

15/96
الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التربية الوطنية
MINISTERE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

البنية الوطنية المتعددة التخصصات
المكتبة - BIBLIOTHEQUE
Ecole Nationale Polytechnique

DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

SUJET

**CODAGE ET DECODAGE CONVOLUTIFS
ETUDE, SIMULATION ET EVALUATION.**

Proposé par :

**B. DERRAS
Z. TERRA**

Etudé par :

**F. FLITTI
Y. OMAROUYOUN**

Dirigé par :

**B. DERRAS
Z. TERRA**

PROMOTION

SEPTEMBRE 96

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التربية الوطنية

MINISTERE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

SUJET

**CODAGE ET DECODAGE CONVOLUTIF
ETUDE, SIMULATION ET EVALUATION.**

Proposé par :

**B. DERRAS
Z. TERRA**

Etudié par :

**F. FLITTI
Y. OMAROUYOB**

Dirigé par :

**B. DERRAS
Z. TERRA**

PROMOTION

SEPTEMBRE 96



DEDICACE

Je dédie ce modeste travail à :

Mes parents.

Ma grande famille .

Et à tous mes amis.

Yassine.

Je dédie ce modeste travail à :

Mes parents.

Toute ma famille .

Tous mes amis, et tous ceux qui le liront.

Farid.

REMERCIEMENTS

Remercions Dieu .

Nous tenons à remercier nos deux promoteurs Mrs. B.Derras et Z.Terra pour leurs assistance durant tout le travail .

Nous remercions nos deux familles pour leurs soutiens.

Enfin nous tenons à remercier Mrs. M.Braikia et O.Yala de l'E.C.O. de Ténès pour l'aide qu'ils nous ont fourni.

SOMMAIRE

INTRODUCTION	1
CHAPITRE -1 : CODAGE DE CANAL	
1-1 Introduction	3
1-2 Modèle d'un système de communication numérique	3
1-3 Modèle de canal	4
1-4 Théorème fondamental du codage de canal	6
1-4-1 Information mutuelle	6
1-4-2 Capacité de canal	6
1-4-3 Théorème	6
1-5 Types d'erreurs	7
1-6 Types de codes	7
1-7 Codes par blocs linéaires	7
1-7-1 Principe	7
1-7-2 Pouvoir de détection et de correction des erreurs	9
1-7-3 Décodage par syndrome	10
1-7-4 Codeurs	11
1-7-5 Décodeurs	12
1-8 Codes cycliques	13
1-8-1 Principe	13
1-8-2 Polynôme générateur	13
1-8-3 Polynôme de parité	14
1-8-4 Codeurs	14
1-8-5 Décodeurs	16
1-9 Présentation de quelques codes cycliques	16
1-10 Conclusion	17
CHAPITRE -2 : CODES CONVOLUTIFS	
2-1 Principe	18
2-2 Représentation matricielle	19
2-3 L'approche dans le domaine temporel, et l'approche dans le domaine des transformées	21
2-3-1 Approche dans le domaine temporel	21
2-3-2 Approche dans le domaine des transformées	23
2-4 Diagramme en arbre, diagramme en treillis et diagramme d'état	24
2-4-1 Diagramme en arbre	24
2-4-2 Diagramme en treillis	25
2-4-3 Diagramme d'état	27
2-5 Fonction génératrice et propriétés de distance des codes convolutifs	28

CHAPITRE -3 :DECODAGE DES CODES CONVOLUTIFS

3-1 Introduction	31
3-2 Décodage à vraisemblance maximale	31
3-3 Algorithme de Viterbi	32
3-4 Décodage séquentiel	34
3-4-1 La métrique de Fano	34
3-4-2 Algorithme de Fano	35
3-5 Performance des codes convolutifs	38
3-6 Conclusion	43

CHAPITRE -4 :SIMULATION, RESULTATS ET INTERPRETATIONS

4-1 Introduction	44
4-2 Modèle d'étude	44
4-2-1 Source d'information	44
4-2-2 Source de bruit	45
4-2-3 Calcul rapport signal sur bruit.	46
4-3 Résultats et commentaires	46
4-4 Conclusion	51

CONCLUSION 53

BIBLIOGRAPHIE 54

INTRODUCTION

La transmission numérique a pour objet la communication d'un point (émetteur) vers un autre (récepteur) d'une information discrète provenant d'une source qui ne dispose que d'un ensemble fini n de caractères appelé «alphabet».

Par une convention préalable, le destinataire a connaissance de l'alphabet utilisé par la source. Il peut donc interpréter l'information qu'il reçoit en fonction de cet alphabet. Il compare les signaux reçus (déformés et perturbés par la transmission dans le canal) à la liste de caractères possibles et en déduit par une «décision» lequel de ces caractères est le plus probablement à l'origine du signal reçu.

L'information numérique transmise peut ainsi, en principe, être régénérée intégralement à la réception. Toute fois, si la distorsion ou la perturbation des signaux reçus est telle qu'elle simule la présence d'un autre caractère, des erreurs de régénération irréversibles apparaissent et leur probabilité (P_e) est le critère principal d'appréciation de la qualité de transmission numérique.

La transition de l'information entre la source et le destinataire ne s'effectue donc pas sans risque d'erreurs à cause du bruit et la distorsion du canal. Afin de remédier à cela et donner une plus grande fiabilité au système de transmission, il existe plusieurs techniques de protection de l'information se basant toutes sur le principe d'ajouter une certaine redondance au message original avant sa transmission. Cette redondance permettra, à la réception, de détecter les erreurs éventuelles dans le message reçu et même de les corriger si elle est en quantité suffisante. Lorsque la redondance introduite ne permet que la détection, la correction exigera une procédure de retransmission de la partie infectée du message, et cela jusqu'à ce que aucune erreur ne soit détectée.

L'opération d'introduire de la redondance dans le but de corriger les erreurs éventuelles s'appelle codage correcteur d'erreurs ou codage de canal.

Les codes correcteurs d'erreurs peuvent être divisés en deux grandes classes : les codes par blocs linéaires et les codes convolutifs. Ces derniers font l'objet de notre présent travail.

Le premier chapitre consiste en des généralités sur le codage de canal en plus de l'exposé des codes blocs linéaires et d'un type particulier de ces derniers appelé codes cycliques.

Le second chapitre est réservé à la théorie du codage convolutif. Trois approches du sujet y sont exposées. La première introduit les codes convolutifs comme étant une généralisation des codes en blocs. La deuxième utilise le fait que le codage convolutif est en réalité une opération de convolution entre la séquence d'entrée et la réponse impulsionnelle du codeur. La troisième est basée sur le fait qu'un codeur convolutif peut être vu comme une machine à état dont l'évolution d'un état à un autre dépend de l'état de départ et l'entrée présente de la machine. Enfin des propriétés de distance de ce type de codes sont étudiées.

Le troisième chapitre est consacré aux procédés de décodage. Nous nous sommes limités aux décodeurs probabiliste à savoir le décodeur de vraisemblance maximale de Viterbi et le décodeur séquentiel de Fano. Les procédés de type déterministe ne sont pas étudiés car ils ne sont valables que pour des types particuliers de codes convolutifs.

Enfin le quatrième chapitre est consacré à la simulation et le test des performances de ces codes.

CHAPITRE 1

CODAGE DE CANAL

1-1 INTRODUCTION

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

Les deux paramètres clés lors de la conception d'un tel système sont:

- La puissance du signal transmis.
- La bande passante du canal.

Ces deux paramètres ensemble avec la densité spectrale de puissance du bruit du récepteur, déterminent le rapport de l'énergie du signal par bit sur la densité spectrale de puissance du bruit (E_b/N_0).

Des considérations pratiques placent une limite sur la valeur que nous pouvons assigner à (E_b/N_0). En conséquence, dans la pratique, nous nous orientons souvent vers un processus de modulation, mais nous trouvons qu'il n'est pas possible de fournir une qualité d'information acceptable.

Pour un rapport (E_b/N_0) fixé, la seule façon pratique valable pour une amélioration de la qualité de transmission, est d'utiliser le codage.

Une autre motivation pratique pour l'utilisation de codage est de réduire le rapport (E_b/N_0) exigé pour un débit d'erreurs par bit fixé. Cette réduction de (E_b/N_0), va permettre de réduire la puissance du signal exigée, ce qui permet aussi de réduire la taille de l'antenne et par conséquent le coût de la transmission.

Le codage de canal est une opération de traitement de signal, qui introduit une redondance contrôlée dans le message. Cette redondance, connue par le décodeur de canal, permet de détecter, voire corriger les erreurs de transmission. Cette fonction spécifique au transmission numérique, représente une partie essentielle d'un système de communication numérique.

L'addition de la redondance pour le message à coder nécessite l'augmentation de la largeur de bande de transmission. En plus, l'utilisation du codage rend le système plus complexe, spécialement l'implémentation des opérations de décodage à la réception. Ainsi, à la conception d'un système de communication numérique à codage contrôleur d'erreurs, on doit tenir compte des considérations de la largeur de bande et de la complexité du système.

1-2 MODELE D'UN SYSTEME DE COMMUNICATION NUMERIQUE

Le modèle est représenté par la figure (1-1) [6].

La transmission d'information numérique actuelle à travers un canal réel est accomplie par un modem(modulateur et démodulateur), qui peut opérer à un débit d'information numérique r_c .

La probabilité d'erreur P_e à la réception , en utilisant seulement le modem , dépend du débit r_c et du rapport signal sur bruit. Cette probabilité dépasse une valeur inférieure prescrite chaque fois qu'on augmente le débit r_c , d'où le besoin d'un recours à la conception des codeurs et de décodeurs de canaux afin d'obtenir des probabilités d'erreurs acceptables.

Nous décrivons ici brièvement les différentes fonctions d'un système de communication numérique.

- La source d'information délivre le message contenant l'information à transmettre.
- Le codeur de source supprime toute redondance dans le message.
- Le codeur de canal introduit une redondance contrôlée dans le message issu du codeur de source. Cette redondance connue aussi par le décodeur de canal permettra la détection et/ou la correction des erreurs dues au bruit inévitable.
- L'émetteur a pour fonction principale de modifier le signal message en une forme plus adaptée à la transmission à travers la voie de transmission.

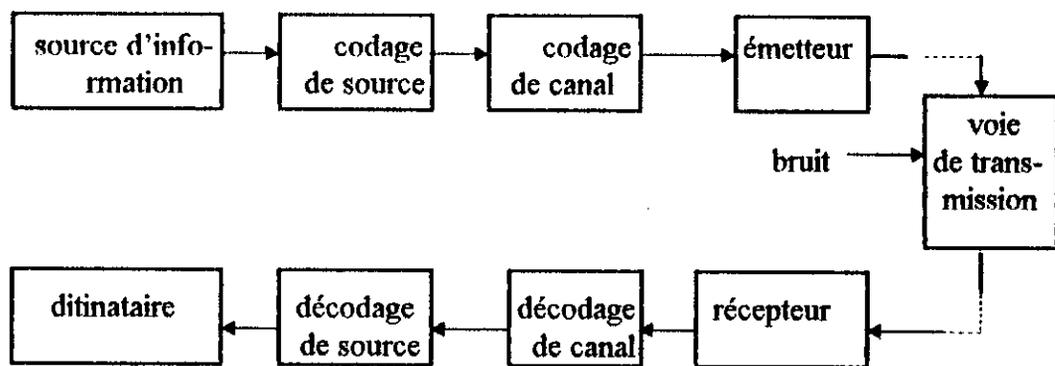


Figure 1-1 Modèle d'un système de communication numérique.

- La voie de transmission représente le milieu physique entre l'émetteur et le récepteur. Elle peut être l'espace libre, un câble coaxial ou une fibre optique.
- Le récepteur, le décodeur de canal et décodeur de source réalisent respectivement, les fonctions inverses de l'émetteur, du codeur de canal et du codeur de source afin de restituer le signal original.

1-3 MODELE DU CANAL

Dans notre schéma de la figure (1-1) , nous n'avons considéré que la perturbation causée par le bruit blanc additif et gaussien. Ce modèle de bruit représente une bonne approximation d'une classe importante des systèmes de communication numériques. La voie de transmission est dite alors canal à bruit blanc additif gaussien (AWGN).

Ce canal (AWGN) associé au modulateur et au démodulateur, représente ce qu'on appelle un canal discret. Cela est dû au fait que l'entrée du modulateur et la sortie du démodulateur prennent toutes les deux leurs valeurs dans deux alphabets finis.

Soit $X=[x_1, x_2, \dots, x_N]$ l'alphabet à l'entrée du canal discret et $Y=[y_1, y_2, \dots, y_M]$ l'alphabet à sa sortie. Lorsque la transformation du symbole x_i introduit à l'entrée, en symbole y_j à la sortie du canal ne dépend pas du temps, le canal est dit stationnaire. Si en plus, la transformation ne dépend pas des transformations qui la précèdent, le canal est dit stationnaire sans mémoire.

Un canal discret, stationnaire et sans mémoire peut être défini par la matrice de transition:

$$[P(X/Y)] = \begin{bmatrix} P(y_1/x_1) & \dots & P(y_1/x_N) \\ \vdots & \ddots & \vdots \\ P(y_M/x_1) & \dots & P(y_M/x_N) \end{bmatrix} \quad (1-1)$$

où, $p(y/x_i)$ est la probabilité d'avoir y_j en sortie lorsque x_i est introduit à l'entrée. Ce canal est schématisé sur la figure (1-2).

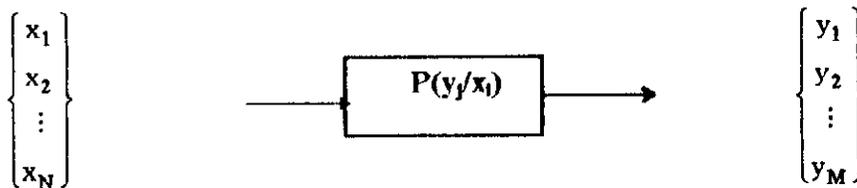


Figure 1-2 Canal discret sans mémoire.

Le canal discret sans mémoire le plus simple résulte de l'utilisation de deux alphabets, chacun d'eux ne contenant que deux symboles (0 et 1). C'est le canal binaire. Il est dit symétrique si:

et

$$\begin{aligned} p(i/j) &= p(j/i) = p; j \neq i \\ p(i/i) &= 1 - p; i = 0, 1 \end{aligned} \quad (1-2)$$

où p est la probabilité d'erreur.

Dans le cas du canal binaire symétrique, le démodulateur est un quantificateur à deux niveaux. Il fait une ferme (hard) décision sur lequel des deux symboles est transmis dans un intervalle donné. Ce type de canal représente la majorité des systèmes de communication, et c'est celui que nous utilisons dans ce qui suit.

Il existe cependant, un autre type de démodulateurs dont la sortie n'est plus binaire mais q-aires. Ce genre de démodulateurs, qui est un quantificateur à plusieurs niveaux, est dit à décision souple ou pondérée (soft décision). Si l'entrée du démodulateur est binaire, le canal est dit binaire à l'entrée, q-aires à la sortie. Les probabilités de transition dans ce cas dépendent de la position des niveaux de quantification, qui elle-même dépend nécessairement du niveau du signal et de la variance du bruit. La réalisation d'un quantificateur à plusieurs niveaux nécessite une commande automatique du gain (CAG) et complique davantage le décodeur.

La décision ferme ayant l'avantage d'une simplicité d'implémentation présente une performance moins bonne que celle de la décision souple.

1-4 THEOREME FONDAMENTAL DU CODAGE DE CANAL

1-4-1 Information mutuelle

Soient deux variables aléatoires X et Y qui prennent leurs valeurs respectivement dans: $X=\{x_1, x_2, \dots, x_N\}$ et $Y=\{y_1, y_2, \dots, y_M\}$.

La réalisation de l'événement $X=x_i$, apporte une information dite mutuelle sur la réalisation de l'événement $Y=y_j$, que l'on peut mesurer par:

$$I(x_i, y_j) = \text{Log}_2 (p(Y = y_j / X = x_i) / p(Y = y_j)) \quad (1-3)$$

L'information mutuelle moyenne est définie par :

$$I(X, Y) = \sum_{j=1}^m \sum_{i=1}^n I(x_i, y_j) P(X = x_i, Y = y_j). \quad (1-4)$$

1-4-2 Capacité d'un canal

La capacité d'un canal est définie comme étant la plus grande quantité d'information mutuelle moyenne qu'il peut transmettre:

$$C = \max I(X, Y). \quad (1-5)$$

Pour un canal binaire symétrique , on peut montrer que[13]:

$$C = 1 + (1-p)\text{Log}_2(1-p) + p\text{Log}_2(p). \quad (1-6)$$

et pour le canal gaussien , on peut aussi montrer que [13]:

$$C = BT\text{Log}_2(1 + S / N)$$

où:

B: Bande passante.

S/N: Rapport signal sur bruit.

1/T: Nombre de symboles transmis par le canal par unité de temps.

1-4-3 Théorème[2]

Avec un codage approprié, il est possible de transmettre les messages d'une source à travers un canal bruité, avec une probabilité d'erreurs arbitrairement faible, pourvu que son débit d'information reste inférieur à la capacité du canal.

Ce théorème n'indique malheureusement pas la façon avec laquelle le codage est réalisé.

1-5 TYPES D'ERREURS

Les erreurs dans les systèmes de communication numériques sont dues au bruit du canal de transmission. Généralement, deux sortes de bruits peuvent être distinguées dans les canaux de transmission.

- Le bruit blanc gaussien : constitue le souci principal dans la conception des modulateurs et des démodulateurs pour la transmission de l'information numérique. Les erreurs de transmission introduites par le bruit blanc gaussien sont considérées comme des erreurs aléatoires .

- Le bruit impulsif : est caractérisé par des intervalles un peu longs et des amplitudes importantes du bruit. Ce type de bruit résulte de plusieurs causes naturelles ou artificielles telles que les interrupteurs , arcs électrique ...etc . Quand un paquet d'erreurs apparaît, il affecte plusieurs symboles ou bits, et il y a d'habitude une dépendance des erreurs dans les symboles affectés. Ces erreurs apparaissent en paquets.

Les schémas de contrôle d'erreurs qui s'occupent d'erreurs aléatoires sont appelés codes correcteurs d'erreurs aléatoires, tandis que les schémas de codage conçus pour corriger les paquets d'erreurs sont appelés codes correcteurs de paquets d'erreurs.

1-6 TYPES DE CODES

Les codes de canal appelés aussi codes correcteurs d'erreurs sont répartis en deux catégories: les codes par blocs linéaires et les codes convolutifs. Il existe un type spécial de code par blocs linéaires appelé codes cycliques. Ils sont des codes par blocs linéaires vérifiant quelques propriétés supplémentaires. Dans la suite de ce chapitre nous allons étudier les codes par blocs linéaires et les codes cycliques. Les codes convolutifs, qui font l'objet de notre travail, vont être traités avec plus de détails dans les chapitres suivants.

