

codage correcteur d'erreurs convolutionnel

| | | |
|-----|--|----|
| 1. | Définition | 2 |
| 2. | représentation en treillis | 3 |
| 3. | Décodage : algorithme de Viterbi..... | 3 |
| 4. | Distance libre..... | 5 |
| 5. | Diagramme d'état. Fonction de transfert..... | 5 |
| 6. | Décodage à décision douce en entrée | 7 |
| 7. | Limitation du décodage à une fenêtre..... | 8 |
| 8. | Codes convolutionnels poinçonnés : | 8 |
| 9. | Choix : comparaison des codes cycliques et des codes convolutionnels . | 9 |
| 10. | Concaténation..... | 9 |
| 11. | Entrelacement..... | 9 |
| 12. | Bibliographie | 10 |

Codes convolutionnels

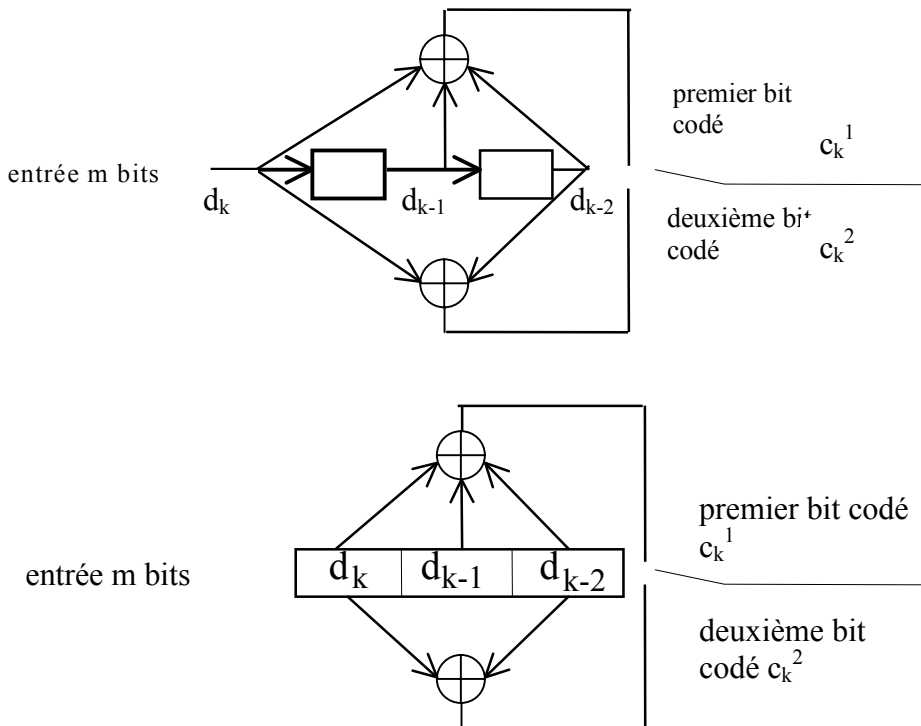
1. Définition

Pour ces codes, chaque bloc de n éléments binaires en sortie dépend non seulement des k éléments binaires présents en entrée mais aussi des m blocs de k éléments binaires précédents. $m+1$ s'appelle la longueur de contrainte, souvent $k=1$.

Le taux de codage est $R= n/k$

Le codage se fait avec des registres à décalage et des ou exclusif.

Exemple :



| bit d'entrée | état du registre $d_{k-1}d_{k-2}$ | contenu du registre | bits de sortie |
|--------------|--------------------------------------|---------------------|----------------|
| 0 | 00 | 000 | 00 |
| 1 | 00 | 100 | 11 |
| 0 | 01 | 001 | 11 |
| 1 | 01 | 101 | 00 |
| 0 | 10 | 010 | 10 |
| 1 | 10 | 110 | 01 |
| 0 | 11 | 011 | 01 |
| 1 | 11 | 111 | 10 |

Dans cet exemple à un é.b d'entrée correspond 2 é.b de sortie qui sont déterminés par les équations suivantes :

$$c_k^1 = d_k + d_{k-1} + d_{k-2} \quad c_k^2 = d_k + d_{k-2}$$

soit en transformée en z :

$$c^1(z) = (1 + z^{-1} + z^{-2})d(z) \quad c^2(z) = (1 + z^{-2})d(z)$$

On remplace souvent Z^{-1} par D (!).
 Et on appelle les polynômes générateurs :

$$1 + D + D^2 \quad 1 + D^2$$

ou en octal [111]=7 et [101]=5

De façon générale le $j^{\text{ième}}$ bit de code a la forme, dans le cas où le bit d'entrée est unique :

$$c^j(k) = \sum_{i=0}^{m-1} g_{ji} d(k-i) \quad \text{où } k \text{ est l'instant d'observation.}$$

