1 Signaux et systèmes en temps discret

$$x(n) = x_{\text{continu}}(nT_s)$$

Avec T_s la période d'échantillonnage

Impulsion unité

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$

Saut unité

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

Durée finie : échantillons égaux à 0 en dehors d'un intervalle donné

Durée infinie : saut unité, oscillation, etc...

Séquence à droite : 0 pour $n < n_0$

Séquence à gauche : 0 pour $n > n_0$

Représentation exponentielle d'un signal périodique

$$e^{\sigma n + jn\omega_0} = e^{\sigma n}\left(\cos(n\omega_0) + j\sin(n\omega_0)\right)$$

Pas d'amortissement si $\sigma = 0$

1.1 Propriétés des systèmes

Lorsqu'on passe un signal x dans un système T on obtient une sortie y

$$y(n) = T\left[x(n)\right]$$

$$\boxed{y(n) = \sum_{k=0}^{q} b(k)x(n-k) - \sum_{k=1}^{p} a(k)y(n-k)}$$

$$\begin{cases} \text{IIR} & a(k) \neq 0 \quad \forall k \in 1, ..., p \\ \text{FIR} & a(k) = 0 \quad \forall k \in 1, ..., p \end{cases}$$

$$p > q \longrightarrow \text{ causal}$$

p est l'ordre du système

Linéarité

$$T\left[ax_1(n) + bx_2(n)\right] = aT\left[x_1(n)\right] + bT\left[x_2(n)\right]$$

Invariance temporelle (ou shift)

$$y(n - n_0) = T\left[x(n - n_0)\right]$$

Causalité $y(n)$ dépend uniquement de $y(n-k)$ et $x(n)$

Stabilité BIBO : borné en entrée et borné en sortie. Vérifié si

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty$$

Inversibilité Si on peut déterminer $x(n)$ à partir de $y(n)$

$$x_1(n) \neq x_2(n) \longrightarrow y_1(n) \neq y_2(n)$$

1.2 Convolution

1.2.1 Propriétés

Linéarité

$$x(n) * (\alpha y(n) + \beta w(n)) = \alpha x(n) * y(n) + \beta x(n) * w(n)$$

Invariance temporelle

$$w(n) = x(n) * y(n) \Longleftrightarrow x(n) * y(n-k) = w(n-k)$$

Commutativité

$$x(n) * y(n) = y(n) * x(n)$$

Associativité

$$\Big(x(n) * h(n)\Big) * w(n) = x(n) * \Big(h(n) * w(n)\Big)$$

Multiplication par une impulsion unité

$$h(n) * d(n) = h(n)$$

1.3 DTFT

$$\boxed{X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{n=+\infty} x(n)e^{-jn\omega}}$$

Cas spéciaux :

$$x(n) = e^{jn\omega_0} \longrightarrow X\left(e^{j\omega}\right) = 2\pi u(\omega - \omega_0) \quad |\omega| < \pi$$

$$x(n) = u(n) \longrightarrow X\left(e^{j\omega}\right) = \frac{1}{1 - e^{-j\omega}} + \pi u(\omega) \quad |\omega| < \pi$$

Impulsion unité :

