Codage canal : codes correcteurs d'erreurs

Hadj-Said Naima, Ali-Pacha Adda, Belgoraf A., Belmekki B.
Université des Sciences et de la technologie d'Oran – USTO- Mohammed BOUDIAF
BP 1505 El M'Naouer Oran 31036 -ALGERIA
E-Mail : nim_hadj@yahoo.fr

1- Introduction

La communication à distance (téléphone, télévision, satellite, etc.) , entre Machines et usagers nécessite des lignes de transmission acheminant l'information sans la modifier. Les lignes utilisées sont en général loin d'être parfaites [3, 6]. Pour cela, l'information devra être codée d'une manière spéciale permettant de déceler les erreurs, ou ce qui est encore mieux de les corriger automatiquement. On a été amené à concevoir des codes détecteurs et correcteurs d'erreurs.

La construction d'un mot de code comportant n bits est effectuée à partir de k bits du message source k-uplet binaires $U=(u_1,u_2,u_3,\ldots..u_k)$, appelé généralement message d'information, et de r bits de redondance. La méthode de codage la plus simple consiste à laisser inchangés les k bits d'information et à les reporter tels quels dans le mot de code en ajoutant les r bits de redondance $\{a_1,a_2,\ldots,a_r\}$. Les codes de ce type sont dits systématiques, un codeur qui transforme chaque message U indépendamment en n-uplet, le vecteur ligne V^T appelé mot code autrement dit il y a 2^k messages différents :

$V^T=[\ v_1\ v_2\ \ldots\ v_n] =[\ u_1,u_2,u_3,\ldots..u_k\ a_1,a_2,\ldots,a_r\ ]$

Les bits de redondance $a=\{a_1,a_2,\ldots,a_r\}$. Sont généralement appelés bits de contrôle.

Lorsque ces derniers sont calculés uniquement à partir des bits d'information du bloc auquel ils appartiennent, le code est appelé code de bloc (n , k), c'est à dire que les n symboles des mots codes sortant dépendent seulement des k bits en entrée correspondants, on dit alors que le codeur est sans mémoire et peut être implémenté par un circuit logique combinatoire.

Lorsque les bits de contrôle sont calculés à partir des bits d'information appartenant à plusieurs blocs, le code est dit convolutionnel ou récurrent.
Le rapport R=k/n est appelé rendement.
On caractérise aussi les codes par leur capacité de correction d'erreurs. En général on a deux types de catégories :
Ceux qui luttent bien contre les erreurs isolées tels que les codes de Hamming, BCH, GOLAY, et Reed Muller, etc.
Ceux qui sont bien adaptés aux coupures (paquet d'erreurs) comme les codes de FIRE, les codes de Reed Solomon, etc.

2. Code en Bloc

Une classe de ses codes est la classe des codes cyclique qui ont la propriété que toute permutation circulaire d'un mot code est un mot code.
Dans cette partie on étudie deux approches de code cyclique de Hamming [1,2,4] le code de Hamming cyclique correcteur d'une erreur CCHS et le code cyclique de Hamming améliorés CCHA.

2.1- C C H S

Un code de Hamming est un code cyclique C(n, k) généré par un polynôme primitif g(x) de degré $m \geq 3$, avec les caractéristiques suivantes :
Longueur du mot de code $n = 2^m-1$.
Nombre de bits de contrôle $m = n - k$.
Nombre de bits d'information $k = 2^m-m-1$.

2.1.1- Principe de codage

Soit $U = (U_0, U_1, \ldots, U_{k-1})$ le k – uplet à coder auquel on associe le polynôme U(x).
Le codage consiste en trois étapes :
a) Pré multiplier le polynôme U(x) associé au k – uplet à coder par x^{n-k}.
b) Obtenir le reste D(x) de la division de $x^{n-k} * U(x)$ par le polynôme générateur g(x).
c) Additionner D(x) et $x^{n-k} * U(x)$ pour obtenir le mot de code $V(x) = D(x) + x^{n-k} * U(x)$.

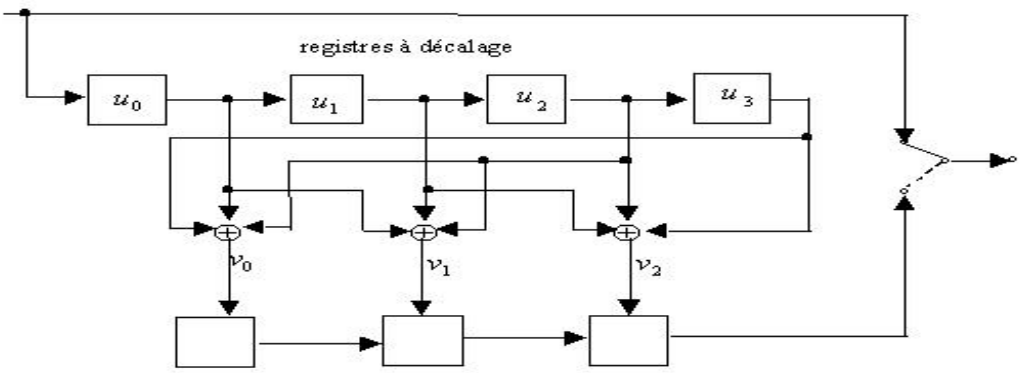

Figure 1 : *Circuit de codage du code C(7,4).*

On note : V(x) est le polynôme du mot code.