Il y a autant de polynômes générateurs que de sorties du code : ces polynômes sont :

$$g_j(D) = \sum_{i=0}^{m-1} g_{ji} D^i$$

2. représentation en treillis

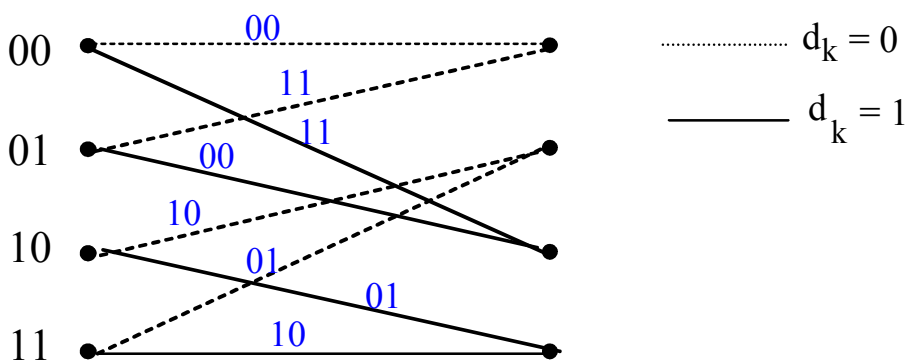
Pour faciliter l'algorithme de décodage, la représentation la plus courante du codage est la représentation en treillis.

L'état du codeur à l'instant k est représenté par l'état $\{d_{k-1}, d_{k-2}, \dots, d_{k-m}\}$. A chaque arrivée d'un élément binaire d_k , une sortie (un mot de code) est générée, puis juste après le codeur passe dans l'état suivant qui est $\{d_k, d_{k-1}, \dots, d_{k-m}\}$.

Le treillis est formé de nœuds reliés par des branches : les nœuds représentent les différents états du codeur possibles : il y en a 2^{m-1} s'il y a une entrée $2^{(m-1)k}$ s'il y a k entrées, les branches représentent les différentes transitions possibles d'un nœud à un autre (ou d'un état du codeur au suivant) lors de l'arrivée d'un bit d'entrée.

Voici le treillis du code précédent : les états sont 00,01,10,11.

Partant, par exemple de l'état 00, l'arrivée d'un 0 mène le codeur à l'état 00 (transition en pointillé pour l'arrivée d'un 0) et l'arrivée d'un 1 mène le codeur à l'état 10 (transition en trait plein pour l'arrivée d'un 1). A chaque branche on peut associer le mot codé soit les 2 bits de code ici.