$$H(e^{j\omega}) = \sum_{n=-\infty}^{+\infty} h(n)e^{-jn\omega}$$

H décrit la réponse fréquentielle du système

1.3.1 Propriétés

Périodicité $X(e^{j\omega})$ est périodique en 2π

Signal réel si $x(n)$ réel alors

$$x(e^{j\omega}) = X^*\left(e^{-j\omega}\right)$$

Décomposition amplitude-phase

$$X(e^{j\omega}) = \left|X(e^{j\omega})\right|e^{j\Phi_x(\omega)}$$

Pour un signal réel $\left|X(e^{j\omega}\right|$ est paire et Φ impaire

1.3.2 Opérations

Convolution

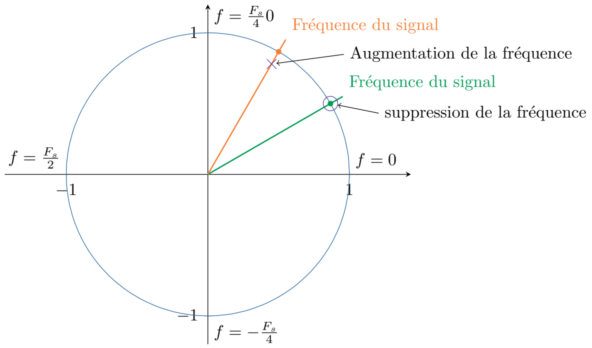
$$y(n) = x(n) * h(n) \Longleftrightarrow Y(e^{j\omega}) = X(e^{j\omega}) \cdot H(e^{j\omega})$$

Modulation

$$y(n) = x(n) \cdot h(n) \Longleftrightarrow Y(e^{j\omega}) = X(e^{j\omega}) * H(e^{j\omega})$$

1.4 Transformée en z

$$X(z) = \sum_{n=-\infty}^{+\infty} x(n) z^n \qquad z = re^{j\omega}$$



Si une fréquence tombe pile sur un zéro (sur le cercle unité) alors elle sera complètement annulée. Un pôle proche du cercle unité va augmenter les fréquences proches de ce pôle.

1.5 Exemple

$f[k]$	$F(z) = Z(f[k])$
$\Delta[k]$	1
$\epsilon[k]$	$\frac{z}{z-1}$
kh	$\frac{hz}{(z-1)^2}$
$\frac{1}{2}(kh)^2$	$\frac{h^2}{2}\frac{z(z+1)}{(z-1)^3}$
a^{kh}	$\frac{z}{z-a^h}$
$e^{-akh}\epsilon[k]$	$\frac{z}{z-e^{-at}}$
$a^{\lvert k\rvert}$	$\frac{1-a^2}{(1-az^{-1})(1-az)}$

1.5.1 Propriétés

Linéarité

$$Z\{\alpha x(n) + \beta y(n)\} = \alpha X(z) + \beta Y(z)$$

Décalage temporel

$$Z\{x(n-N)\} = z^{-N} X(z)$$

	$x(n)$	$X(z)$
Délai	$x(n-n_0)$	$z^{-n_0}X(z)$
Multiplication par α^n	$a^n x(n)$	$X(z/\alpha)$
Conjugué	$x^*(n)$	$X^*(z^*)$
Inversion de temps	$x(-n)$	$X(z^{-1})$
Convolution	$x(n) * w(n)$	$X(z)W(z)$
Multiplication par n	$nx(n)$	$-z\frac{d}{dz}X(z)$

Table 1: Opérations

$$Z\{\delta(n)\} = 1$$

1.6 Équation aux différences

$$\underset{\text{degré relatif}}{d} = \underset{\text{deg(denominateur)}}{n} - \underset{\text{deg(numérateur)}}{m}$$

$$G(z) = \frac{Y(z)}{X(z)}$$

$$G(z) = \frac{b_0 z^{-d} + b_1 z^{-d-1} + b_2 z^{-d-2} + \cdots + b_m z^{-d-m}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}$$

$a \longrightarrow$ poles, $b \longrightarrow$ zéros

$$|a_i| < 1 \longrightarrow \text{stable} \quad \forall i \in [1, n]$$

2 Algèbre linéaire

2.1 Matrices

$$x = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

$$y(n) = h^T x(n) = x^T h \qquad \text{(un élément à la fois)}$$

2.1.1 Transposée hermitienne

$$\left(A^H\right)^H = A$$

$$(A+B)^H = A^H + B^H$$

$$(AB)^H = B^H A^H$$

2.1.2 Rang

Le rang d'une matrice A donne le nombre de colonnes linéairement indépendantes qu'elle contient. Si le rang est **plein** alors

