Codes convolutionnels et techniques de codage

Mostafa Belkasmi

2009-2010 ENSIAS

Plan

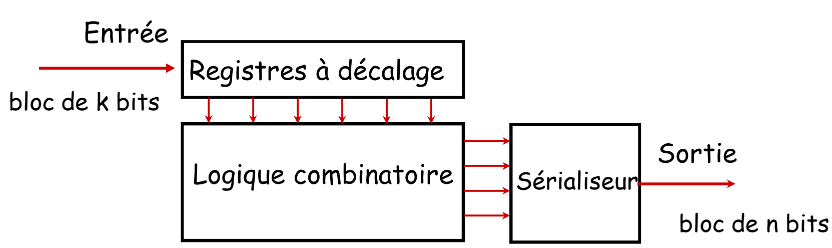
- Codes Convolutionnels
- Autres techniques
- Ingénierie des codes

Introduction

- Codes Convolutionnels (ou convolutifs) aussi importants que les codes cycliques
- Différences :
  - chaque bloc de k bits dépend de 'm+1' blocs de k bits
  - remplacer une séquence d'information de longueur variable par une séquence de code de longueur variable
- Un code est donc l'ensemble des séquences de code.

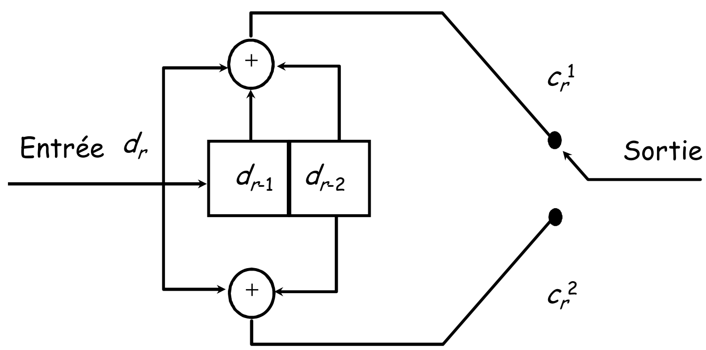
Codes convolutionnels

- Codeur :



- Déf
  - Longueur de contrainte K=m+1
  - Taux de codage (rendement du code) R=k/n

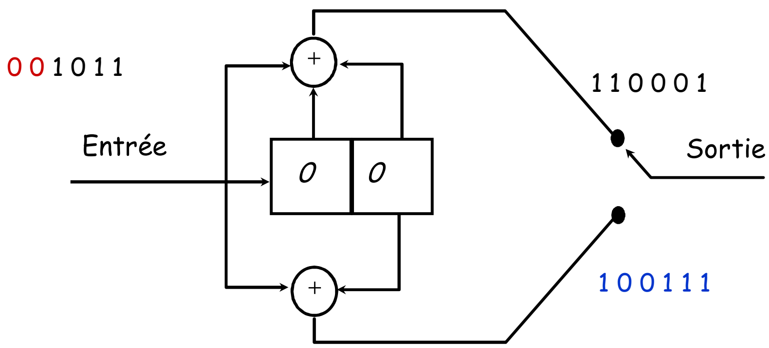
Exemple 1



Exemple 1 paramètres

- Code de taux R = ½ et K=3 (k=1, n=2, m=2)
- Les séquences génératrices du code
  $\mathbf{g}_1$ = [1 1 1]= $7_{(8)}$
  $\mathbf{g}_2$ = [1 0 1]= $5_{(8)}$

Exemple 1 codage



- Séquence info : 1 0 1 1
- Séquence codée : 1 1 0 1 0 0 1 0 1 0 1 1

Caractéristique

- Codes convolutionnels puisqu'il y a un produit de convolution :

Sortie codeur = suite binaire entrée* réponse du codeur

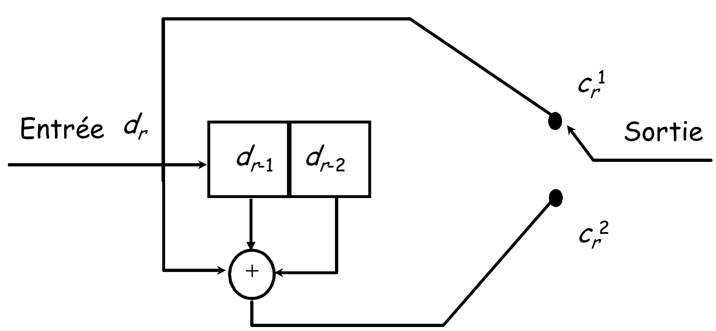
Où la réponse du codeur représente les séquences génératrices du code