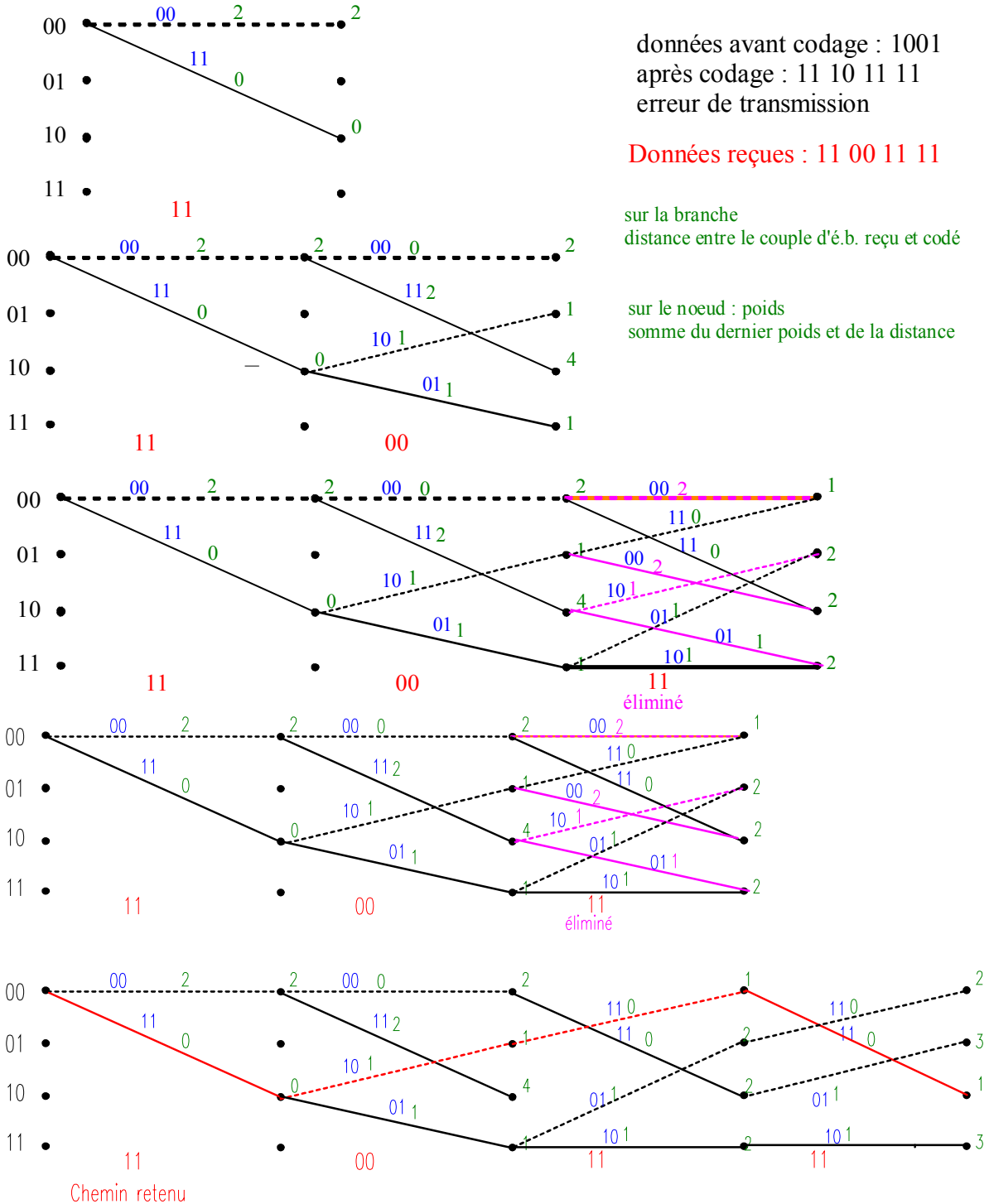


3. Décodage : algorithme de Viterbi

Le décodage le plus courant est basé sur l'algorithme de Viterbi.

Il consiste à rechercher dans l'arbre le chemin qui correspond à la séquence la plus probable, c'est-à-dire celle qui est à la distance minimale de la séquence reçue ou encore la séquence la plus probable.

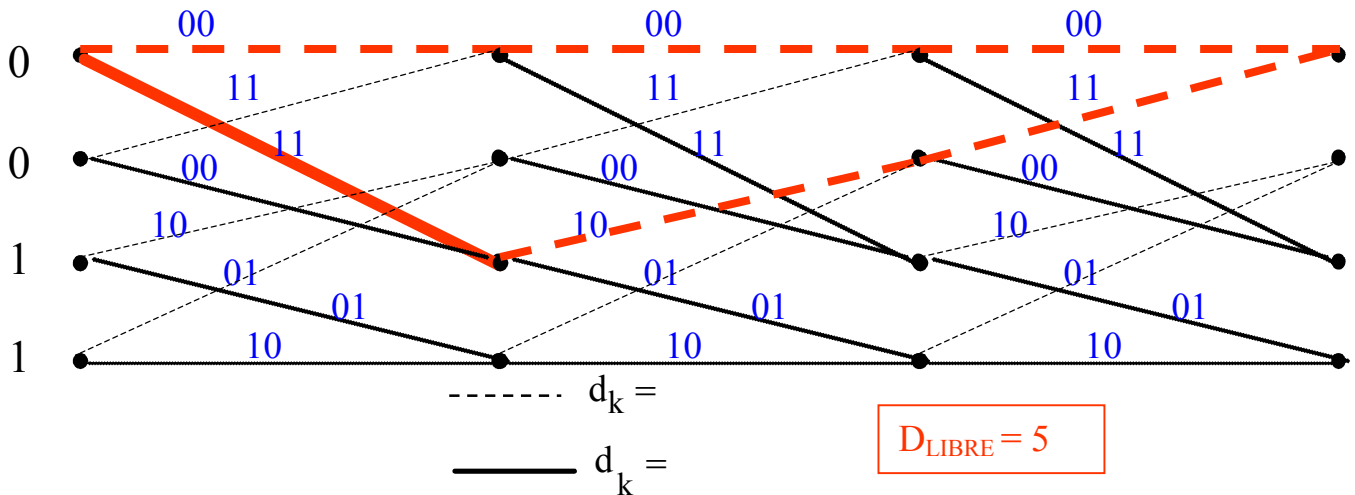
Dans l'exemple qui suit (basé sur le codeur déjà décrit) on suppose que le démodulateur fournit des 0 et des 1 (décision dure). A chaque fois qu'un groupe de 2 é.b. arrive, on examine toutes les branches possibles du treillis, on calcule la distance entre les é.b. affectés aux branches et les é.b. reçus, on ne garde que les branches donnant lieu à une distance minimale (ces branches forment le chemin survivant), et on affecte l'extrémité du chemin survivant d'une métrique égale à la somme de la métrique précédente et de la distance de la branche retenue.



