties (basés sur des codeurs convolutifs avec m' entrées et $(m'+1)$ sorties).

L'ensemble des points signaux $M = 2^{m+1}$ asymétriques est obtenu en ajoutant à $\frac{M}{2}$ points symétriques $\frac{M}{2}$ points mais déphasés ou translatés ou la combinaison des deux par rapport aux $\frac{M}{2}$ points premiers.

Une autre manière de procéder est d'imaginer la partition de la constellation à M points symétriques en deux sous-constellations comportant $\frac{M}{2}$ points avec une séparation maximum et ensuite amélioration d'une adéquate rotation (PSK), translation (AM) ou la combinaison rotation et translation (QAM) d'un sous-ensemble par rapport à l'autre.

Des constellations symétriques et asymétriques sont illustrées dans la figure 3.1 :

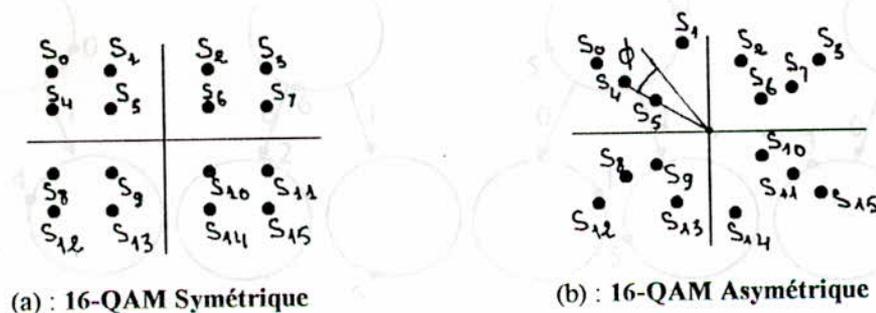


Figure 3.1 : La partition de la 8-PSK Asymétrique

Dans les exemples à considérer, pour raison de simplicité et d'analyse, le critère d'optimisation sera la maximisation de la distance euclidienne d_{free} des configurations TCM. Ensuite, la relation de cette mesure et la probabilité d'erreur moyenne par bit. Chaque m bits d'informations transmis via un codeur en treillis produisent $(m+1)$ symboles qui seront assignés à un signal de la constellation asymétrique à 2^{m+1} signaux, chaque signal complexe C_k transmis à l'instant k est fonction non linéaire de l'état σ_k du codeur et le symbole de sortie X_k de la source, l'état prochain σ_{k+1} du codeur est aussi fonction non linéaire de l'état précédent σ_k et le symbole de sortie de la source X_k comme le montre la figure 1.1.

La borne supérieure de la probabilité d'erreur par bit est donné par la relation :

$$P_b \leq \frac{1}{2m} \operatorname{erfc} \left(\sqrt{\frac{m \cdot E_b \cdot d_{free}^2}{N_0 \cdot 4}} \right) \cdot D^{-d_{free}^2} \cdot \frac{\partial}{\partial D} T(D, I) \Big|_{I=1, D=Z} \quad \dots (3.1)$$

où

$Z = e^{\frac{-m \cdot E_b}{4N_0}}$ est le paramètre de Bhattacharyya.

Asymptotiquement, pour un SNR (rapport signal sur bruit) large, la maximisation de d_{free} est synonyme au minimisation de la probabilité d'erreur moyenne par bit.

Dans certain cas, l'optimisation de la condition asymétrique produit des ensembles de signaux tendant à s'émerger entre eux dans la limite d'un SNR approche de l'infini. Ceci résulte à un codeur catastrophique où des longs trajets apparaît de plus en plus ayant une distance égale ou faiblement large à la distance d_{free} .

Dans ces instants, les performances requises en termes des améliorations dans d_{free} ne sont pas translaté directement en amélioration dans l'exigence du SNR.

3-2-1 Procédure de conception :

La procédure de conception des bons codes en treillis combinés avec des constellations asymétrique est résumée dans les points suivantes :

- utilisons le mapping par méthode de la partition de la constellation.
- Assignons les signaux aux transitions de telle façon à maximiser les distances minimales entre transitions divergeantes ainsi qu'entre transitions remergeantes.
- Maximisons d_{free} ou minimisons la probabilité d'erreur moyenne par bit P_b en considération avec la meilleur rotation Φ (PSK), la translation optimum Δ (AM) ou la combinaisons des deux (QAM).

3-3 Codeurs en treillis et la TCM Asymétrique (ATCM) : [1]

3-3-1 Codeur de rendements $\frac{1}{2}$ combinés avec la 4-PSK Asymétrique :

La partition d'ensemble pour la modulation 4-PSK asymétrique codée en treillis (A 4-PSK) est illustrée dans la figure 3.3 :

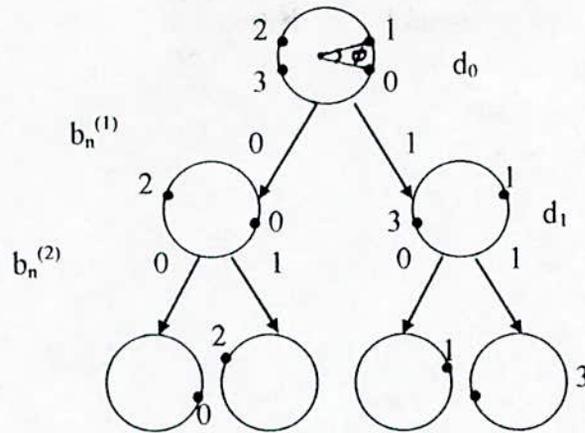


Figure 3.3 : Partition de la 4-PSK Asymétrique

• Le codeur en treillis à 2 états :

Le diagramme en treillis est donné dans la figure 3.4 :

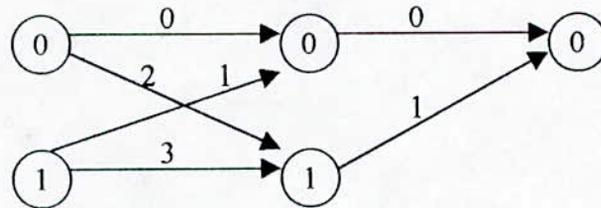


Figure 3.4 : Diagramme en treillis à 2 états pour la 4-PSK

La matrice d'état de transition est :

$$T = \begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \quad \dots(3.2)$$

L'élément $T(i,j)$ de la matrice T représente le symbole de sortie de l'état i vers l'état j . Le treillis de la figure III-4 peut être implémenter en utilisant un codeur convolutif de rendement $\frac{1}{2}$, soit le déphasage Φ effectué (figure 3.3) pendant la partition de la constellation de la modulation 4-PSK Asymétrique. La borne supérieure de la probabilité d'erreur par bit pour cette configuration est donnée par :

$$P_b \leq \frac{0,5 \operatorname{erfc} \left(\sqrt{\left[\frac{4(1+2\alpha)}{(1+\alpha)} \right] \operatorname{Ln}(Z)} \right)}{\left[1 - Z^{\frac{4}{(1+\alpha)}} \right]^2} \quad \dots(3.3)$$

avec :

Z est le paramètre de Bhattacharyya et α est un paramètre donné par :

$$\alpha = \operatorname{tg}^2 \left(\frac{\Phi}{2} \right) \quad \dots(3.4)$$

La valeur optimum de α (équivalent Φ) qui minimise la borne supérieure de P_b est donnée par :

$$\alpha_{opt} \cong -4 \cdot \left(\frac{L\mu Z}{L\mu 3} \right) - 1 \quad \dots(3.5)$$

$$\Phi_{opt} = 2 \cdot \arctan(\sqrt{\alpha_{opt}})$$

En substituant α_{opt} dans la borne supérieure de P_b nous aurons donc la borne optimum de la 4-PSK Asymétrique donnée par :

$$P_b \leq \frac{9}{8} \cdot \operatorname{erfc} \left(\sqrt{\frac{2 \cdot E_b}{N_0}} - L\mu 3 \right) \quad \dots(3.6)$$

Pour la constellation symétrique 4-PSK codée $\left(\Phi = \frac{\pi}{2}, \alpha = 1 \right)$:

$$P_b \leq \frac{0,5 \cdot \operatorname{erfc} \left(\sqrt{\frac{3 \cdot E_b}{2 \cdot N_0}} \right)}{\left[1 - \exp \left(\frac{E_b}{2 \cdot N_0} \right) \right]^2} \quad \dots(3.7)$$

Pour la modulation 4-PSK non codée :

$$P_b \leq 0,5 \cdot \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right) \quad \dots(3.8)$$

La figure 3.5 illustre les trois bornes supérieures 3.6, 3.7, 3.8 en fonction de $\frac{E_b}{N_0}$:

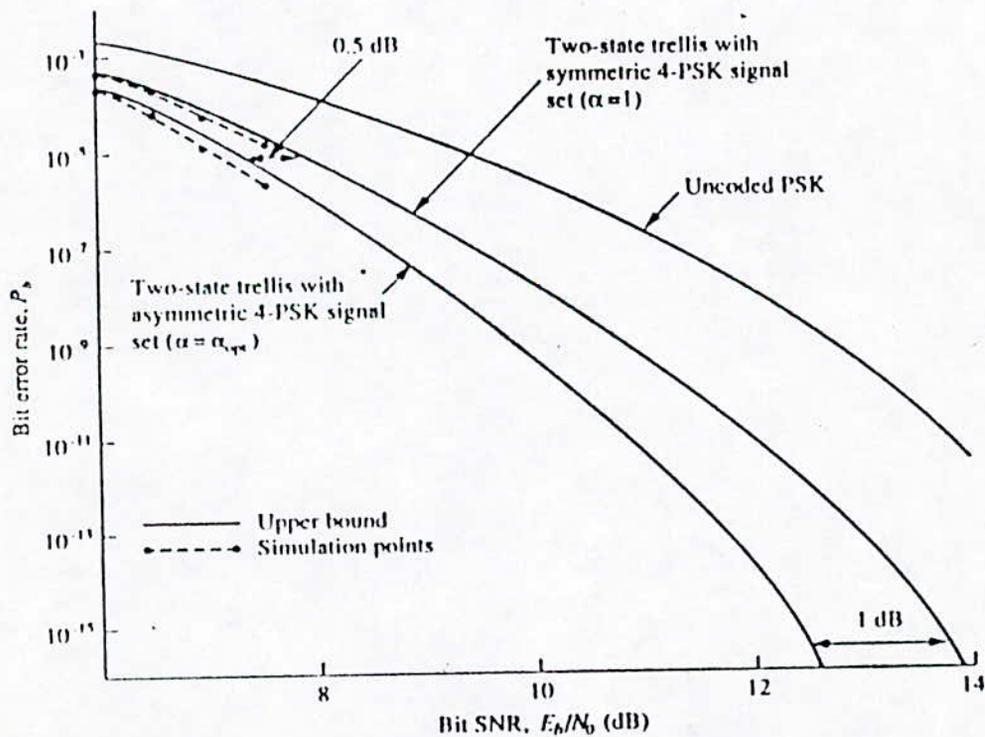


Figure 3.5 : Bornes supérieures des probabilités d’erreur par bit pour la 4-PSK Symétrique et Asymétrique et la 2-PSK non codée

Soit δ_j^2 dénote la distance au carré entre le point 0 et les points 1, 2, 3 de la constellation .

$$\begin{aligned} \delta_1^2 &= d^2(0,1) = 4 \cdot \sin^2\left(\frac{\Phi}{2}\right) \\ \delta_2^2 &= d^2(0,2) = 4 \dots\dots(3.9) \\ \delta_3^2 &= d^2(0,3) = 4 \cdot \cos^2\left(\frac{\Phi}{2}\right) \end{aligned}$$

de la figure 3.4, nous remarquons que la distance minimum d_{free} du chemin de longueur 2 pour la 4-PSK Asymétrique :

$$d_{free}^2 = \delta_2^2 + \delta_1^2 = 4 \cdot \left(1 + \frac{\alpha}{\alpha + 1}\right) \dots\dots(3.10)$$

Pour la 4-PSK codée symétrique ($\alpha = 1$) alors : $d_{free}^2 = 6$.

L'amélioration apportée par la 4-PSK Asymétrique par rapport à 4-PSK symétrique est exprimée par :

$$\eta = 10 \cdot \text{Log} \left(\frac{d_{free}^2(asy)}{d_{free}^2(sym)} \right) = 10 \cdot \text{Log} \left(\frac{2 \cdot (1 + 2 \cdot \alpha)}{3 \cdot (1 + \alpha)} \right) \dots\dots(3.11)$$

Au lieu de minimiser P_b , nous sélectionnons α (i.e. $-\phi$) qui maximise d_{free} , la valeur de ϕ est indépendante du SNR. De l'équation (3.10), d_{free} est maximum pour $\phi = \pi$ ceci signifie que les points 1 et 2 avec les points 0 et 3 dans la constellation , $d_{free}(\Phi = \pi) = 8$ et donc le gain $\eta = 1,25 \text{ dB}$. Le gain de la 4-PSK Asymétrique par rapport à la 2-PSK non codée est 3 dB tandis que la gain de la 4-PSK par rapport à la 2-PSK non codée est de 1,76 dB. En particulier, la modulation Asymétriques gagne 0,5 dB pour une probabilité d'erreur de 10^{-5} .

• **Le codeur en treillis à 4 états :**

Pour un codeur de rendement $\frac{1}{2}$ à 4 états le diagramme en treillis est illustré dans la figure 3.6 :

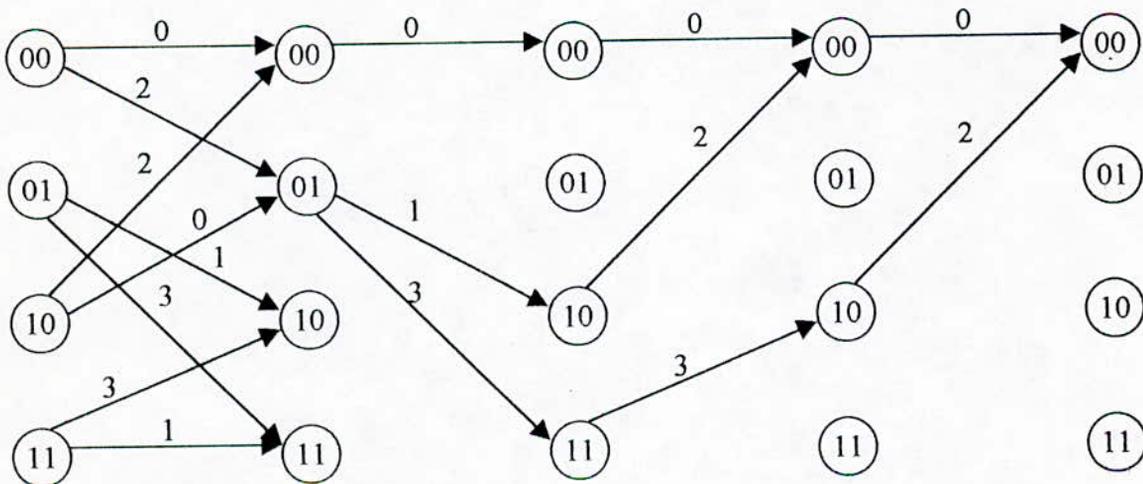


Figure 3.6 : Diagramme en treillis à 4 états pour la 4-PSK

De ce diagramme, nous remarquons deux chemins minimaux qui sont exprimés par :

$$d^2(2,3,3,2) = 4 + 8 \cdot \cos^2\left(\frac{\Phi}{2}\right) + 4$$

.....(3.12)

$$d^2(2,1,2) = 4 + 4 \cdot \sin^2\left(\frac{\Phi}{2}\right) + 4$$

Pour des valeurs petites de ϕ , le chemin de longueur 4 correspondant à 2, 3, 3, 2 est dominant tandis que pour ϕ voisin de π le chemin de longueur 3 correspondant à 2, 1, 2 est dominant.

La recherche de la valeur optimale de ϕ consiste à égaliser les deux distances alors nous aurons $\operatorname{tg}^2\left(\frac{\Phi}{2}\right) = 2$ ce qui donne $\Phi = 109^\circ$ et $d_{free}^2 = 10,67$.

Pour la 4-PSK symétrique $d_{free}^2 = 10$ $\left(\Phi = \frac{\Pi}{2}\right)$ ce qui donne $\eta = 0,28$ dB (le gain apporté par la modulation 4-PSK Asymétrique à 4 états par rapport à la 4-PSK codée symétrique).

Le gain relatif à la modulation 2-PSK non codée :

$$\begin{aligned} \eta(asy) &= 10 \cdot \operatorname{Log} \frac{d_{free}^2(asy)}{d_{min}^2} \\ &= 10 \cdot \operatorname{Log} \frac{10,67}{4} = 4,26 \text{ dB} \\ \eta(sym) &= 10 \cdot \operatorname{Log} \frac{d_{free}^2(sym)}{d_{min}^2} \\ &= 10 \cdot \operatorname{Log} \frac{10}{4} = 3,98 \text{ dB} \end{aligned}$$

.....(3.13)

où d_{min} est la distance minimum entre les points dans la constellation de la modulation 2-PSK non codée.

- **Codeur en treillis à 8 états :**

Le diagramme en treillis à 8 états pour la A 4-PSK est illustré dans la figure 3.7

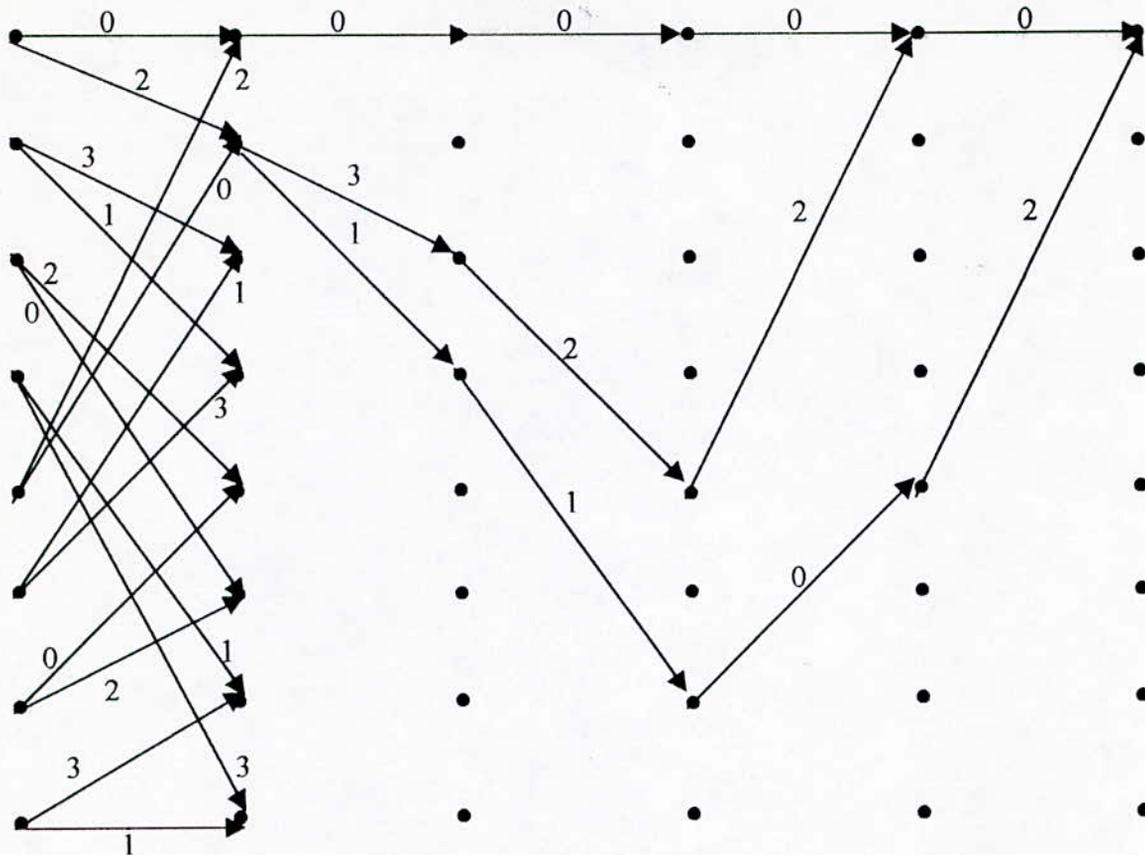


Figure 3.7 : Diagramme en treillis à 8 états pour la 4-PSK

Le carré de la distance des chemin de longueurs 4 et 5 sont donnés respectivement par :

$$d^2(2,3,2,2) = 4 + 4 \cdot \cos^2\left(\frac{\Phi}{2}\right) + 4 + 4 = 12 + 4 \cdot \cos^2\left(\frac{\Phi}{2}\right) \quad \dots(3.14)$$

$$d^2(2,1,1,0,2) = 4 + 4 \cdot \sin^2\left(\frac{\Phi}{2}\right) + 4 \cdot \sin^2\left(\frac{\Phi}{2}\right) + 0 + 4$$

La recherche du déphasage optimum est faite en égalisant les distances des deux chemins ce qui donne :

$$\sin^2\left(\frac{\Phi}{2}\right) = \frac{2}{3} \Rightarrow \Phi = 70,5^\circ \quad (1,23 \text{ rd}) \quad \dots(3.15)$$

La valeur de la distance libre est donc :

$$d_{free}^2(asym) = 13,33 \quad \dots(3.16)$$

Pour la 4-PSK symétrique à 8 états $\left(\Phi = \frac{\pi}{2}\right)$:

$$d_{free}^2(sym) = 12 \quad \dots(3.17)$$

Le gain apporté par la TCM asymétrique par rapport à TCM symétrique est donc :

$$\eta = 10 \cdot \text{Log} \frac{13,33}{12} = 0,46 \text{ dB} \quad \dots (3.18)$$

Le gain par rapport à la 2-PSK non codée :

$$\eta(\text{Asym}) = 10 \cdot \text{Log} \frac{13,33}{4} = 5,23 \text{ dB}$$

$$\eta(\text{Sym}) = 10 \cdot \text{Log} \frac{12}{4} = 4,77 \text{ dB} \quad \dots (3.19)$$

3-3-2 Codeur de rendement 2/3 combiné avec la 8-PSK :

La partition d'ensemble pour la 8-PSK Asymétrique est illustrée dans la figure 3.2.