Exemple 1 suite

0	0	1	0	1	1	0	0	→ 1 1 0 0 0 1
					1	1	1	
				1	1	1		
			1	1	1			
		1	1	1				
	1	1	1					
1	1	1						

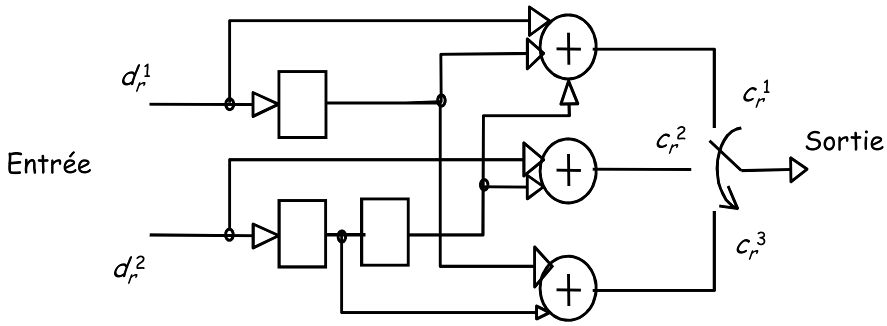
0	0	1	0	1	1	0	0	→ 1 0 0 1 1 1
					1	0	1	
				1	0	1		
			1	0	1			
		1	0	1				
	1	0	1					
1	0	1						

Exemple 2



- Code systématique si les k bits d'information se trouvent parmi les n bits du mot de code courant

Exemple 3

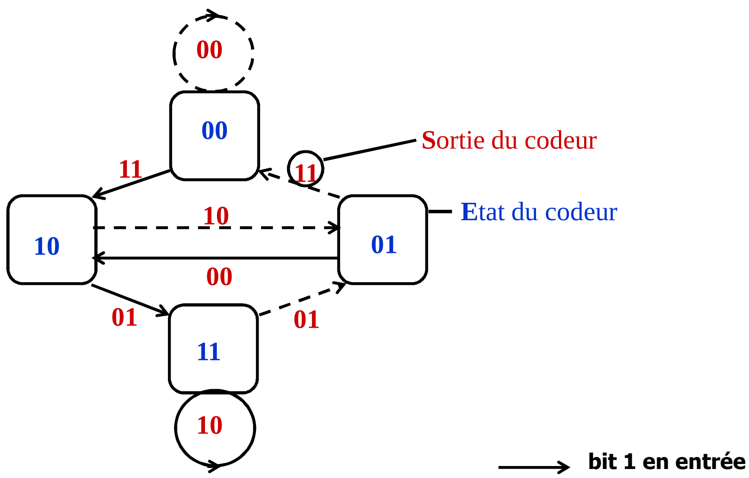


- k=2, n=3, taux=2/3, m=2 et K=3

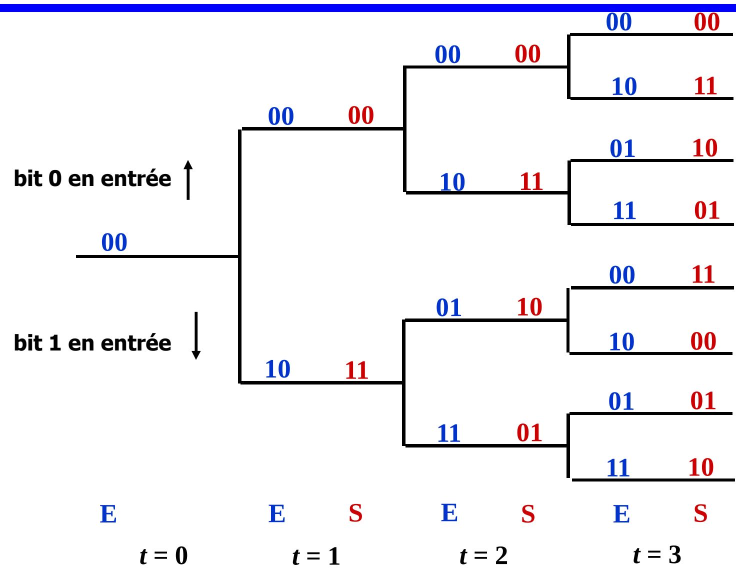
Représentation des codes cv

- Le codeur introduit un effet de mémoire
- On adopte alors la représentation sous forme:
  - —de *diagramme d'état,*
  - —d'*arbre,*
  - —de *treillis.*