4. Distance libre

C'est la distance minimale entre deux chemins divergents partant du même point et y revenant le « plus vite possible ».

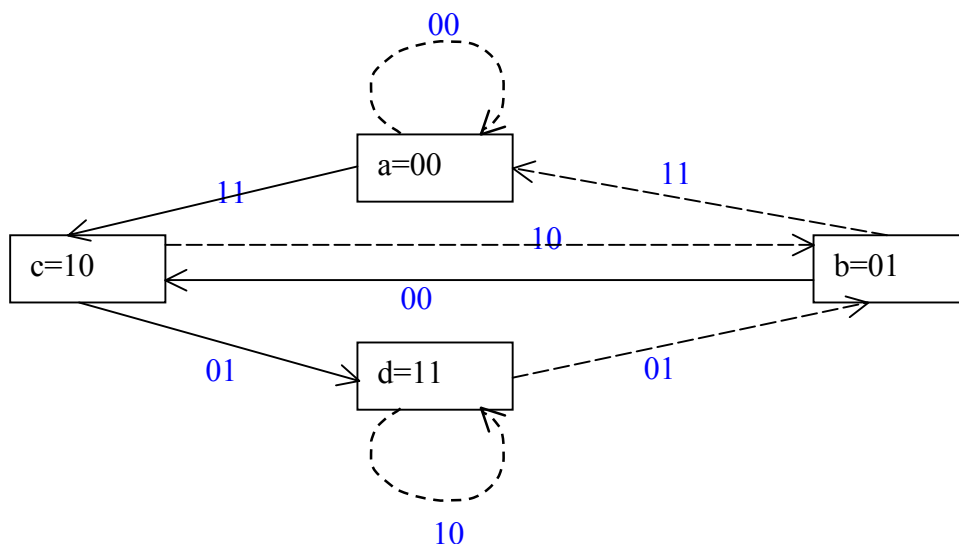
Comme le code est linéaire on peut ne considérer que des chemins partant par exemple de l'état 00. Et la distance libre est le poids de la séquence partant de 00 et y revenant le plus vite possible.



Cette distance représente la distance entre deux chemins les plus proches sans erreur. Plus elle est grande, plus le code sera performant.

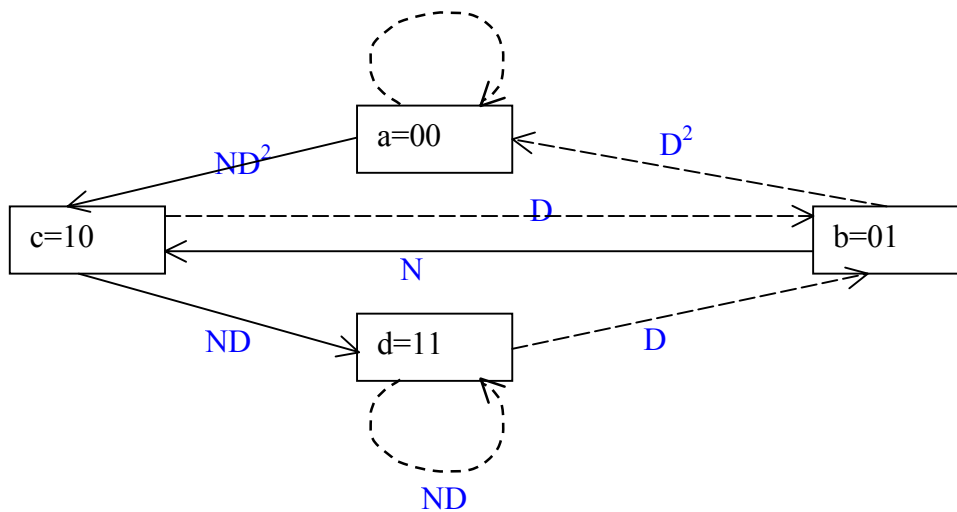
5. Diagramme d'état. Fonction de transfert

