

1 Codage de source

1.1 Entropie

Symboles $\{a_0, a_1, a_2, \dots, a_{n-1}\}$

Probabilités : $\{P(a_0), P(a_1), P(a_2), \dots, P(a_{n-1})\}$ Information contenue dans un message :

$$I(a_k) = -\log_2(P(a_k))$$

Entropie de la source (moyenne du contenu d'information):

$$H = -\sum_{i=0}^{n-1} P(a_i) I(a_i)$$

1.2 Lempel Ziv

Entrée : AABABBBABAABBBABBABB
Ségmentation : A|AB|ABB|B|ABA|ABAB|BB|ABBA|BB

Position (Adresse)	1	2	3	4	5	6	7	8	9
Séquence	A	AB	ABB	B	ABA	ABAB	BB	ABBA	BB
Représentation	A	1B	2B	B	2A	5B	4B	3A	7
Code	0	11	101	001	0100	1011	1001	0110	0111

2 Transmission sans-fil

2.1 Formule de Friis

Dans un cas idéal, sans trajets multiples

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi R} \right)^2$$

en dB :

$$\underbrace{(P_r)_{\text{dB}} - (P_t)_{\text{dB}}}_{Att_{\text{dB}}} = (G_t)_{\text{dB}} + (G_r)_{\text{dB}} + 20 \log_{10} \left(\frac{\lambda}{4\pi R} \right)$$

$$(x)_{\text{dB}} = 10 \log_{10}(x)$$

A noter que la puissance de 2 a été enlevée et le 10 log remplacé par 20 log

2.2 Capacité du canal et efficacité spectrale

$$\frac{C}{B} = \log_2 \left(1 + \frac{S}{N} \right) = \log_2 \left(1 + \frac{E_b R}{N_0 B} \right)$$

La limite est donnée par

$$\frac{E_b}{N_0} = B \frac{2^{\frac{C}{B}} - 1}{R}$$

2.3 Autres

$$1 \text{ W} = 30 \text{ dBm}$$

3 Codage de canal

$$\begin{array}{c} \overbrace{\quad k \quad} \\ 010110 \end{array} \longrightarrow \begin{array}{c} \overbrace{\quad n \quad} \\ 11010010 \end{array}$$

$$\begin{array}{l|l} k & \text{Nombre de bits à transmettre (information)} \\ n & \text{Nombre de bits transmis } n \geq k \end{array}$$

Information (bits à transmettre)

$$x = (x_{k-1} \ x_{k-2} \ \dots \ x_1 \ x_0) \in \mathbb{R}^{1 \times k}$$

Matrice systématique (identité à gauche).

$$G_S = (I_k \ P) = \left(\begin{array}{cccc|ccc} 1 & 0 & \dots & 0 & x & \dots & x \\ 0 & 1 & \dots & 0 & x & \dots & x \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & x & \dots & x \end{array} \right) \in \mathbb{R}^{k \times n}$$

$$H_s = (P^T \ I_{n-k}) \in \mathbb{R}^{n-k \times n}$$

3.1 Propriétés

$$G_S H_S^T = 0$$

3.2 Codes linéaires

La somme de deux codes valides donne un nouveau code valide

3.3 Codes cycliques

Un décalage vers la gauche ou vers la droite donne un autre code valide.

Un décalage d'un mot-code de longueur n dans $\mathbb{R}_n[X]$ est similaire à une multiplication par X

3.3.1 Générateur

Un générateur permet, par des décalages cycliques et des sommes d'obtenir tous les mots-code.

Pour trouver un générateur

3.3.2 Encodage

Encodage systématique :

$$x \xrightarrow{\text{encodage}} xX^{n-k} + \text{reste} \left(\frac{xX^{n-k}}{g(x)} \right) = y$$

Encodage "standard" :

$$x \xrightarrow{\text{encodage}} xg(x)$$

3.3.3 Vérification

On va simplement chercher à savoir si le message est correct ou non.

Vérification systématique (on vérifie simplement que le message soit égal au reste, comme pour l'encodage)

$$y[0 : k] == \text{reste} \left(\frac{y[0 : k]X^{n-k}}{g(x)} \right)$$

Vérification "standard"

$$s(x) = e(x) = \frac{y}{g(x)} \quad s(x) = 0 \longrightarrow \text{ok}$$

3.3.4 Correction d'erreurs

On va utiliser une matrice de contrôle (typiquement pour les codes BCH) qui va indiquer la position de(s) erreur(s)

3.3.5 Décodage

Dans le cas où le message est correct, on va effectuer le décodage.

Décodage systématique (on récupère simplement les bits dans le message tels-quels):

$$\hat{x} = y[0 : k]$$

Décodage "standard" :

$$\hat{x} = \frac{y}{g(x)}$$

4 Algorithme de Viterbi (décodage)

3.4 Générateur \leftrightarrow matrice

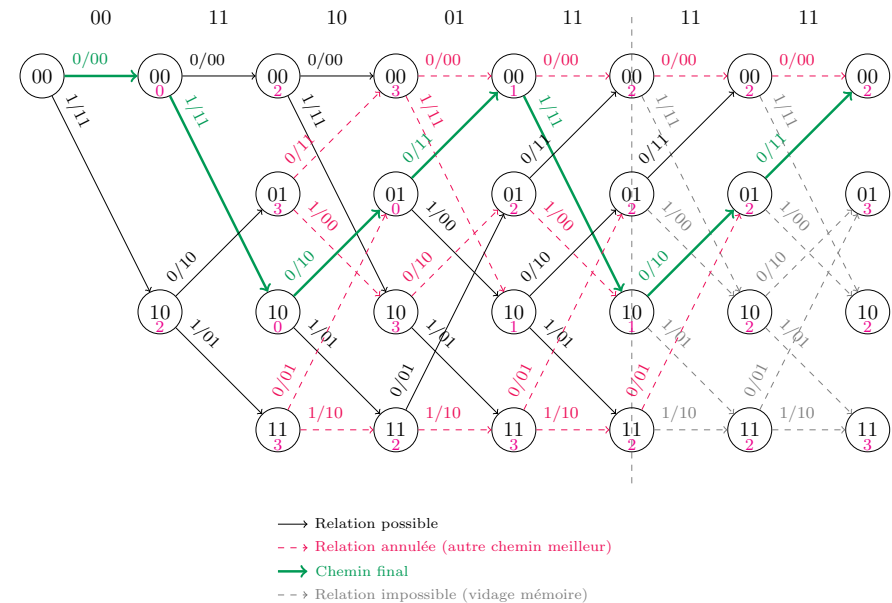
On peut créer la matrice en effectuant des décalages cycliques du générateur $g(x)$

$$\begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0 \\ 0 & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{pmatrix}_{k \times n}$$

Avec le nombre de lignes correspondant à la longueur du message. Cette matrice n'est pas systématique. Il faut

la manipuler si on veut obtenir la version systématique. On effectue ensuite des combinaisons linéaires sur les lignes pour obtenir la matrice systématique.

3.5 Codage convolutionnel



Soit le code reçu 00 11 10 01 11 11 11

Le code corrigé est donc 00 11 10 11 11 10 11. Qui correspond à la séquence 0100100 (le message est 01001, les deux derniers bits sont le vidage de la mémoire).