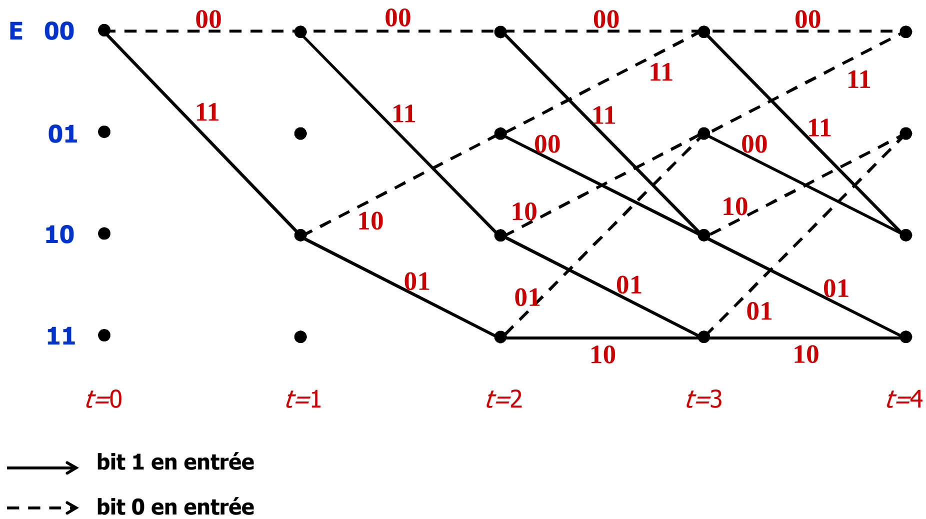
Représentation 1 : Diagramme d'état



Représentation 2 : L'arbre



Représentation 3 : Treillis



Intérêt :

- A toute séquence du code correspond un et un seul chemin dans le treillis partant de l'état 0 et revenant à l'état 0 en fin de parcours : <0, e1, e2, …,0>
- On passe ainsi du langage manipulant des séquences (ou vecteurs) à un langage graphique.

Décodage des CC

- d = (d_1, ..., d_L) séquence d'information avec $d_i = d_i^{(1)} \dots d_i^{(k)}$
- c = (c_1, ... c_N) séquence du code correspondante avec N =L+K-1 et

$$c_i = c_i^{(1)} \dots c_i^{(n)}$$

- Supposons que la séquence c du code a été transmise et désignons par r la séquence reçue :

  r = (r_1, ..., r_N) = c + e

- Le canal est à 2 entrées (0 et 1) et à q sorties.

Métrique de chemin

- Hypothèse : canal discret est sans mémoire. D'où d'après l'indépendance entre les symboles transmis:

$$\Pr(r/c) = \prod_{i=1}^{N} \Pr(r_i / c_i) = \prod_{i=1}^{N} \prod_{j=1}^{n} \Pr(r_i^{(j)} / c_i^{(j)})$$

- La séquence reçue r étant fixe → métrique pour chaque séquence c du code

$$m(r/c) = \log \Pr(r/c) = \sum_{i=1}^{N} \sum_{j=1}^{n} \log \Pr(r_i^{(j)} / c_i^{(j)})$$

- C'est aussi la métrique du chemin correspondant à la séquence c.

Décodage à maximum de vraisemblance:

- *Règle* : "Parmi toutes les séquences du code choisir celle qui maximise la métrique m(r/c) comme étant celle qui a été transmise".

- C'est une méthode optimale mais très complexe en nombre de calcul.

Algorithme de Viterbi

1. Démarrer à t = K - 1. Calculer les métriques de l'unique chemin arrivant à chaque état du treillis. Enregistrer le chemin et la métrique pour chaque état.

2. Incrémenter t. Calculer les métriques de tout les chemins arrivant à un état et ne conserver qu'un seul (celui qui maximise la métrique). Enregistrer le chemin et la métrique pour chaque état.

3. Si t < N = L + K - 1 aller à 2. Sinon stop.

Décision de l'algorithme

- A la fin de l'exécution de cet algorithme, il doit nous rester que 1 seul chemin
  → C'est la séquence du code correspondante à ce chemin qui sera choisie par le décodeur .