On déduit de l'exemple du treillis précédent le diagramme d'état dans lequel le temps n'intervient pas et qui représente les différentes transitions possibles entre états.



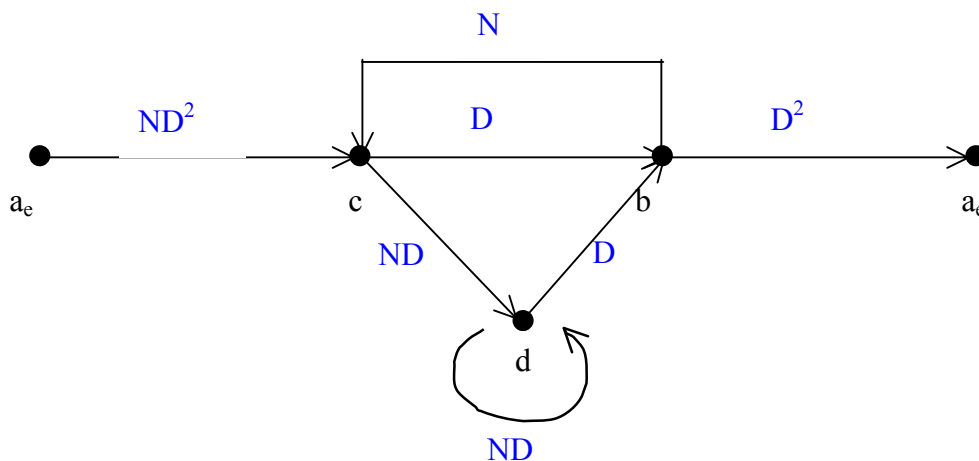
On peut à partir de ce diagramme calculer une fonction de transfert qui dépend des poids des séquences codées, des poids des séquences d'information(entrée).

Pour cela, on introduit, à chaque transition, les variables D affecté de l'exposant égal au poids du mot de code et N affecté de l'exposant égal au poids du mot d'information



On va ouvrir ce diagramme pour avoir un nœud d'entrée et un nœud de sortie : ces deux nœuds seront les mêmes de façon à observer les chemins qui partent d'un nœud et qui y retournent.

Par exemple on prend a_e et a_s le nœud a d'entrée et de sortie :



Les équations de passage sont :

$$c = ND^2 a_e + Nb$$

$$b = Dc + Dd$$

$$d = NDc + NDd$$

$$a_s = Db$$

Après résolution de système :

$$T(D,N) = \frac{a_s}{a_e} = \frac{D^5 N}{1 - 2DN}$$

Soit en développant le numérateur de la fonction de transfert en série :

$$T(D,N) = \sum_{k=0}^{\infty} 2^k D^{k+5} N^{k+1}$$

On en déduit que :

-il existe 2^k chemins partant d'un nœud et aboutissant au même de poids $k+5$ dont la séquence d'entrée a le poids $k+1$

-en particulier si $k=0$, il existe 1 chemin de poids le plus faible 5 : c'est la distance libre.