R(x) est le polynôme reçu.

E(x) est le polynôme d'erreur.

V est un mot de code si et seulement si [2] $V(\alpha) = 0$ avec α racine de g(x) (i.e. $g(\alpha) = 0$).

2.1.2- Principe de Décodage

Le décodage est l'opération la plus complexe puisqu'elle consiste à vérifier si le mot reçu est correct ou non.

Dans le cas où le mot reçu est erroné, le décodeur doit corriger les erreurs puis extraire l'information utile. Le processus de décodage est divisé en deux étapes :

- Détection d'erreurs dans le mot reçu.
- Correction de ces erreurs dans le cas échéant.

2.1.2.1- Détection d'erreurs (calcul du syndrome)

Le syndrome de R(x) noté S(x) est définit par :

$$S(x) = reste(x^{n-k} * R(x) / g(x)) \quad (1)$$

Avec g(x): le polynôme générateur et $R^{(i)}(x)$: le polynôme obtenu après le $i^{ième}$ décalage à droite du mot reçu R(x).

Le syndrome correspondant au $i^{ième}$ décalage cyclique de R(x) peut être calculer par :

$$S^{(i)}(x) = reste(x^{n-k} * R^{(i)}(x) / g(x)). \quad (2)$$

$S^{(i)}(x)$ peut être calculer par [7] :

$$S^{(i)}(x) = \begin{cases} reste(xS^{(i-1)}(x)/g(x)) & si \ \ i > 0 \\ S(x) \quad si \quad i = 0 \end{cases} \qquad (3)$$

De (2) on montre que les $S^{(i)}(x)$ peuvent être calculer à partir de S(x) en utilisant l'équation de récurrence et par conséquent on est pas obligé de réaliser des décalages sur R(x) pour les calculer.

La Formation du syndrome est :

$$S(x) = S^{(0)} + S^{(1)}x + S^{(2)}x^2 + ... + S^{(n-k-1)}x^{n-k-1} \quad (4)$$

La détection se fait suivant la valeur de S(x), on peut dire s'il y a des erreurs ou non, pour cela on a deux cas :

- S(x) = 0, R(x) est un multiple de g(x) donc le mot reçu est considéré comme le mot émis.
- S(x) <> 0, R(x) n'est pas un mot de code c'est à dire que V(x) = R(x) +E(x), on va corriger les erreurs.

S(x) ne dépend que de la configuration d'erreurs E introduite et pas de V.

En effet on a :

$$S(x) = reste(x^{n-k} * R(x)/g(x)) = reste(x^{n-k}E(x)/g(x)) \qquad (5)$$

2.1.2.2 Décodeur de MEGGIT

Le décodeur de Meggit est un décodeur très puissant, il est utilisé pour le piégeage de l'erreur dans le mot reçu.

La distance minimum d'un CCHS est $d_{min} = 3$, et il corrige t erreurs (une erreur simple) :

$$t = [(d_{min} - 1)/2] = 1 \qquad (6)$$

Soit $E(x) = x^{n-1} = x^{2m-2}$, Le principe de correction de E(x).

a) Détection de E(x) (calcul de S(x))

$$S(x) = reste(x^{n-k} * R(x)/g(x))$$

$$= reste(x^{n-k} * E(x)/g(x)) \qquad (7)$$

$$= reste(x^{m} * x^{2m-2}/g(x)) \ .$$

$$= reste(x^{m-1} * x^{2m-1}/g(x)) \qquad (8)$$

On a n= 2^m - 1 ➔ $x^n = x^{2m-1}$ ➔ $1 + x^n = 1 + x^{2m-1}$; donc $1+x^{2m-1}$ est divisible par g(x):

$1+x^{2m-1} = a(x).g(x)$ (9)

Donc le reste (x^{2m-1} / g(x)) = 1.

Et puisque le degré de g(x) = m on a alors $S(x) = x^{m-1}$.

Donc si le symbole erroné est r_{n-1} alors le syndrome correspondant est $S(x) = x^{m-1}$.