Comportement de l'algorithme

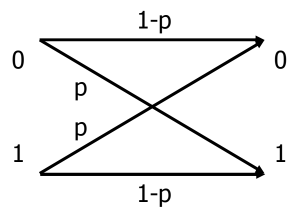
- Déf : branche survivante = permet de prolonger un chemin arrivant à un état à l'instant t-1, en un chemin arrivant à un état à l'instant t
- Donc c'est celle faisant partie du chemin ayant la meilleure métrique arrivant à un état donnée.

Exemples d'application

- Deux cas d'application de cet algorithme sont envisagés:

1. Canal avec décision binaire q = 2 --> CBS.
2. Canal avec décision pondérée q > 2. (par exemple q = 4).

Cas 1 : Canal binaire symétrique

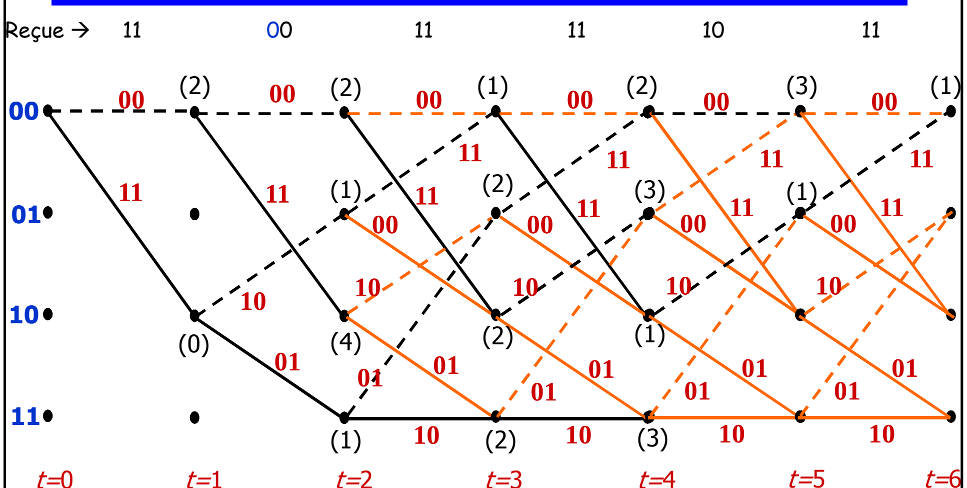


- On démontre que maximiser la métrique revient à minimiser la distance de Hamming entre les chemins.

Exemple

- Séquence info = (1 0 0 1)
- Séquence du code c = (11 10 11 11 10 11)
- Séquence reçue r = (11 00 11 11 10 11) = c+e

Exemple suite



Branche orange : branche non survivante

Décision

- Le chemin optimal (décodé) est :

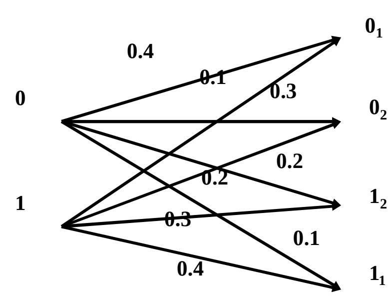
  < 00, 10, 01, 00, 10, 01, 00>

  en remontant petit à petit à partir de l'état final qui est 00.

- Information décodée : (1 0 0 1)

  ➔ correction de l'erreur simple introduite initialement

Cas 2 : Canal avec décision pondérée q = 4.



- Canal entrée binaire et sortie quaternaire

Les métriques

$m(r^{(j)}\|c^{(j)})$	0_1	0_2	1_1	1_2
0	-0.4	-052	-0.7	-1
1	-1	-0.7	-0.52	-0.4

$\mu(r^{(j)}\|c^{(j)})$	0_1	0_2	1_1	1_2
0	10	8	5	0
1	0	5	8	10

- Les métriques de la 2e table sont obtenues par :