$$\text{rank}(A) = \rho(A) = n$$

2.1.3 Inverse

$$(AB)^{-1} = A^{-1}B^{-1}$$

$$(A^H)^{-1} = (A^{-1})^H$$

Inversible si $\det(A) \neq 0$

2.1.4 Déterminant

$$\det(AB) = \det(A)\det(B)$$

$$\det\left(A^T\right) = \det(A)$$

$$\det(\alpha A) = \alpha^n \det(A)$$

$$\det\left(A^{-1}\right) = \frac{1}{\det(A)}$$

2.1.5 Pseudo-inverse

Forme ”norme minimale” (sous-déterminé)

$$A \in \mathbb{R}_{n\times m} \qquad n < m$$

$$\boxed{A^+_{nm} = A^H(AA^H)^{-1}}$$

Permet de calculer la solution minimale

$$x = A^+b$$

Forme ”moindres carrés” (sur-déterminé)

$$\boxed{A^+_{mc} = (A^HA)^{-1}A^H}$$

2.1.6 Norme

$$||x||_1 = \sum_{i=1}^{N} |x_i|$$

$$||x||_2 = \sqrt{\sum_{i=1}^{N} |x_i|^2}$$

Permet aussi de calculer la distance : $||x-y||_2$

$$||x||_\infty = \max |x_i|$$

2.1.7 Produit scalaire

$$\langle a, b\rangle = a\cdot b = ||a||||b||\cos\theta$$

2.1.8 Espace vectoriel

$$\mathbf{v} = \sum_{i=1}^{N} \alpha_i \mathbf{v}_i$$

Si la base est orthogonale et si on utilise autant de coefficients qu'il y a de points dans le signal, alors la reconstruction est parfaite.

2.1.9 Projection

$$\hat{b} = A^+_{mc}x_0 = P_Ab$$

Par exemple

$$\begin{bmatrix}2 & 1\\1 & 2\\1 & 2\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}1\\1\\1\end{bmatrix}$$

$$A^HA = \begin{bmatrix}6 & 5\\5 & 6\end{bmatrix}$$

Solution des moindres carrés donnée par

$$x_0 = (A^H)^{-1}A^Hb = \frac{1}{11}\begin{bmatrix}6 & -5\\-5 & 6\end{bmatrix}\begin{bmatrix}2 & 1 & 1\\1 & 2 & 1\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = \frac{4}{11}\begin{bmatrix}1\\1\end{bmatrix}$$

$$e = b - Ax_0 = \frac{1}{11}\begin{bmatrix}-1\\-1\\3\end{bmatrix}$$

2.1.10 Valeurs propres

$$Av = \lambda v$$

Pour trouver les valeurs propres, on cherche les racines de

$$p(\lambda) = \det(A - \lambda I) = 0$$

$$V = \begin{bmatrix}v_1 & v_2 & \cdots & v_n\end{bmatrix} \qquad \Lambda = \mathrm{diag}\{\lambda_1, \lambda_2, ..., \lambda_n\}$$

$$AV = V\Lambda$$

Vecteurs propres On va chercher les vecteurs v tels que

$$Av = \lambda v \longleftrightarrow (A - \lambda I)\,\vec{v} = \vec{0}$$

Reconstruction à partir des valeurs et vecteurs propres

$$M = V\Lambda V^{-1}$$

2.2 Approximation

Autant de lignes que de points et autant de colonnes que de fonctions de base

$$A = \begin{bmatrix}1 & x_{11} & \sin(2\pi x_{11}) & \cos(2\pi x_{11})\\1 & x_{12} & \sin(2\pi x_{12}) & \cos(2\pi x_{12})\\\vdots & \vdots & \vdots & \vdots\\1 & x_{1n} & \sin(2\pi x_{1n}) & \cos(2\pi x_{1n})\end{bmatrix}$$

Vecteur au sens des moindres carrés :

$$\Theta = (A^TA)^{-1}A^Tb = A^+b$$

Pour une ellipse avec l'équation

$$ax^2 + by^2 + cx + dy + exy = 1$$

on a

$$A = [x_i^2, y_i^2, x_i, y_i, x_iy_i] \qquad b = \begin{pmatrix}1\\\vdots\\1\end{pmatrix}$$

3 Processus aléatoires à temps discrets

Variable aléatoire (pile ou face)

$$P_r\{l\} = 0.5 \qquad P_r\{H\} = 0.5$$

Ω est l'ensemble des possibilités.