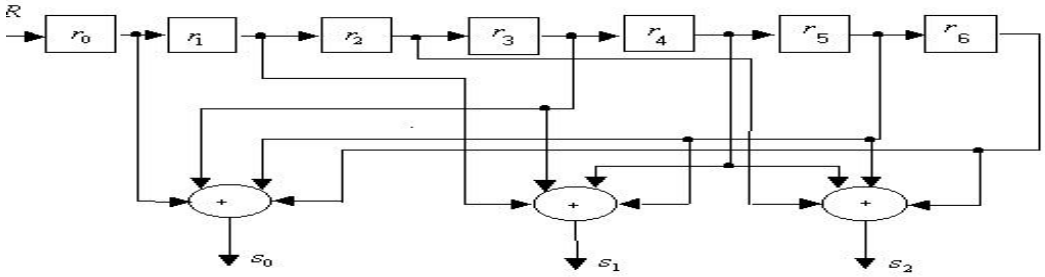

Figure 2 : *Calcul du syndrome du code C(7 , 4)*

b) Correction de E(x)

Si $S(x) = x^{m-1}$ alors la configuration d'erreur $E(x) = x^{2m-2}$ est reconnu le bit r_{n-1} peut être corrigé par :

$$r_{n-1} = r_{n-1} \oplus 1 \qquad (10)$$

avec $$V(x) = R(x) + x^{n-1} \qquad (11)$$

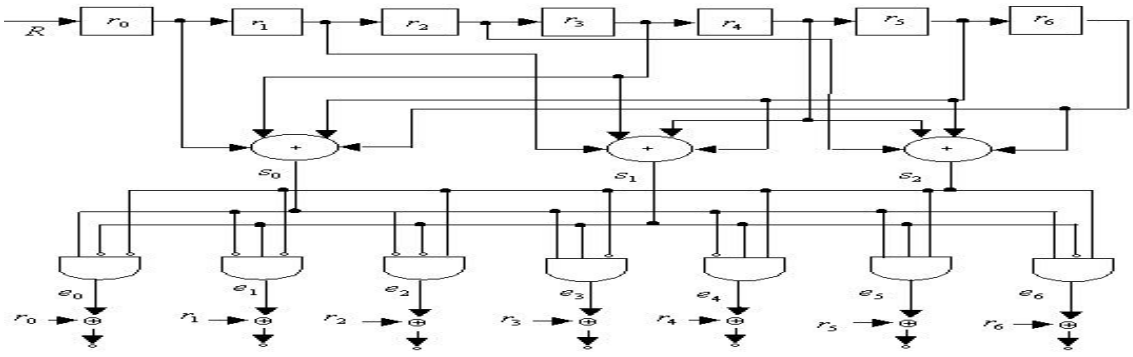

Figure 3*: circuit de décodage du code C(7, 4)*

c) Correction des autres configurations d'erreur

Supposons que $E(x) = x^p$ avec p = 0, … , n-1 , le bit erroné dans R(x) est r_p , en effectuant des décalages cycliques successifs sur R(x), on peut déplacer r_p jusqu'à la position (n-1), pour cela (n-p-1) décalages cycliques sont nécessaires.

Soit $R^{(n-p-1)}(x)$ le mot obtenu en décalant cycliquement R(x) de (n-p-1) positions, le polynôme d'erreur correspondant à $R^{(n-p-1)}(x)$ est $E(x) = x^{n-1}$.

Si on connaît le syndrome $S^{(n-p-1)}(x)$ du mot $R^{(n-p-1)}(x)$ on peut reconnaître l'erreur et par conséquent corriger r_p, en effet si 'i' représente le nombre de décalages nécessaires pour déplacer l'erreur de la position p jusqu'à la position (n-1) (i.e. i= n-p-1).

Alors l'erreur est repérée par la position p = n-i-1 et le mot de code émis est :
$V(x) = R(x)+x^p$.

2.2- C C H A

Le code cyclique de Hamming présenté ci-dessus peut être améliorer pour corriger chaque erreur simple (isolée) et en même temps détecter chaque combinaison de deux erreurs, schématiquement la détection de l'erreur double est rendue possible grâce à un bit de contrôle (redondance) supplémentaire.

Le code CCHA est un code cyclique généré par le polynôme

$g(x) = (1+x) P(x)$ (12)

P(x) est un polynôme primitif de degré m ($m \geq 3$) ; avec les caractéristiques :

- Longueur du mot de code : $n = 2^m-1$.
- Nombre des symboles de contrôle : n-k = m+1.
- Nombre des symboles d'information : $k = 2^m-m-2$.

Le code CCHA se compose de tous les mots du code CCHS qui ont un poids paire [6].

2.2.1- Décodage

Le principe de décodage du décodage du code CCHA peut être inspiré à partir du décodeur précèdent (décodeur du code CCHS).