1-7 CODES PAR BLOCS LINEAIRES

1-7-1 Principe

Un code linéaire par blocs est une application entre des blocs de K symboles qui constituent les messages, et des blocs de N symboles ($N > K$) appelés mot-codes. Le rapport $r = K/N$ est appelé rendement du code. L'opération de codage est illustré par la figure (1-3).

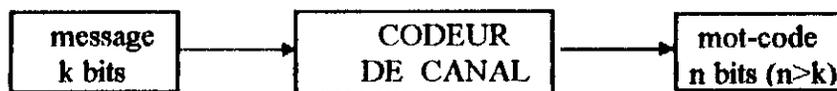


Figure 1-3 Principe de codage des codes par blocs

Lorsque les bits d'information sont placés au début ou à la fin du mot-code sans modifications, le code est dit systématique. De plus, si chaque mot-code peut être exprimé comme une combinaison linéaire d'autres mot-codes, le code est dit systématique linéaire. L'opérations de codage consiste en deux étapes de bases:

-La séquence d'information est segmentée en blocs messages de k bits chacun .

-Le codeur transforme chaque bloc en un autre plus large de n bits ($n > k$) suivant une loi prédéterminée. Les (n-k) bits de redondance sont générés à partir d'une combinaison linéaire des k bits messages. La linéarité des codes par blocs conduit à utiliser la notation matricielle pour les représenter. On passe du message M au mot-code C par une matrice génératrice G:

$$C = MG \quad (1-7)$$

où:

$$M = [m_0, m_1, \dots, m_{k-1}] \text{ et } C = [c_0, c_1, \dots, c_{n-1}].$$

Dans ce qui suit on s'intéressera au codes systématiques à cause de la simplicité de leurs structures et leur mise en oeuvre.

Dans ce cas :

$$C = \begin{cases} m_i & ; \quad i = 0, 1, \dots, k-1 \\ \text{bits de parité} & ; \quad i = k, \dots, n-1 \end{cases} \quad (1-8)$$

Soit b_i un bit de parité, on peut écrire:

$$b_i = p_{i0} m_0 + p_{i1} m_1 + \dots + p_{i, k-1} m_{k-1} \quad (1-9)$$

On peut combiner (1-7) et (1-9) en une seule représentation:

$$[c_0, c_1, \dots, c_{n-1}] = [m_0, m_1, \dots, m_{k-1}] \begin{bmatrix} 1 & 0 & \dots & 0 & p_{00} & p_{01} & \dots & p_{0, k-1} \\ 0 & 1 & \dots & 0 & p_{10} & p_{11} & \dots & p_{1, k-1} \\ \vdots & \vdots \\ 0 & 0 & \dots & 1 & p_{k-1, 0} & \dots & \dots & p_{k-1, n-k-1} \end{bmatrix} \quad (1-10)$$

ou bien

$$C = M \begin{bmatrix} I_{k \times k} & P_{k \times (n-k)} \end{bmatrix}$$

I_k : matrice unitaire $k \times k$

P: matrice $k \times (n - k)$, choisie de sorte que l'implémentation du codeur soit facile ainsi que pour avoir une bonne capacité de correction et un rendement élevé.

Un code de k bits messages et n-k bits de contrôle est symbolisé par (n,k).

N.B.: Toutes les opérations sont modulo-2.

Exemple 1-1

Soit la matrice génératrice G:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Les mots-codes correspondant aux blocs message donnés dans le tableaux(1-1), sont calculés selon la relation (1-10)

Tableau 1-1 Blocs messages et mots-codes.

Messages	Mot-codes
0 0 0	0 0 0 0 0 0
0 0 1	0 0 1 1 1 0
0 1 0	0 1 0 1 0 1
0 1 1	0 1 1 0 1 1
1 0 0	1 0 0 0 1 1
1 0 1	1 0 1 1 0 1
1 1 0	1 1 0 1 1 0

On peut montrer qu'il est toujours possible de trouver une matrice non nulle H, orthogonale à G, c'est à dire:

$$GH^t = 0; \quad (1-11)$$

avec $H = [P^t : I_{n-k}]_{(n-k) \times n}$

La matrice H, appelée matrice de contrôle, permet, comme nous verrons, de détecter les erreurs de transmission.

1-7-2 POUVOIR DE DETECTION ET DE CORRECTION DES ERREURS[7]

Tout d'abord, nous allons définir quelques notions de base:

-Le poids de Hamming d'un mot-code C est le nombre de composantes différentes de zéro .

-La distance de Hamming entre deux mot-codes C1 et C2 est le nombre de composantes par lesquelles C1 et C2 diffèrent

-La distance minimale d'un code linéaire par blocs est la plus petite distance de Hamming entre deux mot-codes.

On peut montrer que la distance minimale d'un code linéaire par blocs est égale au poids de Hamming du mot-code, différent du mot tous zéro, dont le poids est le plus petit .

Un code linéaire par blocs, avec une distance minimale d_{min} , peut corriger jusqu'à $\lfloor (d_{min}-1)/2 \rfloor$ erreurs, et détecter jusqu'à $(d_{min}-1)$ erreurs dans chaque mot-code, où $[x]$ représente la partie entière de x .

De ce qui précède, on déduit que pour un n et k fixés, le but est d'avoir un code linéaire par blocs dont la distance minimale est la plus grande possible.

1-7-3 DECODAGE PAR SYNDROME

Le principe de détection des erreurs est basé sur le fait que parmi les 2^n mots de n bits, seuls 2^k mots ont un sens. Ce sont les 2^k mot-codes correspondants aux 2^k mots messages de k bits délivrés par la source .

Lorsque, à cause du bruit, un mot-code est transformé en un mot de n bits n'ayant pas un sens, l'erreur commise est détectée par le décodeur. Si par contre l'action du bruit transforme un mot-code en un autre mot-code, l'erreur ne sera plus détectable.

Soit le mot erroné reçu à l'entrée du décodeur:

$$R=C+E \quad (1-12)$$

où:

C est le mot-code transmis et E représente le mot erreur.

On définit le syndrome comme étant un vecteur à $(n-k)$ composantes, donné par:

$$S=RH^t \quad (1-13)$$

Alors (1-13) peut s'écrire comme:

$$S=EH^t \quad (1-14)$$

Car :

$$\begin{aligned} S &= RH^t = (C+E)H^t \\ &= CH^t + EH^t \\ &= M(GH^t) + EH^t \\ &= EH^t. \end{aligned}$$

où: M représente le mot-code et G représente la matrice génératrice du code.

La relation (1-14) montre que le syndrome S ne dépend que de la configuration des erreurs. S est donc nul si et seulement si R est un mot-code (E est nulle).

La réception d'un S non nul suffit à la détection d'erreurs. La correction se fait en estimant le mot erreur E à partir de S. Donc l'algorithme de décodage est le suivant:

- Pour un mot reçu R, calculer le syndrome: $S=RH^t$.
- A partir de S, estimer le mot erreur E, à partir d'un tableau préalable.
- Calculer le mot corrigé: $C=R+E$.

1-7-4 CODEURS

Les codeurs linéaires par blocs doivent stockés la matrice G, ou bien la matrice P, et faire des opérations arithmétiques pour générer les bits de contrôle. Dans le cas où n est petit, on peut utiliser des simples registres à décalage pour implémenter l'opération de codage.

Exemple 1-2[12]

Soit le code de Hamming (7,4), la matrice P correspondante est donnée par:

$$P = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

D'après la relation (1-9), les bits de parité sont donnés par:

$$\begin{cases} b_0 = m_0 + m_1 + m_2 \\ b_1 = m_1 + m_2 + m_3 \\ b_2 = m_0 + m_1 + m_3 \end{cases}$$

Ce codeur est schématisé par la figure(1-4).

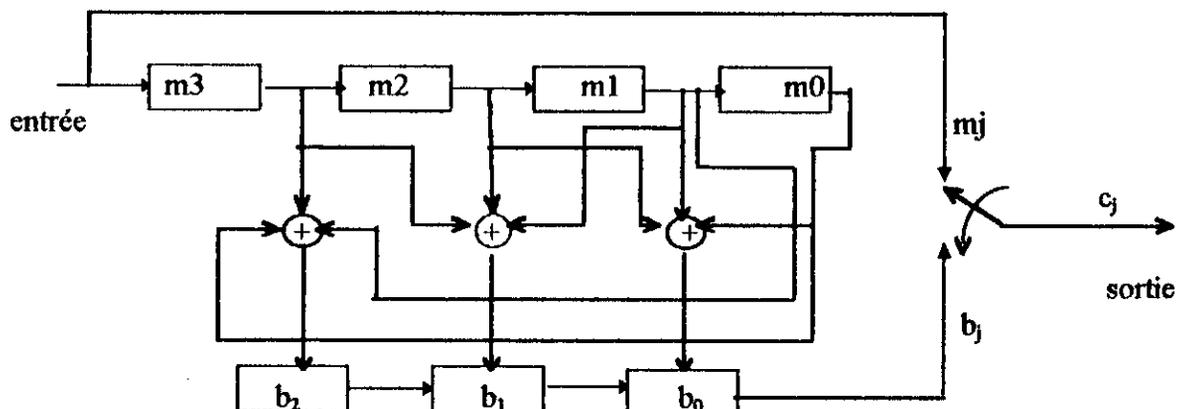


Figure 1-4 Implémentation du codeur Hamming (7,4) systématique.

On voit que le codeur se réduit à un registre à décalage de 4 bits, trois additionneurs modulo-2

à 3 entrées, un registre à décalage de 3 bits, et enfin un commutateur qui sélectionne d'abord les 4 bits d'information, puis les trois bits de redondance.

1-7-5 DECODEURS

Considérons un tableau constitué de 2^{n-k} lignes, chacune contenant 2^n mots de n bits: La première ligne est constituée des 2^k mot-codes, débutant par l'élément nul. Le premier élément de la deuxième ligne est un mot dépourvu de sens, dont le poids de Hamming est le minimum. La deuxième ligne est complétée en additionnant son premier élément aux mot-codes de la première ligne comme indiqué dans le tableau (1-2).

On fait la même chose pour les autres lignes et cela jusqu'à ce que tous les 2^n mots de n bits seront épuisés.

Tableau 1-2 Tableau de décodage

C_0	C_1	C_2	C_{k-1}
E_1	C_1+E_1	C_2+E_1	$C_{k-1}+E_1$
E_2	C_1+E_2	C_2+E_2	$C_{k-1}+E_2$
.
.
.
E_2^{n-k}	$C_1+E_2^{n-k}$	$C_2+E_2^{n-k}$	$C_{k-1}+E_2^{n-k}$

Avec le tableau (1-2), le décodage sera fait en reconnaissant la colonne où figure le mot réceptionné, et le premier élément de cette colonne correspond au mot-code transmis par la source. Ceci explique l'appellation du tableau(1-2) tableau de décodage.

Les éléments d'une ligne s'appellent coset. Chaque coset est caractérisé par un seul syndrome car:

$$[C_i + E_j]H^t = E_j H^t ; \forall C_i$$

Exemple 1-3[14]

Considérons $2^3 = 8$ mots, dont 2 mots sont des mot-codes. Le tableau(1-3) représente le tableau de décodage.

Tableau 1-3 Tableau de décodage

0	0	0	1	1	1
0	0	1	1	1	0
0	1	0	1	0	1
1	0	0	0	1	1

Si par exemple on reçoit le mot 100, on décidera que le mot émis par la source est 000 (premier mot de la colonne). Autrement dit on pourra affirmer qu'une erreur est intervenue en position une.

Le tableau de décodage décrit précédemment nécessite le stockage de 2^{n-k} syndromes de longueur $(n-k)$ bits et 2^{n-k} mot erreurs de longueur n bits. Ainsi le stockage nécessaire est de l'ordre de $2^{n-k}[(n-k)]$ bits. Une fois que n augmente, le codeur devient très complexe et le tableau de décodage devient aussi très grand. Remarquons aussi que c'est en augmentant n qu'on augmente la distance minimale et donc les performances du code, d'où la limitation dans l'utilisation de tels codes. Nous allons introduire dans ce qui suit une sous-classe des codes linéaires par blocs, c'est celle des codes cycliques.

1-8 CODES CYCLIQUES

1-8-1 Principe

Un code cyclique (n,k) est un code linéaire par blocs possédant la propriété suivante: Si $C_0=(C_0, C_1, \dots, C_{n-1})$ est un mot-code, alors le mot $C_1=(C_{n-1}, C_0, C_1, \dots, C_{n-2})$, obtenu par une permutation circulaire à droite, est aussi un mot-code. Cela entraîne que le mot $C_i=(C_{n-i}, C_{n-i+1}, \dots, C_{n-1})$ obtenu par i permutations à droite est aussi un mot-code. Chaque mot-code de n bits peut être représenté par un polynôme de degré $(n-1)$, dont les coefficients correspondent aux composantes du mot-codes.

$$C=(C_0, C_1, \dots, C_{n-1}) \Leftrightarrow C(x)=C_0 + C_1 X + \dots + C_{n-1} X^{n-1} \quad (1-15)$$

Puisque les coefficients du polynôme sont binaires, toutes les opérations effectuées doivent être modulo-2.

On peut montrer [7] que le polynôme $C_i(X)=C_{n-i} + C_{n-i+1} X + \dots + C_0 X^{n-1} + \dots + C_{n-i-1} X^{n-1}$ est le reste de la division du polynôme $X^i C(X)$ par (X^n-1) , c'est à dire:

$$X^i C(X) = Q(X) (X^n-1) + C_i(X) \quad (1-16)$$

1-8-2 Polynôme générateur

Un code cyclique (n,k) est caractérisé par l'ensemble complet des polynômes de degrés $(n-1)$ ou moins. Cet ensemble contient un polynôme de degré minimum $(n-k)$. Ce polynôme noté $g(x)$ est appelé polynôme générateur. Il est équivalent à la matrice génératrice G donnée par (1-7).

Propriétés[7]

-Le polynôme générateur $g(x)$ est unique, Il est de la forme:

$$g(x) = 1 + \sum_{i=1}^{n-k-1} g_i x^i + x^{n-k} \quad (1-17)$$

où $g_i \in \{0,1\}$.

-Tout multiple du polynôme générateur correspond à un mot-code, c'est à dire:

$$C(X) = A(X) g(X) \text{ mod}(X^n - 1) \quad (1-18)$$

-Pour un code (n,k) systématique, le mot-code est de la forme:

$$C = b_0 b_1 \dots b_{n-k-1} m_0 m_1 \dots m_k$$

où: b_i est un bit de parité et m_i est un bit message.

avec :

$$\begin{aligned} C(X) &= b(X) + X^{n-k} m(X) \\ b(X) &= b_0 + b_1 X + \dots + b_{n-k-1} X^{n-k-1} \\ m(X) &= m_0 + m_1 X + \dots + m_k X^k \end{aligned} \quad (1-19)$$

de (1-18) et (1-19) et en divisant par $g(X)$ on obtient:

$$X^{n-k} m(X) / g(X) = A(X) + b(X) / g(X)$$

$b(X)$ est donc le reste de la division de $X^{n-k} m(X)$ par le polynôme $g(X)$.

En résumé, l'opération de codage cyclique se fait en trois étapes:

- Multiplication du polynôme message par X^{n-k}
- Division de $X^{n-k} m(X)$ par le polynôme générateur $g(X)$.
- Addition du reste $b(X)$ à $X^{n-k} m(X)$ pour obtenir le polynôme du mot-code.

1-8-3 Polynôme de parité:

Un code cyclique (n,k) est également caractérisé par un polynôme $h(X)$ de degré k, appelé polynôme de parité.

On montre que le reste de la division de $h(X) g(X)$ par $(X^n - 1)$ est nul [7], c'est à dire:

$$h(X) g(X) \text{ mod}(X^n - 1) = 0.$$

1-8-4 Codeurs

L'opération du codage décrite ci-dessus (1-8-2), consiste en une division du polynôme $X^{n-k} m(X)$ par le polynôme $g(X)$ dont le reste est le polynôme $b(X)$. Cette division est réalisée en utilisant un simple registre à décalage comme le montre la figure (1-5) [12].

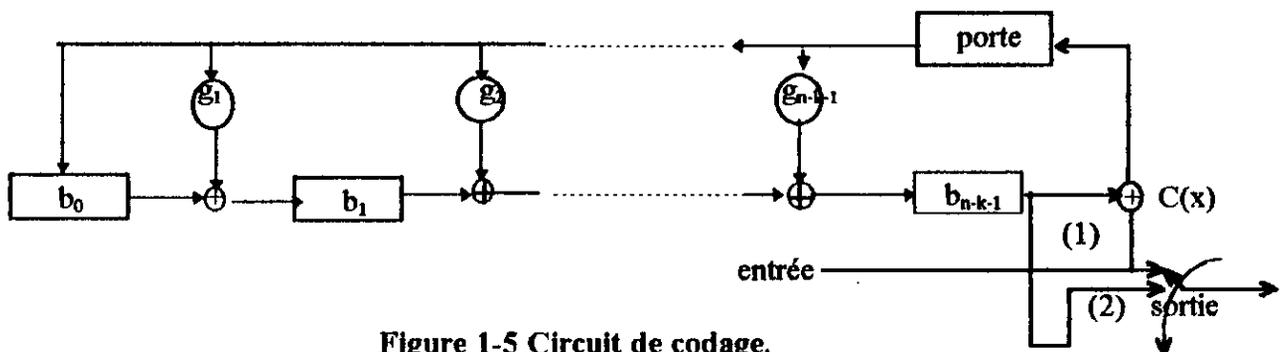


Figure 1-5 Circuit de codage.

où: +: correspond à une addition modulo-2.

g: le ième coefficient du polynôme générateur, il représente une connexion s'il est égal à 1 et dans le cas contraire il n'y a pas de connexion.

Le fonctionnement du codeur est le suivant [13]:

- La porte est ouverte, le commutateur est en position (1). Dans ce cas les k bits d'information sont transmis directement à la sortie et au circuit de division au même temps.

- Après la transmission du dernier bit message, la porte est fermée, le commutateur est mis en position (2), c'est au tour des (n-k) bits résultant de l'opération de division qui sont restés dans les (n-k) cellules du registre, d'être transmis à la sortie.

Exemple 1-4[13]

Considérons le code cyclique (7,4) dont le polynôme générateur est donné par:

$$g(X)=1 + X + X^3.$$

et le circuit de codage est représenté sur la figure (1-6).