3.1 Fonction de répartition

Par exemple l'intégrale d'une gaussienne

$$F_x(\alpha) = P_r\{x \leq \alpha\}$$

3.2 Densité de probabilité

Par exemple une gaussienne

$$f_x(\alpha) = \frac{d}{d\alpha}F_x(\alpha)$$

$$\int_{-\infty}^{\infty} f_x(\alpha)d\alpha = 1$$

3.2.1 Exemple

$$x = \{1, 2, 3, 4, 5, 6\}$$
$$P_r\{x = 1\} = \frac{1}{6}$$
$$F_x(x) = P_r\{x \leq \alpha\}$$
$$P_r\{x = 3\} = \int_{3-\epsilon}^{3+\epsilon} f_x(\alpha) d\alpha$$

3.3 Espérence

Moyenne pondérée de chaque possibilité

$$E\{x\} = 3.5 \qquad \text{(pour un dé)}$$
$$E\{x\} = \sum_k \alpha_k P_r\{x = \alpha_k\}$$

3.4 Variance

$$\sigma^2 = \text{var} = E\{(x(n) - \underbrace{E\{x\}}_{m_x(n)})^2\}$$

Avec σ la déviation standard. On a aussi

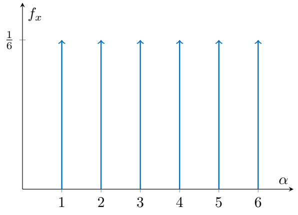
$$\text{var}(x) = \sigma^2 = E\{x^2\} - E^2\{x\}$$
$$\text{var}\{x + y\} = \text{var}\{x\} + \text{var}\{y\}$$

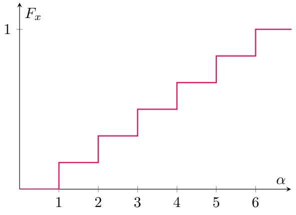
3.5 Biais

$$B = \theta - E\{\hat{\theta}_N\}$$

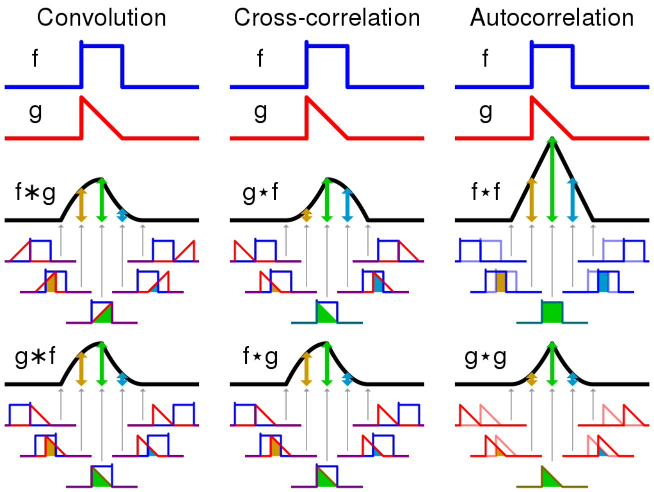
Différence entre la moyenne mesurée (par exemple sur un ensemble de pile ou face) et l'espérance théorique (0.5 dans ce cas)

$$\lim_{N\to\infty} E\{\hat{\theta}\} = \theta \longrightarrow \text{ non biaisé}$$





3.6 Corrélation



$$r_{xy}(k, l) = E\{x(k)y^*(l)\}$$

Si x et y sont indépendants :

$$r_{xy}(k, l) = 0$$

Avec * le conjugué complexe (si les valeurs sont complexes). Si la corrélation est le produit des moyennes, alors les deux variables sont indépendantes