La distance minimale du code CCHA est $d_{min} = 4$,

il peut corriger $t=[(d_{min}-1) / 2]=1$ erreur.

Et détecter $d_{min} / 2 = 2$ erreurs.

a) Calcul du syndrome est détection d'erreurs

En divisant $x^m * R(x)$ par P(x) et R(x) par (1+x) On obtient :

$$x^m * R(x) = a_1(x) . P(x) + S_p(x) \quad (13)$$

$$R(x) = a_2(x) . (1+x) + u \quad (14)$$

Avec le degré $S_p(x) \leq m-1$ et $u \in GF(2)$ On a

$$u = \text{reste}(R(x) / (1+x)) = WG (R(x)) \quad (15)$$

ou WG désigne le poids du polynôme reçu R.

2.2.2- Tests et décisions

-t_1: si $(u = 0)$ et $(S_p(x) = 0)$: le décodeur considère qu'il n'y a pas d'erreurs dans le mot reçu.

-t_2: si $(u = 0)$ et $(S_p(x) \neq 0)$: le décodeur considère qu'il y a double erreurs dans le mot reçu, aucune action de correction n'est prise.

-t_3: si $(u = 1)$ et $(S_p(x) = 0)$:: Dans ce cas on a un nombre impaire d'erreurs, le polynôme erreur est divisible par P(x).

-t_4: si $(u \neq 0)$ et $(S_p(x) \neq 0)$: le décodeur considère qu'il y a une erreur simple dans le mot reçu, un processus similaire à celui du code CCHS est activé avec $g(x) = (1+x) P(x)$.

2.3- Algorithmes de protection : l'entrelacement

En dehors du masquage d'erreurs utilisant la redondance interne de l'information, il existe des méthodes simples qui permettent de lutter contre les erreurs de transmission, sans considération de codage, parmi ces méthodes il y a l'entrelacement.

Avant la transmission, on mémorise une suite de mots de codes dans une mémoire ayant la forme d'une matrice de λ lignes et de n colonnes. On y fait entrer ligne par ligne, λ mots de codes de longueur n, puis on lit la mémoire colonne par colonne afin de transmettre tous les symboles.

La transmission consiste à transmettre les colonnes Colonne 1, Colonne 2, …, Colonne n.

A la réception on fait entrer les symboles reçus dans une mémoire identique colonne par colonne et les mots de codes sont restitués en lisant la mémoire ligne par ligne.

Soient V(0), V(1), …, V(λ), λ mots de codes à entrelacer, l'entrelacement consiste à transmettre ces mots de codes dans l'ordre des symboles suivant :

Colonne 1	**Colonne 2**	**. . .**	**Colonne** n
$V_{n-1}(1)$	$V_{n-2}(1)$	. . .	$V_0(1)$
	.	. . . .	. . .
$V_{n-1}(\lambda)$	$V_{n-2}(\lambda)$	. . .	$V_0(\lambda)$

***Tableau** 1 : Entrelacement*

Ce procédé très simple permet de fragmenter les coupures en disséminant les symboles en erreur dans plusieurs mots de code, ce procédé présente deux inconvénients :

- nécessité d'un volume de mémoire non négligeable pour des codes de longueur même modeste.
- possibilité de constitution de coupures à partir d'erreurs isolées . il est possible de proposer un entrelacement à profondeur variable, appelé scrambling. Mais la gestion des mots devient lourde.

L'effet de l'entrelacement est de répartir sur des mots différents les erreurs d'un même paquet d'erreurs, l'entrelacement n'augmente pas la capacité de correction d'un code, il ne fait qu'adapter la puissance de correction aux configurations d'erreurs les plus fréquentes.

Etant donné un code cyclique C(n, k), il est possible de construire un code cyclique C'(λ*n, λ*k) par entrelacement ceci est accompli en arrangeant m mots de codes du code original en m lignes d'un tableau (tableau 1 : Entrelacement) et en les transmettant colonne par colonne, le code obtenu est appelé code entrelacé.

3. Code Convolutionnel

3.1-Présentation du codeur

Le codeur des codes convolutionnels est un codeur (n, k, m) qui a des blocs de k bits pour la séquence d'information U et produit une séquence codée V de n symboles (k entrée et n sorties).

Dans ce type de codes chaque bloc codé ne dépend pas seulement des blocs de message de k bits correspondant dans la même unité de temps mais aussi des m précédents blocs de messages d'où l'existence de codeur à mémoire d'ordre m qui doit être implémenté avec un circuit logique séquentiel c'est à dire à l'aide des multiplexeurs pour rendre en série l'information en entrée et en sortie un modulo additionneurs (ou exclusif) et des registres à décalage :

- **n** : le nombre de sorties,
- **k** : le nombre d'entrées,
- **m** : le nombre de cellules de tous les étages.