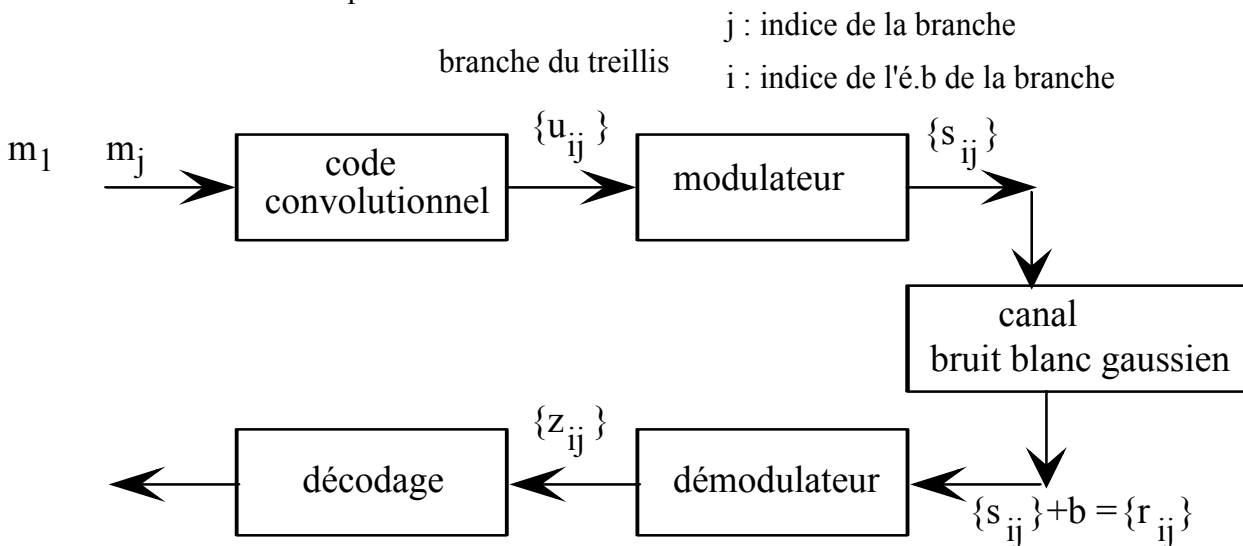
En général :

$$T(D,N) = \sum_{k=d_{lib}}^{\infty} a(h) D^h N^{w(h)}$$

6. Décodage à décision douce en entrée

Un des intérêts de l'algorithme de Viterbi est de pouvoir faire le décodage sans être obligé de prendre une décision en sortie du démodulateur : décision douce.

La chaîne de transmission peut être modélisée comme suit :



Le canal introduit du bruit : donc en sortie du canal on a des signaux non binaires avec des valeurs analogiques (continues).

Le décodage présenté précédemment suppose que le démodulateur a pris une décision 0 ou 1 : décision dure.

On peut effectuer le décodage de Viterbi en supposant que le démodulateur n'a pas pris de décision ou que la valeur du signal est codée sur plusieurs bits : décision souple.

L'algorithme garde le même principe mais les distances utilisées sont les distances euclidiennes :

si (z_k^1, z_k^2) est le couple d'échantillons en sortie du démodulateur à l'entrée du decodeur à $t=k$ et (a_k^1, a_k^2) le couple binaire associée à une branche du treillis

$$M_k = \sum_{i=1}^2 (z_k^i - a_k^i)^2$$

Pour un couple (z_k^1, z_k^2) reçus, on cherche le couple (a_k^1, a_k^2) qui en est à la distance la plus faible.

Par exemple pour une modulation de phase à 2 états (a peut être -1 si 0 est envoyé et +1 si 1 est envoyé), il suffit de comparer les métriques $M_k = -2 \sum_{i=1}^2 (z_k^i a_k^i)$

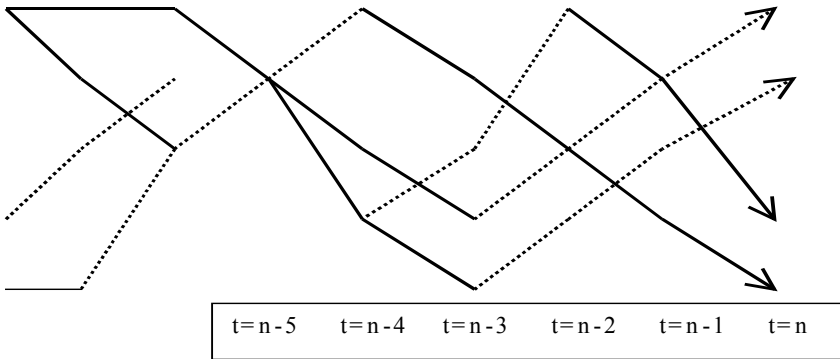
Le décodage corrige alors plus d'erreurs.

7. Limitation du décodage à une fenêtre

En pratique, on n'attend pas que toute la séquence binaire émise soit reçue pour commencer le décodage (trop de mémoire)

On s'aperçoit qu'au bout d'un certain temps $t = n$, les chemins survivants convergent vers un même chemin en amont en $t = n-x$.