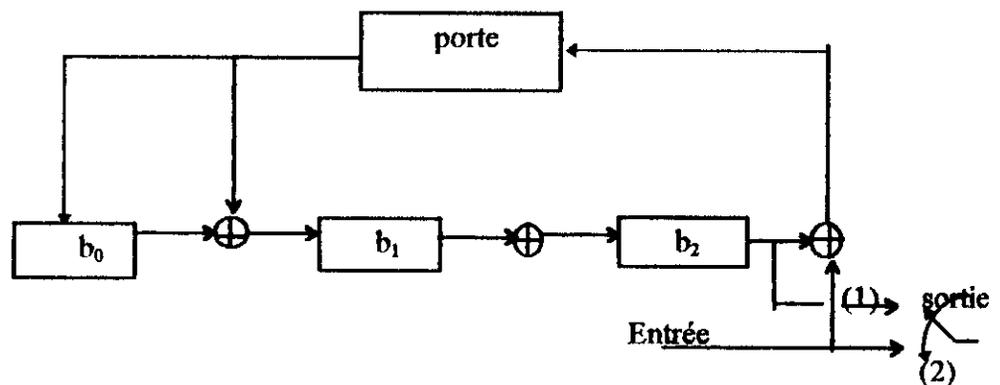


Figure 1-6 Codeur cyclique (7,4), dont le polynôme générateur est: $g(X)=1+X+X^3$.

L'opération de codage sur les bits d'information 1001 est illustrée dans le tableau (1-4).

Tableau 1-4 Operation de codage

décalage	entrée	contenu du registre
état initial	0 0 0
1	1	1 1 0
2	0	0 1 1
3	0	1 1 1
4	1	0 1 1

1-8-5 Décodeurs

L'opération de décodage des codes cycliques consiste en général à calculer le syndrome du mot reçu, à déterminer le mot erreur le plus probable qui est en relation biunivoque avec le syndrome, et enfin de calculer le mot décodé par addition modulo-2 du mot erreurs au mot reçu.

La figure (1-7) illustre ce procédé. Le registre à décalage à n bits stock temporairement le mot reçu. En même temps le mot reçu est transmis au circuit de division par le polynôme générateur. Le reste de cette division modulo(X^n-1) est le syndrome $S(X)$, qui sera transmis au circuit détecteur d'erreurs. Ce dernier fera une correspondance entre le syndrome et l'erreur qui est lui liée. L'addition modulo-2 du mot erreur avec le mot reçu est faite bit par bit, le résultat est transmis en sortie du décodeur.

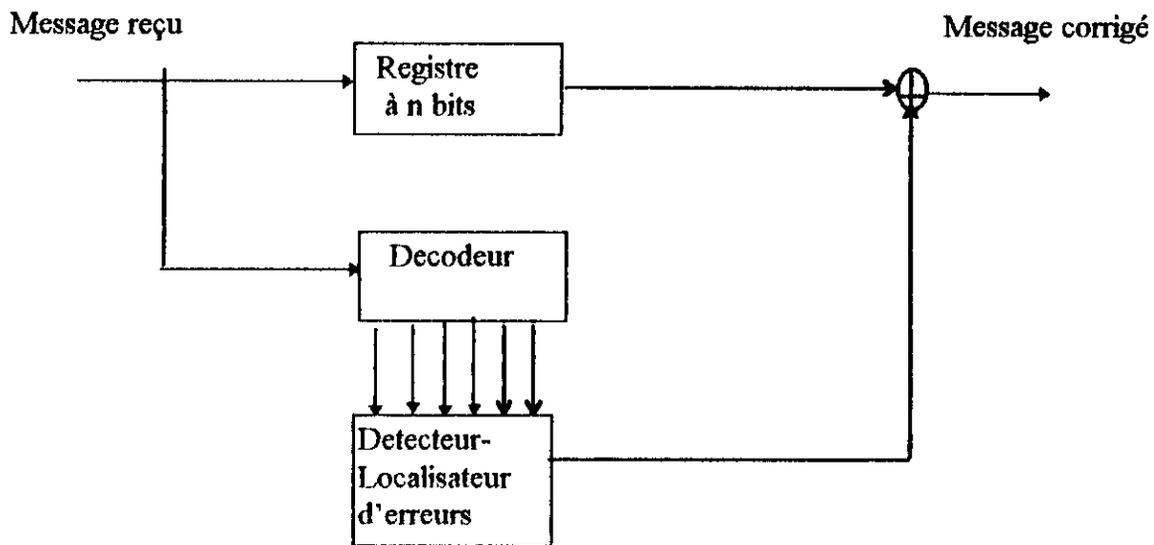


Figure 1-7 Forme générale d'un décodeur de code cyclique.

La correspondance entre le syndrome et le mot erreur est faite à l'aide d'une table dans laquelle on doit stocker les 2^{n-k} mots de k bits qui sont les syndrome et 2^{n-k} mots de n bits qui sont les mots erreurs. On remarque donc que la complexité du décodeur croît exponentiellement avec (n-k).

1-9 PRESENTATION DE QUELQUES CODES CYCLIQUES

Code Hamming correcteur d'une seule erreur

On peut définir un code de Hamming à partir de sa matrice de contrôle H ou de son polynôme générateur $g(X)$. Les colonnes de la matrice H sont les représentations binaires de tous les nombres de 1 à n. Chaque colonne étant constituée de (n-k) symboles binaires. Les paramètres de ce code sont :

- Longueur du mot-code $n=2^{n-k}-1$.

- Longueur des messages k .

- Distance minimale $d_{\min}=3$.

Ce code permet de corriger une seule erreur dans un bloc de n bits [2].

Codes Bose-Chaudhuri-Hocquenghem (BCH)

Ces codes ont pour paramètres :

- Longueur du mot-code: $n=2^m - 1$, m longueur du mot message.

- Nombre de symboles de redondance: $n-k \leq m \times t$.

- Distance minimale : $d_{\min} \geq 2t+1$.

Ils permettent de corriger t erreurs dans un bloc de n bits.

Codes de Reed-Solomone (RS)

Les codes RS sont des codes à symboles q -aires, où q est une puissance de 2 ($q=2^p$). Leurs paramètres sont:

- Longueur de mot-code : $n=q-1$.

- Nombre de symboles de redondance : $n-k=2 \times t$.

- Distance minimale : $d_{\min}=2 \times t+1$.

Ils permettent de corriger t erreurs dans un bloc de n bits.

1-10 CONCLUSION

Bien qu'ils sont faciles à implémenter, surtout les codeurs, les codes cycliques deviennent trop complexes dès qu'on cherche à corriger un nombre important d'erreurs. Cela est dû au fait que dans la correction d'un nombre important d'erreurs, on doit avoir un mot-code trop grand. Il en résulte, en tenant compte de ce qui a été dit avant, un décodeur très complexe. C'est la raison pour laquelle ce genre de codes est surtout utilisé pour la détection d'erreurs seulement, ou la correction d'un nombre réduit d'erreurs. Pour un grand nombre d'erreurs à corriger, les codes convolutifs, qui feront l'objet des prochains chapitres, sont les mieux adaptés.

CHAPITRE 2

CODES CONVOLUTIFS

2-1 Principe

Dans le cas des codes par blocs, les bits de redondance d'un bloc sont calculés uniquement à partir des bits d'information du même bloc.

Pour les codes convolutifs, ou codes récurrents, chaque bloc de longueur n est codé non seulement en fonction des k bits d'information du bloc, mais également en fonction des bits d'information qui appartiennent aux blocs précédents.

Les codes convolutifs présentent l'avantage d'être plus simples à implémenter que les codes par bloc, lorsque la longueur du mot-codé devient élevée.

Cette technique est bien adaptée à la réalisation de codes à grande capacité de détection et de correction. Les codes convolutifs sont aujourd'hui utilisés pour des applications spéciales qui exigent une bonne capacité de correction, ou pour accroître le débit dans les canaux existants. Contrairement aux codes par blocs, les codes convolutifs non systématiques sont préférés.

Un code convolutif est généré en faisant passer la séquence d'information dans un registre à décalage linéaire, de M étages, à nombre d'états fini.

Soit un codeur convolutif à b entrées et n sorties. Dans ce type de codes, une séquence de L bits produit, à la sortie, une séquence codée de longueur: $n[(L/b) + M]$.

où:

b : est le nombre d'entrées du codeur.

n : est le nombre de sorties du codeur.

M : est la mémoire du codeur, elle est égale au nombre d'étages du registre à décalage constituant le codeur..

On définit le débit ou le rendement (r) du codeur comme suit:

$$r = \frac{L}{n[(L/b) + M]} \quad \text{bits/Symbole.}$$

Généralement, on a: $L \gg M$, donc le rendement se simplifie à: $r \approx b/n$ bits/symbole.

La longueur de contrainte d'un code convolutif, exprimée en terme de bits messages, est définie comme le nombre d'étages du registre à décalage M plus «un» soit:

$$K = M + 1.$$

En général, le registre à décalage comporte M étages (de b bits) et n fonctions algébriques, comme le montre la figure (2.1)[12].

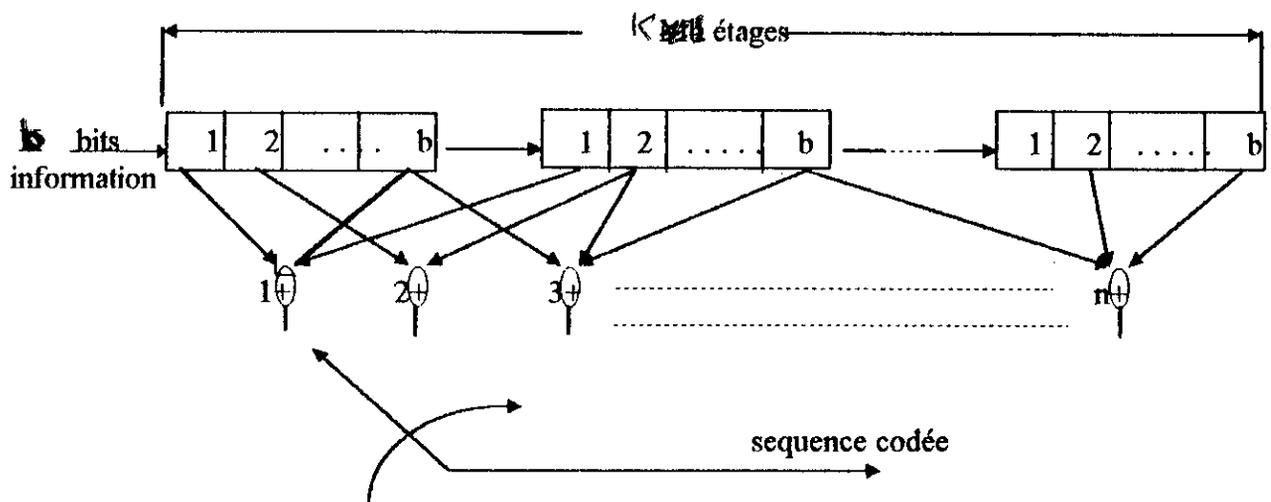


Figure 2-1 Forme générale d'un codeur convolutif.

Il existe trois façons de décrire un code convolutif:

- La représentation matricielle.
- L'approche temporelle ou dans le domaine des transformées.
- Le diagramme en arbre, le diagramme en treillis et le diagramme d'état.

2-2 REPRESENTATION MATRICIELLE[1]

La séquence d'information à coder peut être représentée par un vecteur (m) comportant un nombre infini d'éléments m_i de k bits, soit:

$$m = [m_0, m_1, \dots, m_i, \dots], \text{ avec } m_i = [m_i(1), m_i(2), \dots, m_i(k)].$$

$m_i(j)$ est le bit d'indice j du i ème mot d'information de longueur k de la séquence d'information.

La séquence codée $X = [X_0, X_1, \dots, X_i, \dots]$, constituée d'une suite de blocs X_i de longueur n , est construite à partir de m avec la relation:

$$X = m.G = [X_0, X_1, \dots, X_i, \dots] \quad (2.1)$$

où: G est la matrice génératrice du code, de dimension infinie avec:

$$G = \begin{bmatrix} G_0 & G_1 & G_2 & \dots & G_{M-1} \\ G_0 & G_1 & G_2 & \dots & G_{M-1} & G_M \\ & & & & & & G_0 & G_1 & G_2 & \dots & G_{M-1} \end{bmatrix} \quad (2.2)$$

Soit par exemple la séquence message: $m = (1,0,0,1,1)$ en utilisant la réalisation (2.1), on a:

$$X = [1,0,0,1,1,0,0,0] \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$X = [11,10,11,11,01,01,11].$$

2-3 L'APPROCHE DANS LE DOMAINE TEMPOREL, ET L'APPROCHE DANS LE DOMAINE DES TRANSFORMEES [8]

2-3-1 Approche dans le domaine temporel

Dans ce cas, le codeur convolutif de rendement $1/n$ est considéré comme un système linéaire qui peut être défini par l'ensemble de n réponses impulsionnelles.

Ces réponses impulsionnelles, appelées aussi séquences génératrices, sont obtenues en déterminant les n séquences de sortie du codeur qui résultent de l'application de la séquence $(1,0,0,\dots)$ à l'entrée du codeur.

La séquence codée est obtenue en effectuant le produit de convolution de la séquence message avec les réponses impulsionnelles du codeur .

Le codeur de la figure(2-2) de rendement $1/2$, à deux réponses impulsionnelles caractérisant son comportement dans le domaine temporel .

La séquence $(g^{(1)}_0, g^{(1)}_1, \dots, g^{(1)}_m)$ représente la réponse du «trajet haut entrée-sortie du sommateur » du codeur, et la séquence $(g^{(2)}_0, g^{(2)}_1, \dots, g^{(2)}_m)$ représente la réponse du «trajet bas entrée-sortie du sommateur » du codeur.

Soit (m_0, m_1, m_2, \dots) la séquence du message qui entre dans le codeur de la figure (2-2), un seul bit par unité de temps, en commençant par m_0 .

Le codeur génère deux séquences de sortie notées par $\{X^{(1)}_i\}$ et $\{X^{(2)}_i\}$, par convolution de la séquence message avec les deux séquences génératrices du code:

$$x_i^{(1)} = \sum_{e=0}^M g_e^{(1)} m_{i-e} \quad ; i = 0, 1, \dots \quad (3-4)$$

avec $m_{i-e} = 0$ pour tout $e > i$.

$$x_i^{(2)} = \sum_{e=0}^M g_e^{(2)} m_{i-e} \quad ; i = 0, 1, \dots$$

avec $m_{i-e} = 0$ pour tout $e > i$. (3-5)

Après la convolution, les deux séquences sont multipliées, et on obtient à la sortie du codeur:

$$\{X_i\} = \{X^{(1)}_0 X^{(2)}_0, X^{(1)}_1 X^{(2)}_1, \dots, X^{(1)}_m X^{(2)}_m\} \quad (3-6)$$

Exemple 2-1[8]

Soit le codeur de la figure (2-2) caractérisé par les deux séquences génératrice suivantes:

$(g^{(1)}_0, g^{(1)}_1, g^{(1)}_2) = (1, 1, 1)$: trajet haut de l'entrée-sortie.

$(g^{(2)}_0, g^{(2)}_1, g^{(2)}_2) = (1, 0, 1)$: trajet bas de l'entrée-sortie

Appliquons la séquence message suivante à l'entrée du codeur

$$(m_0, m_1, m_2, m_3, m_4) = (1, 0, 0, 1, 1).$$

La relation (3.4), nous donne:

$$X^{(1)}_0 = g^{(1)}_0 \cdot m_0 = 1 \cdot 1 = 1.$$

$$X^{(1)}_1 = g^{(1)}_0 \cdot m_1 + g^{(1)}_1 \cdot m_0 = 1 \cdot 0 + 1 \cdot 1 = 1.$$

$$X^{(1)}_2 = g^{(1)}_0 \cdot m_2 + g^{(1)}_1 \cdot m_1 + g^{(1)}_2 \cdot m_0 = 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 1 = 1.$$

$$X^{(1)}_3 = g^{(1)}_0 \cdot m_3 + g^{(1)}_1 \cdot m_2 + g^{(1)}_2 \cdot m_1 = 1 \cdot 1 + 1 \cdot 0 + 1 \cdot 0 = 1.$$

$$X^{(1)}_4 = g^{(1)}_0 \cdot m_4 + g^{(1)}_1 \cdot m_3 + g^{(1)}_2 \cdot m_2 = 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 = 0.$$

$$X^{(1)}_5 = g^{(1)}_1 \cdot m_4 + g^{(1)}_2 \cdot m_3 = 1 \cdot 1 + 1 \cdot 0 = 0.$$

$$X^{(1)}_6 = g^{(1)}_2 \cdot m_4 = 1 \cdot 1 = 1.$$

Donc la sortie haute est : $\{X^{(1)}_i\} = \{1111001\}$.

De la même manière, pour la sortie basse, la relation (3-5), nous donne :

$$X^{(2)}_0 = g^{(2)}_0 \cdot m_0 = 1 \cdot 1 = 1.$$

$$X^{(2)}_1 = g^{(2)}_0 \cdot m_1 + g^{(2)}_1 \cdot m_0 = 1 \cdot 0 + 0 \cdot 1 = 0.$$

$$X^{(2)}_2 = g^{(2)}_0 \cdot m_2 + g^{(2)}_1 \cdot m_1 + g^{(2)}_2 \cdot m_0 = 1 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 = 1.$$

$$X^{(2)}_3 = g^{(2)}_0.m_3 + g^{(2)}_1.m_2 + g^{(2)}_2.m_1 = 1.1 + 0.0 + 1.0 = 1.$$

$$X^{(2)}_4 = g^{(2)}_0.m_4 + g^{(2)}_1.m_3 + g^{(2)}_2.m_2 = 1.1 + 0.1 + 1.0 = 1.$$

$$X^{(2)}_5 = g^{(2)}_1.m_4 + g^{(2)}_2.m_3 = 0.1 + 1.1 = 1.$$

$$X^{(2)}_6 = g^{(2)}_2.m_4 = 1.1 = 1.$$

On obtient alors : $\{X^{(2)}_i\} = \{1011111\}$.

le multiplexage des deux séquences, permet d'obtenir la séquence codée:
 $\{X_i\} = \{11,10,11,11,01,01,11\}$.

2-3-2 Approche dans le domaine des transformées

En se basant sur la théorie des filtres linéaires dans laquelle le produit de convolution décrivant l'opération de filtrage dans le domaine temporel, est remplacée par la multiplication des transformées de Fourier dans le domaine fréquentiel.

Sachant aussi que le codeur convolutif est une machine à état finis, invariante dans le temps et linéaire, nous pouvons simplifier le calcul de la séquence codée par l'application d'une transformée appropriée.