$$\mu(r^{(j)}|c^{(j)}) = K1 \log \Pr(r^{(j)}|c^{(j)}) + K2$$

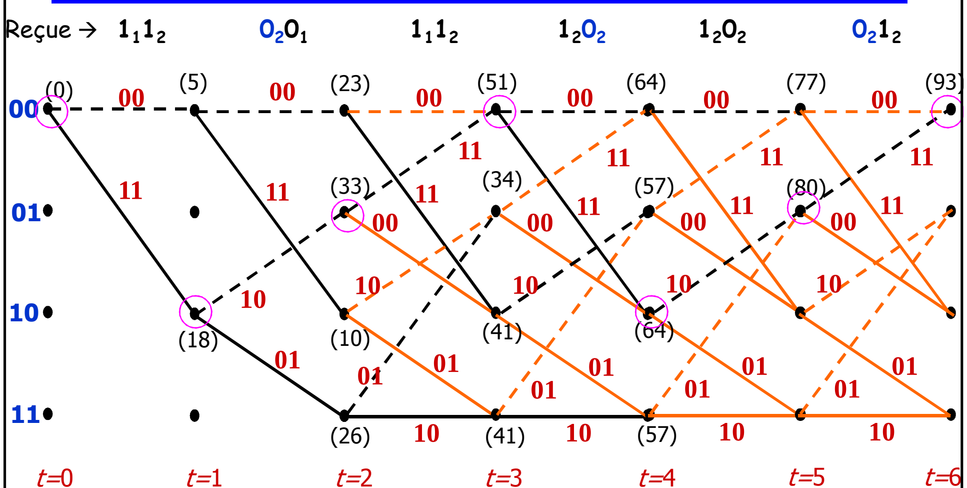
Exemple

- **Soit d = 1 0 0 1 séquence d'info.**
- **c = 1 1 1 0 1 1 1 1 1 0 1 1 séquence du code**
- **r = $1_1\ 1_2\ 0_2\ 0_1\ 1_1\ 1_2\ 1_2\ 0_2\ 1_2\ 0_2\ 0_2\ 1_2$ séquence reçue**

Métrique de branche

$\mu(r\,|c\,)$	$1_1 1_2$	$0_2 0_1$	$1_2 0_2$	$0_2 1_2$
0 0	5=0+5	18=8+10	13	13
0 1	8	8	10	16
1 0	15	15	16	10
1 1	18	5	13	13

Déroulement Viterbi



Branche orange : branche non survivante

Décision

- Le chemin optimal (décodé) est :
  < 00, 10, 01, 00, 10, 01, 00>
  en remontant petit à petit à partir de l'état final qui est 00.

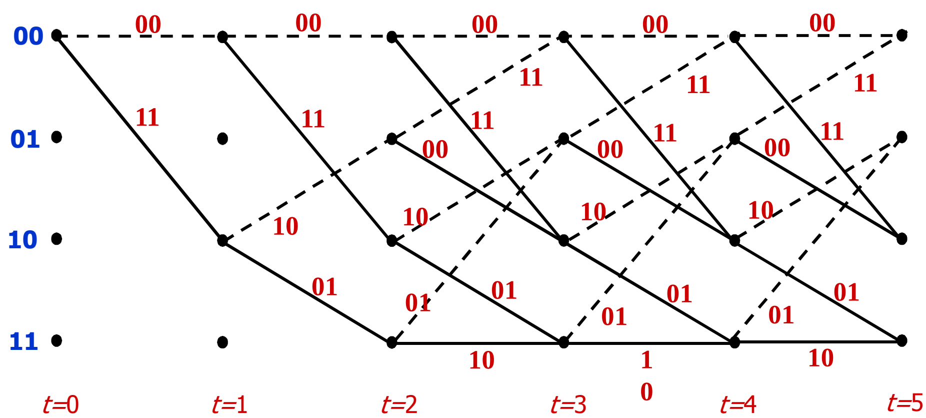
- Information décodée : (1 0 0 1)
  ➔ correction des trois erreurs introduite initialement

Distance libre

- **d_{lib} = min d(c,c') = min W_H(c)**
  avec c et c' deux séquences du code
  et W_H(c) désigne le poids de la séquence c du code.