• Codeur en treillis à 2 états :

Le diagramme en treillis est illustré dans la figure 3.8 :

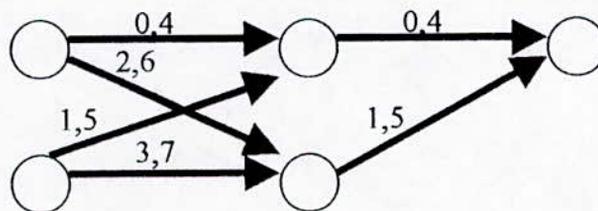


Figure 3.8 : Diagramme en treillis à 2 états pour la 8-PSK

Le carré de la distance d_{free} est :

$$d_{free}^2 = 4 - 2 \cdot \cos \Phi \quad \dots (3.20)$$

Nous remarquons que d_{free} est maximum pour $\Phi = \frac{\pi}{2}$ où les points 1, 3, 5, 7 coïncident avec les points 2, 4, 6, 0 respectivement. Pour la 8-PSK symétrique ($\Phi = \frac{\pi}{4}$) ce qui donne $d_{free}^2 = 4 - \sqrt{2}$, le gain apporté par la modulation asymétriques par rapport à la modulation symétrique est donc :

$$\eta = 10 \cdot \text{Log} \frac{4}{4 - \sqrt{2}} = 1,895 \text{ dB} \quad \dots (3.21)$$

Le gain par rapport à la modulation 4-PSK non codée :

$$\eta(\text{Asym}) = 10 \cdot \text{Log} \frac{4}{2} = 3,01 \text{ dB}$$

$$\eta(\text{Sym}) = 10 \cdot \text{Log} \frac{4 - \sqrt{2}}{2} = 1,116 \text{ dB} \quad \dots (3.22)$$

La figure 3.9 illustre les probabilités d'erreur par bit pour 8-PSK symétrique, 8-PSK asymétriques pour un codeur en treillis de rendement 2/3 ainsi que leurs bornes supérieures

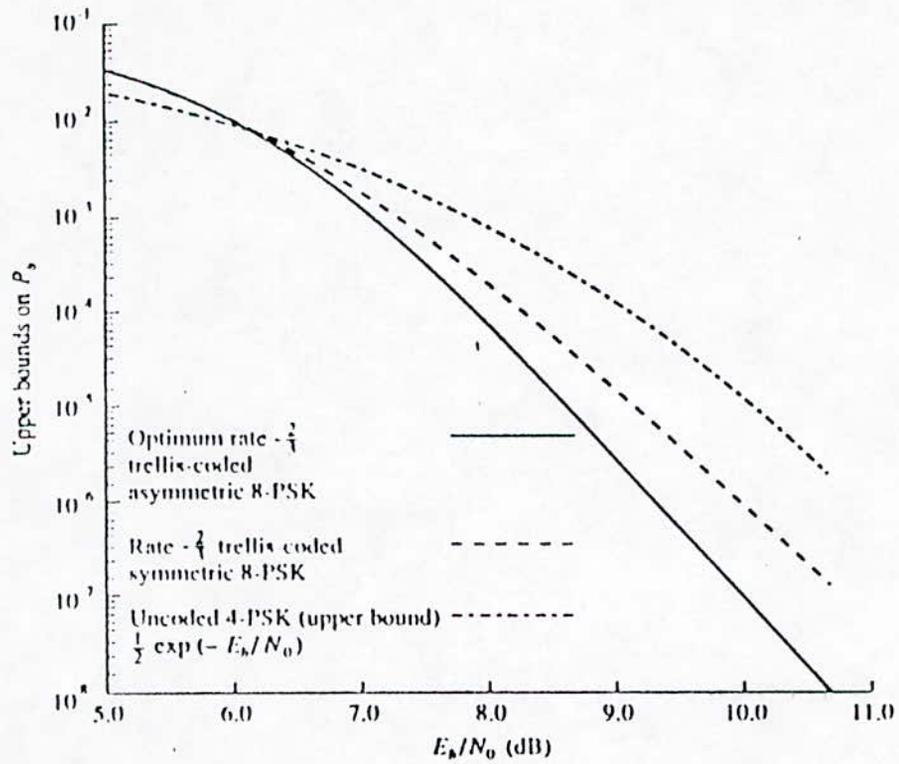
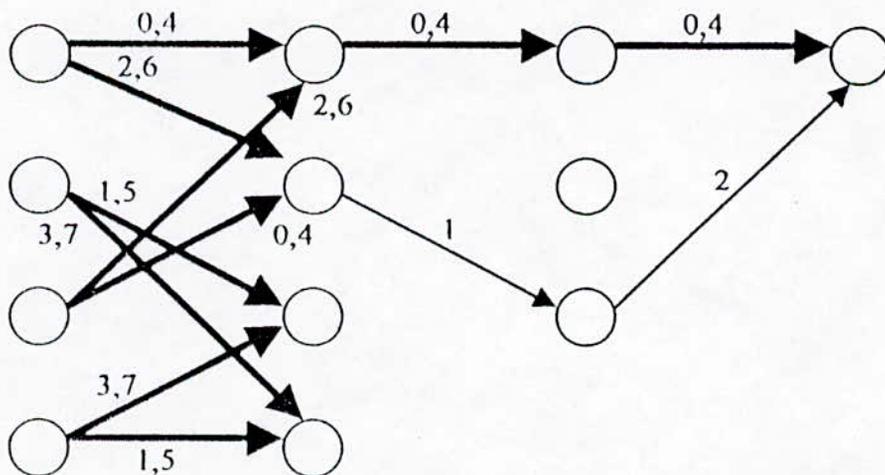


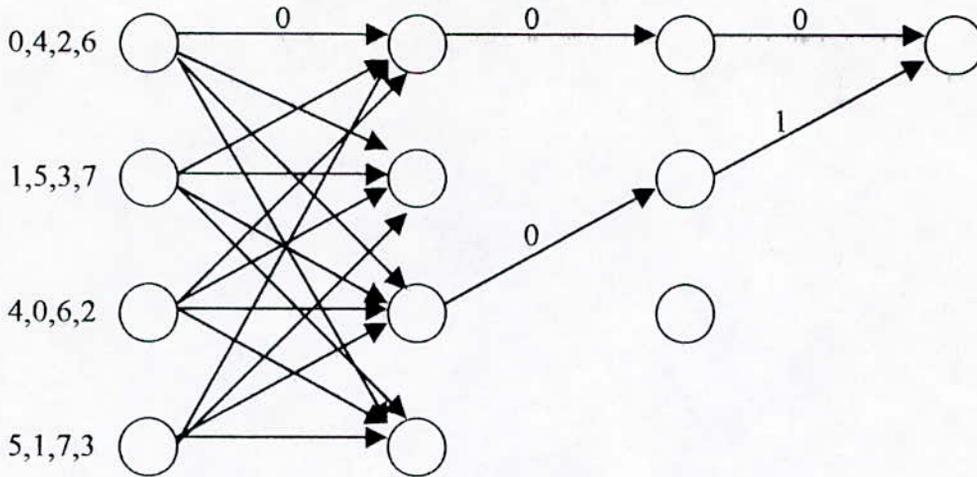
Figure 3.9 : Probabilité d'erreur par bit pour 8-PSK Symétrique et Asymétriques

• Codeur en treillis à 4 états :

Dans ce paragraphe nous allons étudier un codeur en treillis à 4 états avec et sans transitions parallèles. Les deux diagrammes en treillis sont illustrés dans la figure 3.10 a, b :



(a) Treillis avec chemins parallèles



(b) Treillis sans chemins parallèles

Figure 3.10 : Diagrammes en treillis à 4 états pour la 8-PSK

Du diagramme en treillis de la figure 3.10 b, la distance minimale du treillis est donnée par le chemin de longueur 3 :

$$d^2(2,0,1) = 4 - 2 \cdot \cos \Phi \quad \dots (3.23)$$

La valeur optimum du déphasage ϕ qui maximise $d^2(2,0,1)$ est $\Phi = \frac{\pi}{2}$ ce qui donne $dfree^2 = 4$.

Pour le cas symétrique $\Phi = \frac{\pi}{4}$ ce qui donne $dfree^2 = 4 - \sqrt{2}$.

Nous remarquons que ces résultats sont les mêmes que pour un codeur en treillis à deux états donc aucun gain n'est apporté et une complexité est ajoutée.

Du diagramme en treillis de la figure 3.10 a à transitions parallèles la distance minimale du treillis est donnée par le chemin de longueur 3 :

$$\begin{aligned} d^2(2,1,2) &= 6 - 2 \cdot \cos \Phi \\ d_{\max}^2(2,1,2) &= 6 \end{aligned} \quad \dots (3.24)$$

La distance entre transitions parallèles est :

$$d_p^2 = 4 \quad \dots (3.25)$$

La distance libre est donnée par :

$$dfree^2 = \min(d_{\max}^2(2,1,2), d_p^2) = 4 \quad \dots (3.26)$$

La distance libre est déterminée par les transitions parallèles, elle est indépendante de ϕ .

En comparaison avec la modulation asymétriques codée en treillis combinée avec le codeur en treillis à 2 états , aucun gain n'est apporté.

• **Codeur en treillis à 8 états :**

Le treillis est illustré dans la figure 3.11 :

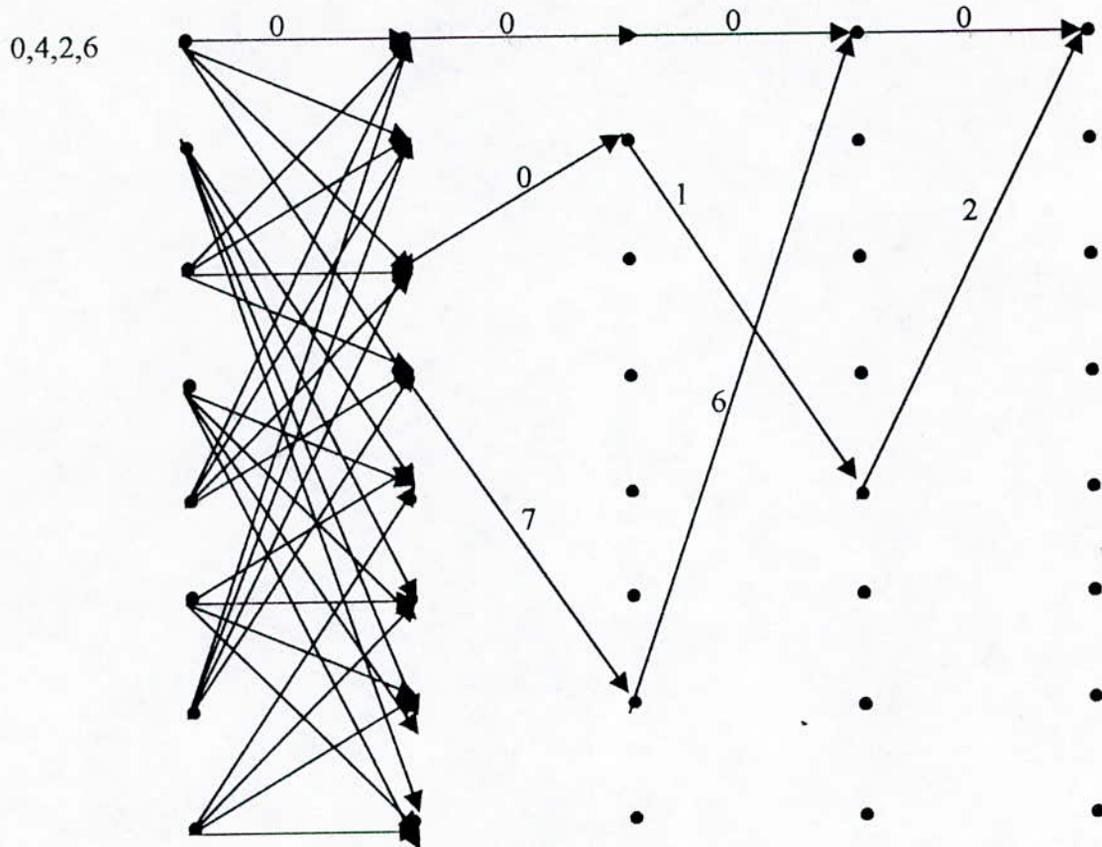


figure 3.11 : Diagramme en treillis à 8 états pour la 8-PSK

$$d^2(6,7,6) = 8 - 2 \cdot \sin \Phi \quad \dots(3.27)$$

$$d^2(2,0,1,2) = 6 - 2 \cdot \cos \Phi$$

En égalisant les deux distances nous aurons $\Phi = \frac{\pi}{4}$ d'où :

$d_{free}^2 = 6 - \sqrt{2} = 4,586$ et donc aucun gain n'est apporté par cette configuration asymétriques par rapport à la configuration symétrique $\left(\Phi = \frac{\pi}{4} \right)$.

Le gain par rapport à la modulation 4-PSK non codée est :

$$\eta = 10 \cdot \text{Log} \frac{6 - \sqrt{2}}{2} = 3,6 \text{ dB} \quad \dots(3.28)$$

3-3-3 Codeur de rendement 3/4 combiné avec la 16-PSK Asymétrique :

• Codeur en treillis à 4 états :

Le diagramme en treillis est illustré dans la figure 3.12 :

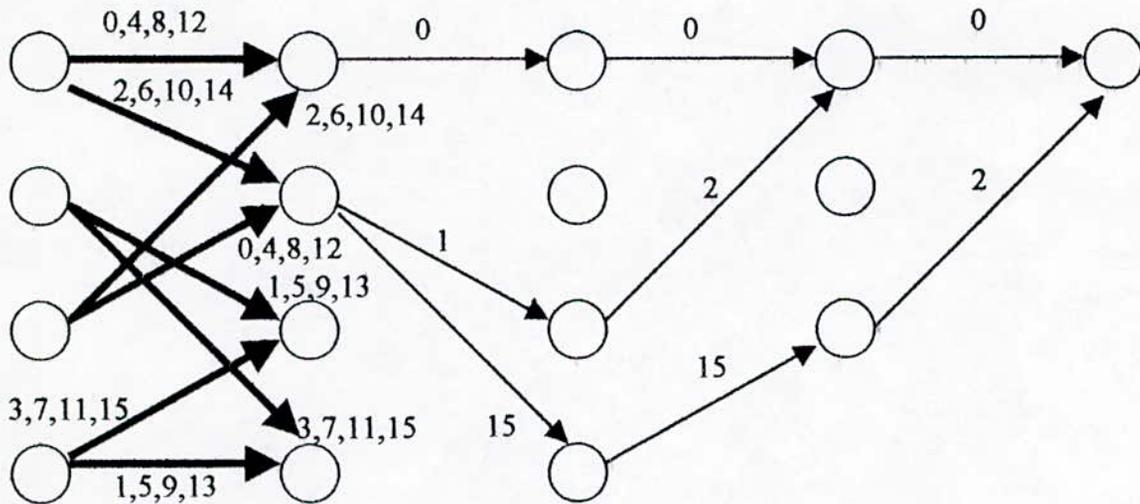


Figure 3.12 : Treillis à 4 états pour la 16-PSK

Du diagramme en treillis nous distinguons deux chemins minimaux de longueur 3,4 ayant des distances données respectivement par :

$$d^2(2,1,2) = 6 - 2\sqrt{2} - 2\cos\Phi$$

$$d^2(2,15,15,2) = 8 - 2\sqrt{2} - 4\cos\left(\frac{\pi}{4} - \Phi\right) \quad \dots(3.29)$$

En égalisant les deux distances nous aurons $\Phi = 26,37$ (0,46 rd) ce qui donne $d_{free}^2 = 1,38$, pour la modulation 16-PSK symétrique codée en treillis à 4 états $\left(\Phi = \frac{\pi}{8}\right)$ nous aurons donc alors : $d_{free}^2 = 1,324$.

Le gain en distance du à cette modulation asymétriques est donné par :

$$\eta = 10 \cdot \text{Log} \frac{1,38}{1,324} = 0,18 \text{ dB} \quad \dots(3.30)$$

Le gain relatif à la modulation 8-PSK non codée :

$$\eta(\text{asym}) = 10 \cdot \text{Log} \left(\frac{1,38}{2 - \sqrt{2}} \right) = 3,72 \text{ dB}$$

$$\eta(\text{sym}) = 10 \cdot \text{Log} \left(\frac{1,324}{2 - \sqrt{2}} \right) = 3,54 \text{ dB}$$

.....(3.31)

Le tableau 3.1 résume l'étude sur les modulations 4-PSK, 8-PSK, 16-PSK combinées avec des codeurs en treillis à 2, 4, 8, 16 états en spécifiant le nombre de transitions parallèles entre états pour chaque configuration, la distance minimum de la modulation non codée de référence, la distance libre pour chaque configuration symétrique et asymétrique, le déphasage optimum ϕ_{opt} et le gain dû à la modulation Asymétrique.

Nombre de niveaux $M=2^{m+1}$	Nombre de transitions parallèles entre états	Nombre D'états dans le treillis	La distance Minimum pour la modulation non codée M/2 niveaux	La distance Libre D_{free}^2 M-PSK Symétrique	la distance libre d_{free}^2 M-PSK Asym	Φ_{opt} (rd)	Gain de la M-PSK sym codée par rapport à la M/2 PSK non codée (dB)	Gain de la M-PSK Asym codée par rapport à la M PSK sym codée (dB)
4	1	2	4	6	8	π	1,76	1,25
4	1	4	4	10	10,67	1,91	3,98	0,28
4	1	8	4	12	13,33	1,23	4,77	0,46
8	2	2	2	2,586	4	$\pi/2$	1,116	1,895
8	2	4	2	4	4	$\pi/4$	3,01	0,00
8	1	8	2	4,586	4,586	$\pi/4$	3,6	0,00
8	1	16	2	5,172	5,20	0,64	4,13	0,024
16	4	2	0,586	0,738	1,172	$\pi/4$	1,00	2,01
16	4	4	0,586	1,324	1,38	0,46	3,54	0,18
16	4	8	0,586	1,476	1,589	0,32	4,014	0,319
16	4	16	0,586	1,628	1,628	$\pi/8$	4,440	0,00

Tableau 3.1

D'après le tableau 3.1, nous remarquons que :

- Nous avons un gain maximum pour les modulations M-PSK (M=4, 8 et 16) Asymétriques combinées avec le codeur à 2 états par rapport aux modulation M-PSK (M=4, 8 et 16) Symétriques. Ce résultat théorique n'est jamais obtenu en pratique car ces codeurs sont catastrophiques.
- Il n'y a aucun gain apporté par les modulations 8-PSK Asymétrique combinée avec les codeurs à 4 et à 8 états et la 16-PSK Asymétrique combinée avec le codeur à 16 états par rapport aux modulations 8 et 16-PSK Symétriques combinées avec les mêmes codeurs.
- Pour les autres configurations de l'ATCM, il y a une légère amélioration.

3-3-4 La modulation 4-AM Asymétrique :

La constellation de la 4-AM Asymétrique est illustrée dans la figure 3.13 où chaque point est translaté de $\pm \Delta$ de sa position dans le cas d'une constellation symétrique.

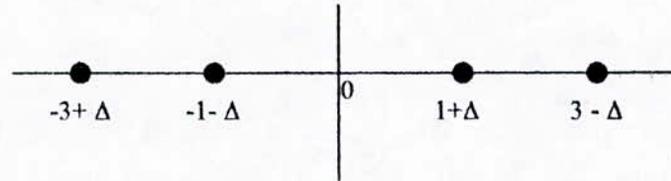


Figure 3.13 : Constellation de la 4-AM Asymétrique

Pour $\Delta = 0$ la constellation est symétrique, la puissance moyenne de l'ensemble des signaux de la figure 3.13 est :

$$P_{av} = 5 - 2\Delta + \Delta^2 \quad \dots(3.31)$$

- **Codeur en treillis à 2 états :**

Pour obtenir l'ensemble de signaux de la 4-AM asymétriques optimum, nous choisissons le décalage Δ qui minimise la probabilité par bit.

Pour un rapport signal sur bruit largement suffisant, l'optimisation de l'ensemble de signaux est équivalent à maximiser la distance libre d_{free} .

La distance libre d_{free} est donnée par :

$$d_{free}^2 = \frac{16 + 4(1 + \Delta)^2}{P_{av}} \quad \dots(3.32)$$

La valeur maximum de d_{free} égale à 8 est atteinte pour $\Delta = 1$, nous notons que lorsque cette limite est atteinte les deux signaux à gauche et à droite de la constellation asymétriques coïncident entre eux ce qui résulte à un codeur catastrophique, en conclusion nous essayons d'approcher de la distance libre maximum mais ne jamais l'atteindre pour éviter le codeur catastrophique.

La figure 3.14 illustre la probabilité d'erreur moyenne par bit en fonction du rapport signal sur bruit SNR pour la 4-AM Asymétriques, symétrique codées en treillis à 2 états ainsi que pour la modulation 2-AM non codée :

Le tableau 3.2 résume l'étude sur les modulations 4-AM, 8-AM, 16-AM combinées avec des codeurs en treillis à 2, 4, 8, 16 états en spécifiant le nombre de transitions parallèles entre états pour chaque configuration, la distance minimum de la modulation non codée de référence, la distance libre pour chaque configuration symétrique et asymétriques, le décalage optimum Δ_{opt} et le gain dû à la modulation Asymétriques.

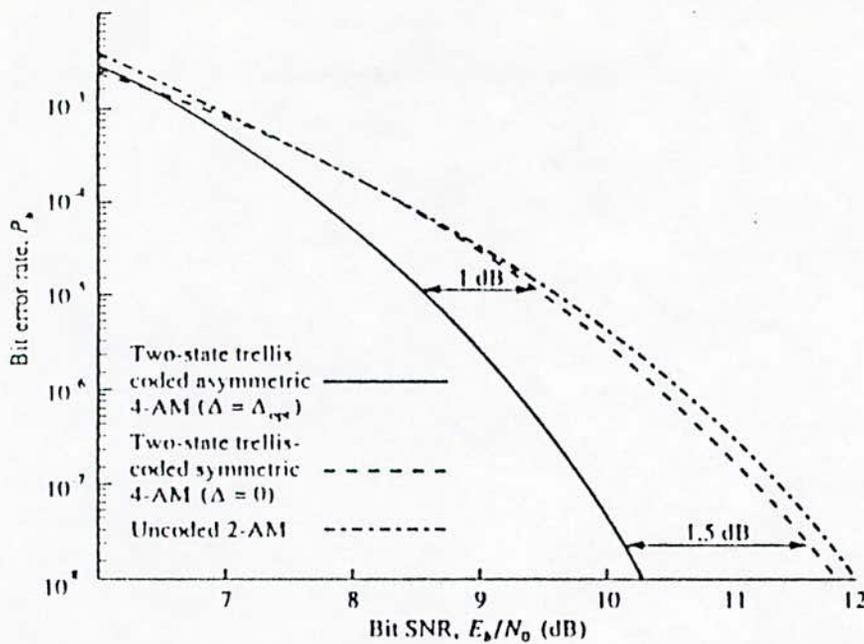


Figure 3.14 : Bornes supérieures pour la 4-AM Symétrique et Asymétrique et la 2-AM non codée

Nombre de niveaux $M=2^{m+1}$	Nombre De Transitions Parallèles Entre états	Nombre D'états dans le treillis	La distance Minimum pour la modulation non codée $M/2$ niveaux	La distance libre d_{free}^2 M-AM symétrique	la distance libre d_{free}^2 M-AM Asym	Δ_{opt}	Gain de la M-AM sym codée par rapport à la $M/2$ AM non codée (dB)	Gain de la M-AM Asym codée par rapport à la M AM sym codée (dB)
4	1	2	4	4	8	1	0	3,01
4	1	4	4	36/5	8	0,17	2,552	0,458
4	1	8	4	8	9,17	0,17	3	0,593
8	2	2	4/5	20/21	8/5	1	0,76	2,24
8	2	4	4/5	12/7	1,812	0,17	3,31	0,24
8	1	8	4/5	40/21	2,078	0,17	3,77	0,38
16	4	2	4/21	20/85	8/21	1	0,918	2,092
16	4	4	4/21	36/85	0,442	0,17	3,47	0,192
16	2	8	4/21	40/85	2,511	0,17	3,47	0,329

Tableau 3.2

Du tableau 3.2, nous constatons que :

- Les modulations M-AM Asymétriques combinées avec le codeur à 2 états apporte un gain maximum par rapport aux modulations M-AM Symétriques combinées avec le même codeur. Ce résultat n'est jamais réalisé en pratique car ces codeurs sont catastrophiques.

- Pour les autres, il y a une amélioration des configurations TCM Asymétriques par rapport aux configurations TCM Symétriques.

3-4 Conclusion :

Dans ce chapitre nous avons vu que l'introduction de la modulation Asymétrique est une procédure qui augmente la distance libre et donc améliore la modulation codée en treillis.

Par conséquent, nous aurons une diminution de la probabilité d'erreur par bit pour un rapport signal sur bit donné. L'analyse des tableaux 3.1 et 3.2 nous a montré cette performance.

Toutefois dans certains cas l'amélioration asymptotique dans la réduction du rapport signal sur bruit peut seulement être réalisée dans la limite où les points de signaux de la constellation émergent entre eux et le codeur en treillis devient catastrophique. Comme exemple le codeur à deux états qui pour toute valeur de m (nombre d'entrées du codeur) résulte asymptotiquement à un gain de 3 dB sur la même bande passante du système non codé, mais en réalité cette performance ne peut jamais être réalisée.

Un autre problème c'est qu'en ayant les points de signaux trop serrés dans la constellation le système devient moins robuste en présence d'erreur de phase de synchronisation.

Pour remédier à ce problème et améliorer en plus le gain de codage, il faut utiliser les configurations TCM multidimensionnelles.

CHAPTER

4

Chapitre 4

Implémentation de l'algorithme de recherche des codeurs optimaux de la TCM

4-1 Introduction :

La simulation permet l'évaluation des performances des systèmes de communication numériques et constitue aussi un outil puissant qui aide à l'étude et la conception de tels systèmes.

Nous citons quelques raisons montrant l'importance de cet outil :

- quelques niveaux d'analyse ne sont pas faisable avec les outils mathématiques traditionnels surtout qu'on est en face d'un modèle de canal et d'un système complexes.
- il est facile de passer d'un modèle de simulation à son implémentation matérielle, et pour cela la distinction entre un modèle de simulation et un prototype d'un système donné devient minimale, d'où l'intérêt économique de la simulation qui réduit considérablement le temps et le coût d'un système et qui se prête plus facilement à la correction et à l'amélioration qu'un prototype.