Les réponses impulsionnelles de chaque trajet dans le codeur sont remplacées par des polynômes dont les coefficients sont représentés par les éléments des réponses impulsionnelles.

Pour bien illustrer cela, prenons l'exemple d'un codeur de rendement 1/2 et de longueur de contrainte M+1. On définit pour le trajet haut de l'entrée-sortie du codeur, le polynôme:

$$g^{(1)}(D) = g^{(1)}_0 + g^{(1)}_1.D + \dots + g^{(1)}_M.D^M$$

où : $g^{(1)}_0, g^{(1)}_1, \dots, g^{(1)}_M$ sont les éléments de la réponse impulsionnelle.

- D : une variable représentant un opérateur de retard unité. La puissance de D définit le nombre des unités de retard de temps par lesquelles le bit associé dans la réponse impulsionnelle est retardé par rapport au premier bit.

De même, on définit pour le trajet bas de l'entrée-sortie du codeur, le polynôme:

$$g^{(2)}(D) = g^{(2)}_0 + g^{(2)}_1.D + \dots + g^{(2)}_M.D^M$$

$g^{(1)}(D)$ et $g^{(2)}(D)$ sont appelés polynômes générateurs de code.

La séquence d'origine est représentée par le polynôme:

$$m(D) = m_0 + m_1.D + \dots + m_{L-1}.D^{L-1}$$

Les deux polynômes représentant les sorties des deux trajets sont données par :

$$X^{(1)}(D) = g^{(1)}(D).m(D) \tag{3-7}$$

$$X^{(2)}(D) = g^{(2)}(D).m(D) \tag{3-8}$$

La séquence de sortie $\{X_i\}$ est obtenue par une simple lecture des coefficients de $X^{(1)}(D)$ et $X^{(2)}(D)$.

Exemple 2-2[8]

On prend le même exemple (2-1). La réponse impulsionnelle du trajet haut de l'entrée-sortie du codeur est (1,1,1), le polynôme générateur correspondant est :

$$g^{(1)}(D) = 1 + D + D^2$$

La réponse impulsionnelle du trajet bas de l'entrée-sortie du codeur est (1,0,1), le polynôme générateur correspondant est :

$$g^{(2)}(D) = 1 + D^2.$$

Pour la séquence message (1,0,0,1,1), nous avons la représentation polynomiale:

$$m(D) = 1 + D^3 + D^4.$$

En utilisant les relations (3-7) et (3-8), on trouve:

$$\begin{aligned} X^{(1)}(D) &= (1 + D^3 + D^4) \cdot (1 + D + D^2) \\ &= (1 + D + D^2 + D^3 + D^6) \end{aligned} \quad \Leftrightarrow \{X^{(1)}_i\} = \{1111001\}$$

$$\begin{aligned} X^{(2)}(D) &= (1 + D^2) \cdot (1 + D^3 + D^4) \\ &= (1 + D^2 + D^3 + D^4 + D^5 + D^6) \end{aligned} \quad \Leftrightarrow \{X^{(2)}_i\} = \{1011111\}$$

Donc: $\{X_i\} = \{11,10,11,11,01,01,11\}$.

2-4 Diagramme en arbre, diagramme en treillis et diagramme d'état[15]

Nous pouvons considérer un codeur convolutif comme une machine à nombre d'états fini, dont la structure peut être mise en évidence à l'aide de l'un quelconque des trois diagrammes en arbre, en treillis, ou le diagramme d'état.

Nous allons illustrer ces trois représentations à l'aide de l'exemple de figure(2.2).

2-4-1 Diagramme en arbre

Le diagramme en arbre de l'exemple considéré est représenté par la figure(2.3). Nous pouvons y représenter à la fois la suite des symboles entrant dans le codeur et celle qui en sorte.

La séquence d'entrée est représentée par un chemin dans le diagramme, le parcourant de gauche à droite. En partant du noeud de départ, on choisit la branche supérieure si le bit d'entrée correspondant est «zéro», sinon on choisit la branche inférieure.

la séquence de sortie est celle des symboles écrits sur les branches de l'arbre, de ce même chemin.

Ainsi, la suite entrante(1,0,0,1,1) est représentée par le chemin descendant la première fois, puis montant à la deuxième étape, même chose pour la troisième étape, et descendant à la quatrième et la cinquième étape. Elle provoque l'émission par le codeur des symboles lus successivement sur chaque branche parcourue, soit:(11,10,11,11,01). Nous pouvons donc établir sur la figure(2.3) toutes les suites sortantes correspondantes aux 32 suites possibles des 5 premiers symboles d'information entrants. Ce diagramme met clairement en évidence aussi qu'après les trois premières étapes, la structure de l'arbre devient répétitive. Nous pouvons voir immédiatement, en

effet, qu'au delà de la troisième branche, les symboles codés associés aux branches partant des deux noeuds marqués a sont identique et de même pour tout les couples marqués de la même lettre. La raison en apparaît évidente à l'examen du codeur: lorsque le troisième symbole d'information pénètre dans le codeur, le premier à y avoir été introduit sort du dernier étage, à droite, et par conséquent cesse de contribuer à déterminer les symboles codés sortants. Par conséquent, les suites de données 100 x y ..., et 000 x y ..., engendrent les mêmes symboles codés au-delà de la troisième branche. Les deux noeuds marqués (a) peuvent être donc confondus.

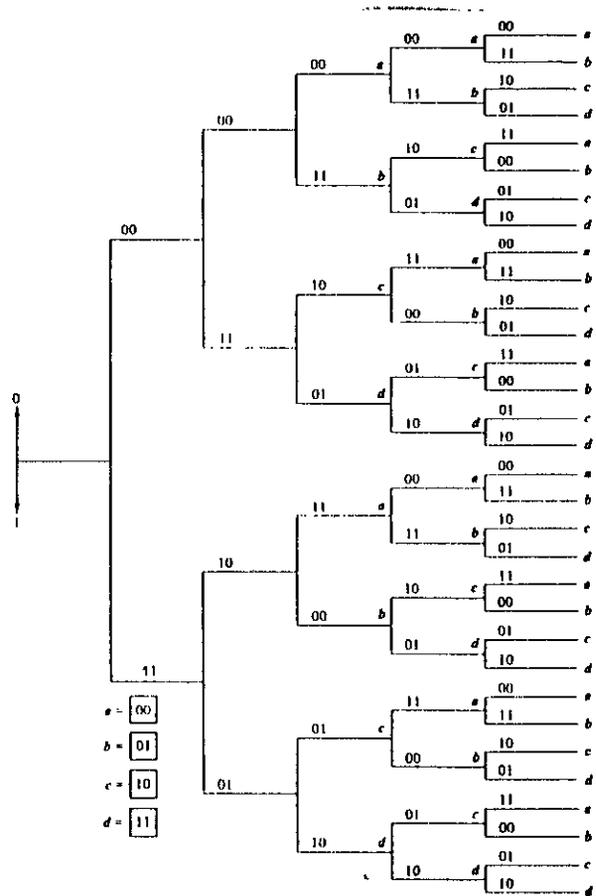


Figure 2-3 Diagramme en arbre du codeur de la Fig(2-2).

2-4-2 Diagramme en treillis

Le fait de confondre les noeuds marqués par la même lettre nous conduit à redessiner le diagramme en arbre selon la figure(2.4). Le diagramme de cette figure est appelée diagramme en treillis.

Nous adoptons la convention que les branches qui correspondent à un symbole entrant «Zéro» sont dessinés en trait plein, tandis que celles qui correspondent à un «un» le sont en tirets. Le diagramme en treillis est plus instructif que le diagramme en arbre parcequ'il montre explicitement le fait que notre codeur est une machine à nombre d'états fini. Nous définissons l'état d'un codeur convolutif de rendement 1/n comme le nombre binaire formé des (K-1) plus

anciens symboles binaires présents dans son registre, le plus récent de ces symboles étant écrit en dernier.

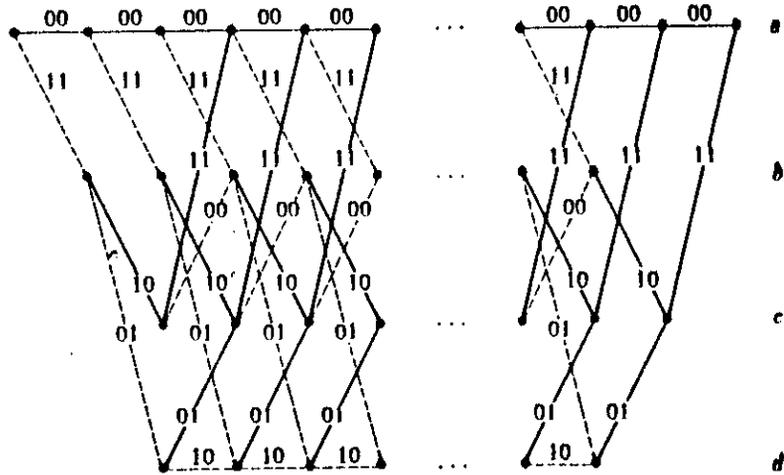


Figure 2-4 Diagramme en treillis associé au codeur de la Fig(2-2)

Le diagramme en treillis contient $(L+K)$ niveaux, où K est la longueur de contrainte du code et L est la longueur du mot message. Les premiers $(K-1)$ niveaux correspondent au départ du codeur de l'état initial (a). Nous remarquons aussi que nous ne pouvons pas atteindre tous les états dans cette partie du treillis.

Cependant, dans la partie centrale du treillis, pour laquelle le niveau j reste borné par : $K-1 \leq j \leq L$, tous les états du codeur peuvent être atteints.

Nous pouvons aussi noter que la partie centrale du treillis conserve une structure fixe et périodique. Considérons la portion du treillis qui correspond à la transition entre les deux niveaux j et $j+1$, avec: $j \geq 2$, de sorte que l'état le plus récent peut être l'un des quatre états (a), (b), (c) ou (d). Cette portion est donnée par la figure (2.5).

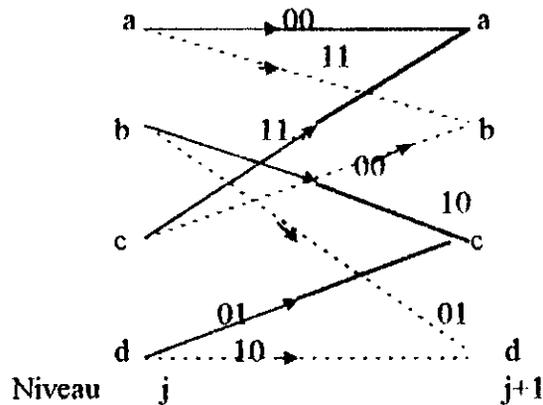


Figure 2-5 Portion centrale du treillis de la figure (2.4)

2-4-3 Diagramme d'état

En fusionnant les noeuds des niveaux j et $j+1$ de la figure (2-5), nous obtenons le digramme d'état représenté sur la figure (2-6).

Les noeuds de la figure représentent les quatre états possibles du codeur. Pour chaque noeud, on a deux branches entrantes et deux autres sortantes. La transition d'un état à un autre causée par l'entrée «zéro» est représentée par un trait plein, et la transition causée par l'entrée «Un» est représentée par des tirets.

Le nombre binaire sur chaque transition correspond à la sortie du codeur qui correspond à cette transition. Supposons par exemple l'état récent du codeur est (01), l'application à l'entrée du codeur d'un «Un», fait transiter celui-ci à l'état (11), et la sortie codée est (01).

A l'aide du diagramme d'état, on peut déterminer la sortie du codeur pour n'importe qu'elle entrée.

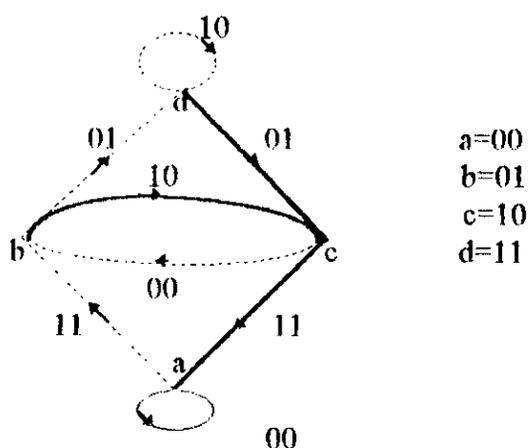


Figure 2-6 Diagramme d'état du codeur de la figure (2-2).

N.B. Ne pas confondre les états avec les sorties.

Considérons par exemple la séquence message (1,0,0,1,1). Pour cette entrée, on suit le chemin abcabd, d'où la séquence codée: (11,10,11,11,01).

Pour généraliser à des codes convolutifs à taux d'émission b/n , remarquons simplement que le diagramme en arbre comporte dans ce cas 2^b branches issues de chaque noeud. L'effet de la longueur de contrainte finie K est le même que ci-dessus, et donc au-delà des K premières branches, les chemins vont commencer à converger par groupe de 2^b , plus précisément, tous les chemins correspondant à $b(K-1)$ mêmes symboles de données convergent en un même noeud, conduisant à un treillis ayant $2^{b(K-1)}$ états, où les divergences en chaque noeud ont lieu par groupe de 2^b branches. K représente ici le nombre des b -uplets emmagasinés dans le registre du codeur. Par conséquent, le diagramme d'état aura aussi $2^{b(K-1)}$ noeuds, chacun ayant 2^b branches qui en partent et autant qui arrivent.

2-5 FONCTION GENERATRICE ET PROPRIETES DE DISTANCE DES CODES CONVOLUTIFS

Nous pouvons calculer l'ensemble des poids de Hamming associés aux chemins du code, ou de façon équivalente, l'ensemble des distances entre le chemin associé à la suite codée formée de tous les «zéros» et tous ceux qui s'en sont séparés, à l'aide du diagramme d'état. Pour simplifier les choses, considérons le même exemple traité depuis le début de ce chapitre, celui de la figure (2.2), avec $K = 2$, et $r = 1/2$.

Le diagramme d'état du codeur est montré sur la figure (2-6). Nous commençons par la modification de ce diagramme en un autre représenté par la figure (2.7). Nous scindons en deux états, l'état (a), un état initial(a_0) et un état final(a_1), et sa propre boucle est éliminée.

Un autre changement, c'est que chaque branche est dénommée. L'exposant de D sur les branches décrit le poids de Hamming de séquence de sortie correspondant à cette branche. L'exposant de I décrit le poids de Hamming de séquence d'entrée correspondante. Donc, pour l'entrée «Zéro» nous avons: $I^0 = 1$, et pour l'entrée «un», nous avons: $I^1 = I$. L'exposant de L est toujours égal à «un», correspondant en fait que la longueur de chaque branche est «un».

Par exemple, le chemin $a_0 \rightarrow b \rightarrow c \rightarrow a_1$ est noté par $D^2 L I^3$, c.à.d que le poids de Hamming de la séquence de sortie correspondante est 7, la longueur du chemin est 6, et le poids de Hamming de l'entrée est 3.

Nous définissons un chemin fondamental comme le chemin qui commence en (a_0) et se termine en (a_1). Soit $T_{d,l,i}$ le nombre de chemins de (a_0) vers (a_1) notés $D^d \cdot L^l \cdot I^i$. La fonction génératrice est définie comme étant:

$$T(D, L, I) = \sum_{d=1}^{\infty} \sum_{e=1}^{\infty} \sum_{i=1}^{\infty} T_{d,e,i} D^d L^e I^i \quad (3 - 9)$$

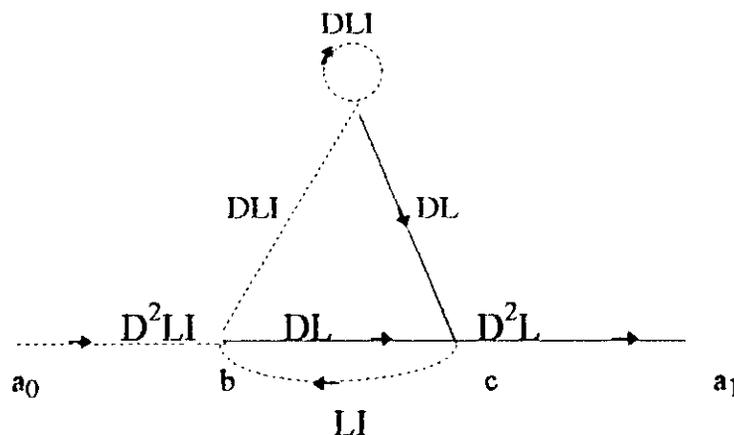


Figure 2-7 Diagramme d'état modifié

Pour le codeur de notre exemple, nous pouvons calculer la fonction génératrice, en examinant le diagramme d'état modifié comme étant un graphe de fluence à une seule entrée et une seule sortie.

La fonction génératrice peut être comme la fonction de transfert du graphe de fluence. Ainsi nous traitons les noeuds comme des jonctions de sommation et les notations de branches comme des gains, nous pouvons écrire:

$$\left. \begin{aligned} X_b &= D^2 L X_{a_0} + L X_c \\ X_c &= D L X_b + D L X_d \\ X_d &= D L X_b + D L X_d \\ X_{a_1} &= D^2 L X_c \end{aligned} \right\} \quad (3-10)$$

$$\text{et: } T(D,L,I) = X_{a1}/X_{a0} \quad (3.11)$$

En résolvant le système (3.10), on trouve

$$T(D,L,I) = \frac{D^5 \cdot L^3 \cdot I}{1 - D \cdot L \cdot I(1+L)} \quad (3.12)$$

En utilisant le développement en série de Maclaurin, on peut réécrire l'expression(3.12) en série de puissances:

$$T(D,L,I) = D^5 \cdot L^3 \cdot I + D^6 \cdot L^4 \cdot I^2 (1+L) + D^7 \cdot L^5 \cdot I^3 \cdot (1+L)^2 + \dots \quad (3.13)$$

Nous pouvons faire les remarques suivantes:

- Il existe un seul chemin fondamental à la distance 5 du chemin nul. Il diverge de celui-ci sur les trois branches qui précèdent le noeud à partir duquel il se confond avec lui. En plus, il diffère du chemin nul en un seul bit d'entrée.
- Il y a deux chemins fondamentaux à la distance 6 du chemin nul, dont l'un a divergé quatre branches en amont et l'autre cinq, et les deux diffèrent du chemin nul en deux bits d'entrée, et ainsi de suite.

En général, pour un code convolutif, la plus petite distance de Hamming de n'importe quel chemin fondamental par rapport au chemin nul est égale au distance limitée (Free distance) du code. Ainsi, le code convolutif, dont la fonction génératrice est donnée par (3.12) a la distance limite(d_{min}) qui est égale à 5; $d_{min} = 5$. Cela entraîne que deux erreurs quelconque peuvent être corrigées, puisque la suite reçue est à la distance 2 de la suite émise, à la distance 3 au moins de toute autre suite codée possible.