- ***Exemple :***
  CC (n=2, k=1, K=3) ----> d_{lib} = 5

Exemple



Distance libre

- **Table de CC de taux 1/2 :**

K	g1,g2 (octal)	d_{lib}
3	7, 5	5
4	17, 15	6
5	35, 23	7
6	75, 53	8
7	171, 133	10
8	371, 247	10
9	753, 561	12

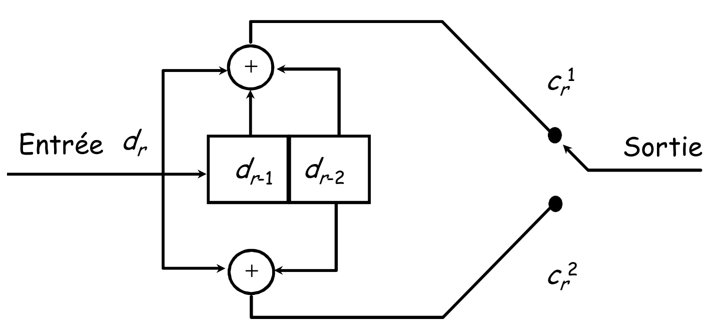
Autres représentations des CC

- Polynômes générateurs
- Matrice génératrice polynomiale
- Matrice génératrice binaire

Polynômes générateurs

- Exemple 1 CC(k=1, n=2,K=3)
- Séquences génératrices :

$\mathbf{g}_1$ = [1 1 1]= $7_{(8)}$ $\mathbf{g}_2$ = [1 0 1]= $5_{(8)}$



Polynômes générateurs

- Séquences génératrices :
  $\mathbf{g}_1$ = [1 1 1]= $7_{(8)}$ $\mathbf{g}_2$ = [1 0 1]= $5_{(8)}$

- Polynômes générateurs :
  $g_1(D)=g_{10} + g_{11}D + g_{12}D^2 = 1 + D + D^2$
  $g_2(D)=g_{20} + g_{21}D + g_{22}D^2 = 1 + D^2$

Polynômes générateurs

- Séquence info : 1 0 1 1
- Séquence codée : 1 1 0 1 0 0 1 0 1 0 1 1
- $d(D)= 1 + D + D^3$
- $c^1(D)= d(D)*g_1(D) = 1 + D^4 + D^5$
- $c^2(D)= d(D)*g_2(D) = 1 + D + D^2 + D^5$

Matrices génératrices

- Matrice génératrice polynomiale

$$G = (1 + D + D^2 \quad 1 + D^2)$$

- Matrice génératrice binaire semi-infinie :

$$G' = \begin{pmatrix} 11\ 10\ 11 & & \\ 00\ 11\ 10\ 11 & & 0 \\ 00\ 00\ 11\ 10\ 11 & & \\ 00\ 00\ 00\ 11\ 10\ 11 & & \\ \text{--------------------} & & \end{pmatrix}$$

Matrices génératrices

- *Séquence info :* 1 0 1 1
- *Séquence codée :* 1 1 0 1 0 0 1 0 1 0 1 1

$$\begin{array}{c} 1 \\ 1 \\ 0 \\ 1 \end{array} \begin{pmatrix} 11\ 10\ 11\ 00\ 00\ 00 \\ 00\ 11\ 10\ 11\ 00\ 00 \\ 00\ 00\ 11\ 10\ 11\ 00 \\ 00\ 00\ 00\ 11\ 10\ 11 \end{pmatrix}$$

- ----------------------
- 11 01 01 00 10 11

Cas général

- Code de paramètres (k ,n, K)

$$G' = \begin{bmatrix} \overset{n}{\leftrightarrow} & & & & & & & \\ k\updownarrow \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \mathbf{g}_{K-1} & & & & \\ & \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \mathbf{g}_{K-1} & & & \\ & & \ddots & & & \ddots & & \\ & & & \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \mathbf{g}_{K-1} & \\ & & & & & & \cdots & \end{bmatrix}$$

- Exemple 3 :

$$\mathbf{g}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{g}_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{g}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

Codes Catastrophiques

- **Définition :**

  Ce sont des codes tels qu'un nombre fini d'erreurs en entrée du décodeur induit un nombre infini d'erreurs en sortie du décodeur.

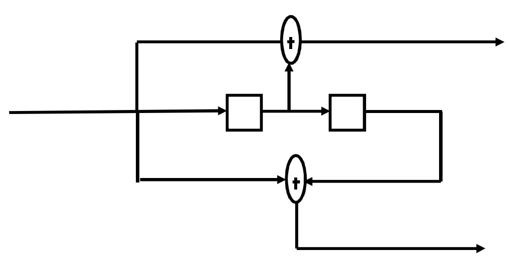
- **Propriétés:**

  Un *CC* est catastrophique s'il existe un chemin fermé de poids nul dans son diagramme d'états.