Dans ce chapitre, nous allons implémenter l'algorithme, proposé par Ungerboeck, de recherche des codes optimaux pour différents types de modulation codée en treillis à deux dimensions.

4-2 Modèle de simulation :

Le modèle de simulation est constitué par les blocs suivants :

- Une source d'information binaire.
- Un bloc d'aiguillage qui partage l'information en bloc de m bits et faisant la répartition.
- Un bloc faisant l'embrouillage (scramble) de la séquence d'entrée pour obtenir une nouvelle séquence plus aléatoire qui attaque le codeur systématique à feed-back.
- Un bloc faisant l'aiguillage qui partage la nouvelle séquence en bloc de m bits.
- Un codeur systématique à feed-back.
- Un bloc de calcul de la distance d_{free} et les paramètres de la matrice de parité.

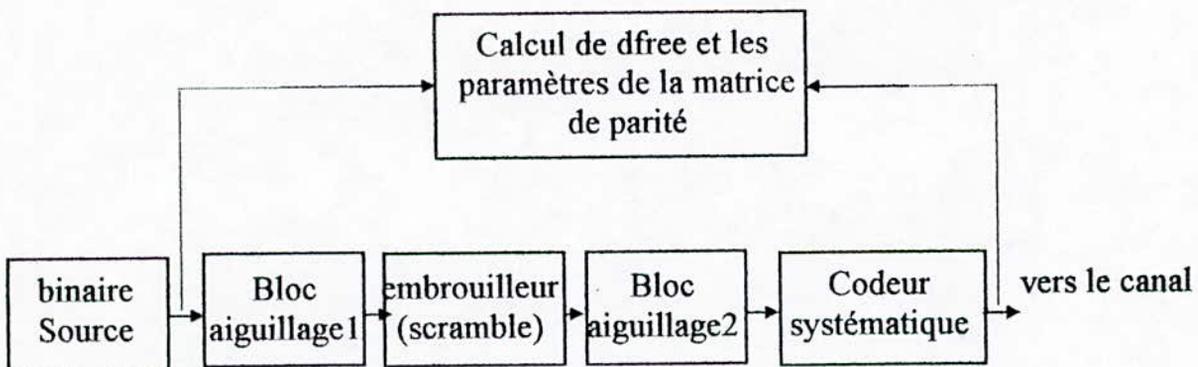


Figure 4.1 : Modèle de simulation

4-2-1 Source d'information :

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

La méthode adoptée par la CCITT, consiste à utiliser une séquence pseudo-aléatoire, séquence périodique dont les propriétés statistiques sont voisines de celles d'un trafic réel (aléatoire).

Le générateur d'une telle séquence est constitué d'un registre à décalage comportant m étages, et des additionneurs modulo-2. Le choix de m détermine la période de la séquence qui est égale à $2^m - 1$.

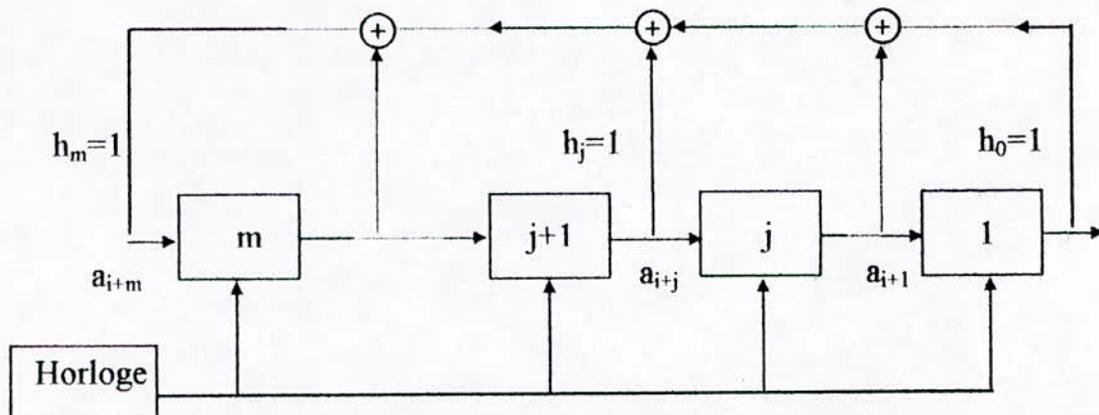


Figure 4.2 : Générateur pseudo-aléatoire

La séquence générée comporte 2^{m-1} bits égaux à 1 et $2^{m-1} - 1$ bits égaux à 0 ce qui donne une répartition presque uniforme des 1 et des 0.

4-2-2 Bloc aiguillage1 :

Ce bloc partage la séquence pseudo-aléatoire en parties contenant chacune m bits d'information distribués aux m entrées.

4-2-3 Bloc d'embrouillage (Scramble) :

Ce bloc est constitué de registres à décalage, il effectue un embrouillage sur la séquence d'entrée pour la rendre plus aléatoire.

4-2-4 Bloc aiguillage2 :

Ce bloc partage la séquence embrouillée en parties contenant chacune m bits d'information distribués aux m entrées du codeur systématique à boucle de retour.

4-2-5 Codeur systématique :

Ce codeur à boucle de retour constitué de v éléments mémoires de rendement $(m/m+1)$.

4-2-6 Bloc de calcul :

Dans ce bloc s'effectue le calcul de la distance libre d_{free} et les paramètres de la matrice de parité.

4-3 L'organigramme de recherche des codes optimaux :

L'organigramme étudié dans le chapitre 2 (2-7-6) est représenté dans la figure 4.3.

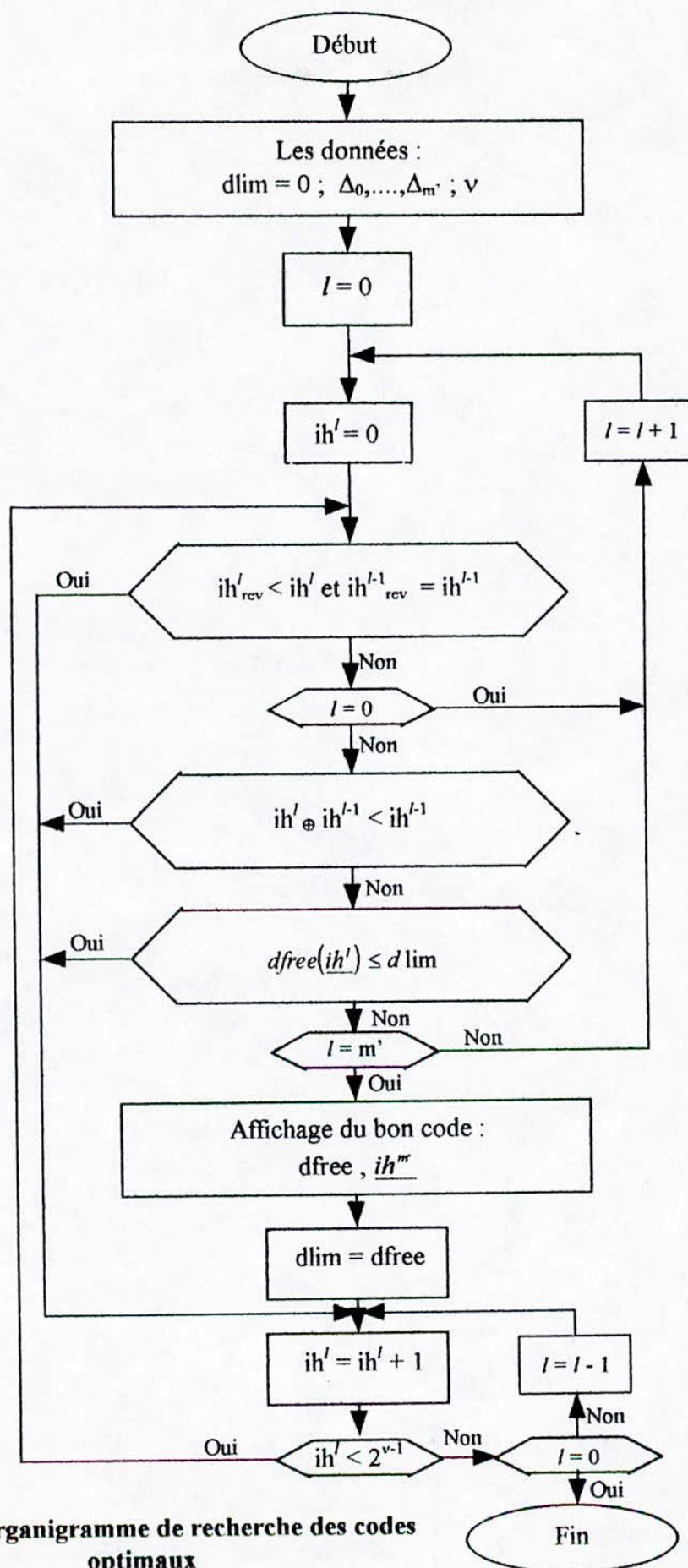


Figure 4.3 : Organigramme de recherche des codes optimaux

4-3-1 Langage utilisé :

Vue la complexité de l'organigramme qui a été proposé par Ungerboeck, les boucles ne sont pas conventionnelles, il y a un chevauchement, l'utilisation de l'instruction GOTO est indispensable, donc nous avons utilisé le langage F 32 (Fortran à 32 bits) qui présente une grande souplesse par rapport au fortran 77. F32 travaille dans un environnement sous windows.

4-3-2 Programmation :

L'implémentation de cet organigramme nous a conduit à la conception de :

- 11 fonctions.
- 10 subroutines

4-3-3 Les fonctions :

Les fonctions utilisées sont :

- **la fonction Bit** : A pour paramètres l'entier ih et le numéro de bit ib, cette fonction donne le bit numéro ib de l'entier ih. Si $ih = 6, ib = 2 \Rightarrow \text{Bit}(ih, ib) = 1$.
- **La fonction rev** : A pour paramètres l'entier ih et le nombre d'éléments mémoires v, l'entier ih s'écrit en (v-1) bits, cette fonction donne l'entier réversible de l'entier ih. Ceci nous a permis d'implémenter la règle 1. Si $ih = 2, v = 3 \Rightarrow \text{Bit}(ih, v) = 1$.
- **La fonction xor2** : A pour paramètres les entiers ih1, ih2 et v. Cette fonction donne l'entier issu du xor de ih1 et ih2 bit par bit. Ceci nous a permis d'implémenter la règle 2. Si $ih1 = 2, ih2 = 3, v = 3 \Rightarrow \text{xor2}(ih1, ih2, v) = 1$.
- **La fonction ixor** : A pour paramètres deux bits i1, i2; elle donne le xor de ces deux bits.
- **La fonction Lastbit** : Cette fonction donne le bit de poids faible de la sortie du codeur systématique à feed-back.
- **Les fonctions abaque** : Il y a 6 fonctions, une pour chaque type de modulation, elles donnent les distances entre les sous-ensembles de la partition d'alphabet en connaissant les sorties du codeur systématique et le type de la modulation (M-AM, M-PSK, QAM), ceci permet de réaliser le bloc de calcul de d_{free} .
- **La fonction GCD** : Une fois qu'on a les polynômes de la matrice de parité $H^0(D), \dots, H^m(D)$, ensuite en les écrivant en entiers, cette fonction nous donne leur plus grand diviseur commun et ceci nous permet d'accepter ou de rejeter le code, donc d'implémenter la règle 4.

4-3-4 Les subroutines :

Les subroutines utilisées sont :

- **makesbpa** : génère une séquence pseudo-aléatoire.
 - **makexaig0 et makexaig01** : permet de réaliser le bloc aiguillage1 selon le nombre d'éléments mémoires.
 - **makexdot0 et makexdot01** : permet de réaliser le bloc embrouilleur (Scramble).
-

- **mattovect** : passage matrice – vecteur.
- **makexdot1** : simule le bloc aiguillage2.
- **extraire** : permet d'extraire l'entier de la matrice de parité spécifié par le niveau (numéro de la ligne).
- **affecter** : permet d'écrire l'entier dans la matrice de parité sous forme de bits en spécifiant le numéro de la ligne.
- **update** : la mise à jour du bit de poids faible de la sortie du codeur systématique à feed-back.

4-4 Les modulations traitées :

Dans notre recherche, nous avons utilisé les modulations suivantes :

- La modulation 4-AM.
- La modulation 8-AM.
- La modulation 16-AM.
- La modulation 8-PSK.
- La modulation 16-PSK.
- La modulation QAM.

Ces modulations sont combinées avec des codeurs à 4,8,16 et 32 états pour $v = 2,3,4$ et 5 respectivement.

Au total nous avons conçu 24 codes .

4-5 Résultats :

Nous avons réalisé les programmes de recherche des codes optimaux, pour chaque modulation nous donnons son propre programme principal. Ce genre de programmation est appelée programmation structurée.

Les difficultés rencontrées lors de l'implémentation de cet organigramme sont :

- Le bloc de calcul de la séquence embrouillée pour cela nous avons utilisé l'équivalence entre un codeur systématique à feed-back et un codeur d'Ungerboeck.
- Le bloc de calcul de la distance libre d_{free}

Données :

Dans ces tableaux nos résultats seront comparés aux résultats d'Ungerboeck.

- Le nombre d'entrées du codeur en treillis m qui spécifie la modulation utilisée.
- Le nombre d'entrées du codeur convolutif m' .
- Le nombre d'éléments mémoires v .

Paramètres à calculer :

- Les polynômes de la matrice de parité $H^0(D), \dots, H^m(D)$.
- La distance libre d_{free} .

Représentation des résultats :

Les résultats obtenus par cette simulation sont groupés dans les tableaux ci-dessous. Les représentations des codeurs optimaux sont faites selon le type de modulation utilisé et le nombre d'éléments mémoires.

Nous donnerons pour chaque v trois représentations :

- Représentation du codeur systématique typique.
- Représentation du codeur simulé.
- Représentation du codeur d'Ungerboeck.

4-5-1 La modulation 8-PSK :

Données			Résultats de la simulation				Résultats d'Ungerboeck			
M	m'	v	H ⁰ (D)	H ¹ (D)	H ² (D)	d _{free}	H ⁰ (D)	H ¹ (D)	H ² (D)	D _{free}
2	1	2	5	2	---	1.414	5	2	---	2
2	2	3	11	02	04	2.141	11	02	04	2.141
2	2	4	21	02	06	2.274	23	04	16	2.273
2	2	5	41	02	10	2.399	45	16	34	2.399

Tableau 4.1

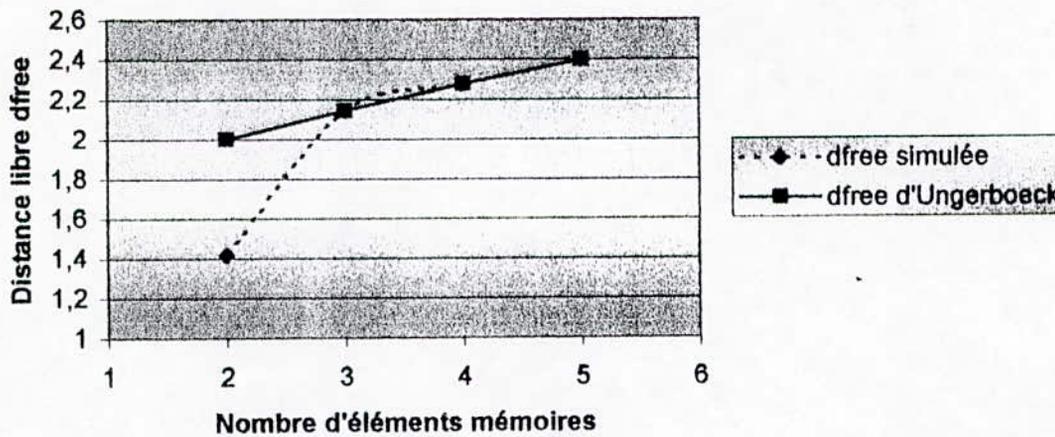


Figure 4.4 : Les distances libres simulée et d'Ungerboeck en fonction du nombre d'éléments mémoires

$v=2$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent de manière générale :

$$H^0(D) = D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_1^1 \cdot D^1 + 0$$

où h_1^0 et h_1^1 sont les paramètres qu'il faut trouver.

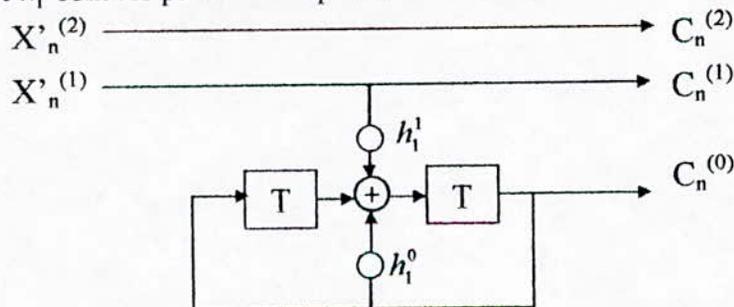


Figure 4.5 : Codeur systématique Typique pour la 8-PSK avec $v=2$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=5$$

$$H^1(D)=2$$

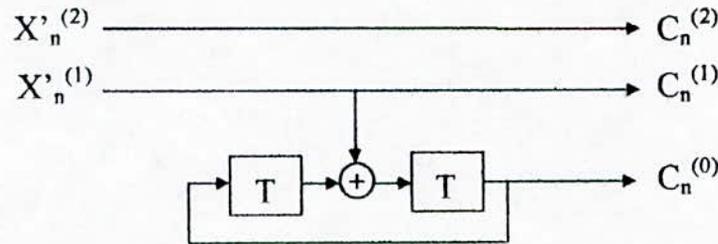


Figure 4.6 : Codeur de la simulation pour la 8 PSK avec v=2

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$$H^0(D)=5$$

$$H^1(D)=2$$

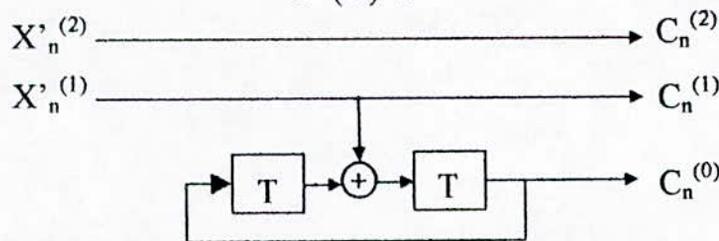


Figure 4.7 : Codeur d'Ungerboeck pour la 8 PSK avec v=2

v = 3 : Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

$$H^2(D) = 0 + h_2^2 \cdot D^2 + h_1^2 \cdot D^1 + 0$$

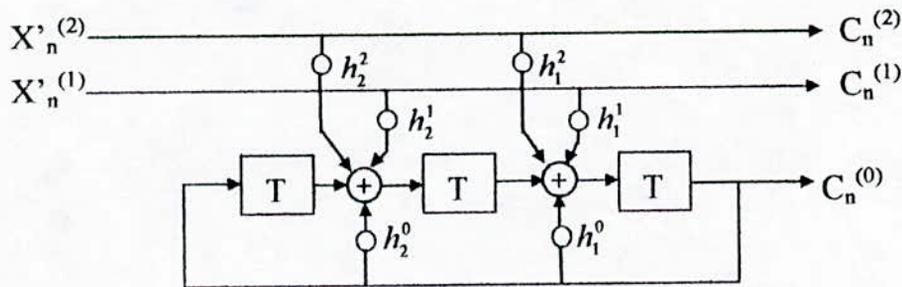


Figure 4.8 : Codeur systématique pour la 8-PSK avec v = 3

Codeur simulé :

Les polynômes que nous avons trouvés :

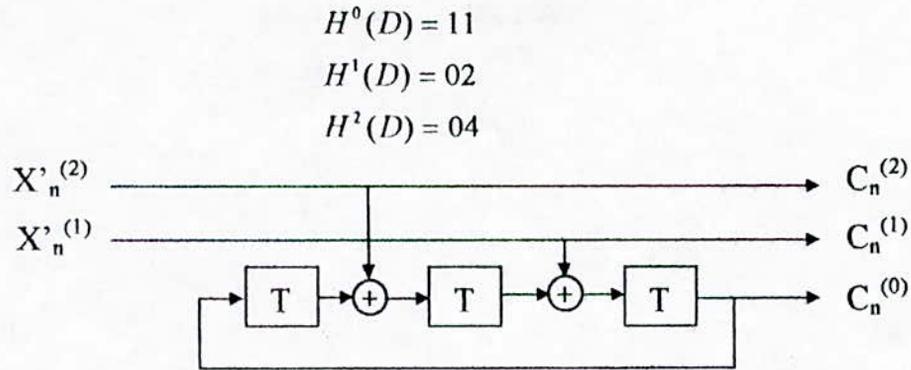


Figure 4.9 : Codeur de la simulation pour la 8 PSK avec $v=3$

Codeur d'Ungerboeck :

Les polynômes donnés par Ungerboeck :

$$H^0(D) = 11$$

$$H^1(D) = 02$$

$$H^2(D) = 04$$

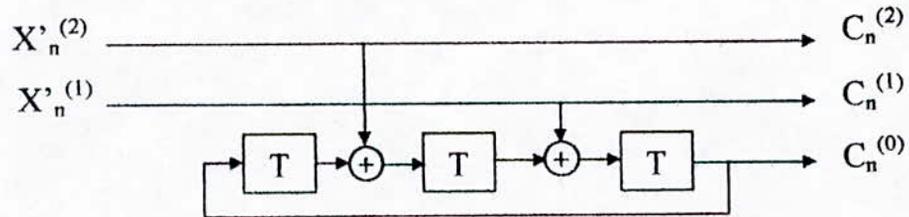


Figure 4.10 : Codeur d'Ungerboeck pour la 8 PSK avec $v=3$

$v = 4$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^4 + h_3^0 \cdot D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_3^1 \cdot D^3 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

$$H^2(D) = 0 + h_3^2 \cdot D^2 + h_2^2 \cdot D^2 + h_1^2 \cdot D^1 + 0$$

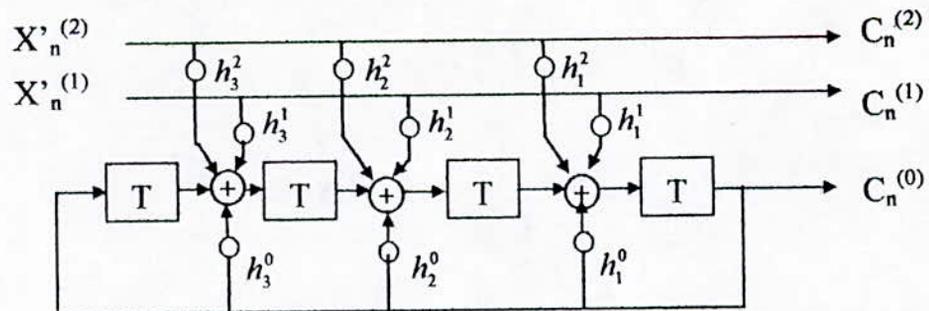


Figure 4.11 : Codeur systématique pour la 8-PSK avec $v=4$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D) = 21$$

$$H^1(D) = 02$$

$$H^2(D) = 06$$

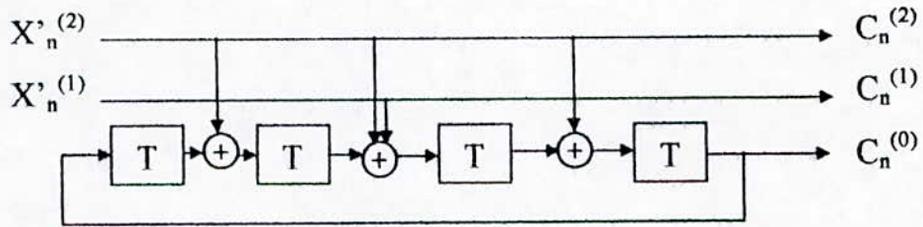


Figure 4.12 : Codeur de la simulation pour la 8 PSK avec $v=4$

Codeur d'Ungerboeck :