Il nécessite par suite un décodeur à mémoire d'ordre m [1, 2].

Les r =(n–k) bits redondants qui sont ajoutés à la séquence d'information croit avec m d'où k est strictement inférieur à n ou bien le rendement R <1.

Pour illustrer ce dernier on prend comme exemple un codeur (2,1,3) figure 4 :

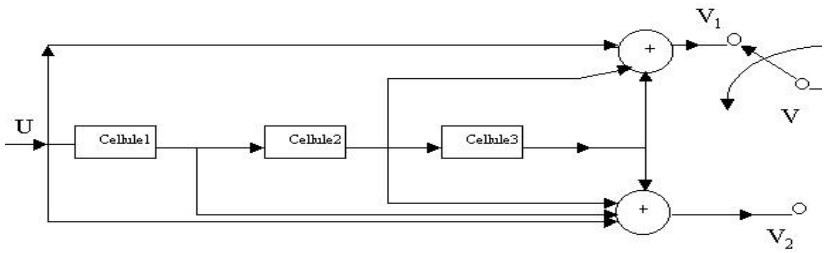

Figure.4: *CODEUR (2, 1, 3).*

Considérons la séquence d'information *U =(10110…..)* avec le bit le plus à gauche est supposé entré le premier. En utilisant les règles de l'addition avec le ou exclusif *(1+1=0)* ainsi que le multiplexeur qui sert à rendre la séquence codée en série.
On obtient en sortie donc la séquence en sortie V=(11,01,00,01,……..).

3.2- Analyse des Codes Convolutionnels

Pour analyser les codes convolutionnels. On préfère donc représenter graphiquement le fonctionnement du codeur, l'étude du codage convolutionnel par un diagramme arborescent [6] devient vite impraticable dès que la séquence à coder dépasse quelques bits, puisque la dimension de l'arbre double à chaque étage.

On préfère donc représenter le fonctionnement du codeur à l'aide d'un diagramme d'état ou le diagramme en treillis. On suppose que les registres du codeurs sont initialisées à zéro.

3.2.1- Diagramme d'état

Le diagramme d'état dans lequel chaque état représente un état particulier des registres du codeur et chaque branche représente le changement d'état du codeur en fonction de l'arrivée d'un nouveau bit. Les différentes branches sont repérées par la valeur du bit d'entrée qui provoque le changement d'état et par le mot de code produit à l'arrivée de ce bit.
Dans le cas de la figure 4 correspondant au codeur (2,1,3), le codeur comprend trois registres et peut prendre huit états auxquels correspond huit nœuds du diagramme d'état figure 5.
Les états S_i correspondant aux états des cellules à décalage du codeur (2, 1, 3) sont $S_0 = 000$, $S_1 = 100$, $S_2 = 010$, $S_3 = 110$, $S_4 = 001$, $S_5 = 101$, $S_6 = 011$ et $S_7 = 111$.
On procède en posant S_0 comme étant un état initial et final à la fois, on étiquette chaque branche avec le bit qui cause la transition et les deux bits sortants.

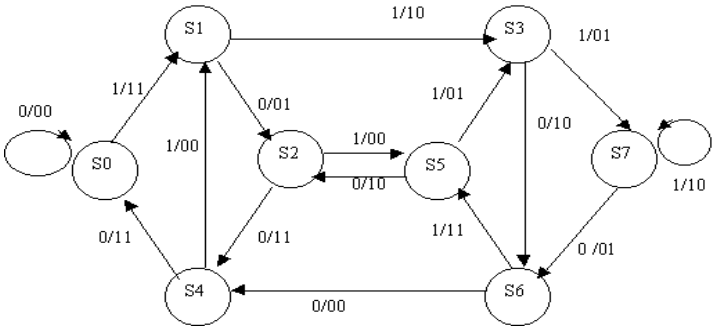

figure 5 : *Diagramme d'état du codeur (2, 1, 3).*

Si l'information à coder est U=(11101) son mot code correspondant est V=(1110010111101111) de longueur n*(L+m) donc 16.
Pour connaître le mot code d'une quelconque séquence d'information il suffit de suivre le chemin tout au long du diagramme et donc on constate qu'il revient toujours à S_0.

3.2.2- Diagramme en Treillis

Le fonctionnement d'un codeur convolutionnel peut être également représenté par un diagramme en treillis du type de celui de la figure 6 qui correspond au codeur de la figure 4, et dans lequel le cheminement du point sur le graphe s'effectue de la gauche vers la droite.

Le treillis provient de l'étude des automates d'états finis conçus à l'intention d'expliquer le fonctionnement interne de l'algorithme de viterbi pour le décodage des codes convolutionnels. A ce jour les treillis pour le décodage de Viterbi restent la motivation fondamentale pour la recherche.
Chaque transition d'un état de registre à l'état suivant est repérée par une ligne pleine ou en pointillé selon que le bit qui l'a provoqué est 1 ou 0, et comporte l'indication du mot de code produit.