On garde tous les chemins survivants de $n-x$ à n et à $t=n$, on décide de décoder l'échantillon de $t=n-x$.



Pour un rendement $1/2$ $n \sim 5$ à 6 fois la longueur de contrainte.

8. Codes convolutionnels poinçonnés :

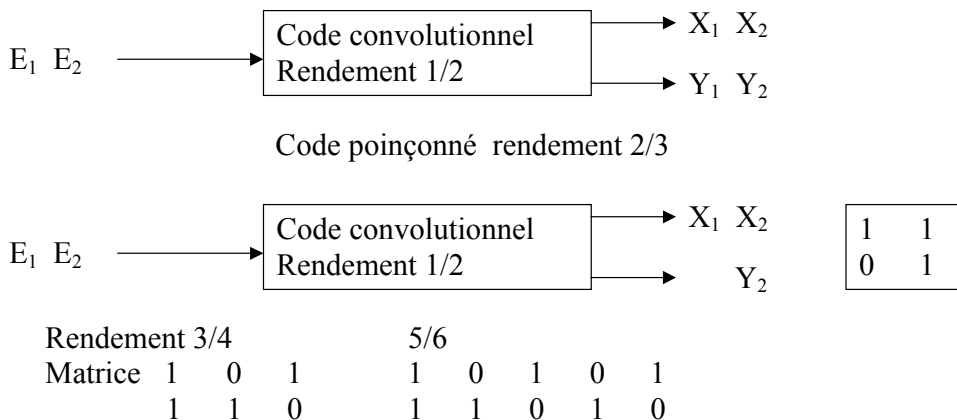
Plus le rendement est faible et plus la longueur de contrainte est grande, plus le code est performant.

Mais, plus le rendement est faible, plus la largeur de bande nécessaire à la transmission doit être importante.

Les rendements des codes décrits sont de la forme $1/n$, donc très faibles, d'ailleurs, on voit rarement des codes convolutionnels de rendement inférieur à $1/2$

Pour fabriquer des codes convolutionnels de rendement supérieur on utilise des codes poinçonnés : ils sont obtenus à partir des codes normaux auxquels on supprime un bit de sortie périodiquement :

ex :



Un code rendement $1/2$ dont on supprime un bit de sortie sur 4 devient un code de rendement $2/3$.

De même pour construire un code de rendement $\frac{3}{4}$, on ne transmet que X_1, Y_1, Y_2, X_3 etc...

9. Choix : comparaison des codes cycliques et des codes convolutionnels

Rappel : rendement = nb bits d'information / nb bits de code

- Le rendement des codes en blocs est élevé ($> 0,9$), celui des codes convolutionnels est assez faible malgré la technique de poinçonnage.
- Les codes cycliques particulièrement les codes de Reed-Solomon corrigent des erreurs arrivant par paquets, ou par rafales (burst) Le code de Reed-Solomon (204,188,8) utilisé dans les normes DVB peut corriger un paquet 8 octets soit 64 bits par trame.
- Si les rafales d'erreurs ont une taille supérieure à la possibilité du code, on effectue un entrelacement des bits avant le décodage : le but est de mélanger les bits pour « casser » les paquets d'erreurs.

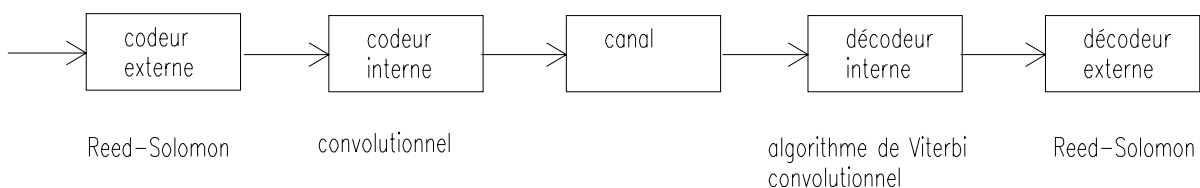
Les rafales d'erreurs peuvent être provoquées par le canal, en particulier les bruits impulsifs (souvent dû aux interrupteurs) dans les câbles, ou par des multi-trajets dont la somme entraîne un signal presque nul.

- Les avantages des codes convolutionnels sont leurs bonnes performances grâce au décodage à décision souple et leur faculté des corriger des erreurs isolées.