Les polynômes que Ungerboeck à trouver :

$$H^0(D) = 23$$

$$H^1(D) = 04$$

$$H^2(D) = 16$$

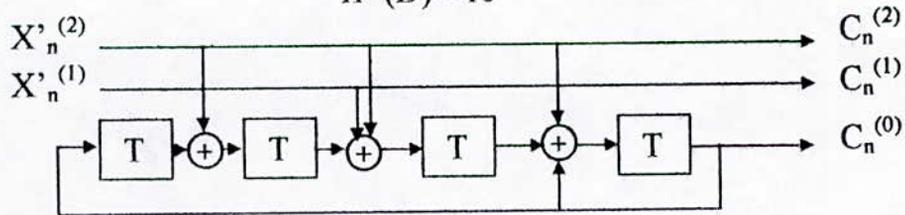


Figure 4.13 : Codeur d'Ungerboeck pour la 8 PSK avec $v=4$

$v = 5$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^5 + h_4^0 \cdot D^4 + h_3^0 \cdot D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_4^1 \cdot D^4 + h_3^1 \cdot D^3 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

$$H^2(D) = 0 + h_4^2 \cdot D^4 + h_3^2 \cdot D^3 + h_2^2 \cdot D^2 + h_1^2 \cdot D^1 + 0$$

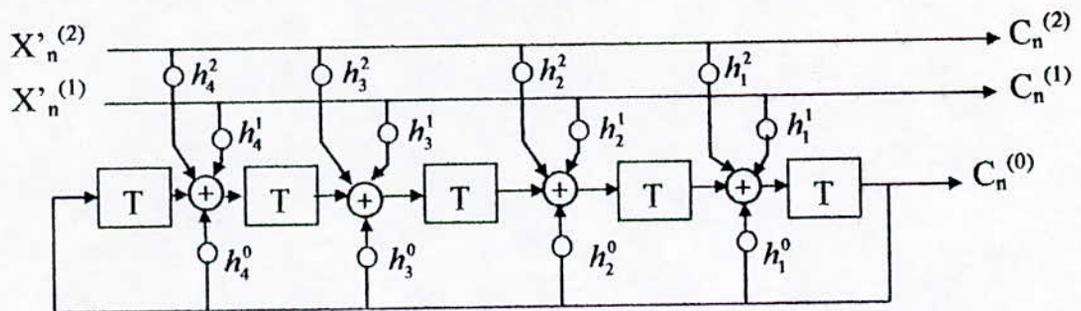


Figure 4.14 : Codeur systématique pour la 8-PSK avec $v=5$

Codeur simulé :

Les polynômes que nous avons trouvés :

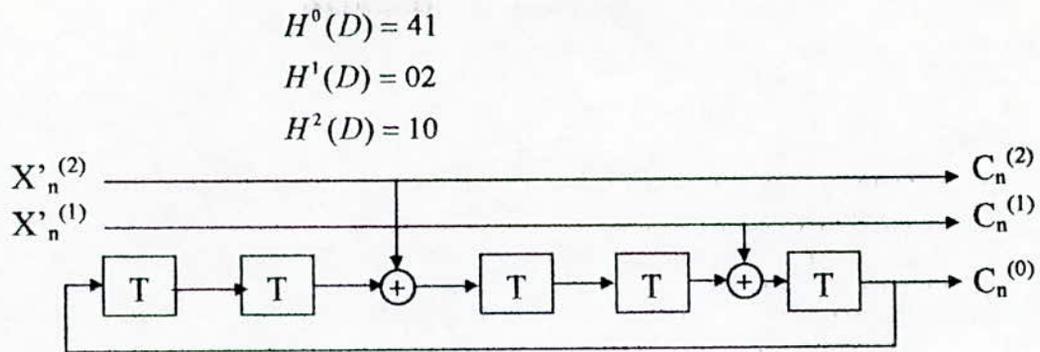


Figure 4.15 : Codeur de la simulation pour la 8 PSK avec $v=5$

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$H^0(D)=45$

$H^1(D)=16$

$H^2(D)=34$

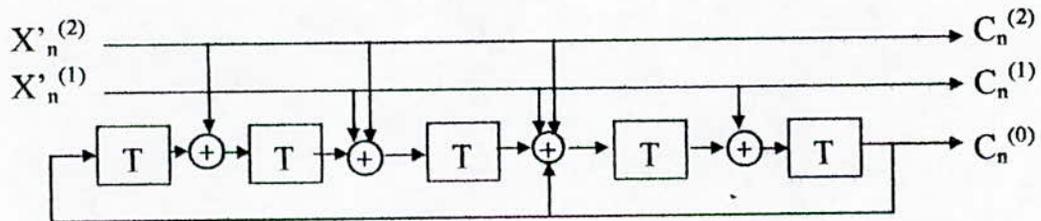


Figure 4.16 : Codeur d'Ungerboeck pour la 8 PSK avec $v=5$

4-5-2 La modulation 16 PSK :

Données			Résultats de la simulation				Résultats d'Ungerboeck			
M	M'	v	$H^0(D)$	$H^1(D)$	$H^2(D)$	Dfree	$H^0(D)$	$H^1(D)$	$H^2(D)$	dfree
3	1	2	5	2	---	1.15	5	2	---	1.15
3	1	3	11	02	---	1.224	13	04	---	1.214
3	1	4	21	04	---	1.28	23	04	---	1.275
3	1	5	41	06	---	1.295	45	10	---	1.38

Tableau 4.2

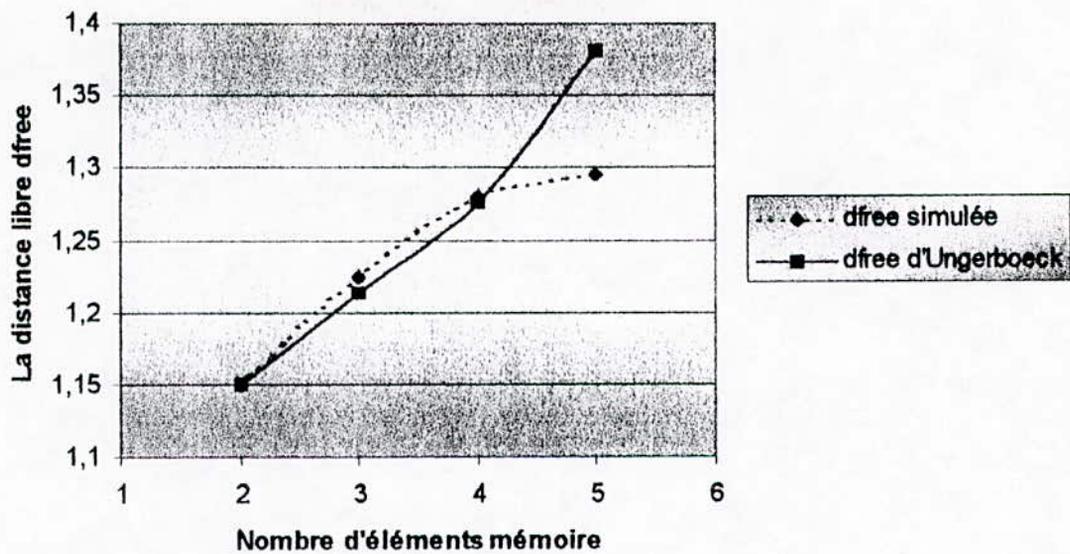


Figure 4.17 : Distances libres simulée et d'Ungerboeck en fonction du nombre d'éléments mémoires

v = 2 : Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_1^1 \cdot D^1 + 0$$

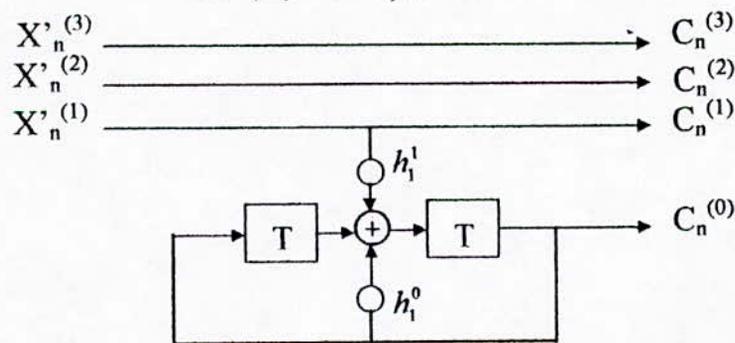


Figure 4.18 : Codeur systématique pour la 16-PSK avec v=2
Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D) = 5$$

$$H^1(D) = 2$$

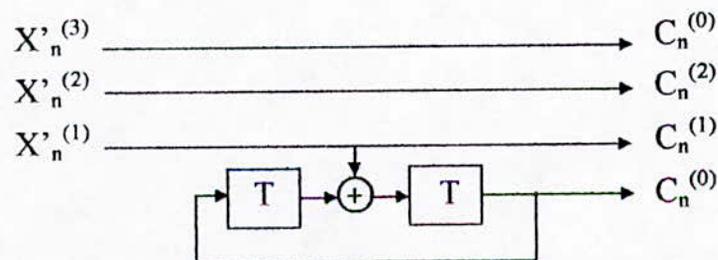


Figure 4.19 : Codeur de la simulation pour la 16 PSK avec v=2

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$$H^0(D)=5$$

$$H^1(D)=2$$

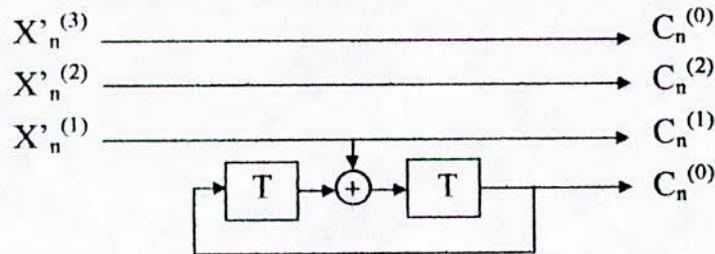


Figure 4.20 : Codeur d'Ungerboeck pour la 16 PSK avec $v=2$

$v = 3$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D)= D^3 + h_2^0.D^2 + h_1^0.D^1 + 1$$

$$H^1(D)=0+h_2^1.D^2 + h_1^1.D^1 + 0$$

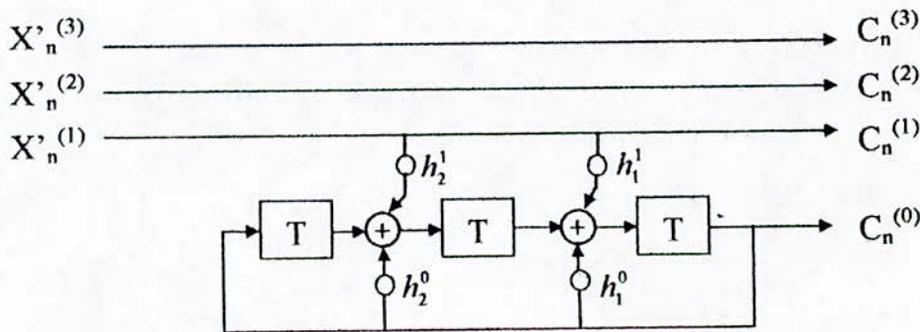


Figure 4.21 : Codeur systématique pour la 16-PSK avec $v = 3$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=11$$

$$H^1(D)=02$$

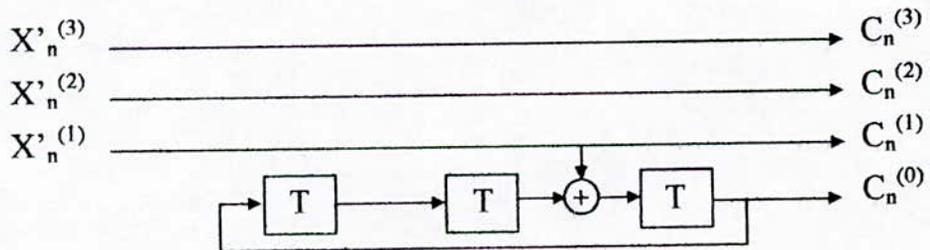


Figure 4.22 : Codeur de la simulation pour la 16 PSK avec $v=3$

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

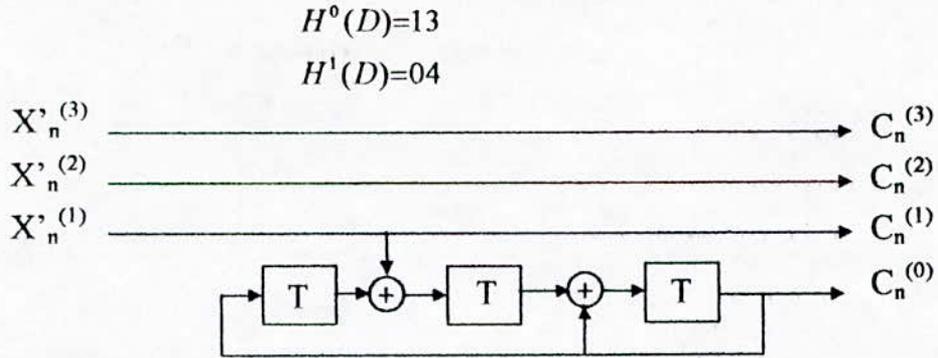


Figure 4.23 : Codeur d'Ungerboeck pour la 16 PSK avec $v=3$

$v = 4$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^4 + h_3^0 \cdot D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_3^1 \cdot D^3 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

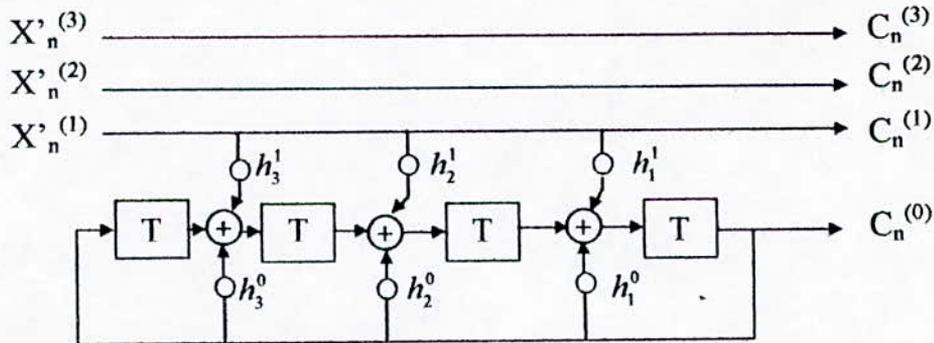


Figure 4.24 : Codeur systématique pour la 16-PSK avec $v=4$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=21$$

$$H^1(D)=04$$

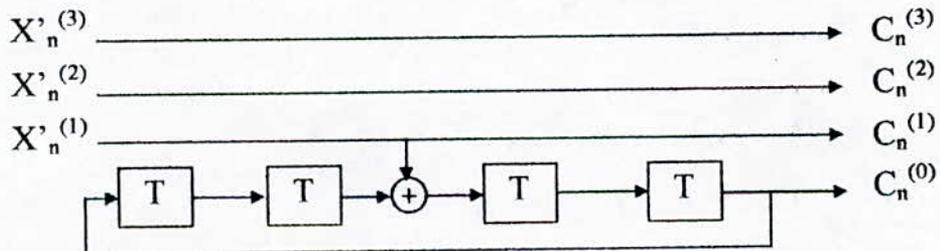


Figure 4.25 : Codeur de la simulation pour la 16 PSK avec $v=4$

Codeur d'Ungerboeck :

Les polynômes donnés par Ungerboeck :

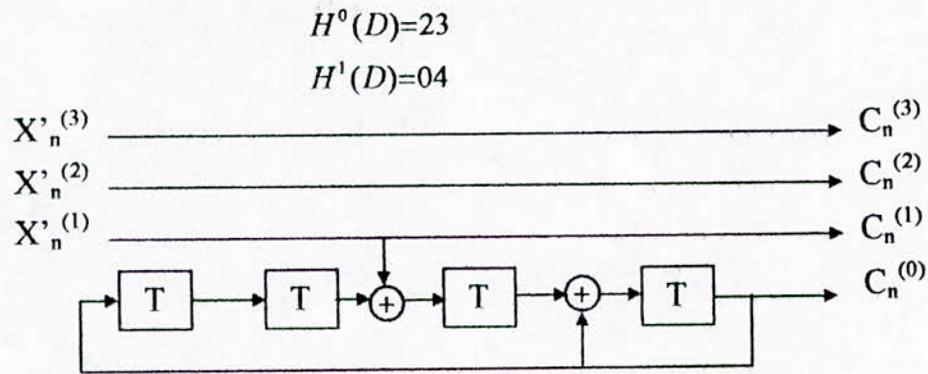


Figure 4.26 : Codeur d'Ungerboeck pour la 16 PSK avec $v=4$

$v = 5$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^5 + h_4^0 \cdot D^4 + h_3^0 \cdot D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_4^1 \cdot D^4 + h_3^1 \cdot D^3 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

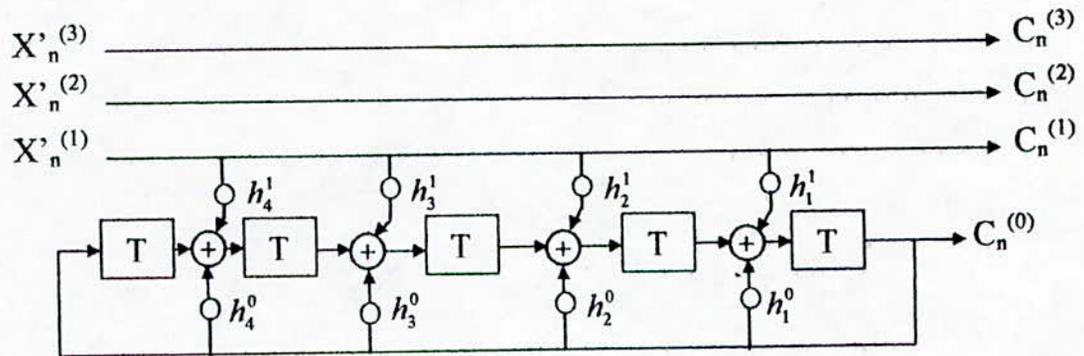


Figure 4.27 : Codeur systématique pour la 16-PSK avec $v = 5$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=41$$

$$H^1(D)=06$$

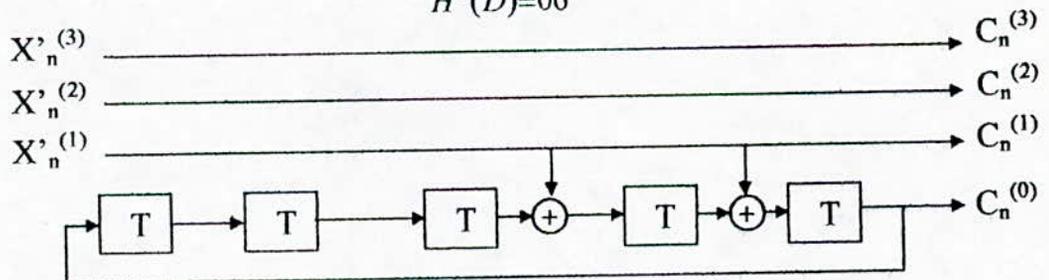


Figure 4.28 : Codeur de la simulation pour la 16 PSK avec $v=5$

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

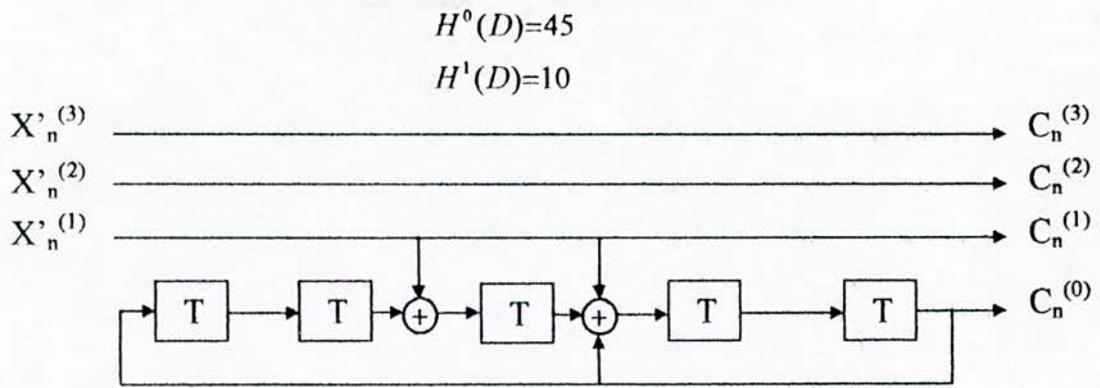


Figure 4.29 : Codeur d'Ungerboeck pour la 16 PSK avec $v=5$

4-5-3 La modulation 4-AM :

Données			Résultats de la simulation				Résultats d'Ungerboeck			
m	M'	v	$H^0(D)$	$H^1(D)$	$H^2(D)$	Dfree	$H^0(D)$	$H^1(D)$	$H^2(D)$	dfree
1	1	2	7	0	---	1.788	5	2	---	2.682
1	1	3	13	04	---	1.999	13	04	---	2.82
1	1	4	21	12	---	2.682	23	04	---	2.965
1	1	5	41	14	---	2.827	45	10	---	3.22

Tableau 4.3

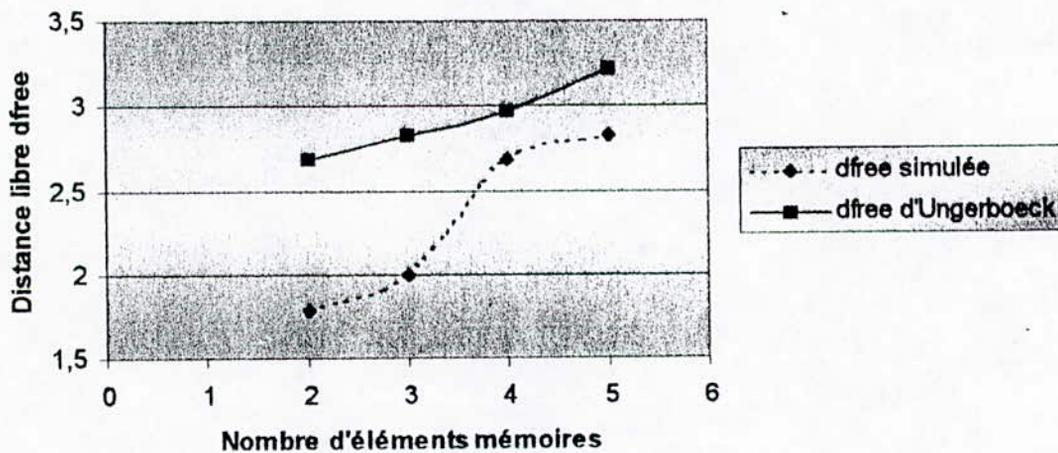


Figure 4.30 : Distances libres simulée et d'Ungerboeck en fonction du nombre d'éléments mémoires

$v = 2$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_1^1 \cdot D^1 + 0$$

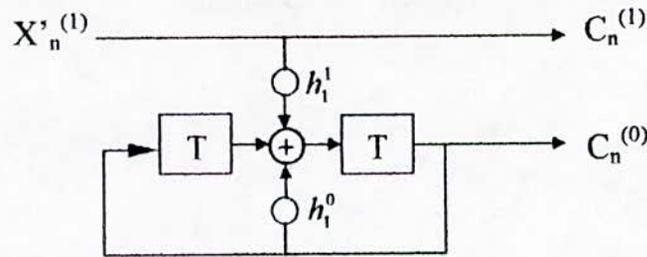


Figure 4.31 : Codeur systématique pour la 4-AM avec $v = 2$
Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=7$$

$$H^1(D)=0$$

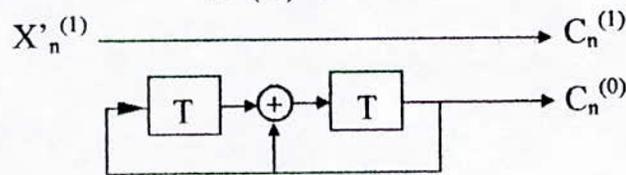


Figure 4.32 : Codeur de la simulation pour la 4-AM avec $v = 2$

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$$H^0(D)=5$$

$$H^1(D)=2$$

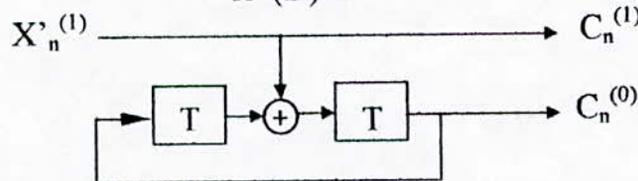


Figure 4.33 : Codeur d'Ungerboeck pour la 4 AM avec $v=2$

$v = 3$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D)=D^3 + h_2^0.D^2 + h_1^0.D^1 + 1$$

$$H^1(D)=0+h_2^1.D^2 + h_1^1.D^1 + 0$$

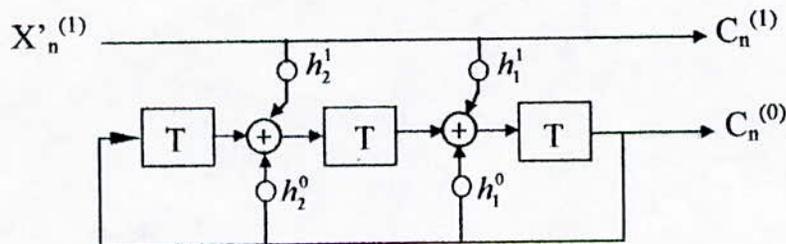


Figure 4.34 : Codeur systématique pour la 4-AM avec $v = 3$

Codeur simulé :