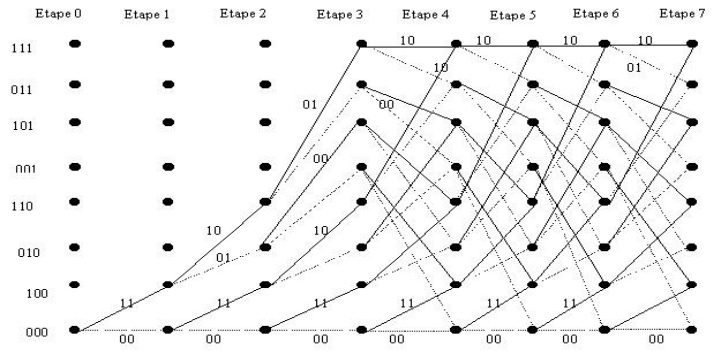

figure 6 : *Diagramme en treillis du codeur (2, 1, 3).*

Cette structure peut être considérée comme un diagramme d'état répété une multitude de fois, le temps que la séquence d'information soit codée. Il permet de connaître un chemin correspondant à un mot code donné.

La complexité du treillis croit exponentiellement avec le nombre d'états $2^{m,}$ c'est à dire avec la longueur de contrainte Lc.

3.3- Décodage des codes convolutionnels :

En pratique, il est possible de tirer un meilleur parti des possibilités de détection et de correction des codes convolutionnels en faisant appel à des méthodes probabilistes [8, 9, 10] dont le plus connue est l'algorithme de Viterbi.

3.3.1- Principe de l'algorithme de Viterbi :

L'algorithme de Viterbi applique au décodage le principe du maximum de vraisemblance et il est basé sur la programmation dynamique.

L'algorithme de Viterbi peut être considéré comme une estimation de l'état du codeur.

Si le codeur comporte (K-1) étages de m bits, l'état Q_i du codeur juste avant l'arrivée du mot d'information $x^T_{i,}$ peut être considéré comme défini par la valeur des (K-1) mots qui précèdent $x^T_{i,}$ avec :

$$Q_i = (x^T_{i-1}, x^T_{i-2}, \ldots; x^T_{i-K+1})$$

Puisque chaque mot d'information a une longueur de m bits Q_i peut prendre $2^{(K-1)m}$ valeurs différentes possibles.
L'arrivée du bloc d'information x^T_i fait passer le codeur à l'état Q_{i+1} en provoquant l'émission d'un mot de code y^T_i de n bits.
Inversement, il est possible de déterminer le mot d'information x^T_i et le mot de code y^T_i sachant que le codeur est passé de l'état Q_i à l'état Q_{i+1}.
Le décodage de la séquence reçue $y^{\wedge T}$ peut être effectué en estimant la séquence des états successifs $Q^{\wedge}$ qui a donné naissance à $y^{\wedge T}$.
Les séquences émission estimées $x^{\sim T}$ et $y^{\sim T}$ sont ensuite déterminées [1, 2] à partir de la séquence $Q^{\wedge}$.

3.3.2- Présentation de l'algorithme de Viterbi

L'algorithme peut être mis en œuvre de façon commode [11, 12, 13] à partir d'un diagramme en treillis en appliquant les règles suivantes :

1. **Etape 1** : Partir du premier étage du diagramme où deux branches arrivent à chaque état, calculer pour chacun des états la distance $d(y^{\sim T}, y^{\wedge T})$ de chacun des deux chemins qui arrivent à cet état, pour chaque état, conserver le chemin dont la distance est la plus faible (le survivant), et éliminer l'autre.

2. **Etape 2** : Répéter l'opération pour chaque étage du treillis.

3. **Etape 3** : Lorsque l'origine de tous les survivants passe par un chemin unique, ce dernier représente une partie du mot décodé.

4. Concaténation des Codes

Dans ce qui suit on va essayer de construire un nouveau code avec deux étages en série figure 7, l'un lutte contre les paquets d'erreurs (Burst) et l'autre lutte contre les erreurs isolées afin de reconstituer l'information émise avec plus de fidélité [5]. Cette technique est connue sous le nom de la **concaténation des codes**.

La concaténation à été proposée pour la première fois par FORNEY dans le but de concevoir des codes à longueur convenable qui peuvent être décodé sans l'utilisation d'équipement complexe, lorsqu'on veut un système de codage puissant, on peut envisager deux (ou plus) niveau de codage [2].