Codes Catastrophiques

- Codes Systématiques ----> codes non catastrophiques
- *Exemple 1* : CC systématique
- Non catastrophique mais d_{lib} =4

Exemple 2 : CC catastrophique



Exemple 2 : CC catastrophique

- d = séquence d'info. = 0
- e = séquence erreur = 11 01 00 ......
- d'= séquence d'info. décodée = 11 11 11 ....

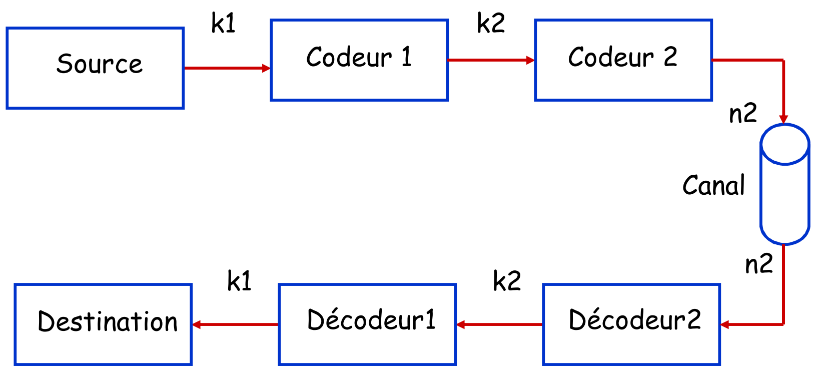
$$G = \begin{pmatrix} 11\ 10\ 01 & & & & & \\ 00\ 11\ 10\ 01 & & & & & \\ 00\ 00\ 11\ 10\ 01 & & & & & \\ 00\ 00\ 00\ 11\ 10\ 01 & & & & & \\ & & - - - - - - - - & & & \end{pmatrix}$$

Autres techniques

- Concaténation
- Entrelacement
- Poinçonnage
- Code Produit

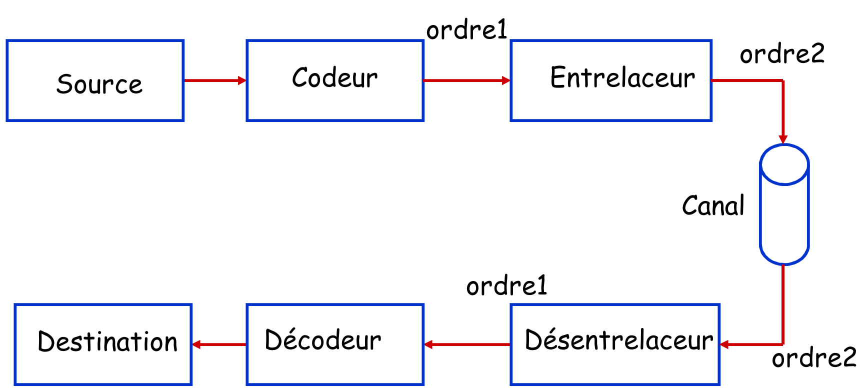
Concaténation

- Pour améliorer encore la qualité de la liaison on met en cascade deux codages successifs.



Entrelacement

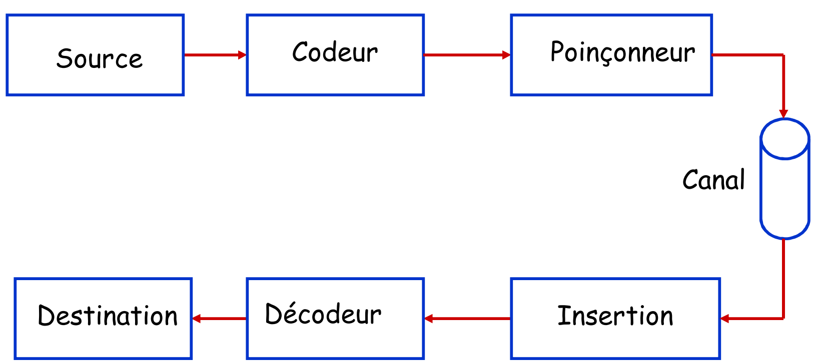
Changer l'ordre des symboles en sortie du codeur
et le remettre juste avant l'entrée du décodeur
afin d'éviter les paquets d'erreurs (bursts)



Poinçonnage

- Le décodeur de Viterbi devient complexe très vite dès qu'on augmente k (2^k branches arrivant à chaque état du treillis)
- On peut réduire artificiellement le taux d'un code de taux 1/2 en éliminant certains bits parmi ceux en sortie du codeur.
- Les bits supprimés sont ensuite remplacés par des zéros en entrée du décodeur.