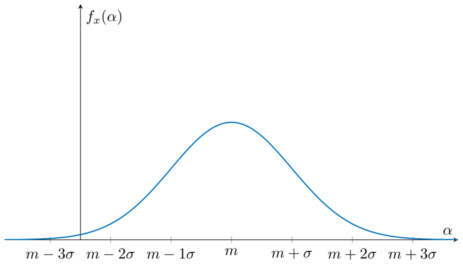
$$r_{xy} = E\{x\}E\{y^*\} = m_x m_y^* \longrightarrow \text{ indépendants}$$

3.7 Coefficient de corrélation

$$\rho_{xy} = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y} \leq 1$$

3.8 Gaussienne

$$f_x(\alpha) = \frac{1}{\sigma_x\sqrt{2\pi}} \exp\left(-\frac{(\alpha - m_x)^2}{2\sigma_x^2}\right)$$



3.9 Bruit blanc

Avec déviation standard de σ

$$P = \sigma^2$$

Le spectre est plat entre $-\frac{F_s}{2}$ et $\frac{F_s}{2}$ avec une amplitude de σ^2

4 Systèmes de communication

Possibilité d'effectuer une compression importante pour des données prévisibles / redondantes. Deux caractéristiques de l'information :

1. Meaning
2. Suprise

Avec E certain, $I(E) = 0$

$$p(E) = 1 \longrightarrow I(E) = 0$$

Avec un événement peu probable F, $I(F) > 0$.
Pour deux événements non-liés :

$$I(E_1 \cap E_2) = I(E_1) + I(E_2)$$

$$\boxed{I(E) = -K \log_a(P(E))}$$

a vaut souvent 2 (binaire).

4.1 Exemple équiprobable

$$p(A) = \frac{1}{2} \quad p(B) = \frac{1}{2} \quad p(C) = \frac{1}{8} \quad p(D) = \frac{1}{8}$$

4.1.1 Entropie H

$$H = \frac{1}{2}1 + \frac{1}{2}2 + \frac{1}{8}3 + \frac{1}{8}3 = \frac{7}{4} \text{ bits par symbole}$$

$H = 0$ pas d'information (certain), $H = H_{max}$ tous les symboles ont la même probabilité.

5 Codage de source

Aucune connaissance de la source, son rôle est de minimiser la redondance

Dilemme : Si on supprime des bits dans la source, on doit en rajouter dans le canal pour augmenter la robustesse.

5.1 Entropie

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

$$I(a_k) = -\log_2(P(a_k)) \qquad \text{[bits]}$$

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

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

De la source → ne pas prendre en compte si on envoie x symboles. On parle uniquement de la source

5.1.1 Combinaison de sources

Si une source S_3 fait un choix entre S_1 et S_2 (avec α la chance de S_1). L'entropie sera

$$H_3 = \alpha\Big(H_1 - \log_2(\alpha)\Big) + (1-\alpha)\Big(H_2 - \log_2(1-\alpha)\Big)$$

Pour construire l'arbre, on multiplie les probabilités des symboles de S_1 par α et les probabilités des symboles de S_2 par $(1-\alpha)$

5.2 Méthodes de codage

1. Longueur fixe (comptage binaire "standard")
2. Optimal (100% efficace) pour puissances de 2 ($\frac{1}{2}$, $\frac{1}{4}$, ..., $2 \times \frac{1}{2^{n-1}}$)

Symbole	prob	Code A	Code B
s_0	1/2	1	0
s_1	1/4	01	10
s_2	1/8	001	110
$\vdots$	$\vdots$	$\vdots$	$\vdots$
s_{n-2}	$1/2^{n-1}$	00...01	11...10
s_{n-1}	$1/2^{n-1}$	00...00	11...11

Les deux codes (A et B) sont valides

5.3 Efficacité du code

Entropie divisée par la longueur moyenne (déterminée à partir de la méthode de codage)

$$\frac{H}{\bar{l}}$$

Si n symboles sont combinés, on divise par n la longueur moyenne (pour connaître la longueur moyenne correspondant à un symbole).