On code d'abord l'information avec un code C_1 (code externe), ainsi on passe d'une suite de longueur k à une suite codée de longueur n, cette nouvelle séquence (de longueur n) est alors considérée comme de l'information pour un second code C_2 (code interne).

Généralement le code externe est plus spécialement construit pour lutter contre les erreurs engendrées par le code interne lorsqu'il se trompe, on obtient ainsi des systèmes de codage très puissants que les systèmes élémentaires [4].

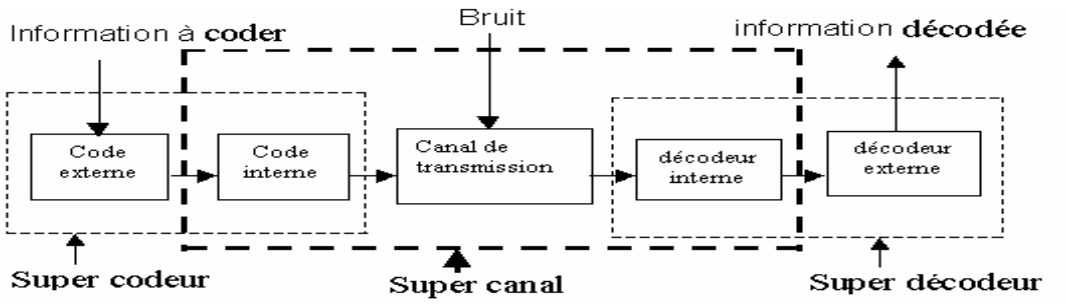

Figure 7: *concept de la concaténation*

4.1- Principe de la Concaténation

Soient $C_1(n_1,k_1)$ un code binaire de distance minimum d_1 et $C_2(n_2,k_2)$ un code non binaire à coefficient dans $GF(2^m)$ de distance minimum d_2 figure 8.

a=$a_0a_1a_2\ldots a_{2-1}$ est l'information à coder avec le code $C_2(n_2,k_2)$. C=$c_0c_1c_2c_3\ldots c_{n2-1}$ est le mot de code obtenu après codage de **a** en utilisant le code externe $C_2(n_2,k_2)$.

D=$d_0d_1d_2\ldots d_{n2-1}$ le mot de code obtenu après codage de C on utilisant le code interne $C_1(n_1,k_1)$.

Les symboles d'informations sont divisés en k_2 blocs contenant chacun k_1 bits d'informations, les k_2 blocs sont codés en utilisant le code C_2 pour former un mot contenant n_2 blocs (la taille de chaque bloc est k_1), chaque k_1 bits de chaque bloc sont ensuite codés en utilisant le code C_1 pour former un mot de longueur n_1 (contenant n_1 bits de GF(2)), donc les mots de codes du code résultant sont caractérisés comme suit [1] :

1. $n = n_1 \times n_2$ la longueur du mot de code.
2. le code C a une distance minimum. $d_{min} \geq d_1 \times d_2$
3. $k = k_1 \times k_2$ la longueur de l'information utile, et les rendements r et R :

$$r = k_1/n_1 \qquad R = k_2/n_2$$

$$R' = r \times R = (k_1 \times k_2) / (n_1 \times n_2)$$

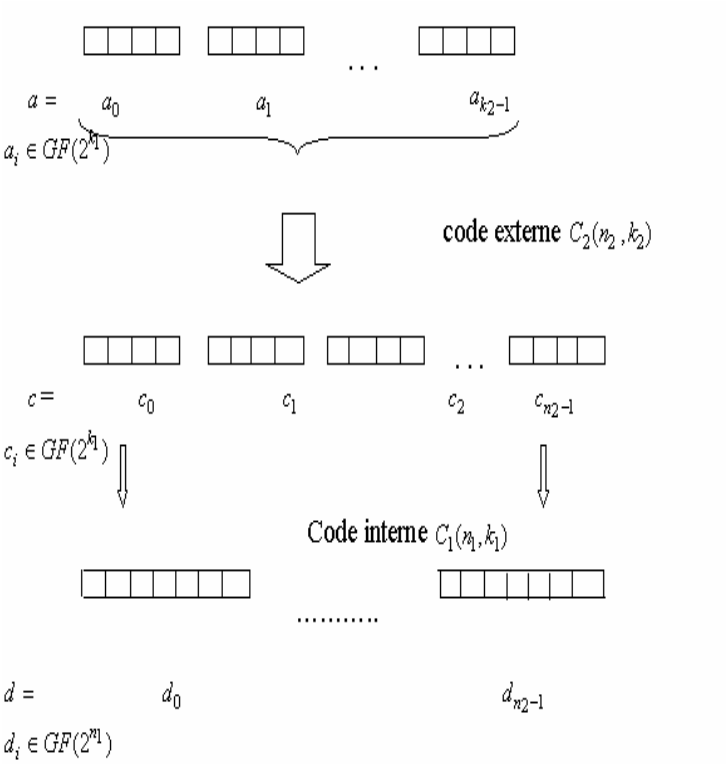

Figure 8 : *Principe de la Concaténation*

4.2- Choix des Codes

Lorsque les choix nécessaire (type de démodulation et stratégie de protection) sont faits, il reste à décider du code à utiliser.