Poinçonnage



L'intérêt d'une telle opération est d'utiliser toujours le même décodeur qui est celui d'un code de taux 1/2 (qui n'est pas très complexe).

Poinçonnage

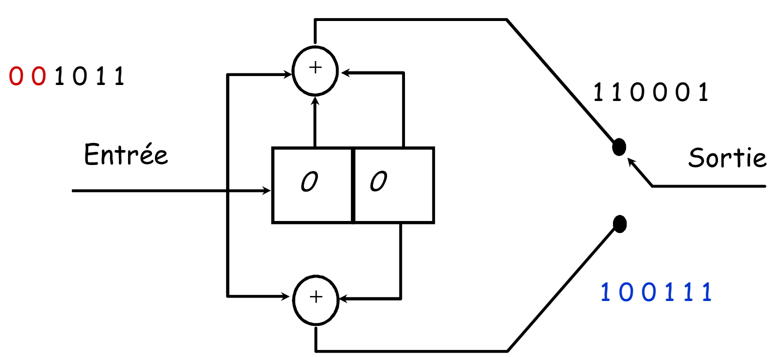
- **Exemple : CC(2,1,3)**
- Ce code engendre 4 bits codés pour chaque paire de bits d'information. Si pour chaque couple de bits d'information au lieu de transmettre les 4 bits, on ne transmet que les 3 premiers. On obtiendrai ainsi un code de taux 2/3.

Emission x x x x x x x x x x x x

Bits supprimés

Réception x x x 0 x x x 0 x x x 0

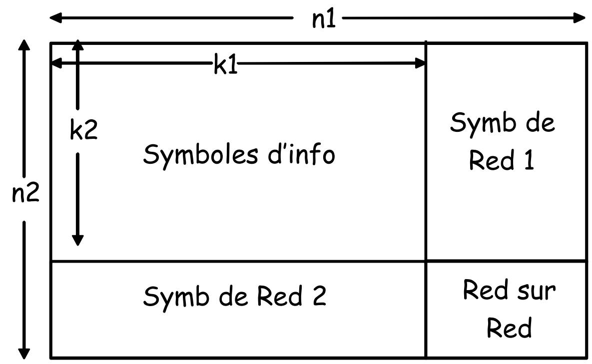
Poinçonnage



- Séquence info : 1 0 1 1
- Séquence codée : 1 1 0 1 0 0 1 0 1 0 1 1
- Séquence transmise : 1 0 1 0 1 0 0 1 1

Produit de codes

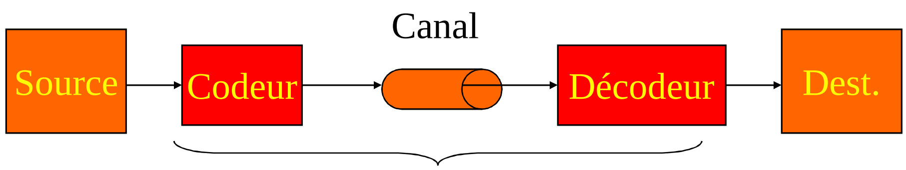
- Produit de deux codes linéaires systématiques C1(n1,k1) et C2(n2,k2).
- Chaque mots du code produit C(n1.n2, k1.k2) est une matrice formée par des lignes dans le code C1 et des colonnes dans le code C2.



Ingénierie du codage

- Conception des schémas de codage appropriés
- Obtention des performances d'un schéma de codage par analyse ou par simulation
- Tracer des courbes donnant le Taux d'erreurs en fonction du rapport signal sur bruit du canal sous l'étude.

Simulation de la chaîne de transmission

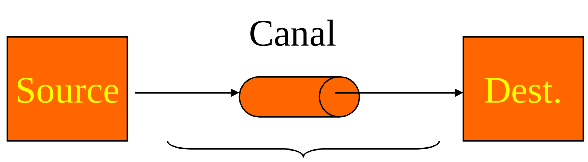


„Taux d'erreurs résiduelles"

Le composant Canal :

- plusieurs modèles existent: Gauss, Rayleigh, GSM, ….
- Ont comme paramètre le RSB

Courbe de performance



„Taux d'erreurs"

On obtient ainsi la courbe de performance de la modulation sur le canal en question mais sans le codage. A un RSB on associe un TEB.

Courbes de Performance