Avec des canaux très bruités (Hertziens) pour profiter des avantages des deux types de codes, on peut les concaténer

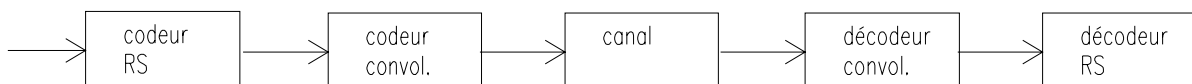
10. Concaténation

Le décodage du code convolutionnel corrige les erreurs indépendantes mais a tendance à laisser des paquets d'erreurs (comme le décodage dépend des données actuelles mais aussi des précédentes, une erreur à t a de l'influence sur le décodage à $t+1$...) qu'un décodage de Reed-Solomon RS pourra alors corriger. Le schéma du codage et du décodage est alors le suivant :



EX TV numérique satellite et terrestre

décodeur externe : RS décodeur interne convolutif $L=7, R=1/2$



On ajoutera un entrelaceur entre les deux codes pour diminuer la taille des paquets d'erreurs à l'entrée de décodeur RS

11. Entrelacement

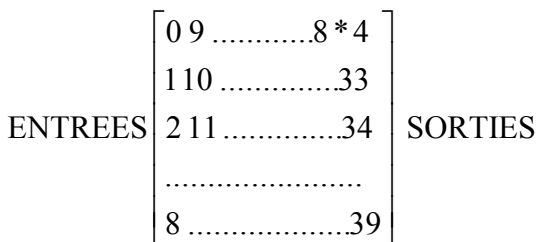
Pour diminuer la taille des rafales pour les erreurs arrivant en rafales, on procède à un entrelacement temporel : ainsi la taille des paquets d'erreurs ne dépassera pas les possibilités de correction du code RS.

Le principe est de changer l'ordre des symboles à l'émission les remettre en ordre à la réception. (opérations duales très similaires)

Ce module est donc inséré entre les deux décodages (et codages)

Principe : opération sur les octets.

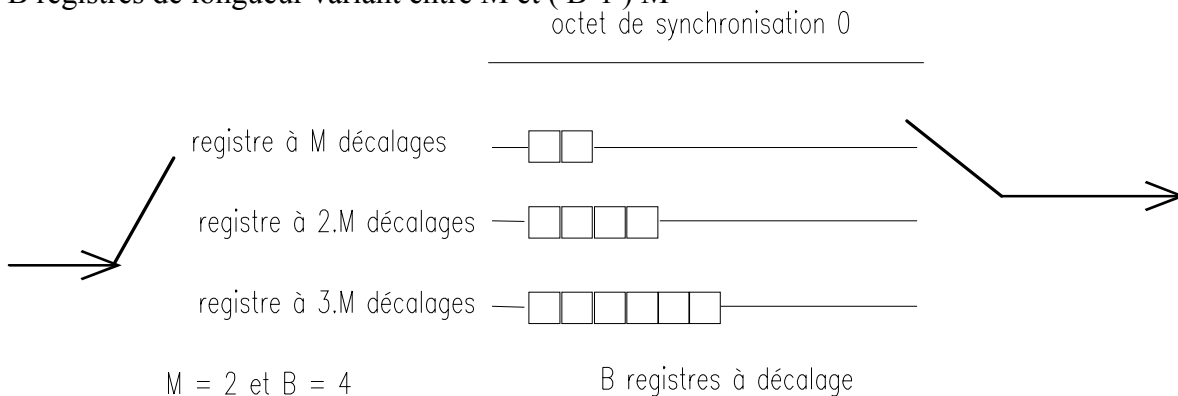
Un premier type d'entrelacement est l'entrelacement par blocs :



Un deuxième type est l'entrelacement convolutionnel

Voici l'exemple de l'entrelacement défini dans les normes DVB télévision satellite et terrestre.

B registres de longueur variant entre M et (B-1) M



Normes DVB : M = 17 et B = 12

Un octet est au plus décalé de 11×17 (soit la taille de la trame excepté l'octet de synchronisation)

12. Bibliographie

- Communications numériques
Alain Glavieux, Michel Joindot Ed Masson
- Introduction aux Codes Correcteurs
Pierre Csillag Ed Ellipses
- Digital Communications : fundamentals and applications
Bernard Sklar Ed Prentice Hall
- Eléments de communications numériques
J.C. Bic, D.D. Duponteil, J.C .Imbeaux Ed Dunod
- La Télévision Numérique MPEG1, MPEG2 et les principes du système européen DVB
Hervé Benoit Ed Dunod