C’est à dire choisir judicieusement le code externe et le code interne afin de constituer un système concaténé.

Un code se choisit en fonction de certains paramètres comme :

- La distance minimale **d** : permet de savoir le nombre maximum **t** d’erreurs corrigibles par mot (d= 2t+1 ou d = 2t) .
- Le polynôme de poids : permet de connaître exactement l’efficacité du code (utile pour le taux d’erreurs en entrée qui sont forts).
- Le taux de redondance : souvent limité dans les applications, on peut indiquer une valeur limite souvent acceptée par les industriels 10%.
- Le type de codage : systématique ou non, on peut souhaiter que l’information n’apparaisse pas en clair.
- Comptabilité ou non avec la cryptographie.
- Complexité de l’électronique du codeur / décodeur.
- Temps d’attente pour avoir le premier mot décodé.
- Débit, coût et volume électronique, architecture classique ou non, longueur des blocs d’information, gain à 10^{-5} (ou un autre niveau).

5. Conclusion

Satellites à défilement, satellites géostationnaires, réseaux internationaux, téléconférence, disque – compact sont nourris d'une information de plus en plus souvent traitée par des méthodes mathématiques qui s'appuyant sur électronique raffinée.

L'amélioration qualitative de la transmission peut également être réalisée en agissant sur le canal (câbles, fibre optique etc.), ainsi que l'information transmise. Cette information est considérer comme des ensembles de symboles génèrent par la source.

Avant de transmettre de tels symboles à travers le canal à perturbation, il est y ajouté une certaine redondance pour l'apport de certains symboles, appelés symboles de contrôles et dont la raison est indiquer au destinataire la présence d'erreurs et même de lui donner la possibilité de les corriger, c'est le cas de codes détecteurs et correcteurs d'erreurs.

Mais, lorsqu'on veut un système de codage puissant, on peut envisager deux (ou plus) niveaux de codage (concaténation série), dans le but d'exploiter les avantages du premier code pour minimiser les inconvénients du deuxième et/ou l'inverse, afin de reconstituer l'information émise avec beaucoup plus de fidélité.

6. Bibliographie

[1] G.C Clark, J.B Cain , “Error Correcting Coding for Digital Communication’’, Plenum Press 1981.

[2] D.J Costello, S.Lin, “Error Control Coding : Fondamentals and Applications’’, Prentice Hall 1983.

[3] Revue Mini & Micro, N°171, Sept 1982.

[4] A. Poli, Li Huguet, « Codes Correcteurs : Théorie et Applications », Masson Paris 1989.

[5] S.Foughali, S.Khelifa, « Concaténation des Codes Cyclique (Reed Solomon – Hamming) Appliquées aux images fixes », Institut d’Informatique, USTO 1998.

[6] M.A. Sahraoui, A. Zebida , « Codes Détecteurs et Correcteurs d’erreurs appliquées aux images fixes », Institut d’Informatique, USTO 1995.

[7] M. Diab, « Contribution à l’étude des architectures systoliques pour les codes correcteurs d’erreurs « , Thèse de doctorat de l’université Paul Sabatier, Sept 92.

[8] A.BENHALLAM ‘ Comparaison Entre Différentes Procédures De Décodages Des Codes Convolutionnels ’ Note Interne ; Institut National Polytechnique De Toulouse 1985.

[9] J.B ANDERSON ‘Sequential Decoding Based On Qn Error Criterion’ IEEE Trans Inf Theory Vol 38 N :3 , pp 987-1001, May 1992.

[10] B.RADOSAVLJEVICI ‘ Sequential Decoding Of Low Density Parity Check Codes By Adaptive Reordering Of Parity Checks’ , IEEE Trans Inf Theory Vol 38 ; N :6 PP 1833-1839 ;Novembre 1992.

[11] M. Benazzouz, A.F. Daouadji ‘Décodage Des Codes Convolutionnels : Algorithme de Viterbi’, Mémoire de fin d’étude d’ingéniorat, Institut d’Informatique, Novembre 1997.

[12] Henri Nussbaumer ‘ Téléinformatique Tome :1’, Presses Polytechniques Romandes, 1987.

[13] H.GHAIB , W. MOSTEGHALMI ‘Décodage Séquentiel Des Codes Convolutionnels : Algorithme De La Pile’, Mémoire de fin d’étude d’ingéniorat, Institut d’Informatique, Novembre 1998.