Les polynômes que nous avons trouvés :

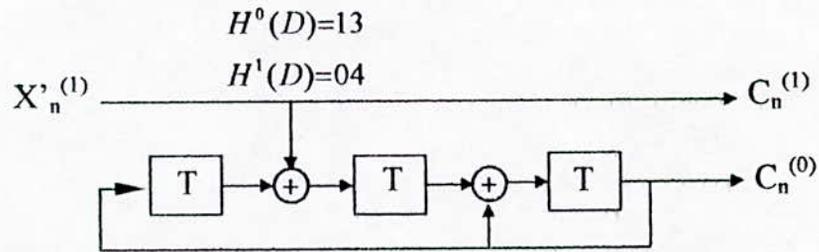


Figure 4.35 : Codeur de la simulation pour la 4-AM avec $v=3$

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$$H^0(D)=13$$

$$H^1(D)=04$$

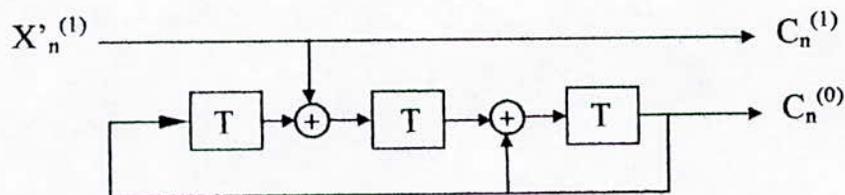


Figure 4.36 : Codeur d'Ungerboeck pour la 4 AM avec $v=3$

$v=4$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^4 + h_3^0 \cdot D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_3^1 \cdot D^3 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

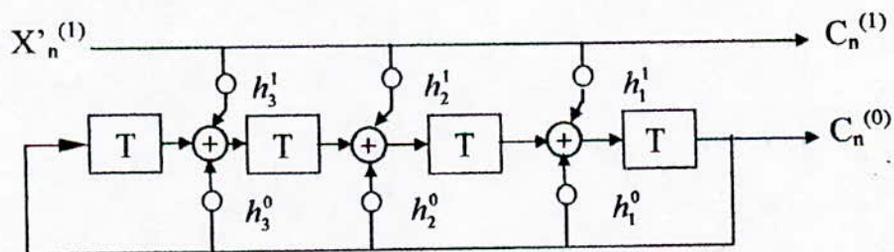


Figure 4.37 : Codeur systématique pour la 4-AM avec $v=4$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=21$$

$$H^1(D)=12$$

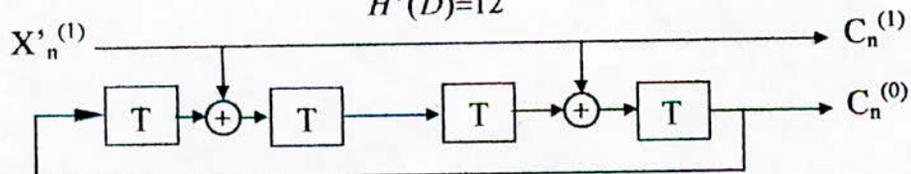


Figure 4.38 : Codeur de la simulation pour la 4-AM avec $v=4$

Codeur d'Ungerboeck :

Les polynômes donnés par Ungerboeck :

$$H^0(D)=23$$

$$H^1(D)=04$$

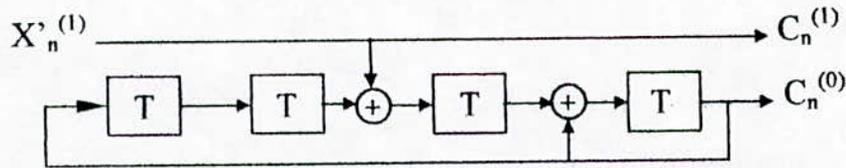


Figure 4.39 : Codeur d'Ungerboeck pour la 4 AM avec v=4

v = 5 : Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D)=D^5 + h_4^0.D^4 + h_3^0.D^3 + h_2^0.D^2 + h_1^0.D^1 + 1$$

$$H^1(D)=0+h_4^1.D^4 + h_3^1.D^3 + h_2^1.D^2 + h_1^1.D^1 + 0$$

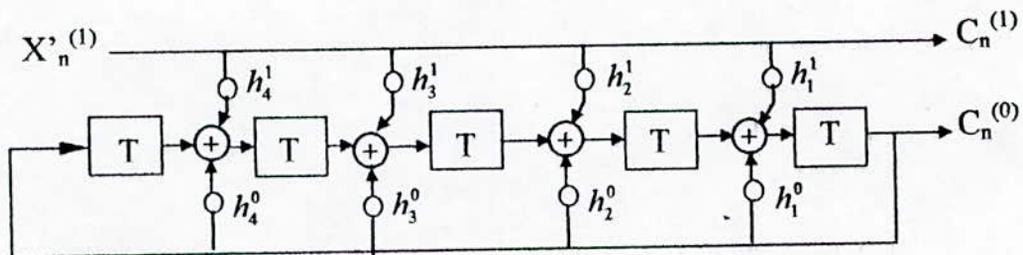


Figure 4.40 : Codeur systématique pour la 4-AM avec v =5

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=41$$

$$H^1(D)=14$$

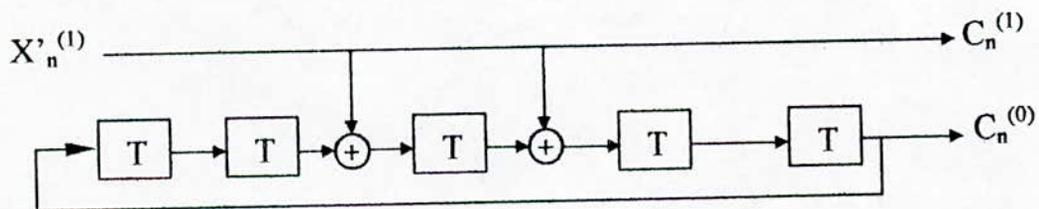


Figure 4.41 : Codeur de la simulation pour la 4 AM avec v=5

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$$H^0(D)=45$$

$$H^1(D)=10$$

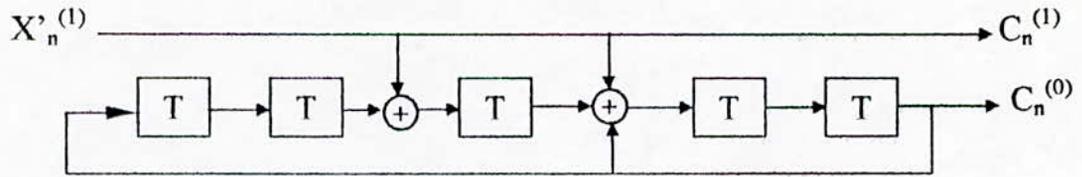


Figure 4.42 : Codeur d'Ungerboeck pour la 4 AM avec v=5

4-5-4 La modulation 8-AM :

Données			Résultats de la simulation				Résultats d'Ungerboeck			
m	m'	v	H ⁰ (D)	H ¹ (D)	H ² (D)	Dfree	H ⁰ (D)	H ¹ (D)	H ² (D)	dfree
2	1	2	5	2	---	0.872	5	2	---	1.34
2	1	3	11	02	---	0.974	11	04	---	1.41
2	1	4	21	04	---	1.067	23	04	---	1.482
2	1	5	41	12	---	1.378	45	10	---	1.61

Tableau 4.4

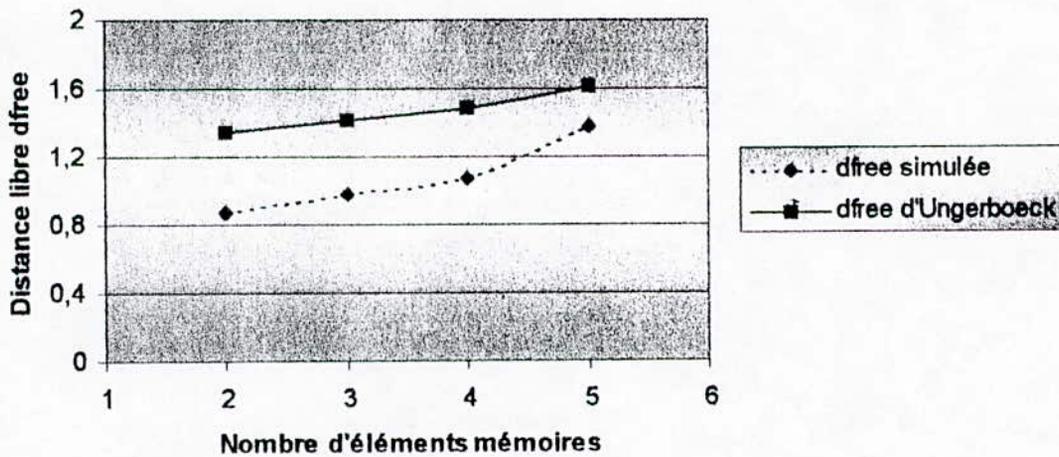


Figure 4.43 : Distances libres simulée et d'Ungerboeck en fonction du nombre d'éléments mémoires

v = 2 : Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_1^1 \cdot D^1 + 0$$

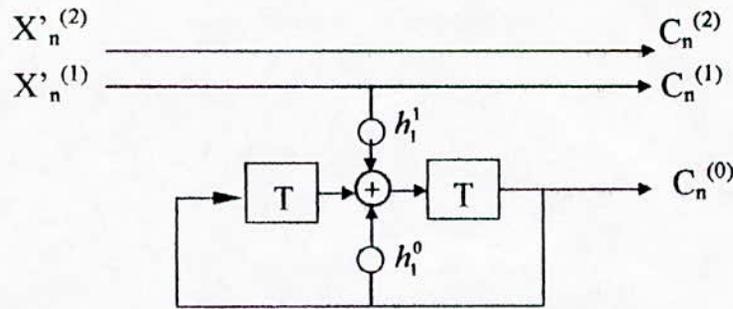


Figure 4.44 : Codeur systématique pour la 8-AM avec $v=2$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=5$$

$$H^1(D)=2$$

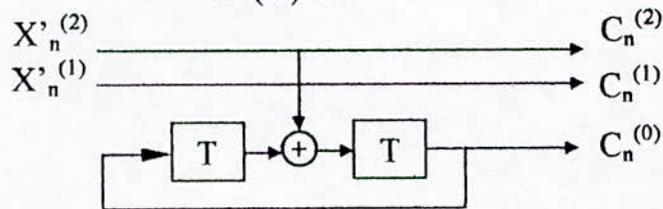


Figure 4.45 : Codeur de la simulation pour la 8 AM avec $v=2$

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$$H^0(D)=5$$

$$H^1(D)=2$$

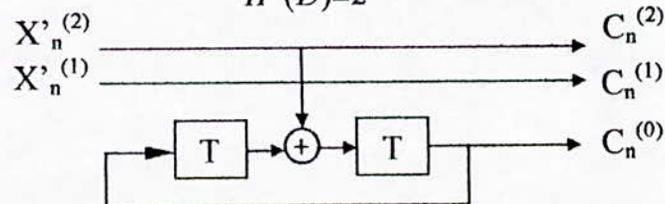


Figure 4.46 : Codeur d'Ungerboeck pour la 8 AM avec $v=2$

$v=3$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D)=D^3 + h_2^0.D^2 + h_1^0.D^1 + 1$$

$$H^1(D)=0+h_2^1.D^2 + h_1^1.D^1 + 0$$

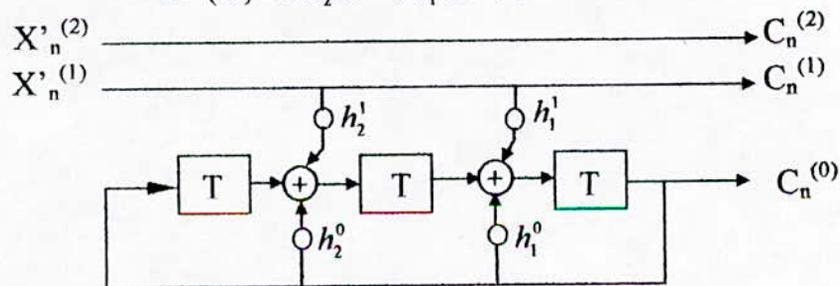


Figure 4.47 : Codeur systématique pour la 8-AM avec $v=3$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=11$$

$$H^1(D)=02$$

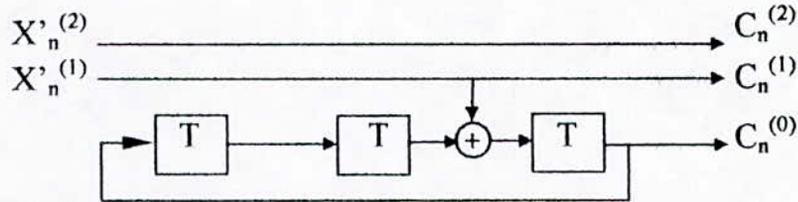


Figure 4.48 : Codeur de la simulation pour la 8 AM avec $v=3$

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$$H^0(D)=13$$

$$H^1(D)=04$$

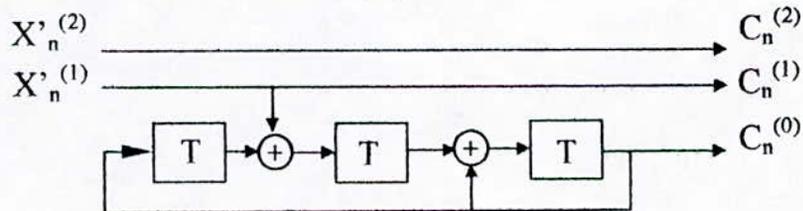


Figure 4.49 : Codeur d'Ungerboeck pour la 8 AM avec $v=3$

$v = 4$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D)= D^4 + h_3^0.D^3 + h_2^0.D^2 + h_1^0.D^1 + 1$$

$$H^1(D)=0+h_3^1.D^3 + h_2^1.D^2 + h_1^1.D^1 + 0$$

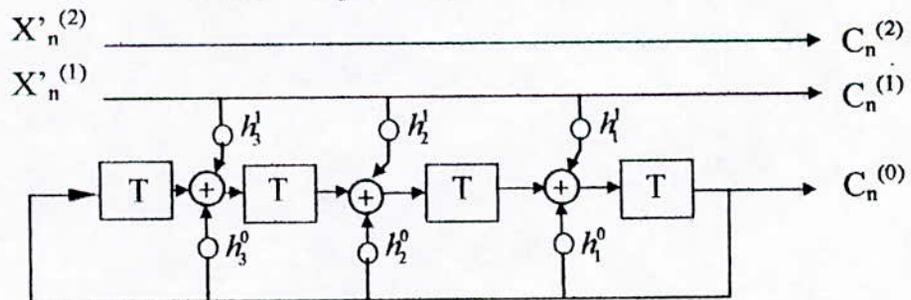


Figure 4.50 : Codeur systématique pour la 8 AM avec $v = 4$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=21$$

$$H^1(D)=04$$

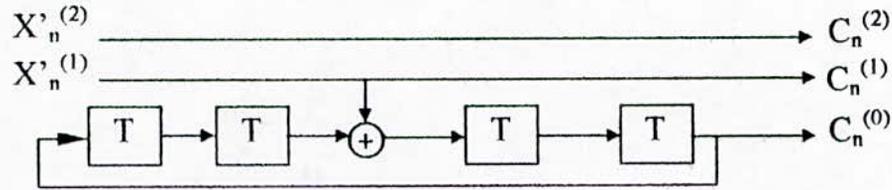


Figure 4.51 : Codeur de la simulation pour la 8 AM avec v=4

Codeur d'Ungerboeck :

Les polynômes donnés par Ungerboeck :

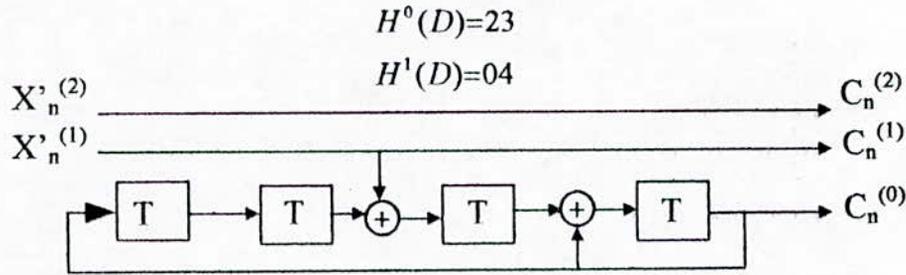


Figure 4.52 : Codeur de la simulation pour la 8 AM avec v=4

v = 5 : Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^5 + h_4^0.D^4 + h_3^0.D^3 + h_2^0.D^2 + h_1^0.D^1 + 1$$

$$H^1(D) = 0 + h_4^1.D^4 + h_3^1.D^3 + h_2^1.D^2 + h_1^1.D^1 + 0.$$

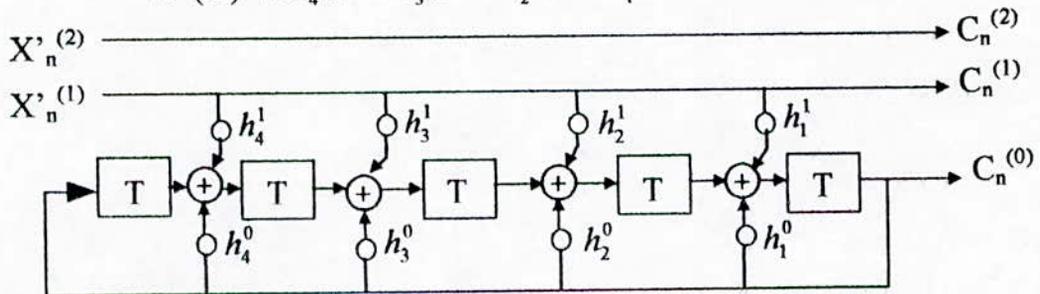


Figure 4.53 : Codeur systématique pour la 8-AM avec v =5

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=41$$

$$H^1(D)=12$$

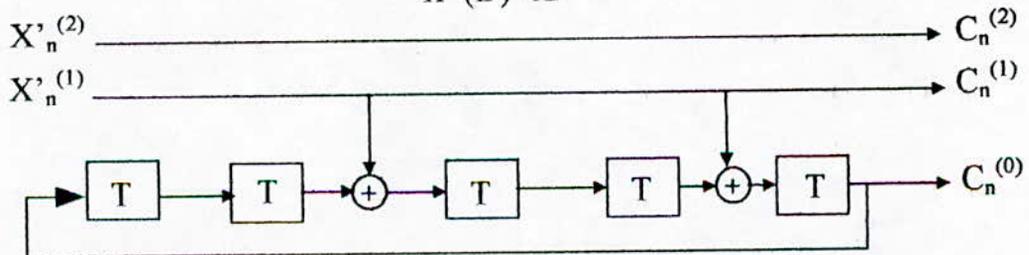


Figure 4.54 : Codeur de la simulation pour la 8 AM avec v=5

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$$H^0(D)=45$$

$$H^1(D)=10$$

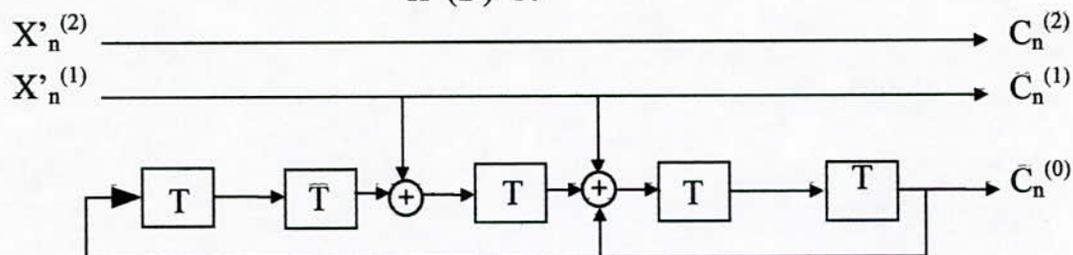


Figure 4.55 : Codeur de la simulation pour la 8 AM avec v=5

4-5-5 La modulation 16-AM :

Données			Résultats de la simulation				Résultats d'Ungerboeck			
m	m'	v	H ⁰ (D)	H ¹ (D)	H ² (D)	Dfree	H ⁰ (D)	H ¹ (D)	H ² (D)	dfree
3	1	2	5	2	---	0.867	5	2	---	0.654
3	1	3	11	02	---	0.97	13	04	---	0.689
3	1	4	21	04	---	0.993	23	04	---	0.723
3	1	5	41	06	---	0.999	45	10	---	0.786

Tableau 4.5

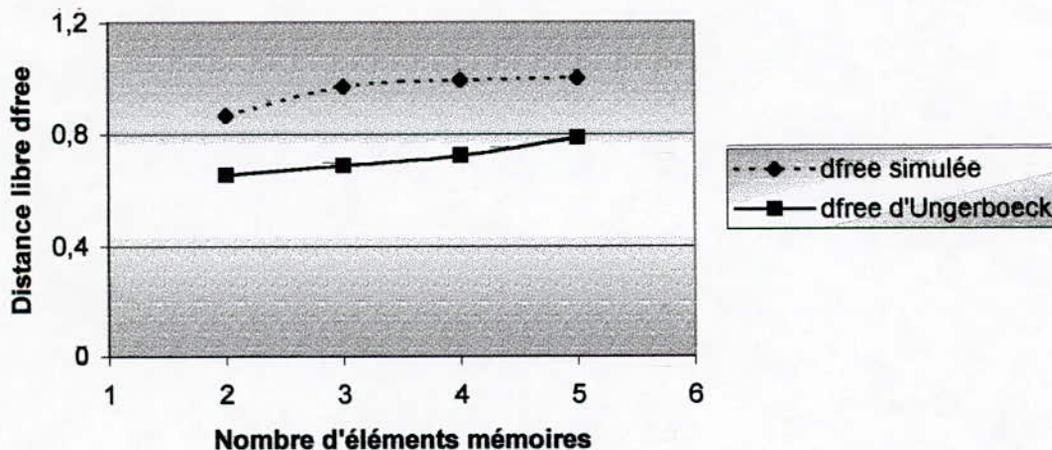


Figure 4.56 : Distances libres simulée et d'Ungerboeck en fonction du nombre d'éléments mémoires

v = 2 : Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_1^1 \cdot D^1 + 0$$

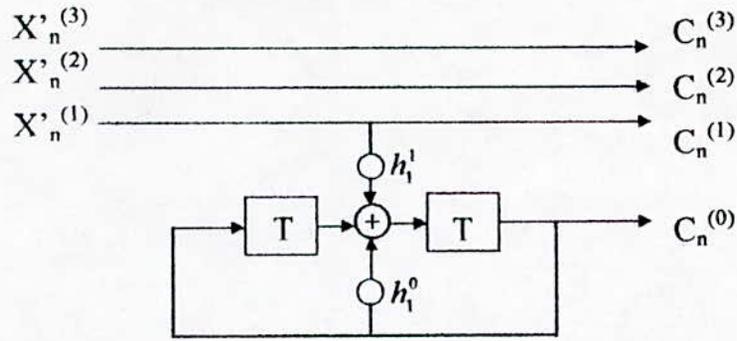


Figure 4.57 : Codeur systématique pour la 16 AM avec $v=2$

Codeur simulé :

Les polynômes que nous avons trouvés :