Dans le prochain chapitre, nous allons voir les deux algorithmes de décodage des codes convolutifs, l'algorithme de Viterbi, puis celui de Fano, ensuite nous parlerons de la probabilité d'erreurs de décodage, où la fonction génératrice joue un rôle très important dans la démonstration d'une borne supérieure.

CHAPITRE 3 DECODAGE DES CODES CONVOLUTIFS

3-1 Introduction

Il existe deux approches pour le décodage des codes convolutifs. La première est une approche déterministe dite décodage à logique majoritaire utilisé pour un type spécial de codes convolutifs [10]. La deuxième est l'approche probabiliste comprenant le codeur optimal de Viterbi, et des codeurs sous-optimaux dont le plus célèbre est celui de Fano. Dans ce qui suit nous exposerons l'algorithme de Viterbi et celui de Fano.

3-2 Décodage a vraisemblance maximale

Avant d'exposer la principe de l'algorithme de Viterbi, nous allons montrer que pour un canal binaire symétrique, le décodage à vraisemblance maximale se réduit au décodage à distance de Hamming minimale [14].

Soit X le mot-code émis à travers le canal discret sans mémoire, et soit Y le mot reçu qui peut être différent du mot émis à cause du bruit du canal. Le décodeur a pour rôle d'estimer le mot-code transmis X' en se basant sur le mot reçu Y .

Le procédé d'estimation de X' est dit optimal si la probabilité d'erreur de décodage est minimale. La fonction de vraisemblance maximale que doit maximiser le décodeur est $P(Y/X)$, probabilité conditionnelle de recevoir Y si X est émis.

Pour un canal binaire symétrique, Y et X sont deux séquences binaires de longueur N chacune. Soient X_i et Y_i les i èmes composantes de X et Y . Nous avons alors:

$$P(Y/X) = \prod_{i=1}^N P(y_i/x_i) \quad (3-1)$$

Si la distance de Hamming entre X et Y est d , et si l'on considère comme c'est le cas dans le canal binaire symétrique, alors (3-1) devient:

$$P(Y/X) = p^d \cdot (1-p)^{n-d} \quad (3-3)$$

En général p est petite, et peut être négligée devant 1, d'où:

$$P(Y/X) \approx p^d \quad (3-4)$$

Si X' est un autre mot-code, distant de Y de d' , et qui n'est pas choisi comme estimé de Y . Donc la fonction de vraisemblance de X' est inférieure à celle de X .

$$P(Y/X') < P(Y/X) \quad (3-5)$$

En tenant compte que: $P(Y/X') \approx p^{d'}$

(3-5) implique: $d' > d$; ($P < 1$).

On conclut que maximiser la vraisemblance revient à minimiser la distance de Hamming. La fonction du décodeur sera donc:

« Choisir l'estimé X' qui minimise la distance de Hamming entre X et Y ».

3-3 Algorithme de Viterbi

Considérons l'exemple du code convolutif étudié au chapitre 3 , dont la représentation en treillis est donnée par la figure (2-4) le décodage à vraisemblance maximale consiste à choisir le chemin correspondant à la séquence binaire qui diffère de la séquence reçue en un nombre minimum de bits.

L'algorithme peut être partagé en deux parties. La première s'étend du début du treillis jusqu'au niveau $j=K-1$, où K est la longueur de contrainte. Dans cette partie, le treillis est identique à l'arbre, à chaque noeud arrive une seule branche. Le décodage consiste à attribuer une distance nulle à l'état initial, et attribuer à chaque autre état X , atteint par transition d'un état Y , une distance qui est la somme de la distance à l'état Y et la distance de transition.

La distance de transition, à un niveau i du treillis entre deux états Y et X , est la distance de Hamming entre les n bits correspondant à cette transition du codeur et n bits de la séquence reçue (allant du bit numéro $((i-1).n+1)$ au bit numéro $(i.n)$).

La deuxième partie s'étend du niveau $j=K$ jusqu'au niveau $j=L+M$, où L est la longueur du message et M la mémoire du codeur (égale à $K-1$, K longueur de contrainte). Dans cette partie à chaque état X arrivent 2^b branches. Le décodage consiste à choisir la branche pour laquelle la distance au noeud X 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 X , alors peu importe on choisira une de ces branches . Cette opération est répétée avec tous les $2^{b(k-1)}$ états, et cela pour tous les niveaux de cette partie. Enfin de cette partie nous obtiendrons $2^{b(k-1)}$ chemins, desquels le codeur choisira celui qui a la distance la plus petite. La séquence correspondante à ce chemin est la plus proche, du sens de Hamming, de la séquence émise.

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

1. Partir du k ième étage (niveau) du diagramme où 2^b branches arrivent à chaque état. Calculer pour chacun des états la distance de chacun des 2^b chemins qui arrivent à l'état. Pour chaque état, conserver le chemin dont la distance est la plus faible, (le survivant) et éliminer les autres .
2. Répéter l'opération pour chaque étage j du treillis tant que $j \leq (L/b) + M$.
3. Choisir le chemin dont la distance est la plus petite, et éliminer tous les autres .

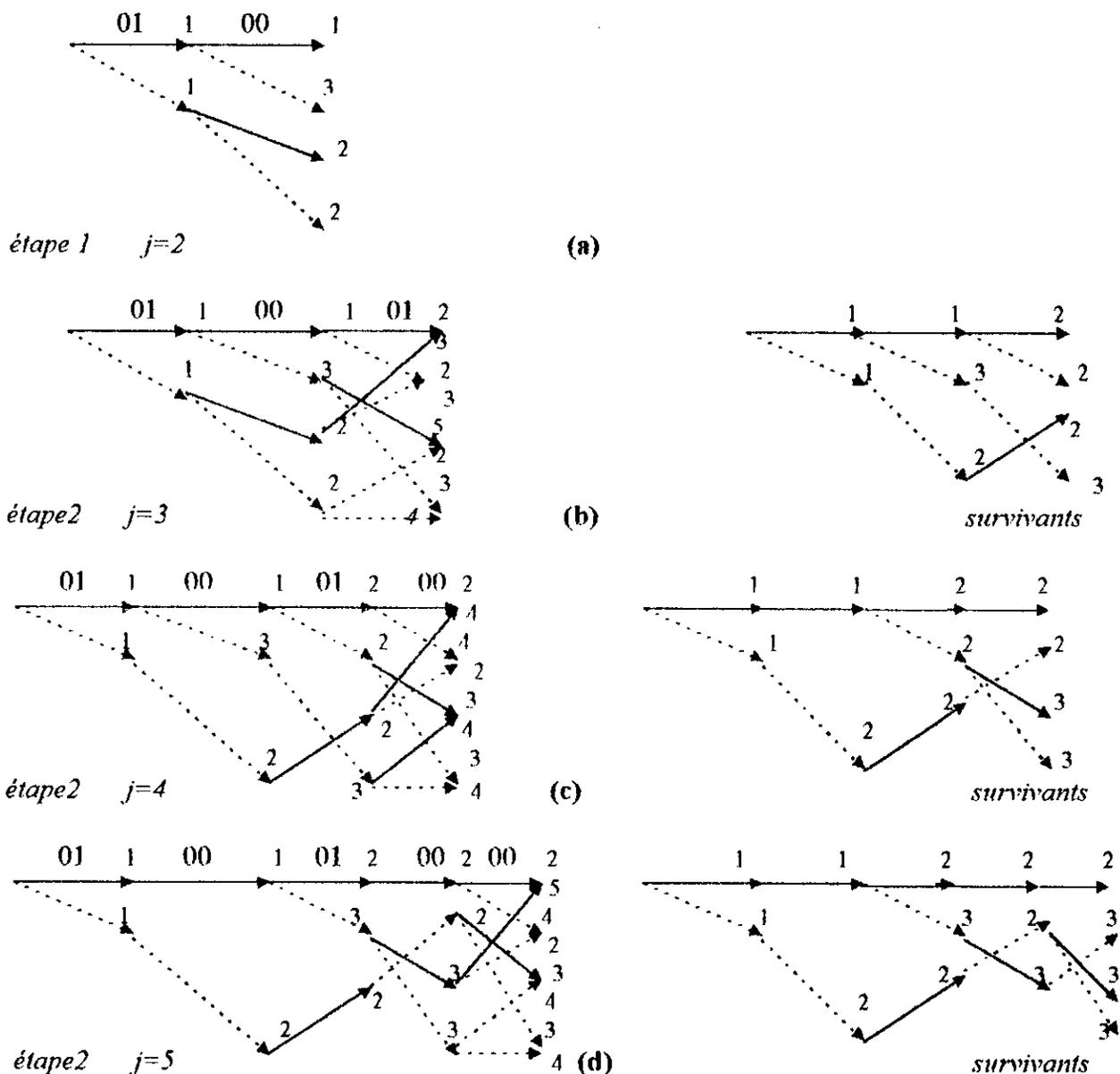


Figure 3-1 Exemple de decodage utilisant l'algorithme de Viterbi.

L'avantage de cet algorithme est que le nombre d'opérations, égal à $L2^{n(K-1)}$ [8], est linéaire en L. Cependant ce nombre présente une fonction exponentielle en K. En notant aussi, qu'il est nécessaire de stocker les $2^{b(K-1)}$ chemins jusqu'à la fin, où la décision est prise, on se rend compte que le décodeur devient très complexe et nécessite une capacité mémoire importante pour des K importants.

Exemple 3-1

Revenons à notre codeur de la figure(2-2). Supposons que ce dernier génère une séquence nulle de n bits(tous zéros), laquelle après transmission à travers le canal donne la séquence reçue (0100010000) comprenant deux erreurs. La figure (3-1-a) montre la première partie du décodage. La distance à chaque état y est indiquée.

La figure(3-1) b, c, d montre la deuxième partie. Chacune de ces figures compte deux treillis, dans celui de gauche on a conservé tous les chemins aboutissant à chaque noeud, tandis que dans celui de droite on n'a gardé que les survivants.

Le chemin conservé après la décision finale est celui dont l'erreur cumulée est la plus petite c.à.d. le chemin (11111...) correspondant à la séquence (tous zéros).

3-4 DECODAGE SEQUENTIEL

L'inconvénient avec le décodage à vraisemblance maximale est que le décodeur devient très complexe, et nécessite une capacité mémoire énorme pour K important. Sachant que la distance minimale augmente avec K [15], on se rend compte qu'en utilisant l'algorithme de Viterbi, la performance du code est limitée par l'espace mémoire que peut avoir le décodeur.

La solution est l'utilisation des algorithmes de décodage séquentiel qui réduisent beaucoup la complexité du décodage et qui devient dépendante linéairement de K[5].

Le décodage séquentiel est une technique de recherche intuitive, par essai et erreur, du chemin le plus vraisemblable dans l'arbre représentant le code. Durant cette recherche, le décodeur se déplace en avant ou en arrière d'un noeud à chaque étape.

La décision d'avancer ou de reculer est prise suivant le résultat de comparaison de la métrique du chemin, variant le long du chemin, avec un seuil lui aussi variable. Dans ce qui suit nous exposerons l'algorithme de Fano, mais avant nous introduirons la métrique utilisée dans cet algorithme.

3-4-1 La métrique de Fano

Considérons un code convolutif de longueur de contrainte K et de débit b/n. Soit C_{ij} le ième bit du mot de n bits associés à la ième branche d'un chemin quelconque dans le diagramme en arbre. Soit Y_{ij} le bit correspondant dans la séquence reçue à ce bit est définis par [8]:

$$\gamma_{ij} = \text{Log}_2 \left[\frac{P(y_{ij} / c_{ij})}{P(y_{ij})} \right] - r \quad (3-6)$$

où: $P(Y_{ij}/C_{ij})$ est la probabilité de transition du canal, et $P(Y_{ij})$ la probabilité d'avoir Y_{ij} en sortie. La métrique de la branche est:

$$\gamma_j = \sum_{i=1}^L \gamma_{ij} = \sum \left[\text{Log}_2 \left(\frac{P(y_{ij} / c_{ij})}{P(y_{ij})} \right) \right] \quad (3-7)$$

γ_j est une valeur entière [8]. Si le chemin comprend L branches, la métrique du chemin est:

$$\Gamma(L) = \sum_{j=1}^L \gamma_j = \sum_{j=1}^L \sum_{i=1}^n \left[\text{Log}_2 \left(\frac{P(y_{ij}/c_{ij})}{P(y_{ij})} \right) \right] \quad (3-8)$$

Pour un chemin de (L+1) branches, on a:

$$\Gamma(L+1) = \sum_{j=1}^{L+1} \gamma_j = \sum_{j=1}^L \gamma_j + \gamma_{L+1} \quad (3-9)$$

$$\Gamma(L+1) = \Gamma(L) + \gamma_{L+1}$$

La dernière expression représente une méthode efficace pour l'implémentation du calcul de métrique.

3-4-2 Algorithme de Fano

Considérons le diagramme en arbre de la figure(2-3), vu au chapitre 2. Supposons que le décodeur se situe dans un noeud quelconque de l'arbre. A partir de ce noeud, le décodeur regarde en avant dans l'arbre, s'il trouve un noeud dont la métrique est supérieure ou égale au seuil courant, il se déplace à ce noeud. Sinon il revient en arrière d'un noeud et essaye d'avancer dans une autre branche de l'arbre.

Le seuil courant T_L est défini comme étant un multiple du seuil de référence Δ qui est un paramètre fixé par le concepteur du décodeur. Lorsque Δ est très petit, le décodeur revient souvent en arrière, même s'il est sur le bon chemin. D'autre part, si Δ est trop grand, les chemins incorrects ne sont identifiés rapidement. Il faut, donc, faire un compromis lors du choix de Δ pour minimiser les calculs.

Pour montrer le mécanisme avec lequel le décodeur se déplace en avant ou en arrière dans l'arbre, considérons la figure (3-2), qui représente une partie du diagramme en arbre d'un code convolutif 1/n.

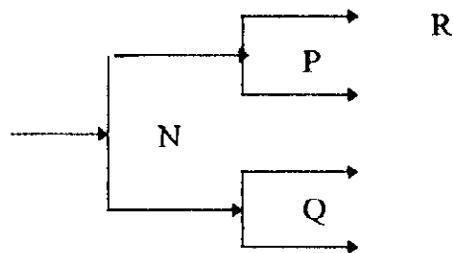


Figure 3-2 Une partie du diagramme en arbre.

Supposons que le décodeur a atteint le point P après avoir parcouru L branches de l'arbre. La métrique en P est $\Gamma_P(L)$ et le seuil est T_L . Du point P le décodeur regarde devant, cherchant le meilleur des noeuds, c'est à dire celui dont la métrique est la plus grande.

Soit R ce noeud et $\Gamma_R(L+1)$ la métrique correspondante. Si $\Gamma_R(L+1) \geq T_L$ le décodeur se déplace en R. si R est visité pour la première fois le seuil est augmenté par le plus grand multiple

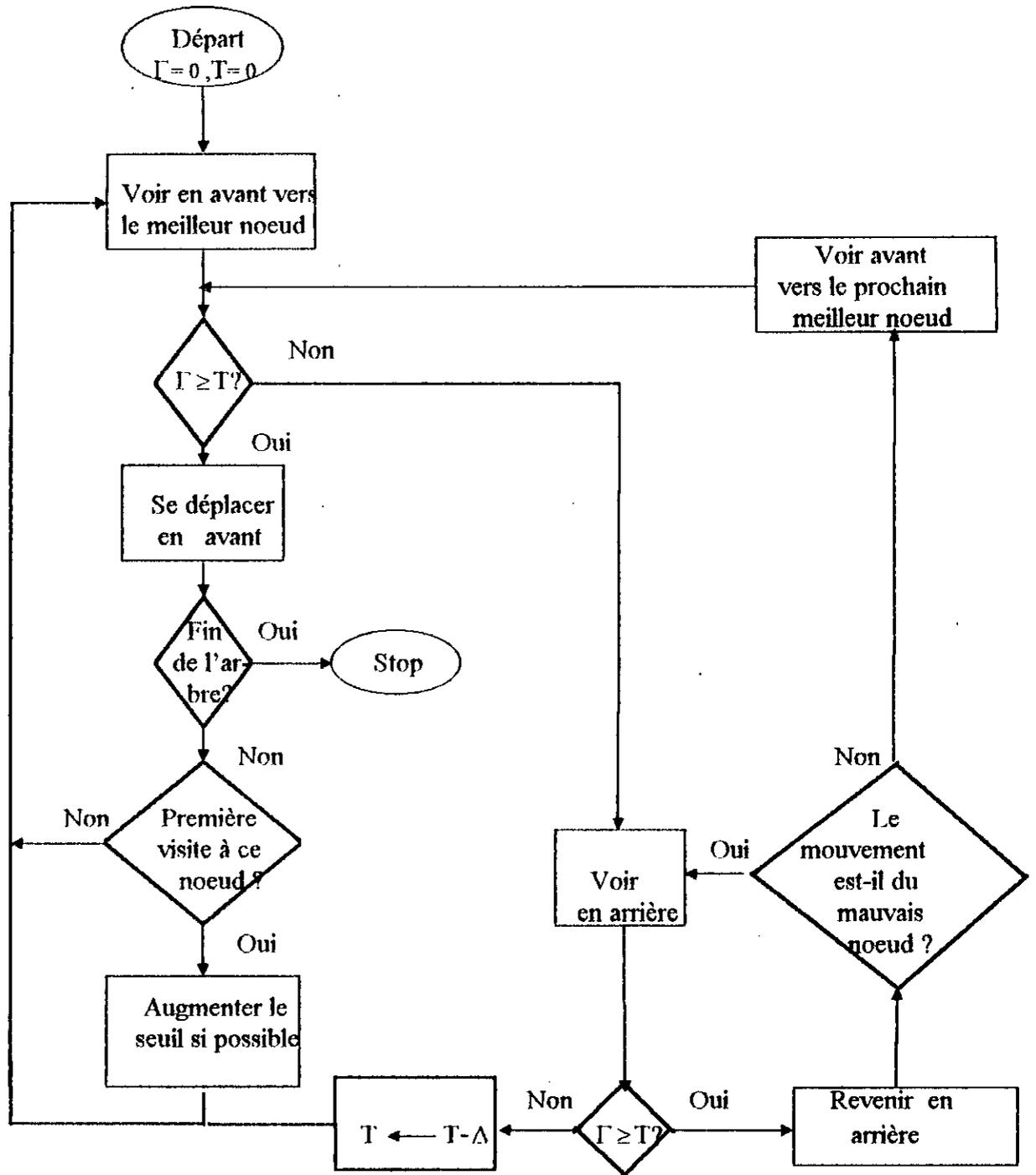


Figure 3-3 Organigramme de l'algorithme de Fano de Δ tel que T_L reste inférieur ou égal à $\Gamma_R(L+1)$. Si le décodeur est déjà passé par R le seuil reste inchangé.

Cependant si $\Gamma_R(L+1) < T_L$, le codeur regarde en arrière vers le noeud N. Soit $\Gamma_N(L+1)$ la métrique en N. Si $\Gamma_N(L+1) \geq T_L$, le codeur se déplace en N. De N le décodeur essaye de se déplacer, si c'est possible, en avant vers le point Q, qui rappelons le est moins bon que P. Si le déplacement vers Q n'est pas possible ($\Gamma_Q(L) < T_L$) alors le décodeur regarde de nouveau en arrière au niveau L-2.

Si $\Gamma_N(L+1) < T_L$, le décodeur ne peut ni avancer ni rebrousser chemin. Dans ce cas le seuil T_L est diminuer de Δ , puis le décodeur essaye de se déplacer de nouveau vers R avec un seuil plus petit. Les conditions initiales de cet algorithme sont:

- A l'état initial, la métrique est nulle: $\Gamma(0) = 0$, et par convention $\Gamma(-1) = -\infty$, ce qui signifie que le décodeur ne pourra jamais se déplacer en arrière dès le départ.

-Le seuil initial est égal à zéro: $T_0 = 0$.

L'organigramme de cet algorithme est donné par la figure (3-3). Deux noeuds spéciaux y sont mentionnés: le « meilleur » noeud et le « mauvais » noeud. Vu à partir d'un noeud particulier de l'arbre, le « meilleur » noeud est celui des deux noeuds profonds d'une branche dont la métrique associée est la plus grande. Le « mauvais » est celui dont la métrique associée est la plus faible.

Le décodeur de Fano, en plus des calculs qu'il doit effectués dans sa recherche, doit stocker la séquence reçue et une copie exacte du codeur.

Exemple 3-2 [8]

Pour illustrer l'algorithme de Fano, considérons le diagramme en arbre de la figure(3-4) correspondant à un code convolutif de longueur de contrainte $K = 7$ et de débit $r = 1/2$ et de nombre de bits message $L = 2$. La figure comporte aussi la métrique du chemin sur chaque noeud de l'arbre. Ces valeurs ont été calculées pour la séquence transmise(00,00,00,00) et la séquence reçue(10,00,01,00). Le seuil de référence est $\Delta = 4$.

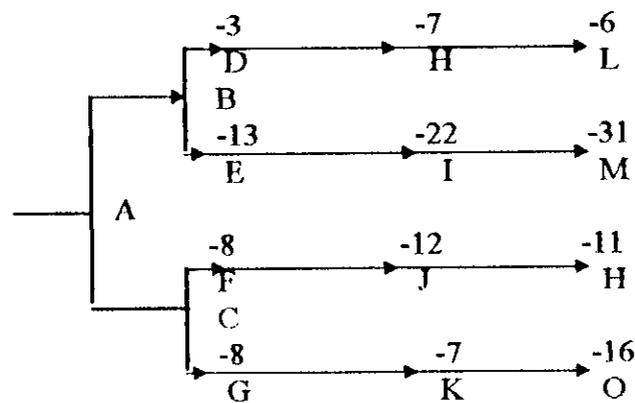


Figure 3-4 Diagramme en arbre pour code convolutif 1/2.

Le tableau (3-1) donne le comportement de l'algorithme. Pour chaque étape le tableau donne le noeud où se situe le décodeur, la métrique correspondante et le seuil courant T

Le codeur commence à l'origine, noeud A de l'arbre avec $\Gamma = 0$ et $T = 0$. Il cherche à se déplacer ou bien en B ou en C. Puisque leurs métriques sont égales et inférieures à T, le décodeur reste en A avec diminution du seuil $T = -4$. Le décodeur peut maintenant choisir l'un quelconque des deux noeuds B ou C.

Dans cet exemple, le décodeur choisit C. Les autres étapes montrent les déplacements en avant et en arrière du codeur, jusqu'à la dernière étape où il arrive en L, et le chemin choisi est A B D H L.

Tableau 3-1 décodage de Fano

Etape	Noeud	Métrique, Γ	Seuil, T
1	A	0	0
2	A	0	-4
3	C	-4	-4
4	A	0	-4
5	B	-4	-4
6	D	-3	-4
7	B	-4	-4
8	A	0	-4
9	A	0	-8
10	C	-4	-8
11	G	-8	-8
12	K	-7	-8
13	G	-8	-8
14	C	-4	-8
15	A	0	-8
16	B	-4	-8
17	D	-3	-8
18	H	-7	-8
19	L	-6	-8
20	STOP		

3-4 PERFORMANCES DES CODES CONVOLUTIFS

On sait bien que la performance d'un code dépend des distances relatives entre les signaux binaires que peut générer le codeur.

Ces distances peuvent être obtenues du diagramme d'état du code. La structure et la complexité de ce diagramme dépend fortement de la longueur de contrainte K , mais pas du tout de la longueur des blocs.

Il apparaît donc que la probabilité d'erreur par bloc n'est pas une mesure pertinente de la qualité de transmission, particulièrement dans le cas, fréquent, où tout le message subit un codage convolutif en bloc unique. C'est la probabilité d'erreur par symbole P_e qui apparaît comme la mesure la plus utile de la qualité.

On la définit comme étant le rapport du nombre moyen attendu de symboles erronés dans une séquence décodée au nombre total de symboles de cette même séquence. Notre but, donc est de borner supérieurement P_b , mais nous considérons en premier une mesure de la qualité plus accessible, qui est la probabilité d'erreur en un noeud, notée P_e .

Le schéma de la figure (3-5) représente deux chemins du treillis du code. Sans perdre de généralité, nous pouvons prendre pour chemin correct celui du haut, et supposer que celui du bas est choisi par le décodeur de vraisemblance maximale.

Pour que cela se produise, il faut que l'accroissement de la métrique du chemin correct, dans la partie où les deux chemins sont distincts, soit plus faible que celui du chemin incorrect considéré. Nous dirons alors qu'il y a erreur aux noeuds i et j . D'où une condition nécessaire, mais non suffisante, pour qu'une erreur ait lieu au noeud j est qu'un chemin incorrect, se séparant en j du chemin correct, ait dans la partie où les deux chemins sont distincts un accroissement de métrique supérieur à celui du chemin correct.



Figure 3-5 Exemple de cas d'erreur.

Nous pouvons donc borner supérieurement la probabilité d'erreur au noeud j par la probabilité qu'un chemin se séparant en j du chemin correct, subisse un accroissement de métrique supérieur à celui du chemin correct, dans la partie où les deux chemins sont distincts

$$P_e(j) \leq \Pr \left[\bigcup_{x'_j \in X'(j)} \{ \Delta M(x'_j, x_j) \geq 0 \} \right] \quad (3-10)$$

Où x'_j est un chemin incorrect se séparant au noeud j du chemin correct x_j , $X'(j)$ est l'ensemble de tous les chemins de ce type, $\Delta M(x'_j, x_j)$ est la différence des accroissements de la métrique des deux chemins x'_j et x_j dans la partie où ils sont distincts.

$$(3-10) \text{ donne: } P_e(j) \leq \sum \Pr[\Delta M(x'_j, x_j) \geq 0] \quad (3-11)$$

Chaque terme de cette sommation est la probabilité d'erreur pour un couple de vecteurs du code dans la partie où les deux chemins sont distincts. Pour un canal binaire à l'entrée et si la distance entre x'_j et x_j est d , la borne de la probabilité d'erreur pour la paire de vecteurs x'_j et x_j , selon Bhattacharyya[15], est donnée par:

$$P_d \leq \exp \left[d \ln \sum_y \sqrt{P(y/0)P(y/1)} \right] \quad (3-12)$$

Où Y désigne le symbole reçu, et $P(Y/i)$, $i \in \{0,1\}$, est la probabilité de transition.

$$\text{Si on pose: } D_0 = \sum_y \sqrt{P(y/0)P(y/1)}$$

$$(3-12) \text{ devient: } P_d \leq D_0^d \quad (3-13)$$

S'il existe $a(d)$ chemins incorrects à la distance de Hamming d du chemin correct (Dans la partie où ils diffèrent), nous tirons de (3-13) et (3-10):

$$P_e(j) \leq \sum_{d=d_{\min}}^{\infty} \Pr \{ \text{erreur provoqué e par l'un quelconque de } a(d) \text{ chemins incorrects à la distance } d \}$$

$$P_e(j) \leq \sum_{d=d_{\min}}^{\infty} a(d) P_d \leq \sum_{d=d_{\min}}^{\infty} a(d) D_0^d \quad (3-14)$$

Où d_{\min} est la distance limite (Free distance).

Nous avons vu au chapitre précédent que l'ensemble des distances entre l'un quelconque des chemins et tous les autres se déduisaient de la fonction génératrice $T(D)$, L , I étant pris égales à 1. Pour illustrer cela, considérons l'exemple correspondant à la figure (2-2). Nous avons alors:

$$T(D) = \frac{D^5}{1-2D} = D^5 + 2D^6 + \dots + 2^{k-5} D^k + \dots = \sum_{d=5}^{\infty} 2^{d-5} D^d$$

C'est à dire que dans ce cas $d_{\min} = 5$ et $a(d) = 2^{d-d_{\min}}$

$$T(D) = \sum_{d=d_{\min}}^{\infty} a(d)D^d \quad (3-15)$$

La borne (3-14) de probabilité d'erreur en un noeud est indépendante du noeud considéré. Si la séquence est supposée infinie, ce que nous ferons dans toute la suite. Nous remarquons que le nombre d'erreurs sur les symboles, dû au choix d'un chemin qui se sépare du chemin correct en j , peut être borné en pondérant chaque terme de l'expression (3-11) par le nombre des erreurs sur les symboles d'information qui surviennent lorsque le chemin correspondant est choisi.

En supposant que le chemin correct est le chemin nul (Sans perdre de généralité pour les canaux binaires symétriques), ce facteur de pondération devient le nombre des « un » dans la suite de données correspondant à l'intervalle où le chemin incorrect est distinct du chemin nul. On obtient pour le nombre de symboles erronés correspondant au chemin incorrect à partir de j la borne suivante.

$$E[n_b(j)] \leq \sum_{i=1}^{\infty} \sum_{d=d_{\min}}^{\infty} ia(d,i)P_d \leq \sum_{i=1}^{\infty} \sum_{d=d_{\min}}^{\infty} ia(d,i)D_0^d \quad (3-16)$$

Où $a(d,i)$ est le nombre de chemins distincts du chemin nul à partir de j , qui a le poids d et correspondent à i symboles d'information dans l'intervalle où ils diffèrent du chemin nul. Mais les coefficients $a(d,i)$ ne sont autres que les coefficients de $D^d I^i$ dans la fonction génératrice généralisée, vue au chapitre précédent, $T(D,I) = T(D,L,I)$, avec $L=1$.

$$T(D,I) = \frac{D^5 I}{1-2DI} = D^5 I + 2D^6 I^2 + \dots + 2^{d-5} D^d I^{d-4} + \dots = \sum_{d=5}^{\infty} 2^{d-5} D^d I^{d-4}$$

et donc:
$$a(d,i) = \begin{cases} 2^{d-5} & ; \text{ Si } i = d - 4, d \geq 5 \\ 0 & ; \text{ Sinon} \end{cases}$$

et on a alors:
$$E[n_b(j)] \leq \sum_{d=5}^{\infty} (d-4) 2^{d-5} D_0^d = \left. \frac{\partial T(D,I)}{\partial I} \right|_{I=1, D=D_0}$$

Il est clair que l'on peut en général développer la fonction génératrice généralisée $T(D,I)$ sous forme:

$$T(D,I) = \sum_{i=1}^{\infty} \sum_{d \geq d_i} a(d,i) D^d I^i \quad (3-17)$$

Et que sa dérivée partielle par rapport à I, pour I=1:

$$\left. \frac{\partial \Gamma(D, I)}{\partial I} \right|_{I=1} = \sum_{i=1}^{\infty} \sum_{d=1}^{\infty} i a(d, i) D^d I^i \quad (3-18)$$

A partir de (3-18) et (3-16) on obtient:

$$E\{n_b(j)\} \leq \left. \frac{\partial \Gamma(D, I)}{\partial I} \right|_{I=1, D=D_0} \quad (3-19)$$

borne supérieure du nombre moyen des erreurs sur les symboles de données dues à un chemin correct à partir de j.

Pour un code 1/n chaque noeud ou branche correspond à un symbole d'information. L'éventualité que le symbole associé au noeud j soit erroné est la réunion des éventualités d'un cas d'erreur à partir du noeud j ou antérieurement, tel qu'une erreur ait lieu en j.

La probabilité $P_b(j)$ que le symbole d'information associé au noeud j soit décodé incorrectement est borné par la somme des probabilités de ces éventualités.

$$P_b(j) \leq E\{n_b(j)\} \leq \left. \frac{\partial \Gamma(D, I)}{\partial I} \right|_{I=1, D=D_0} \quad (3-20)$$

Pour un code de taux b/n nous avons:

$$P_b(j) \leq \frac{E\{n_b(j)\}}{b} \leq \frac{1}{b} \left. \frac{\partial \Gamma(D, I)}{\partial I} \right|_{I=1, D=D_0} \quad (3-21)$$

Exemple 3-3

Pour le canal binaire symétrique:

$$D_0 = 2 \sqrt{p(1-p)}$$

où p est la probabilité de transition.

Pour le canal AWGN:

$$D_0 = e^{-\xi_s / N_0}$$

Où le ξ_s / N_0 est le rapport signal sur bruit. Si le canal AWGN est converti en canal binaire symétrique par décision ferme en sortie, pour $\xi_s / N_0 \ll 1$, on a :

$P \approx 1/2 - (\xi_s / \pi N_0)^{1/2}$, d'où:

$$-\text{Ln}D_0 \approx -\text{Ln}\left[\sqrt{1 - 4(\xi_s / \pi N_0)}\right] \approx -\text{Ln}\left[1 - \frac{2 \xi_s}{\pi N_0}\right] \approx \frac{2 \xi_s}{\pi N_0}$$

C'est à dire que la convention du canal AWGN en canal binaire symétrique entraîne une perte de $2/\pi$, soit approximativement 2dB, en rapport signal sur bruit.

3-6 CONCLUSION

Pour des canaux, qui ne sont pas trop bruités, des codes convolutifs à des longueurs de contraintes petites sont suffisants pour projeter l'information. Dans ce cas l'algorithme de Viterbi est le meilleur choix pour le décodeur, puisque la complexité de l'implantation ne sera pas importante. Mais dès qu'une meilleure performance est exigée, canaux très bruité, l'utilisation de cet algorithme donnera un décodeur très complexe avec capacité de stockage exigée, énorme.

Cela est dû au fait qu'augmenter la performance revient à avoir des longueurs de contraintes importantes. Dans ce cas l'utilisation de décodage séquentiel est plus souhaitable, à cause de la simplicité d'implantation par rapport au décodeur de Viterbi, et cela même s'il sont moins bons que ces derniers.

Remarquons, en fin que l'utilisation des codes par blocs ou convolutifs entraîne une augmentation de la bande fréquence du signal. Si on veut utiliser efficacement la bande de fréquence disponible sur le canal, et optimiser même l'énergie transmise, la modulation et le codage doivent être fait en même temps. C'est la technique appelée modulation avec codes en treillis(trellis-coded modulation) qui sort du cadre de cet exposé.

CHAPITRE 4

SIMULATION, RESULTATS ET INTERPRETATIONS

4-1 Introduction

La simulation permet l'évaluation des performances des systèmes de communication, et constitue aussi un outil puissant qui aide à la conception de tels systèmes . Cette évaluation des performances se fait en calculant la probabilité d'erreurs pour un système de communication donné (figure (1 - 1)) et un bruit donné.

4-2 Modèle d'étude

Le modèle utilisé lors de notre simulation est représenté sur la figure (4 - 1) . Il comporte une source d'information binaire, un codeur convolutif de canal, un canal binaire symétrique supposé idéal (sans distorsions), une source de bruit, un décodeur de canal et enfin un comparateur pour calculer le nombre de bits erronés après le décodage. Ce nombre divisé par le longueur de la séquence donne la probabilité d'erreur P_e (voir la figure(4 -11)). Le codeur et le décodeur sont déjà détaillés dans les précédents chapitres. Nous allons voir maintenant les sources d'information et de bruit.

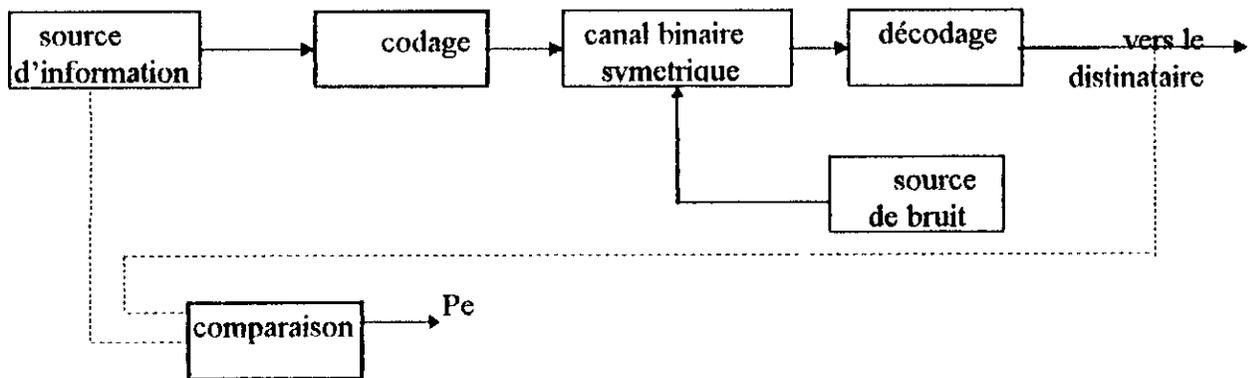


Figure 4-1 Modèle de simulation.

4-2-1 Source d'information

La séquence d'information est simulée par une séquence binaire pseudo-aléatoire [4] présentant des propriétés d'autocorrélation proches de celles d'une séquence aléatoire pure (bruit blanc). Le générateur d'une telle séquence est constitué d'un registre à décalage comportant m étages, et des additionneurs modulo-2 (figure(4 - 2)). Il génère une séquence périodique. Le choix de m détermine la période de la séquence, qui est égale à $2^m - 1$. Il est généralement préférable de générer une séquence très longue, donc d'utiliser un m grand.