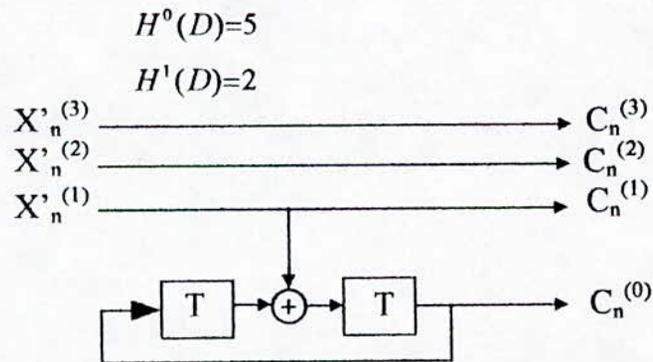


Figure 4.58 : Codeur de la simulation pour la 16 AM avec $v=2$

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

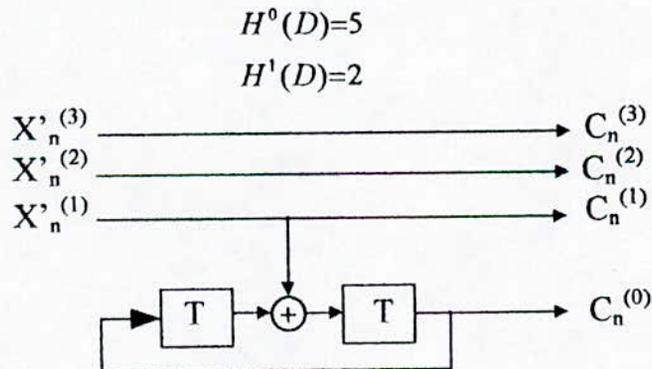


Figure 4.59 : Codeur d'Ungerboeck pour la 16 AM avec $v=2$

$v = 3$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

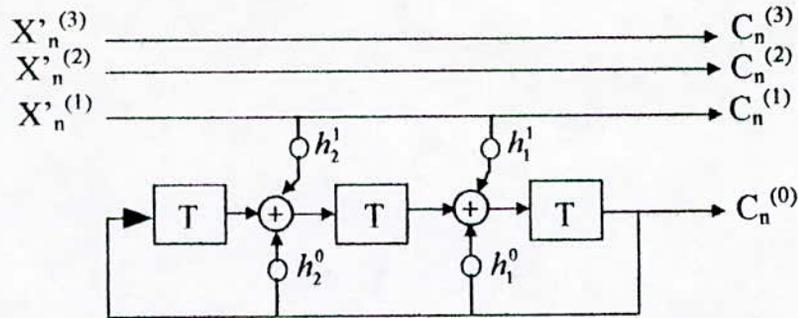


Figure 4.60 : Codeur systématique pour la 16 AM avec $v = 3$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D) = 11$$

$$H^1(D) = 02$$

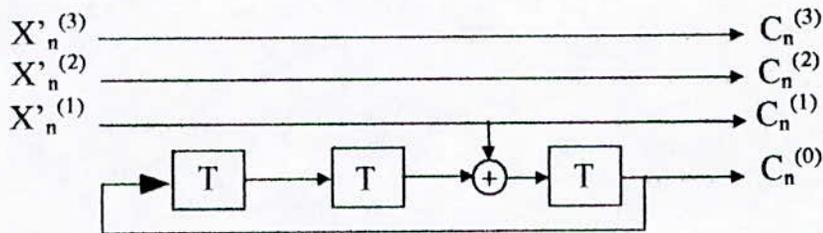


Figure 4.61 : Codeur de la simulation pour la 16 AM avec $v = 3$

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$$H^0(D) = 13$$

$$H^1(D) = 04$$

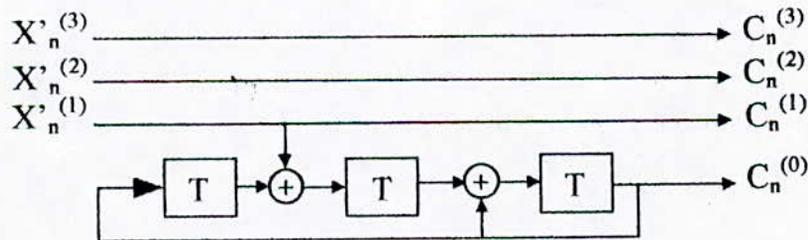


Figure 4.62 : Codeur d'Ungerboeck la simulation pour la 16 AM avec $v = 3$

$v = 4$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^4 + h_3^0 \cdot D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_3^1 \cdot D^3 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

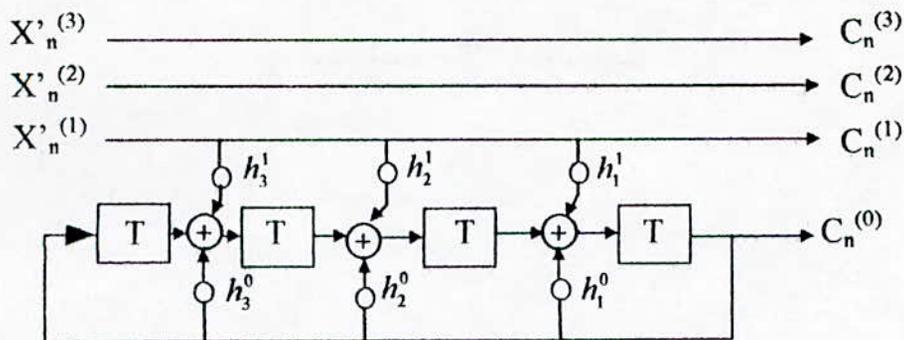


Figure 4.63 : Codeur systématique pour la 16 AM avec $v = 4$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D) = 21$$

$$H^1(D) = 04$$

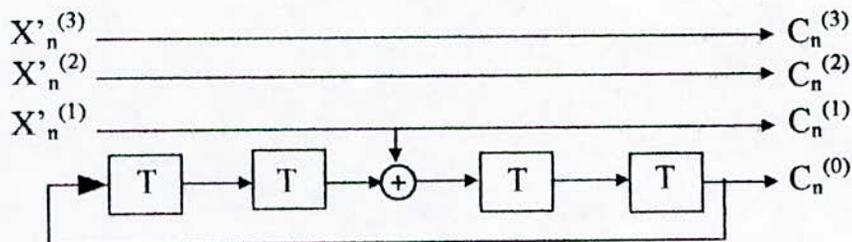


Figure 4.64 : Codeur de la simulation pour la 16 AM avec $v=4$

Codeur d'Ungerboeck :

Les polynômes donnés par Ungerboeck :

$$H^0(D) = 23$$

$$H^1(D) = 04$$

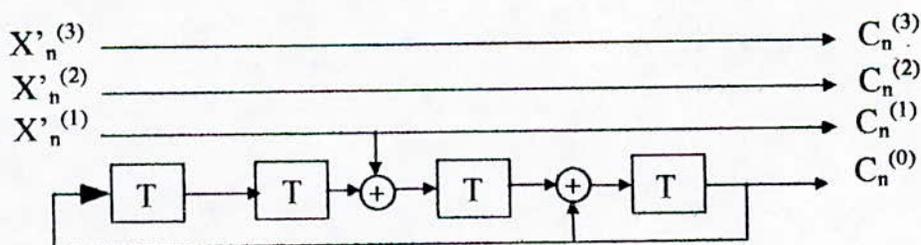


Figure 4.65 : Codeur d'Ungerboeck pour la 16 AM avec $v=4$

$v = 5$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^5 + h_4^0 \cdot D^4 + h_3^0 \cdot D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_4^1 \cdot D^4 + h_3^1 \cdot D^3 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

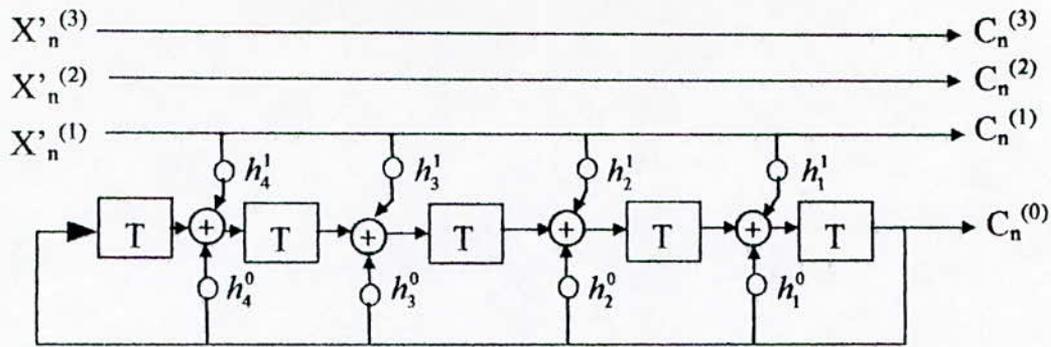


Figure 4.66 : Codeur systématique pour la 16 AM avec $v=5$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=41$$

$$H^1(D)=06$$

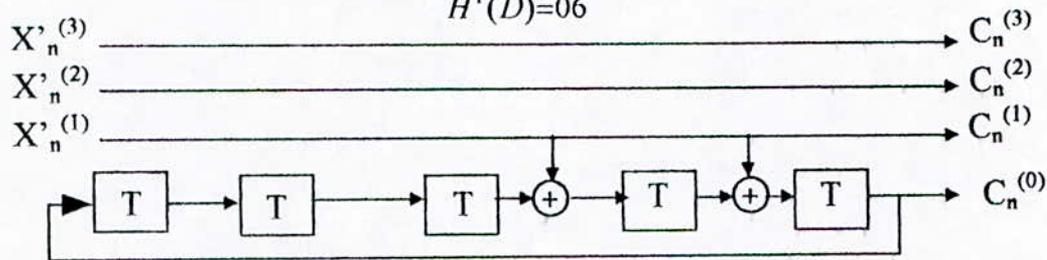


Figure 4.67 : Codeur de la simulation pour la 16 AM avec $v=5$

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$$H^0(D)=45$$

$$H^1(D)=10$$

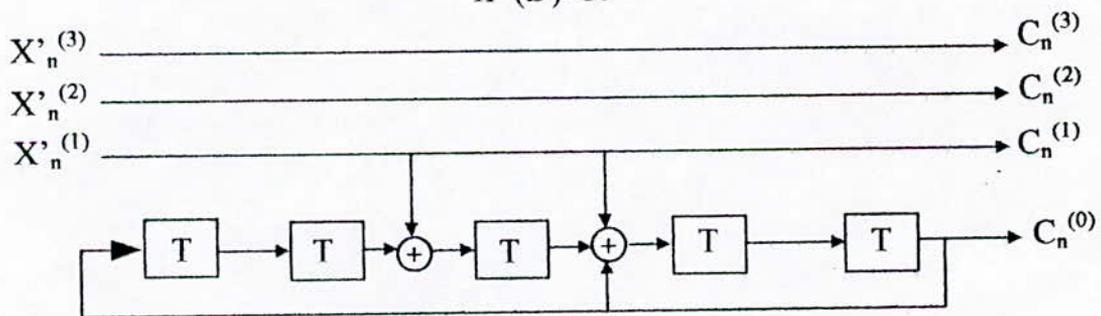


Figure 4.68 : Codeur d'Ungerboeck pour la 16 AM avec $v=5$

4-5-6 La modulation 16-QAM :

Données			Résultats de la simulation				Résultats d'Ungerboeck			
M	m'	v	$H^0(D)$	$H^1(D)$	$H^2(D)$	dfree	$H^0(D)$	$H^1(D)$	$H^2(D)$	dfree
3	1	2	5	2	---	1.263	5	2	---	1.264
3	2	3	11	02	02	1.477	11	02	04	1.413
3	2	4	21	02	04	1.61	23	04	16	1.548
3	2	5	41	02	06	1.67	41	06	10	1.548

Tableau 4.6

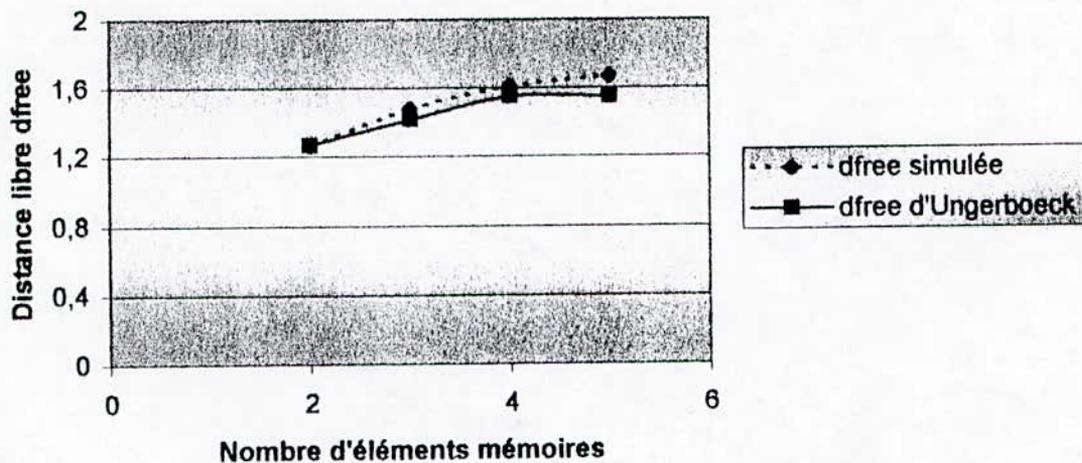


Figure 4.69 : Distances libres simulée et d'Ungerboeck en fonction du nombre d'éléments mémoires

$v = 2$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_1^1 \cdot D^1 + 0$$

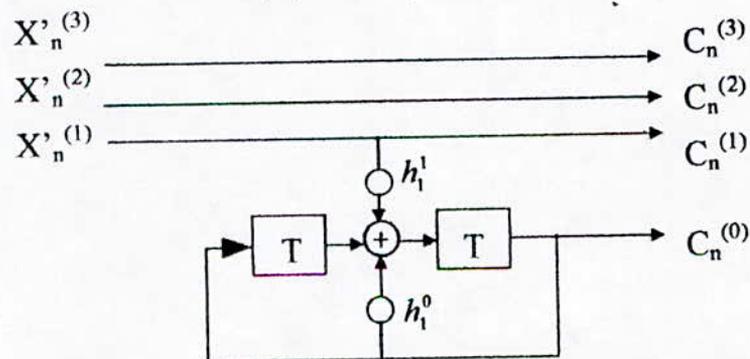


Figure 4.70 : Codeur systématique pour la 16 QAM avec $v=2$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D) = 5_s$$

$$H^1(D) = 2_s$$

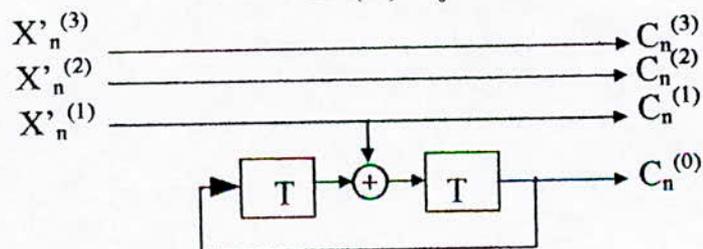


Figure 4.71 : Codeur de la simulation pour la 16 QAM avec $v=2$

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

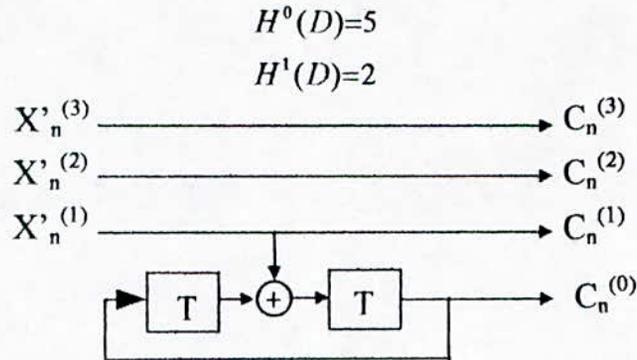


Figure 4.72 : Codeur d'Ungerboeck pour la 16 QAM avec $v=2$

$v = 3$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

$$H^2(D) = 0 + h_2^2 \cdot D^2 + h_1^2 \cdot D^1 + 0$$

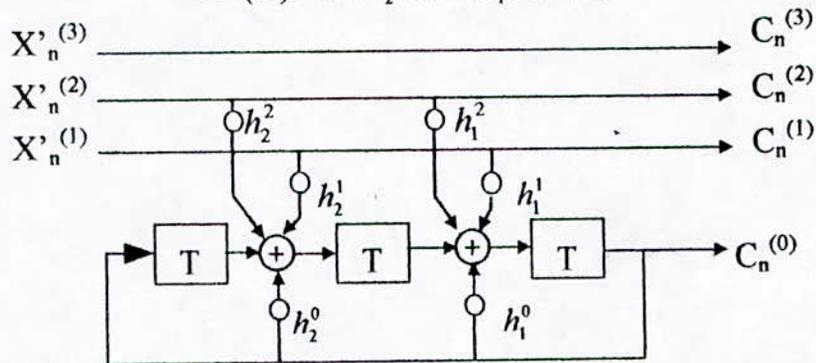


Figure 4.73 : Codeur systématique pour la 16 QAM avec $v = 3$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=11$$

$$H^1(D)=02$$

$$H^2(D)=02$$

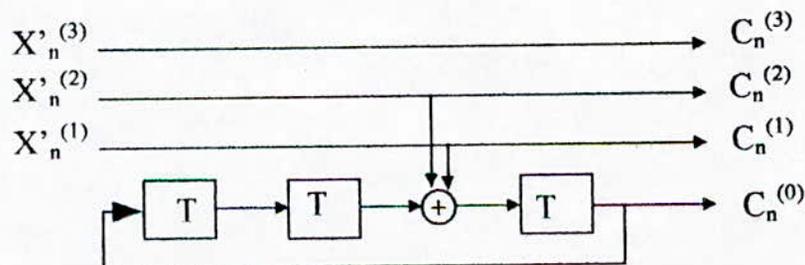


Figure 4.74 : Codeur de la simulation pour la 16 QAM avec $v=3$

Codeur d'Ungerboeck :

Les polynômes donnés par Ungerboeck :

$$H^0(D)=11$$

$$H^1(D)=02$$

$$H^2(D)=04$$

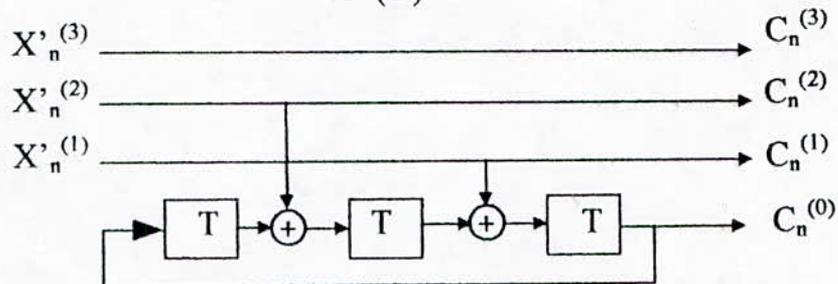


Figure 4.75 : Codeur d'Ungerboeck pour la 16 QAM avec $v=3$

$v = 4$: Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^4 + h_3^0 \cdot D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_3^1 \cdot D^3 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

$$H^2(D) = 0 + h_3^2 \cdot D^2 + h_2^2 \cdot D^1 + h_1^2 \cdot D^0 + 0$$

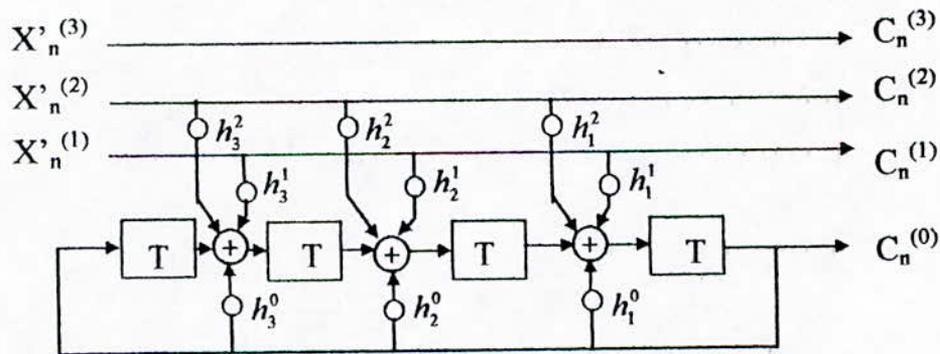


Figure 4.76 : Codeur systématique pour la 16 QAM avec $v=4$

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D)=21$$

$$H^1(D)=02$$

$$H^2(D)=04$$

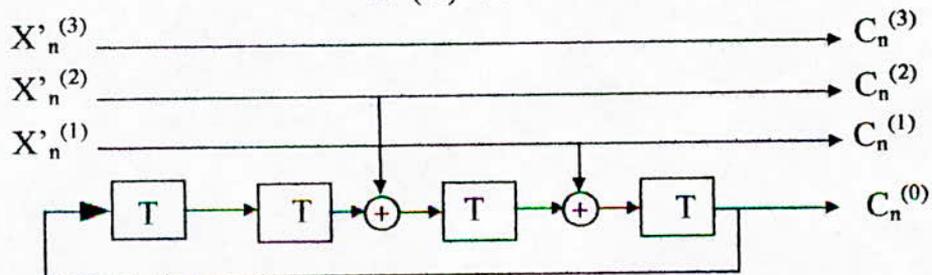


Figure 4.77 : Codeur de la simulation pour la 16 QAM avec $v=4$

Codeur d'Ungerboeck :

Les polynômes que Ungerboeck à trouver :

$$H^0(D) = 23$$

$$H^1(D) = 04$$

$$H^2(D) = 16$$

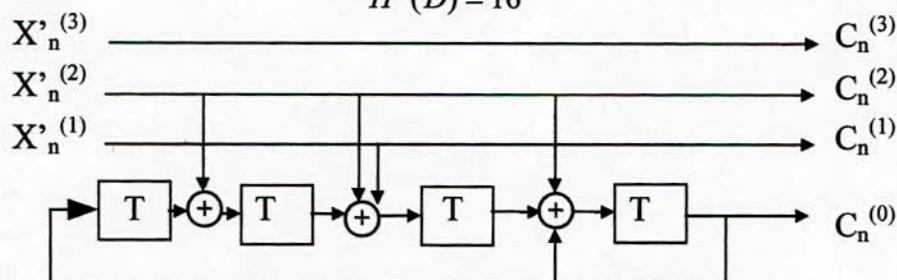


Figure 4.78 : Codeur d'Ungerboeck pour la 16 QAM avec v=4

v = 5 : Codeur systématique typique :

Les polynômes de la matrice de parité s'écrivent :

$$H^0(D) = D^5 + h_4^0 \cdot D^4 + h_3^0 \cdot D^3 + h_2^0 \cdot D^2 + h_1^0 \cdot D^1 + 1$$

$$H^1(D) = 0 + h_4^1 \cdot D^4 + h_3^1 \cdot D^3 + h_2^1 \cdot D^2 + h_1^1 \cdot D^1 + 0$$

$$H^2(D) = 0 + h_4^2 \cdot D^4 + h_3^2 \cdot D^3 + h_2^2 \cdot D^2 + h_1^2 \cdot D^1 + 0$$

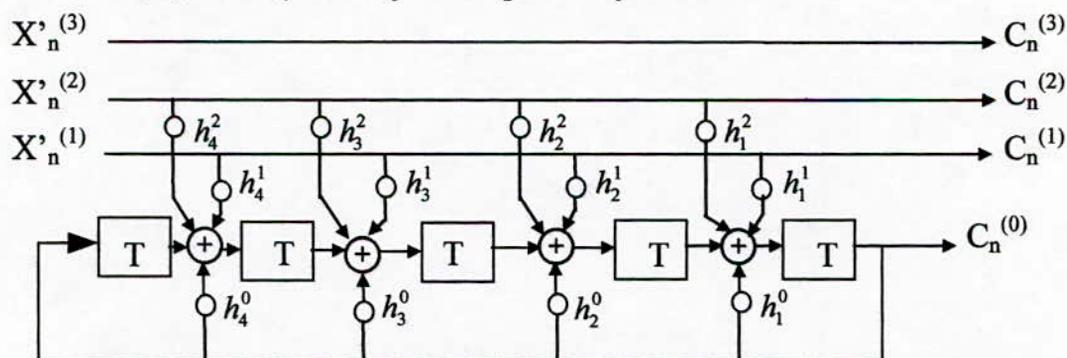


Figure 4.79 : Codeur systématique pour la 16 QAM avec v=5

Codeur simulé :

Les polynômes que nous avons trouvés :

$$H^0(D) = 41$$

$$H^1(D) = 02$$

$$H^2(D) = 06$$

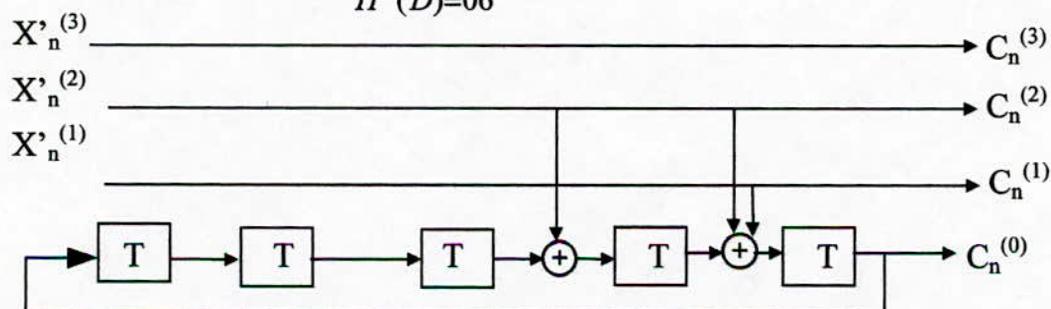


Figure 4.80 : Codeur de la simulation pour la 16 QAM avec v=5

Codeur d'Ungerboeck :

Les polynômes trouvés par Ungerboeck :

$$H^0(D)=41$$

$$H^1(D)=06$$

$$H^2(D)=10$$

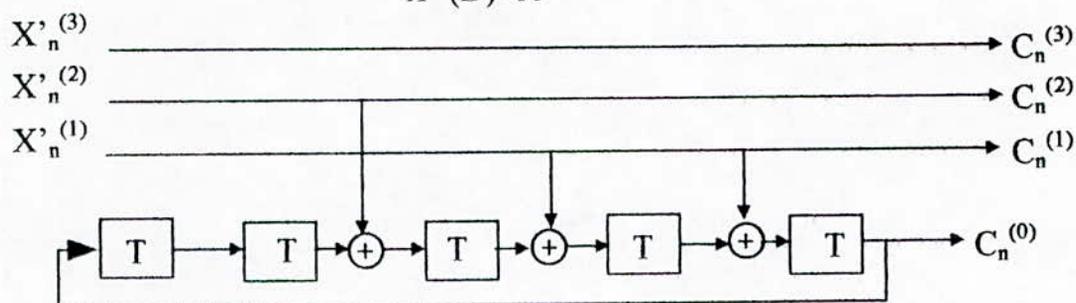


Figure 4.81 : Codeur d'Ungerboeck pour la 16 QAM avec $v=5$

4-6 Commentaires :

La conception des codeurs pour la TCM à 2D, nous a conduit aux calculs de la distance libre d_{free} et les paramètres de la matrice de parité. Les résultats de la simulation et ceux d'Ungerboeck ont été présentés dans les tableaux 4.1 jusqu'au 4.6 ainsi que les courbes d'évaluation de d_{free} en fonction du nombre d'éléments mémoires pour chaque modulation sont présentées par les figures 4.4, 4.17, 4.30, 4.43, 4.56 et 4.69. Les courbes hachurées représentent les résultats de la simulation et les courbes en trait plein représentent les résultats d'Ungerboeck.