Les connexions aux additionneurs modulo - 2 sont déterminées par des polynômes primitifs de degré m ayant la forme:

$$h(x) = \sum_{j=0}^m h_j x^j \quad \text{avec} \quad \begin{cases} h_0 = h_m = 1 \\ h_j = 0 \text{ ou } 1 \text{ pour } j = 1, \dots, m-1 \end{cases}$$

Un polynôme primitif de degré m est un polynôme irréductible qui divise $x^{2^m-1} + 1$ et ne divise pas $x^n + 1$ pour $n < 2^m - 1$.

Chaque coefficient h_j égal à 1 correspond à une connexion.

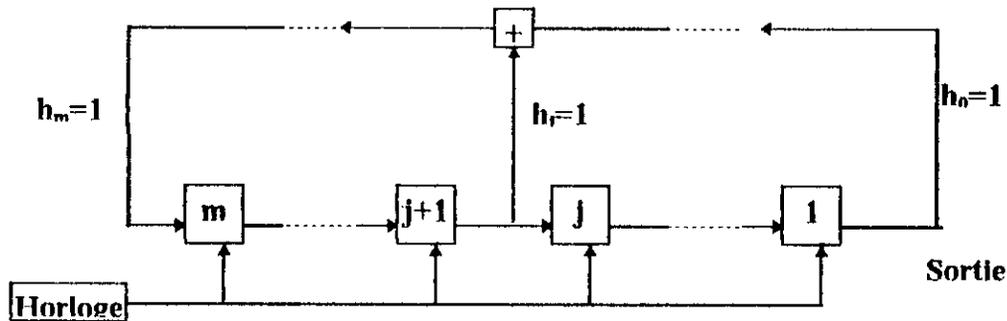


Figure 4-2 Générateur pseudo-aléatoire.

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

4-2-2 Source de bruit

Elle est simulée par une séquence aléatoire de même longueur que la séquence d'information codée, mais dont le pourcentage P des 1 est contrôlé. Cette séquence additionnée modulo-2 à la séquence codée produira une séquence codée dont $P\%$ des éléments sont infectés par du bruit, donc erronés.

Soit une séquence d'information codée de longueur L , qu'on veut bruite avec un pourcentage P . Nous devons donc générer une séquence de bruit de longueur L dont $P\%$ des bits est égal 1 et qui sont répartis d'une manière aléatoire. Pour cela on cherche H , l'entier le plus proche de $2PL/100$ et qui est de la forme 2^m-1 . Ensuite on suit les étapes suivantes:

- 1- Générer, de la même manière que pour la séquence d'information, une séquence $S1$ de longueur H . $S1$ a à peu près $PL/100$ de 1 et $PL/100$ de 0.
- 2- Si $H'=2(H+1)-1$ (égal à $2^{m+1}-1$) est inférieur ou égal à L alors aller à l'étape 3, sinon aller à l'étape 6.
- 3- Générer une séquence de longueur H' de la même manière que pour la séquence d'information.
- 4- Disperser aléatoirement les 1 de $S1$ sur une longueur égale à H' de la manière suivante:

- $j=1$.
- Pour $i=1$ jusqu'à H' lire $S2(i)$:
 - si $S2(i)$ est égal à 1 alors $S3(i)= S1(j)$ et $j=j+1$.
 - sinon $S3(i)= 0$.
- Ecraser $S1$ et renommer $S3,S1$.

5- $H=H'$ et on revient à l'étape 2.

6- Si H est ^{supérieur} inférieur à L , compléter la séquence de longueur H avec des 0 jusqu'à avoir une séquence de longueur L , sinon H est égal à L et $S1$ est la séquence désirée.

Cette démarche donne de bons résultats pour L de la forme 2^m-1 et $P \in A=\{50, 50/2, 50/3, 50/4 \dots\}$. Remarquons qu'on peut aussi avoir toutes les valeurs $P' > 50$ du pourcentage telqu'il existe $P \in A$ vérifiant : $P' = 100 - P$. Cela s'obtient en générant une séquence de bruit, de pourcentage $p < 50$, puis inverser les bits de cette séquence.

4-2-3 Calcul du rapport signal sur bruit

Soit une séquence d'information codée de longueur L dont $P\%$ de ses bits sont contaminés par le bruit. La période du signal est $T= L\tau$, avec τ la durée d'un bit. Si on représente les 1 par une valeur A et les 0 par la valeur $-A$, alors la puissance moyenne du signal utile est:

$$S = \frac{1}{T} \int_0^T A^2 dt = A^2.$$

L'énergie associée à un bit est:

$$e = A^2 \tau.$$

Le bruit affecte $PL/100$ bits de la séquence d'information et donc la puissance moyenne du bruit sur une période est $ePL/100$. Sa puissance est donc:

$$N = \frac{ePL / 100}{T} = \frac{A^2 P}{100}$$

Le rapport signal sur bruit est, donc, donné par:

$$S/N = \frac{100}{P}$$

En dB ce rapport est:

$$S/N(\text{dB}) = 10 \log_{10} \left(\frac{100}{P} \right).$$

4-3 Résultats et commentaires

Les figures 4 -3,4-4 et 4-5 représentent la probabilité d'erreurs P_e en fonction du rapport signal sur bruit (S/N) pour les codes convolutifs de débits $1/2, 1/3, 1/4$ de longueurs de contrainte égales à 3, et cela en utilisant l'algorithme de Viterbi.

La figure 4 -6 donne P_e en fonction de S/N pour le codeur de débit $1/2$ de longueur de contrainte égale à 5, en utilisant toujours l'algorithme de Viterbi.

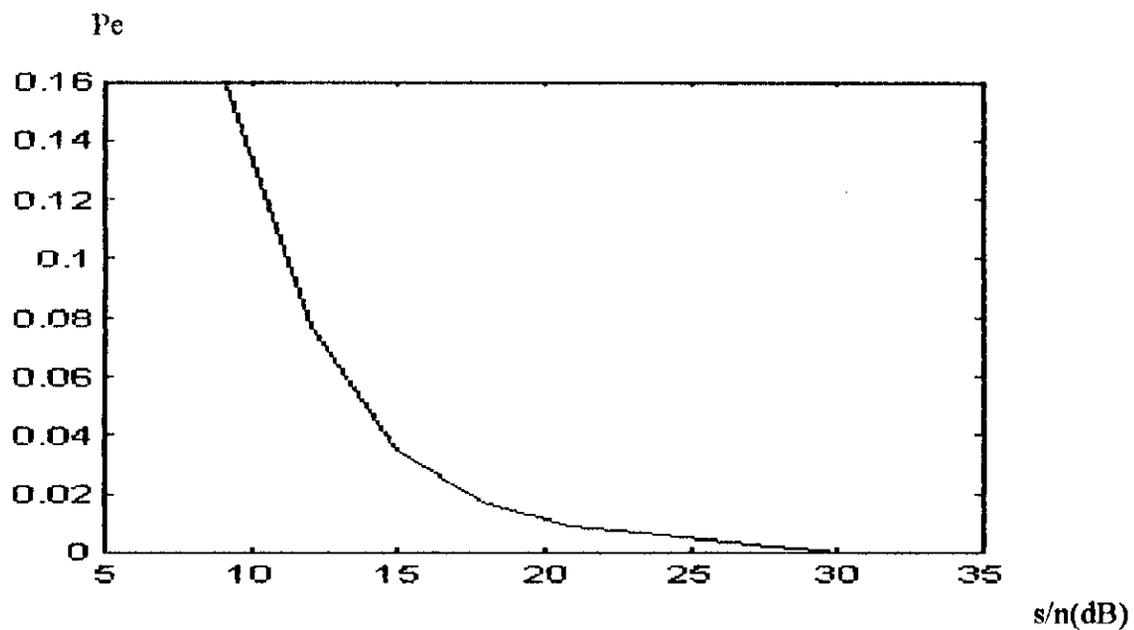


Figure 4 3 codeur 1/2,k=3.

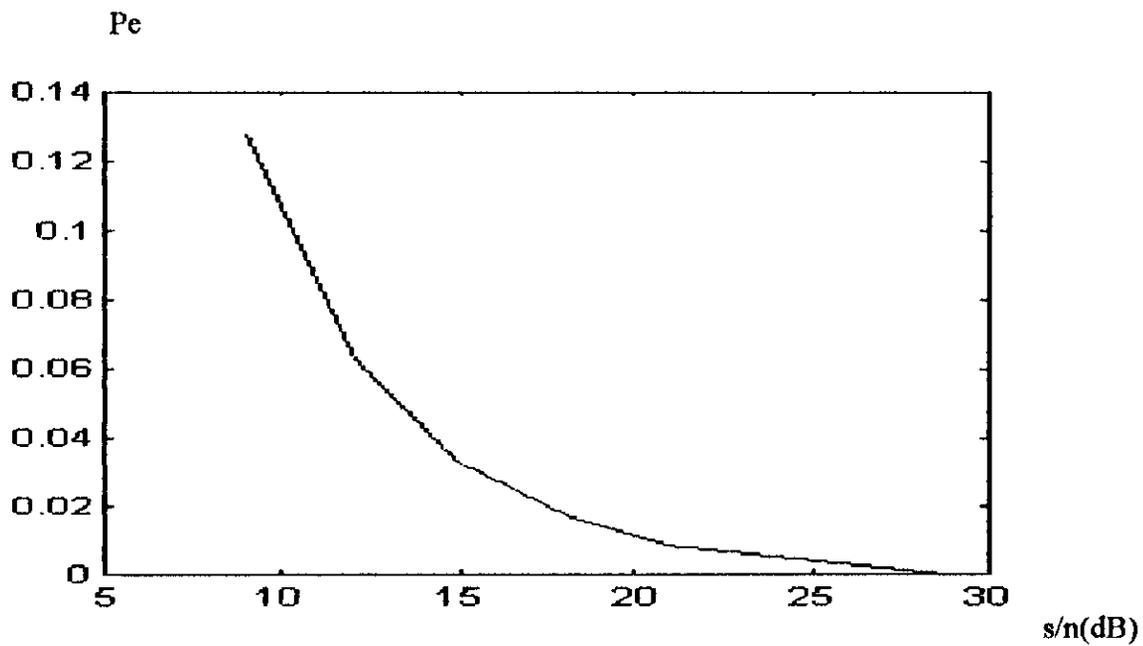


Figure 4 -4 codeur 1/3,k=3.

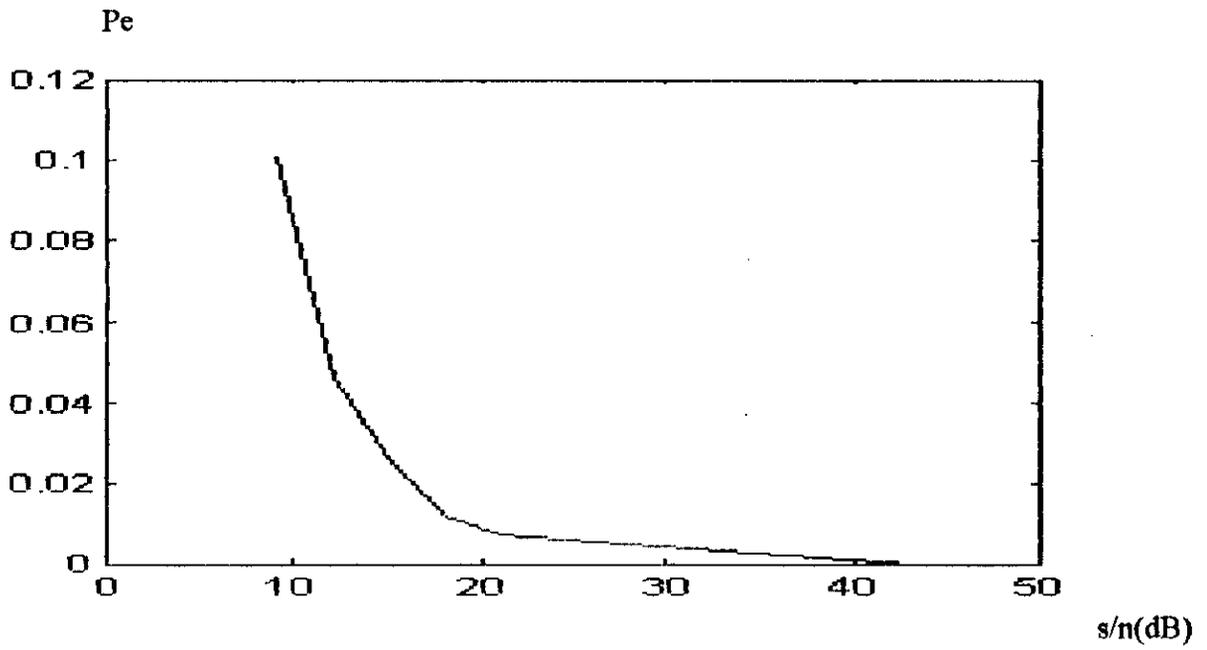


Figure 4 -5 codeur 1/4,k=3.

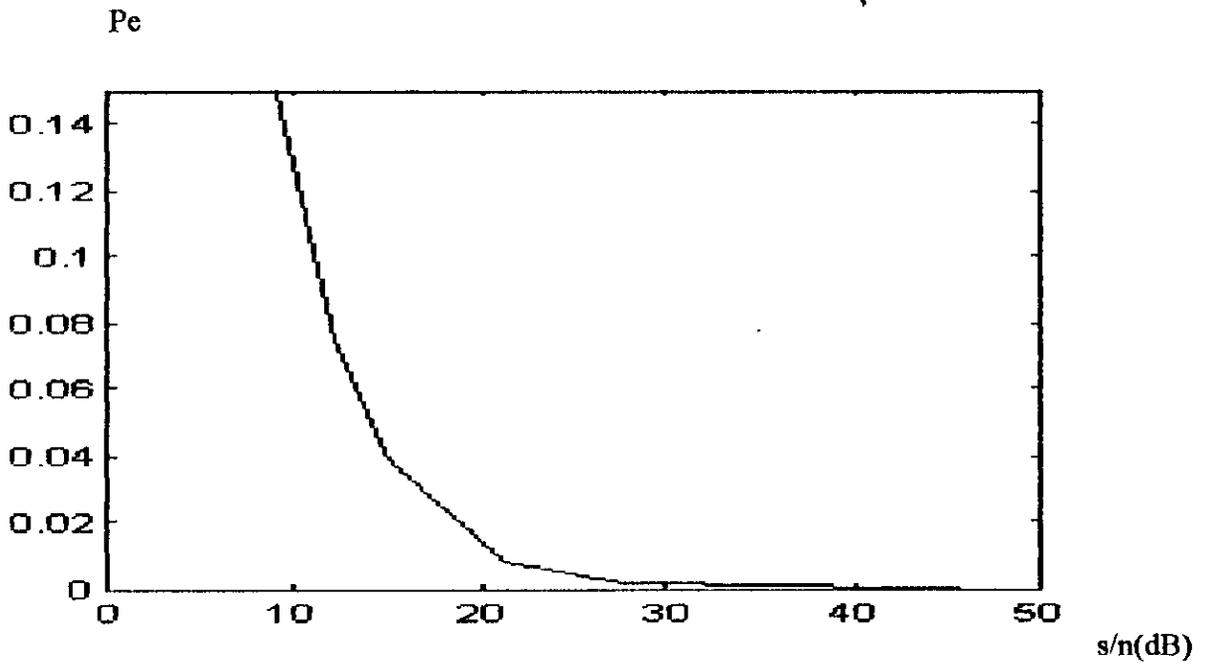


Figure 4 -6 codeur 1/2,k=5.

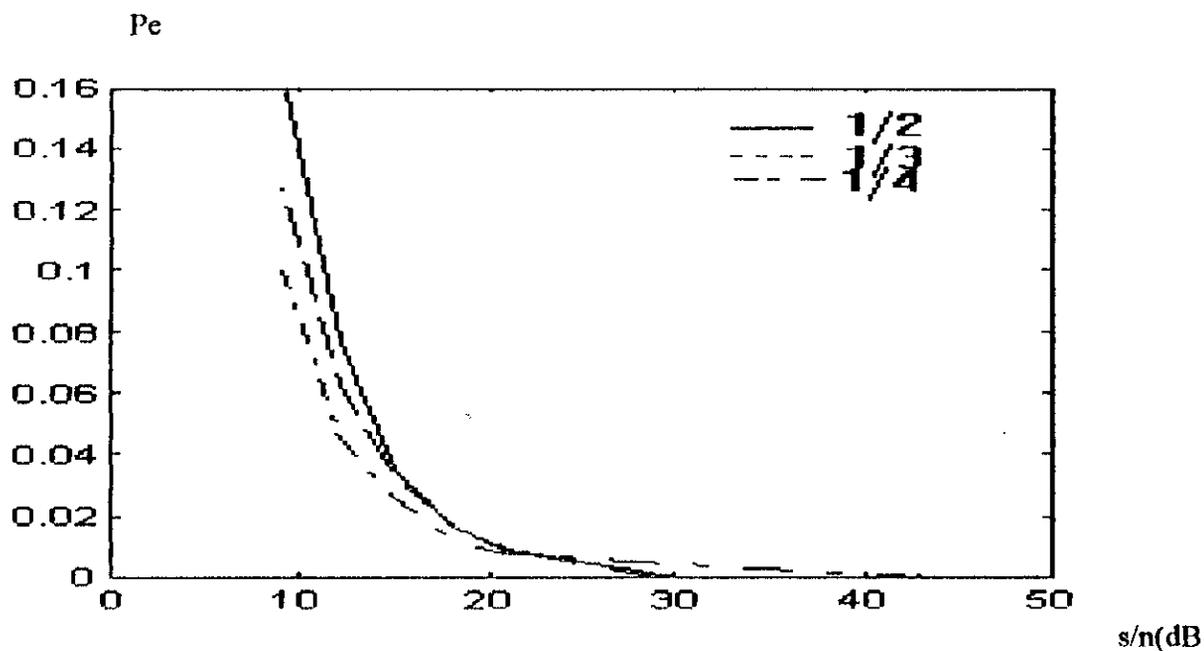


Figure 4 -7 comparaison entre codeurs $1/2, 1/3, 1/4; k=3$ en utilisant l'algorithme de Viterbi

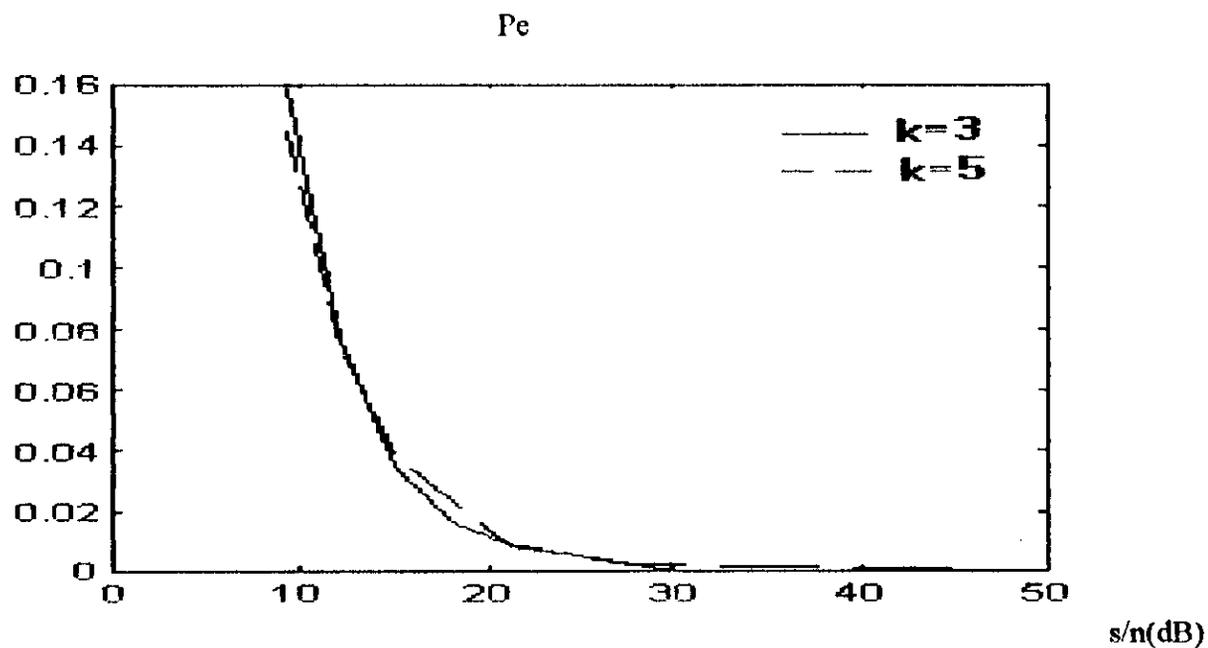


Figure 4 -8 comparaison entre codeurs $1/2; k=3, k=5$ en utilisant l'algorithme de Viterbi.