Nous constatons de ces tableaux que les valeurs simulées de d_{free} sont très proches de celles d'Ungerboeck. Les courbes données par les figures 4.4, 4.17, 4.30, 4.43, 4.56 et 4.69 confirment cette convergence.

Nous constatons encore de ces tableaux que les paramètres de la matrice de parité calculés par simulation sont en conformité avec les paramètres d'Ungerboeck à l'exception du codeur à quatre états pour la modulation 4-AM qui est rejeté pour la simple raison qu'il ne vérifié pas la quatrième règle d'Ungerboeck.

Nous avons remarqué durant le calcul qu'en laissant le programme affiche tous les bons codes, le codeur choisit par Ungerboeck se trouve parmi ces derniers.

Le critère de choix de ces codeurs, par Ungerboeck n'est pas évoqué dans ces publications [2], [10].

Nous remarquons par conséquent que les paramètres choisis par ce dernier dans la publication [2] de 1982 sont légèrement différents de celle de 1987 [10]. Le tableau 4.7 montre cette différence pour la modulation 8-PSK à nombre d'éléments mémoires 7 et 8.

Année	v	$H^0(D)$	$H^1(D)$	$H^2(D)$
1982	7	405	250	176
		435	072	130
1987	8	203	014	016
		277	054	122

4-7 Conclusion :

La simulation a permis de concevoir des codeurs optimaux, pour les modulations 8-PSK, 16-PSK, 4-AM, 8-AM, 16-AM et 16-QAM, les plus utilisées dans la TCM à deux dimensions. Les résultats trouvés sont conformes avec la théorie (les règles empiriques) et sont proches des résultats d'Ungerboeck.

Dans le prochain chapitre, nous allons évalué les performances de la modulation Asymétrique codée en treillis.

CHAPTER

5

Chapitre 5

Evaluation des performances de la modulation Asymétrique codée en treillis

5-1 Introduction :

Dans ce chapitre de simulation nous traitons la modulation codée en treillis Asymétrique. Nous effectuons des mesures de probabilités d'événement d'erreur et probabilités par bit pour différentes configurations TCM Asymétriques. Les codeurs utilisés dans cette simulation de la TCM Asymétrique sont les codeurs obtenus par la simulation réalisée au chapitre 4.

Ces mesures seront comparées avec les résultats de la TCM Symétrique. [3]

5-2 Modèle de simulation :

Le modèle utilisé lors de notre simulation, illustré dans la figure 5.1, comporte :

- une source d'information binaire.
- un codeur en treillis.
- un canal gaussien supposé idéal (sans distorsions).
- une source de bruit.
- un décodeur à maximum de vraisemblance, utilisant l'algorithme de Viterbi.
- un bloc de calcul de la probabilité d'événement d'erreur moyenne $P_e(e)$.
- un comparateur pour calculer le nombre de bits erronés après le décodage.

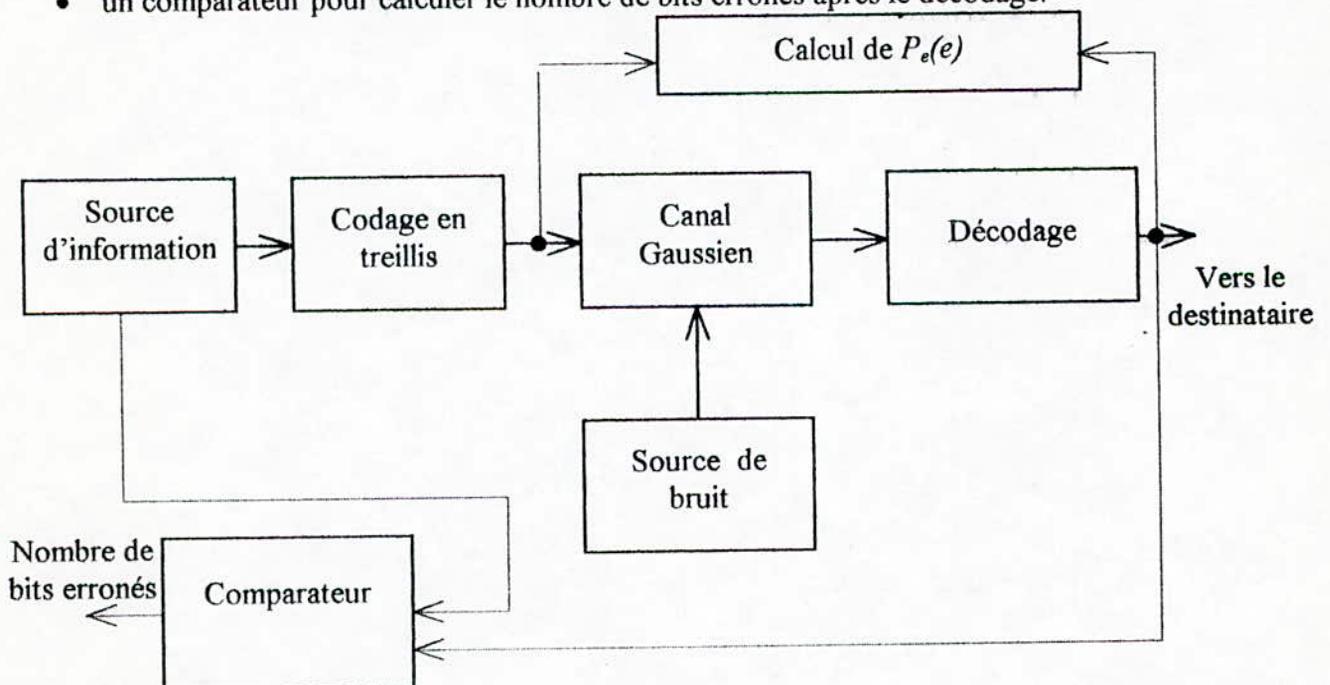


Figure 5.1 : Modèle de simulation

5-2-1 Source d'information :

La génération de la séquence pseudo-aléatoire est faite selon la procédure décrite dans le paragraphe 4-2-1.

5-2-2 Codeur en treillis :

Dans notre simulation nous évitons les codeurs catastrophiques, ces codeurs ne sont pas performants puisqu'il causent des séquences d'erreurs longues après décodage avec une probabilité non négligeable .

5-2-3 Source de bruit :

Le bruit envisagé est (AWGN)

- blanc.
- additif.
- gaussien.

On n'a pas pris en compte de l'effet du filtre de réception, c'est-à-dire que le bruit reste AWGN à la réception (car le filtrage rend ce bruit coloré).

5-2-4 Décodage :

Le décodage est fait selon la méthode de maximum de vraisemblance utilisant l'algorithme de Viterbi.

5-3 Modulations utilisées :

Dans notre simulation, notre choix reste limité aux modulations à deux dimensions, nous avons utilisé :

- La modulation 4-PSK Asymétrique combinée avec le codeur en treillis à 4 états représenté dans la figure 5.2 a.
- La modulation 8-PSK Asymétrique combinée avec le codeur en treillis à 16 états illustré dans la figure 5.2 b.
- La modulation 16-PSK Asymétrique combinée avec le codeur en treillis à 8 états représenté dans la figure 5.2 c.

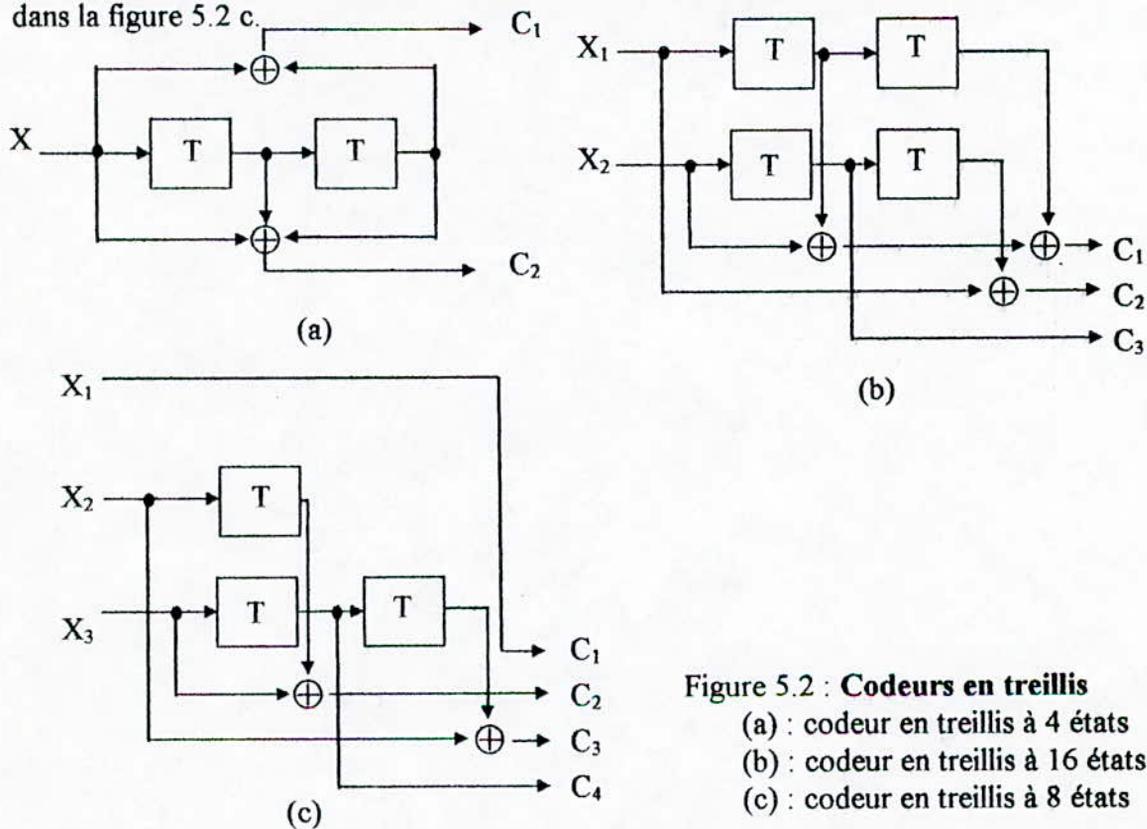


Figure 5.2 : Codeurs en treillis
 (a) : codeur en treillis à 4 états
 (b) : codeur en treillis à 16 états
 (c) : codeur en treillis à 8 états

5-4 Transcodage :

Les bits de sorties du codeur en treillis sont assignés aux symboles emis.

Exemple 5.1 :

Soit le codeur en treillis de la figure 5.2 a combiné avec la 4-PSK Asymétrique, le tableau 5.1 de correspondance entre les bits de sortis et les symboles transmis est :

C ₁	C ₂	S
0	0	1
0	1	Exp(j.φ)
1	0	-1
1	1	-exp(j.φ)

Tableau 5.1

Où φ est le déphasage effectué lors de la construction de la constellation pour la modulation Asymétrique.

5-5 Programmation :

Les programmes sont donnés dans l'annexe B.

5-5-1 Langage utilisé :

Dans le MATLAB (Version sous Windows), on peut générer un bruit blanc gaussien

- de variance égale à l'unité dans chaque dimension (car on a deux dimensions)
- de moyenne nulle.

Pour avoir un bruit de variance σ^2 dans chaque dimension , on multiplie les échantillons générés du bruit par σ .

Le Matlab contient des fonctions prêtes à utiliser, telles que la génération du bruit, la fonction d'erreur, fonction d'autocorrection, etc. Ces fonctions permettent une réduction des programme, d'où l'utilisation du Matlab dans cette simulation.

5-5-2 Calcul de la probabilité $P_e(e)$ et le nombre des bits erronés :

La séquence d'information pseudo-aléatoire générée par la source est de longueur $2^{16} - 1 = 65535$ bits, mais dans notre simulation elle est limitée à 60.000 bits.

On calcule la séquence correcte correspondante à l'information pseudo-aléatoire, et son trajectoire dans le treillis, puis on calcule la séquence incorrecte qui est choisie par le décodeur à maximum de vraisemblance ainsi que sa trajectoire. La trajectoire d'une séquence est complètement définie par les états décrits par la séquence minimum elle même.

En tirant à partir de ces deux séquence et de leurs trajectoires, les événements d'erreur, puis on calcule les probabilités d'erreur correspondantes à ces événements.

La probabilité d'événement d'erreur est donnée par : **[3]**

$$P_e(e) = \frac{1}{fr / N(d_{free})} \sum_{i=1}^{nev} P_i \quad \dots(5.1)$$

où

$$* p_j = \frac{1}{2} \cdot \text{erfc} \left(\frac{d_j}{\sqrt{8} \cdot \sigma} \right) \quad \dots(5.2)$$

- * d_i : la distance de l'événement d'erreur dans le treillis.
- * nev est le nombre des événements d'erreur sur toute la longueur du treillis.
- * fr est la fréquence des événements ayant une distance euclidienne égale à d_{free} . Si on suppose que ces événements sont équiprobables, alors chaque événement parmi ces dernières a une fréquence égale à $fr / N(d_{free})$.
- * P_i est la i ème probabilité d'événement d'erreur, qui est calculée à l'aide de la relation (5.2), $N(d_{free})$ peut être calculer à la main si le codeur est simple, et avec l'ordinateur dans le cas contraire. Si tous les événements sont équiprobables, donc ils ont la même fréquence $fr/N(d_{free})$, la formule (5.1) devient dans ce cas un résultat direct du principe d'union.

Lorsque la longueur du treillis (ou la longueur de la séquence d'information) tend vers l'infini, $P_e(e)$ tend vers sa valeur exacte.

Si on peut calculer le nombre des bits erronés b_i dans l'événement d'erreur numéro i ($i=1, \dots, nev$) la probabilité d'erreur par bit est donnée par :

$$P_b(e) = \frac{1}{m \times fr / N(d_{free})} \sum_{i=1}^{nev} b_i P_i \quad \dots (5.3)$$

où m est le nombre des entrées du codeur en treillis.

Généralement le calcul de $P_b(e)$ est difficile, donc il faut chercher un autre paramètre facile à calculer et qui peut remplacer $P_b(e)$ même d'une manière asymptotique .

La profondeur de décision dp (depth) est pratiquement choisie supérieure ou égale à 6 fois le nombre des éléments mémoires du codeur. Dans notre simulation nous avons pris $dp=2000$, pour que les résultats soient les meilleurs possibles.

A partir de la trajectoire de la séquence incorrecte, on peut compter les bits erronés, donc le pourcentage de ces bits, qui est le nombre de bits erronés sur le nombre total des bits. Ce pourcentage n'est autre que la probabilité des bits erronés, conditionnée par l'apparition de séquence incorrecte, donc on peut calculer la probabilité Pb :

$$Pb = P_e(e) \times \text{le pourcentage de bits erronés} \quad \dots (5.4)$$

qui est donc la probabilité des bits erronés sans conditionnement. Nous allons utiliser ce paramètre pour la comparaison entre les codeurs choisis.

A l'aide des deux relations (5.3) et (5.4) et avec quelques hypothèses statistiques, on peut montrer que pour un rapport signal sur bruit (RSB) grand, on a :

$$Pb \approx \frac{m}{L} \times P_e(e) \quad \dots (5.5)$$

où L est la longueur de séquence d'information.

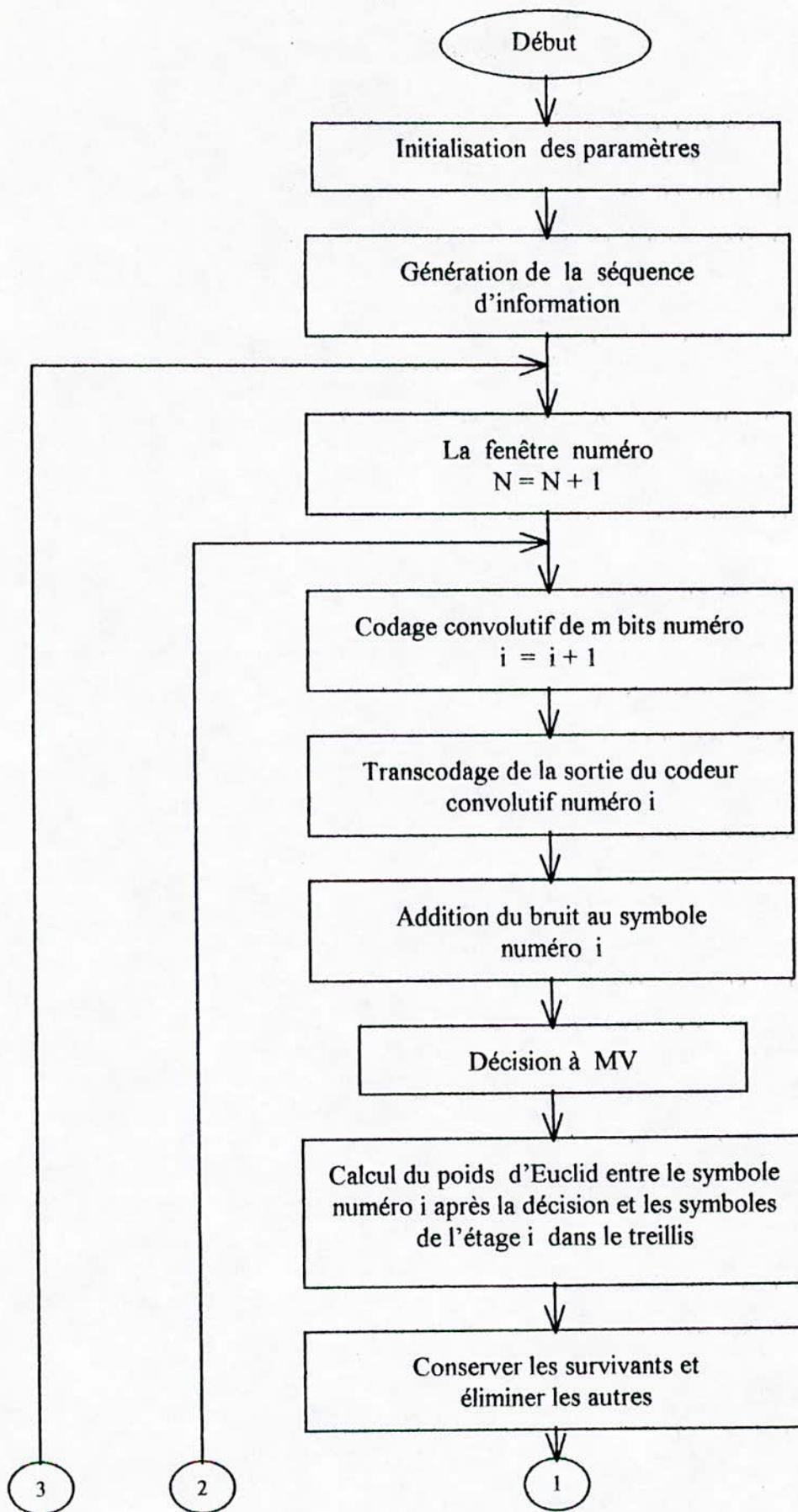
5-5-3 L'organigramme :

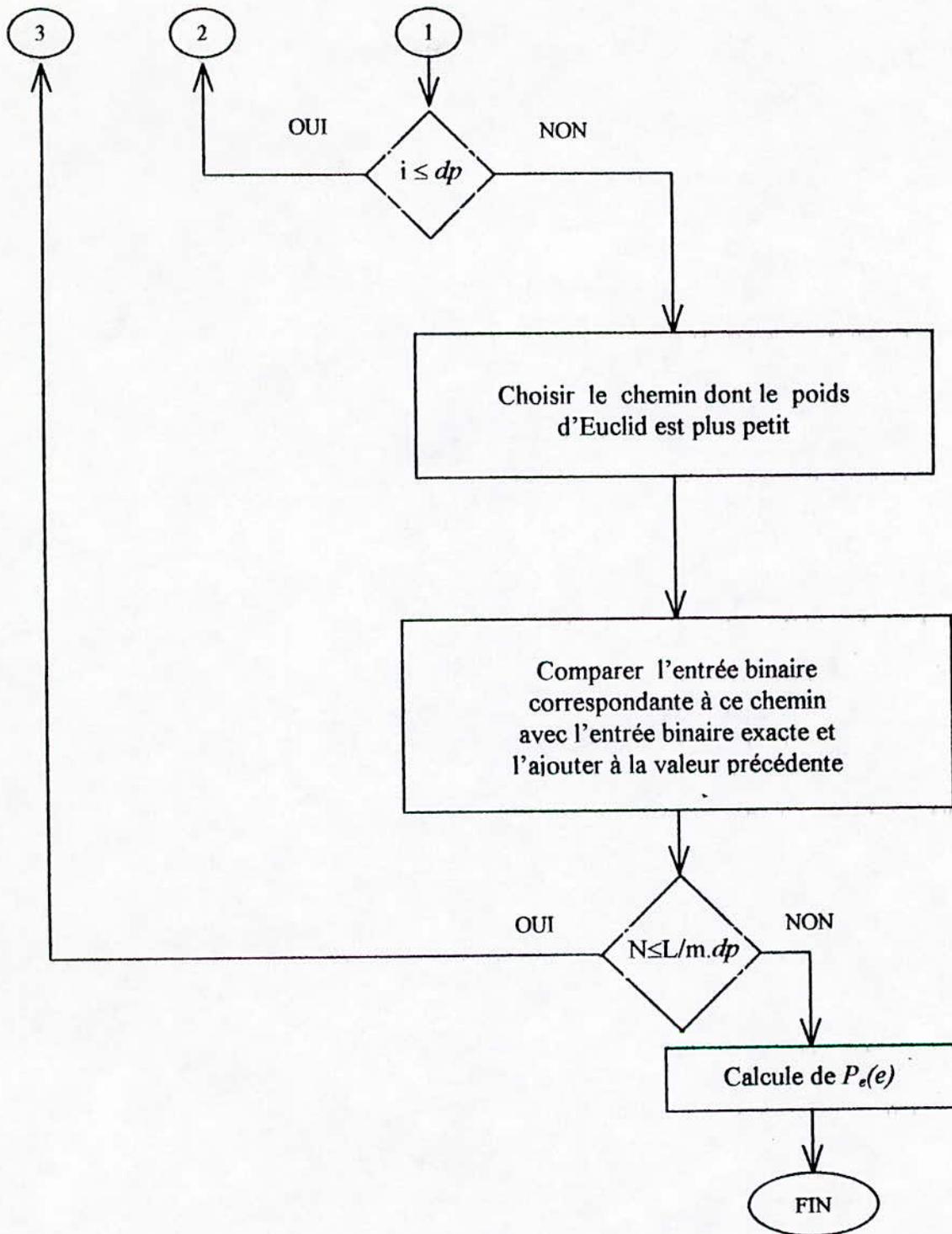
L'organigramme de la simulation est donné par la figure 5.3.