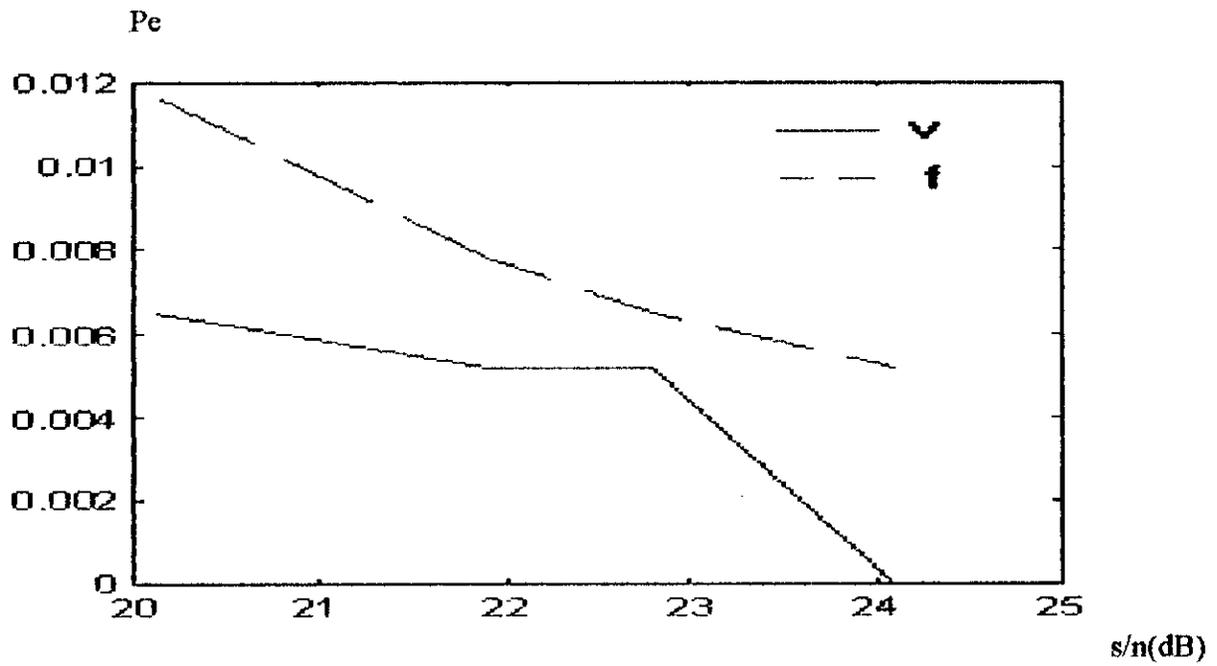


Figure 4 - 9 comparaison entre l'algorithme de Viterbi et celui de Fano.

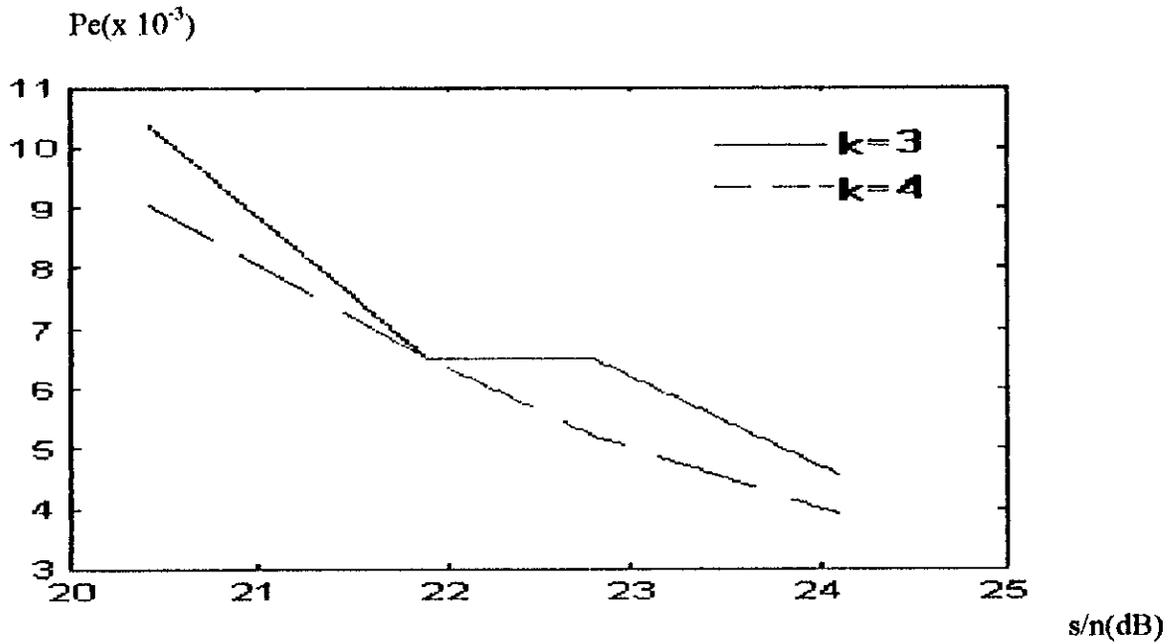


Figure 4 - 10 comparaison entre codeurs 1/2; K=3, K=4 en utilisant l'algorithme de Fano.

Toutes les courbes 4-3,4-4,4-5 et 4-6 ont la même allure (allure exponentielle décroissante): P_e est inversement proportionnelle au rapport S/N. Cela est conforme avec la théorie puisque pour un rapport S/N grand le signal d'information n'est affecté que peu par le bruit, et donc le décodeur aura à son entrée une séquence binaire avec un nombre de bits erronés faible, d'où une probabilité d'erreurs faible pour la séquence décodée, et vice versa.

La courbe de la figure 4 -7 fait la comparaison entre trois codeurs de même longueur de contrainte ($K=3$) et des débits $1/2, 1/3, 1/4$. On remarque que la probabilité d'erreurs P_e , pour un codeur de débit $1/n$, est inversement proportionnelle à n . Cela correspond bien à la théorie puisque pour n grand la redondance introduite par le codeur est importante et donc le signal d'information résiste aux bruits plus que pour n petit où la redondance introduite est faible.

La courbe de la figure 4 -8 fait une comparaison entre deux codeurs de même débit, $1/2$, et de longueurs de contrainte différentes, respectivement $K=3$ et $K=5$. On remarque que la probabilité d'erreurs P_e est inversement proportionnelle à la longueur de contrainte K . Cela est conforme avec la théorie puisque pour une séquence binaire de longueur L , la séquence d'information codée sera pour un codeur $1/n$, de longueur $n(L+K-1)$ et donc la redondance introduite augmente avec K , d'où la probabilité d'erreurs P_e diminue.

La courbe de la figure 4 -9 fait une comparaison entre l'algorithme de décodage de Viterbi et celui de Fano. Elle donne P_e en fonction du rapport S/N et cela en utilisant ces deux algorithmes. On remarque bien que l'algorithme de décodage optimal de Viterbi donne de meilleurs résultats que celui de Fano qui est sous-optimal. Cependant l'importance de l'algorithme de Fano réside dans sa simplicité d'implémentation par rapport à celui de Viterbi pour des longueurs de contraintes importantes. En revanche, son inconvénient est qu'il nécessite le stockage de toute la séquence reçue durant toute l'opération de décodage ce qui pose un problème de capacité mémoire pour des séquences relativement longues.

La courbe de la figure 4 -10 fait la comparaison entre deux codeurs de même débit et de longueurs de contrainte différentes $K=3$ et $K=4$ respectivement, et ce en utilisant l'algorithme de Fano. De même que P_e est inversement proportionnelle à K ce qui correspond parfaitement à la théorie.

4-4 CONCLUSION

En conclusion, nous dirons que pour un canal peu bruité l'utilisation de l'algorithme de Viterbi est préféré puisque dans ce cas des codeurs de longueurs de contrainte faibles donnent d'assez bons résultats. Tandis que si le canal est fortement bruité des codeurs de longueurs de contrainte importantes sont nécessaires et donc l'utilisation de l'algorithme de Viterbi rendra le décodeur très complexe, d'où la préférence d'utilisation de l'algorithme de Fano. Cependant même pour un canal fortement bruité, si les longueurs des messages transmis sont importantes, l'utilisation de l'algorithme de Fano nécessitera une taille mémoire importante. Dans ce cas un compromis doit être fait lors du choix entre ces deux algorithmes.

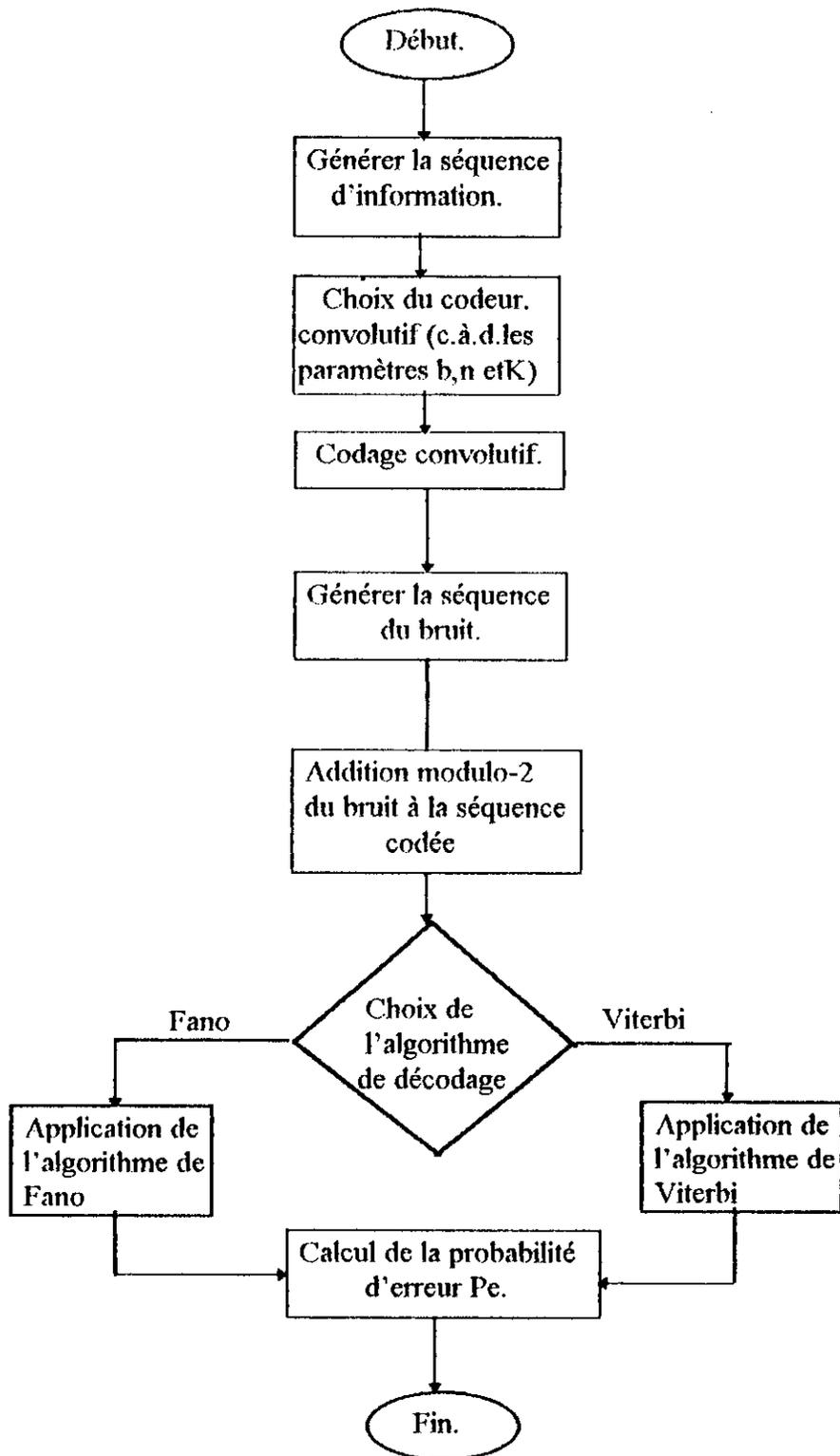


Figure 4-11 Organigramme représentant le plan de simulation.

CONCLUSION

Notre travail consiste en une étude des codeurs convolutifs englobant deux parties, la première est théorique, la deuxième étant une simulation dans le but d'évaluer les performances de tels codes.

Dans la partie théorique on a exposé les théorèmes fondamentaux du codage de canal ainsi que les fondements théoriques des codes convolutifs. Les algorithmes de décodage de Viterbi et celui de Fano y sont aussi détaillés.

Dans la partie de simulation on a essayé d'évaluer les performances des codes convolutifs, et cela en estimant la probabilité d'erreurs P_e de ces codes en fonction du rapport signal sur bruit S/N , décodage étant réalisé à l'aide de l'algorithme optimal de Viterbi. Malheureusement nous n'avons pas pu arriver à des probabilités d'erreurs inférieures à 10^{-3} et cela à cause de la limitation de la mémoire disponible. Cependant l'allure des courbes obtenues correspond bien à la théorie puisqu'on a trouvé que P_e diminue avec l'augmentation du rapport signal sur bruit. Nous avons aussi étudié l'effet des paramètres K et n (pour un codeur de débit $1/n$ et de longueur de contrainte K) sur la performance des codes convolutifs. Les résultats obtenus sont conformes avec la théorie, puisqu'ils montrent la diminution de P_e avec K et n . Enfin nous avons fait la comparaison entre l'algorithme optimal de décodage de Viterbi et celui Fano qui est sous-optimal. Là aussi les résultats sont conformes avec la théorie, puisqu'on a trouvé des résultats meilleurs avec l'algorithme de Viterbi que ceux obtenus avec l'algorithme de Fano.

Remarquons enfin que le fait d'ajouter de la redondance à l'information augmente la bande passante du signal original et diminue ainsi l'efficacité de l'utilisation de la bande de fréquence allouée à la transmission.

Pour remédier à cela une technique combinant la modulation et le codage peut être utilisée; c'est la modulation par codes en treillis (trellis-coded modulation). A la réception l'algorithme de Viterbi peut être utilisé pour la détection de la séquence d'information dont la fonction de vraisemblance est maximale.

BIBLIOGRAPHIE

- [1] Anderson, *Source and Channel Coding: An Algorithmic Approach*, Kluwer Academic Publishers (USA), 1991.
- [2] G. Battail, "Théorie de Codage et Protection Contre les Erreurs," *Techniques de l'Ingenieur.Traité Elecronique*, E-170.
- [3] S. Bellag,M. Guellai,"Simulation,Conception et Evaluation des Systèmes de Communication Numériques , "*Projet de fin d'etude*,Ecole Nationale Polytechnique,Département d'Electronique,El-harrach (Algerie), 1995.
- [4] J.C. Bic, D. Dupenteil et J.C. Imbeaux, *Elements de Communication Numériques*, Dunod, 1986.
- [5] R.M. Fano,"A heuristic discussion of probabilistic decoding,"*IEEE Trans.Inform.Théory*, Vol.II-9, pp.64-74, 1967.
- [6] G.D.Jr. Forney, " The Viterbi Algorithm," *Proc.IEEE*, Vol.61, pp.268-278, 1973.
- [7] A. Glavieux,"Codage de l'Information et Modultion des Signaux,"*Techniques de l'Ingenieur.Traité, Mesure et Cntrôle* R-380.
- [8] S. Haykin, *Digital Communications*, J. Wiley and Soon, New York (USA), 1988.
- [9] Knud, J. Larsen, "Short Convolutional Codes with Maximal Free Distance for Rates 1/2,1/3, and 1/4, "*IEEE Trans.Inform.Théory* , Vol.II-19, pp.371-372, May 1973.
- [10] H. Nussbaumer, *Téléinformatique*, Tome1, Prsses Polytechnique de Romandes, 1987.
- [11] A. Poli et LI. Huguet, *Codes Correcteurs,Théorie et Applications*, Masson, 1989.
- [12] J.G. Proakis., *Digital Communications*, Mc Graw Hill, New York (USA), 1983.
- [13] K. Sam Shanmugam, *Analog and Digital Communication* J.Wiley and Soon, New york (USA), 1985.
- [14] A. Spataru, *Fondement de la Théorie de la Transmission de l'Information*, Presses Polytechniques Romandes, 1987.
- [15] A.J. Viterbi, and J.K. Omura, *Principles of Digital Communication and Coding*, Mc.Graw Hill, (USA), 1979.

- [16] A.J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Inform. Theory*, Vol.IT-13, pp.260-269, 1967.