5-6 Résultats et commentaires :

Nous avons réalisé à l'aide du MATLAB des programmes de simulation pour les modulations M-PSK; $M=4, 8, 16$ Asymétriques combinées respectivement avec les codeurs illustrés dans la figure 5.2.

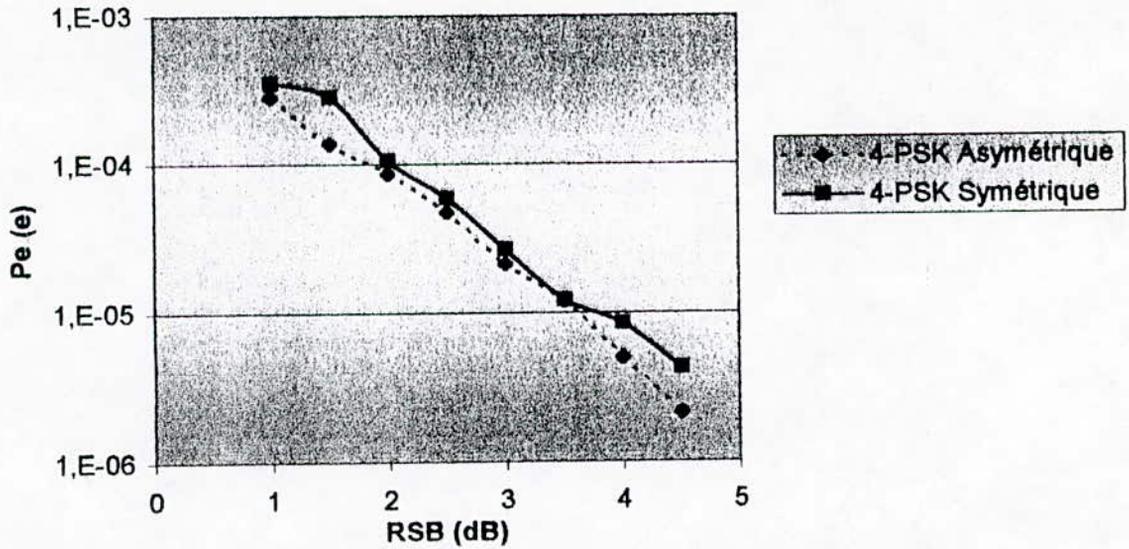
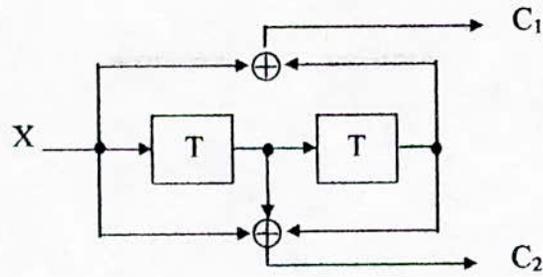
Dans la représentation des résultats, nous donnerons le schéma du codeur/modulateur, la courbe d'évaluation de la probabilité d'événement d'erreur et la courbe d'évaluation de la probabilité d'erreur par bit en fonction du RSB pour les configurations TCM Symétrique et Asymétrique.



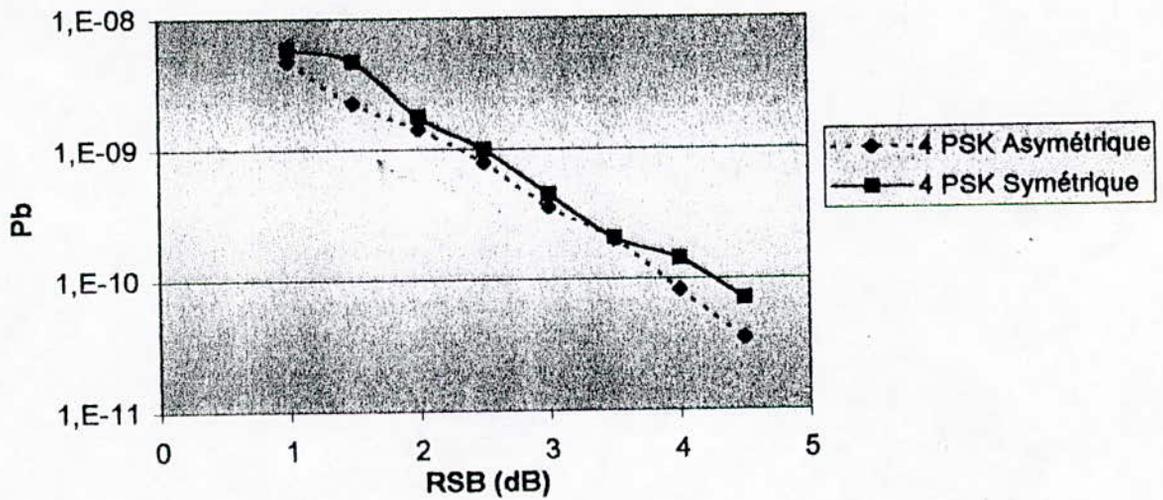


L : la longueur de l'information
 m : le nombre des entrées du codeur en treillis
 dp : le depth

Figure 5.3 L'organigramme

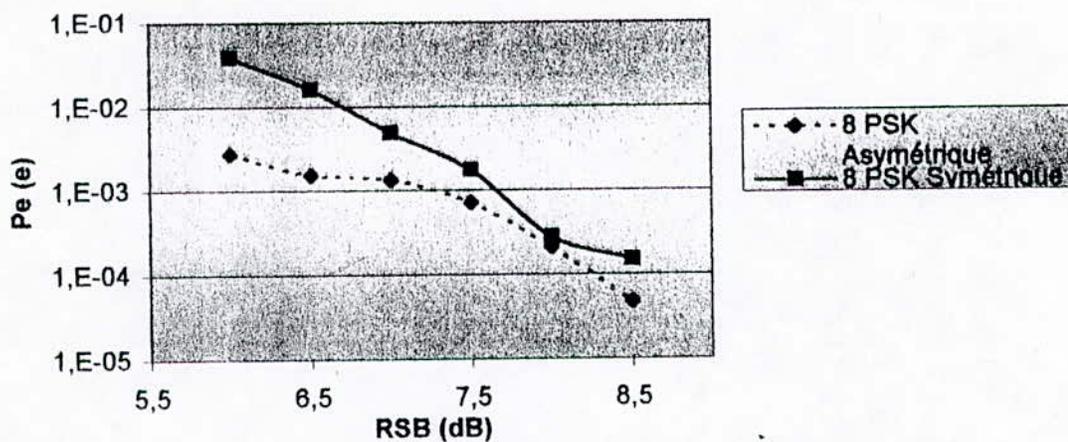
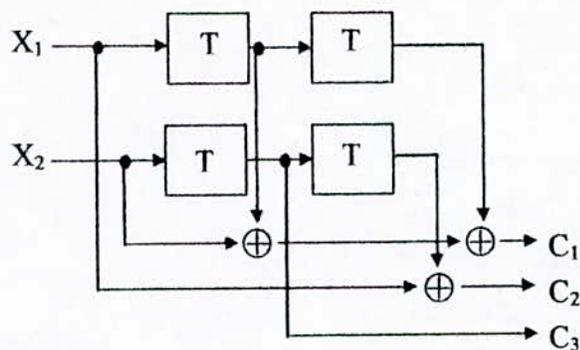


(a) : Probabilités d'événements d'erreur de la 4 PSK Asymétrique et Symétrique combinées avec le codeur en treillis à 4 états.

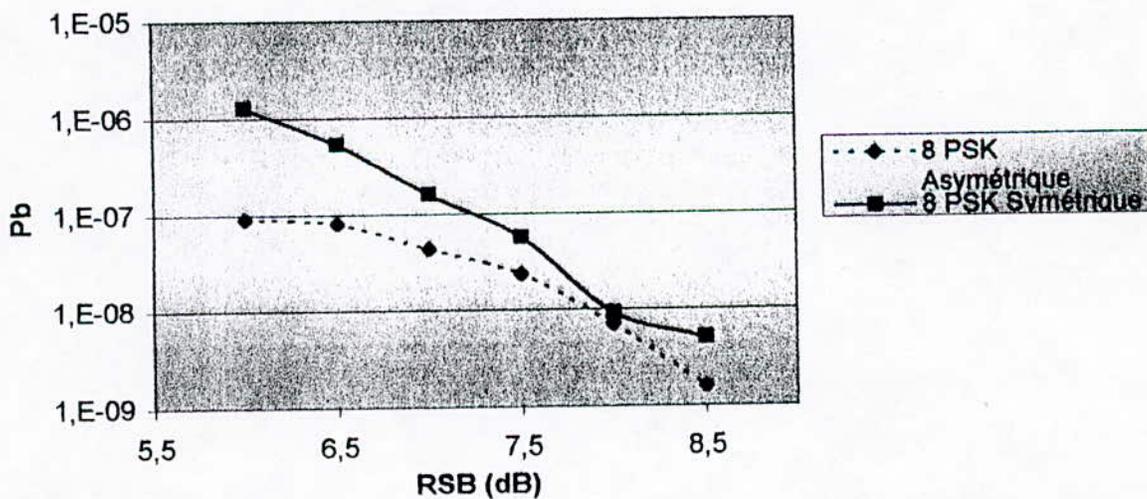


(b) : Probabilités d'erreur par bit pour la modulation 4 PSK Asymétrique et Symétrique combinées avec le codeur en treillis à 4 états.

Figure 5.4

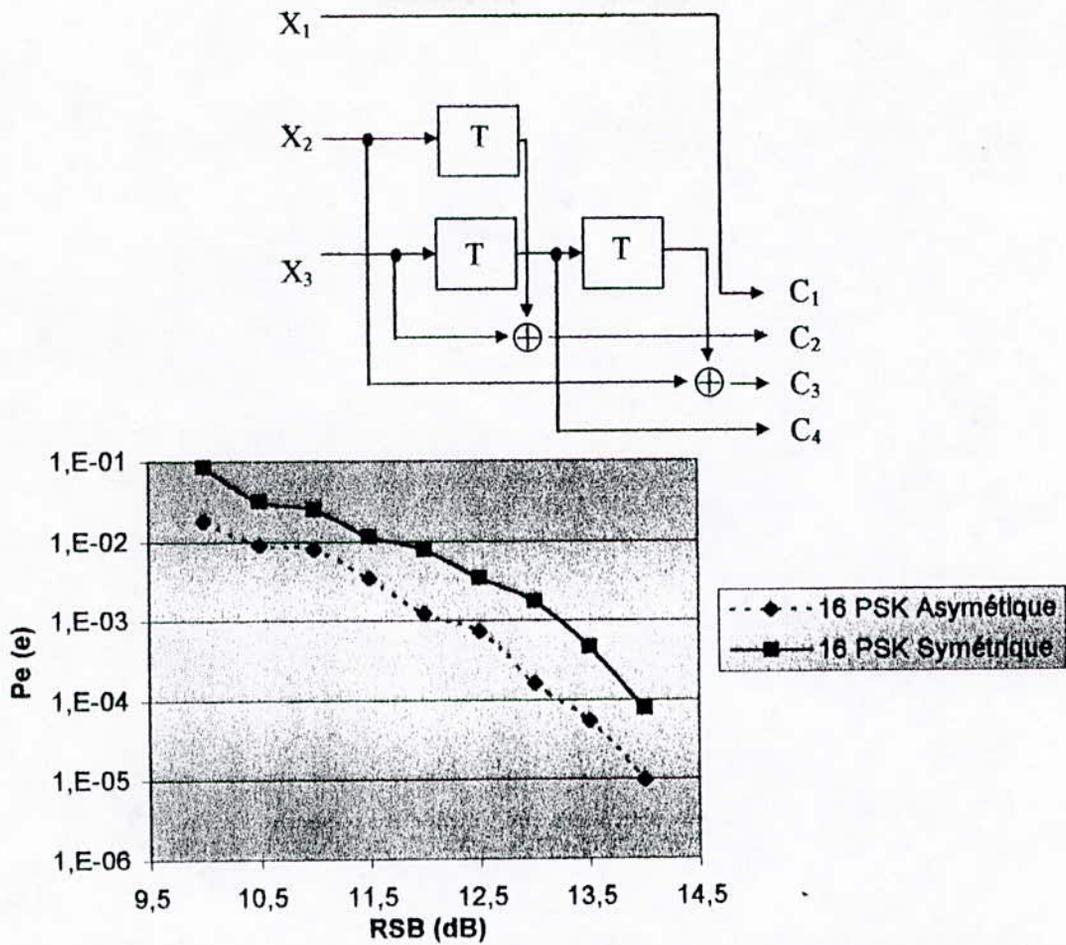


(a) : Probabilités d'événements d'erreur de la 8 PSK Asymétrique et Symétrique combinées avec le codeur en treillis à 16 états.

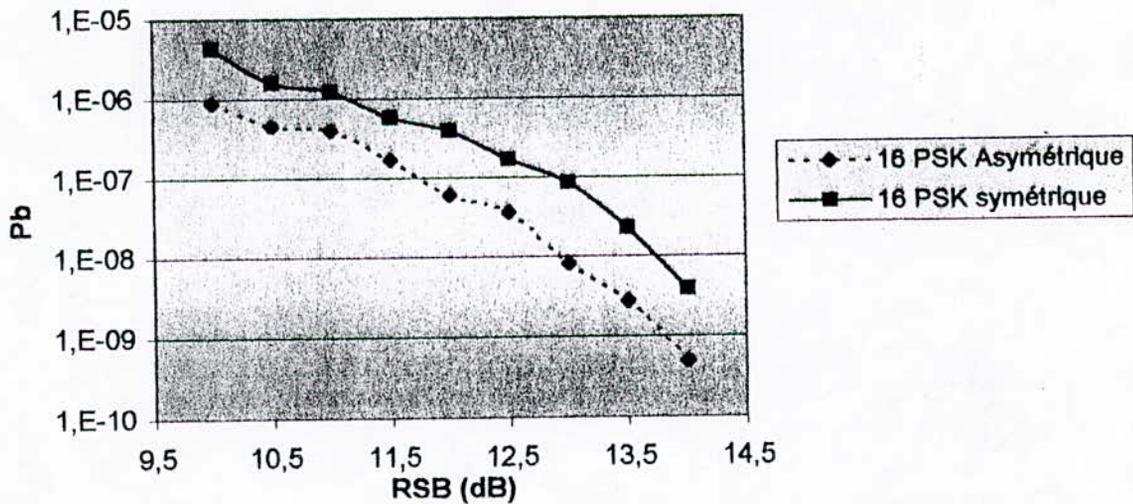


(b) : Probabilités d'erreur par bit pour la modulation 8 PSK Asymétrique et Symétrique combinées avec le codeur en treillis à 16 états.

Figure 5.5



(a) : Probabilités d'événements d'erreur de la 16 PSK Asymétrique et Symétrique combinées avec le codeur en treillis à 8 états.



(b) : Probabilités d'erreur par bit pour la modulation 16 PSK Asymétrique et Symétrique combinées avec le codeur en treillis à 8 états

Figure 5.6

5-6-1 Probabilité d'événement d'erreur $P_e(e)$:

Pour évaluer la performance d'un système TCM, on est amené à mesurer sa probabilité d'événement d'erreur .

Les courbes obtenues par simulation pour les systèmes TCM à modulation Asymétrique sont représentées en hachuré tandis que les courbes en trait plein représentent les systèmes TCM à modulation Symétrique.

L'analyse des figures 5.4 a jusqu'à 5.6 a permet de faire les constatations suivantes :

- La probabilité d'erreur $P_e(e)$ est inversement proportionnelle au rapport signal sur bruit RSB ce qui est conforme avec la théorie.
- Les courbes hachurées sont situées au dessous des courbes en trait plein, ceci confirme bien qu'un système avec codage en treillis combiné avec la modulation Asymétrique est plus performant que celui combiné avec modulation Symétrique.

5-6-2 Probabilité des bits erronés :

Ce paramètre est une indication de la qualité d'un système de transmission.

L'analyse des figures 5.4 b jusqu'à 5.6 b permet de faire les constatations suivantes :

- les courbes hachurées et à trait plein sont des fonctions décroissantes en fonction du rapport signal sur bruit.
- Les courbes hachurées sont situées au dessous des courbes en trait plein, ceci confirme bien qu'un système avec codage en treillis combiné avec la modulation Asymétrique est plus performant que celui combiné avec modulation Symétrique.

5-7 Conclusion :

Notre simulation a permis de calculer les performances des systèmes TCM à modulation Asymétrique à deux dimensions. Les résultats obtenus sont plus performants des résultats obtenus pour les configurations TCM Symétriques.

Notre simulation s'est limitée aux 4-PSK, 8-PSK, 16-PSK Asymétriques, nous avons évité les configurations TCM combinées avec des codeurs catastrophiques.

Pour la transmission des informations avec des grands débits, et avec une performance acceptable, les codeurs convolutifs à grands nombre d'états sont les plus performants.

CONCILIUM
GENERALE

Conclusion générale

Le travail présenté dans ce mémoire concerne la modulation codée en treillis (TCM) a englobé deux parties : une théorique et l'autre une simulation .

L'étude de la partie théorique a pour but de donner les fondements de base de la modulation codée en treillis à deux dimensions, pour cela nous avons étudié les deux réalisations de la TCM à savoir les codeurs d'Ungerboeck et Calderbank-Mazo, les règles de transcodage(mapping) par la partition d'ensemble, le décodage par l'utilisation de l'algorithme de Viterbi .

Une partie importante était réservée à l'étude de l'algorithme de recherche des codeurs optimaux pour les modulations à une et à deux dimensions proposé par Ungerboeck. Afin de faire une évaluation de la modulation asymétrique codée en treillis par rapport à la TCM symétrique, une analyse de cette théorie est faite et des résultats théoriques en termes de gain de codage ont été donné.

La partie simulation a été divisée en deux réalisations :

- Implémentation de l'algorithme de recherche des codeurs optimaux.
- Evaluation des performances de la TCM asymétrique.

Pour la mise en points de ces concepts, nous avons réalisé des programmes en F32 pour implémenter l'algorithme de recherche des codeurs optimaux. Cette simulation nous a permis de concevoir 24 codeurs optimaux combinés avec les modulations 4-AM, 8-AM, 16-AM, 8-PSK, 16-PSK et 16-QAM à deux dimensions, en calculant la distance libre d_{free} et les paramètres de la matrices de parité pour chaque configuration.

La deuxième partie de la simulation était la mise en oeuvre d'une simulation permettant d'évaluer les performances des modulations Asymétriques codées en treillis et cela en estimant la probabilité d'erreur par bit P_b et la probabilité d'événement d'erreur $P_e(e)$ de ces configurations en fonction du rapport signal sur bruit RSB. Pour ce faire, nous avons utilisé les modulations Asymétriques 4-PSK combinée avec le codeur d'Ungerboeck à 4 états, 8-PSK en conjonction avec le codeur à 16 états, 16-PSK combinée avec le codeur à 16 états.

Cette simulation nous a permis de calculer les paramètres d'évaluation des systèmes TCM basés sur les modulations Asymétrique tels que la probabilité d'événement d'erreur $P_e(e)$ et la probabilité d'erreur par bit P_b . Les courbes obtenues de la simulation ont été comparées par rapport aux courbes des configurations TCM symétriques[3], nous constatons que les courbes d'évaluation de l'ATCM se localisent au dessous des courbes d'évaluations de la TCM ceci que la modulation asymétrique codée en treillis est meilleure que la modulation symétrique codée en treillis.

Les TCM à deux dimensions utilisent une constellation de 2^{m+1} pour transmettre m bits d'information par symbole et cette expansion d'alphabet coûte 3 dB.

L'introduction de la modulation Asymétrique est une procédure qui augmente la distance libre et donc améliore la TCM mais toutefois dans certains cas cette amélioration asymptotique dans la réduction du rapport signal sur bruit peut seulement être réalisée dans la limite où les points de signaux de la constellation émergent entre eux et le codeur en treillis devient catastrophique. Comme exemple le codeur à deux états qui pour toute valeur de m résulte asymptotiquement à un gain de 3 dB sur la même bande passante du système non codé, mais en réalité cette performance ne peut jamais être réalisée. Un autre problème c'est qu'en

ayant les points de signaux trop serré dans la constellation le système devient moins robuste en présence d'erreur de phase de synchronisation.

Pour remédier à ces problèmes pratiques, il faut changer la constellation afin d'augmenter le gain de codage des configurations TCM et ceci en utilisant les modulations codées en treillis multidimensionnelles. Ces dernières ont pour but de réduire le facteur d'expansion d'alphabet.

THE GEOGRAPHIC

BIBLIOGRAPHIE

- [1] E. Biglieri, D. Divsalar, P.J. McLane and M.K. Simon, *Introduction to trellis-coded modulation with applications*, Macmillan publishing company, New York, 1994.
 - [2] G. Ungerboeck, " Channel Coding with Multilevel / Phase Signals," *IEEE Trans. Inf. Theory*, Vol. IT-28, pp.56-67, Jan. 1982.
 - [3] A. Chemsas, K. Ghrissi, *Etude et Evaluation des systemes de communication numeriques utilisant la modulation par codes en treillis TCM*, ENP, oct. 1997.
 - [4] S. Haykin, *Digital Communication*, John Wiley & Sons, New York, 1988.
 - [5] H. Sari, " Transmission des Signaux Numériques," *Techniques de l'Ingénieur, Traité Electronique*, E 7100, 1996.
 - [6] E. Zehavi and J. K. Wolf, " On the Performance Evaluation of Trellis Codes," *IEEE Trans. Inf. Theory*, Vol. IT-33, pp. 196-202, Mar. 1987.
 - [7] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, Kluwer Academic Publishers, New York, USA, 1994.
 - [8] A.R. Calderbank and J. Mazo, " A New Description of Trellis Codes," *IEEE Trans. Inf. Theory*, Vol. IT-30, pp. 784-791, Nov. 1984.
 - [9] G. D. Forney, JR. , " Convolutional Codes I : Algebraic Structure," *IEEE Trans. Inf. Theory*, Vol. IT- 6, pp. 720-738, Nov. 1970.
 - [10] G. Ungerboeck, " Trellis-Coded Modulation with Redundant Signal Sets. Part I : Introduction ; Part II: State of the Art," *IEEE Commun. Mag.*, Vol. 25, N° 2, pp.5-21, Feb. 1987.
 - [11] G. D. Forney, JR. , " The Viterbi Algorithm," *IEEE Proc.*, Vol. 61, pp. 268-278, Mar. 1973.
 - [12] S. Benedetto, M. A. Marsan, Albertengo and E. Giachin, " Combined Coding and Modulation : Theory and Applications," *IEEE Trans. Inf. Theory*, Vol. IT-34, pp. 223-236, Mar. 1988.
 - [13] G. D. Forney, JR. R.G. Gallager, G. R. Lang, F. M. Longstaff and S.U. Qureshi, " Efficient Modulation for Band-Limited Channels," *IEEE J. Select. Areas Commun.*, Vol. SAC-2, N° 5 pp. 632-647, Sept. 1984.
-