5.4 Décodage

Décodage instantané : Aucun mot-code n'est prefix d'un autre

Inégalité de Kraft-McMillan

$$\sum_{i=0}^{n-1} 2^{-\text{longueur}(s_i)} \neq 1 \longrightarrow \text{Pas instantané}$$

Si = 1 cela ne veut pas forcément dire que le code est instantané

5.5 Huffman

Attention au 1 en **haut** ou en **bas** (les exemples sont données avec le 1 en haut) (les deux sont utilisés dans le cours). Ensuite on construit l'arbre avec le nouvel élément "en haut" ou "en bas" (précisé dans l'exo en principe).

5.5.1 Longueur moyenne

$$\bar{l} = \sum P(a_k) \cdot \text{longueur du code}(a_k)$$

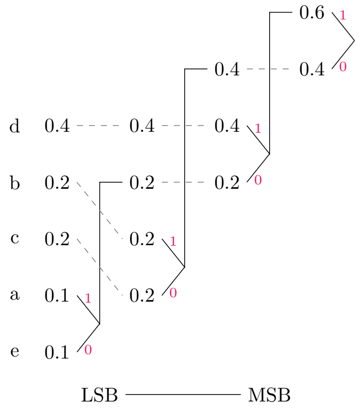
5.5.2 Variance de la longueur

$$\text{var} = \sigma^2 = \sum P(a_k) \cdot \left(\text{longueur du code}(a_k) - \bar{l}\right)^2$$

Si on doit départager deux codes, une variance plus faible est meilleure

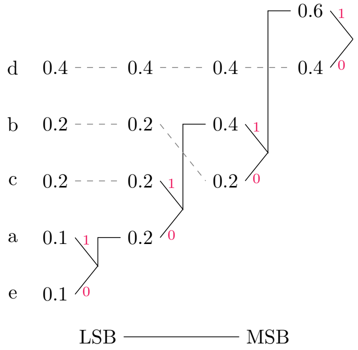
5.5.3 Huffman avec nouvel élément en haut

a	b	c	d	e
0.1	0.2	0.2	0.4	0.1



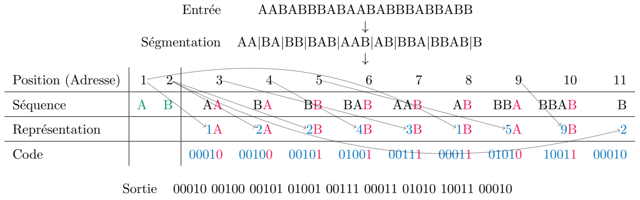
a	b	c	d	e
101	01	00	11	100

5.5.4 Huffman avec nouvel élément en bas



a	b	c	d	e
1101	10	111	0	1100

5.6 Lempel Ziv



Entrée AABABBBABAABABBBABBABB

↓

Ségmentation AA|BA|BB|BAB|AAB|AB|BBA|BBAB|B

↓

Position (Adresse)	1	2	3	4	5	6	7	8	9	10	11
Séquence	A	B	AA	BA	BB	BAB	AAB	AB	BBA	BBAB	B
Représentation			1A	2A	2B	4B	3B	1B	5A	9B	2
Code			00010	00100	00101	01001	00111	00011	01010	10011	00010

Sortie 00010 00100 00101 01001 00111 00011 01010 10011 00010

Taux de compression :

$$\frac{L_{\text{initiale}} - L_{\text{finale}}}{L_{\text{initiale}}} [\%]$$

6 AWGN et Shannon

$$C = \frac{R}{\log_2(M)}$$

6.1 Capacité du canal et efficacité spectrale

Sans bruit

$$\text{eff. spec.} = \frac{C}{B} = 2\log_2(M)$$

Avec bruit

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

La limite est donnée par

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

6.1.1 Capacité maximale du canal (pas de bruit)

$$C_{\max} = 2B \qquad R_{\max} = 2B\log_2(M)$$

6.2 BER

$$\text{BER} = \frac{1}{2}\text{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right)$$

7 Wireless

$$\boxed{\lambda = \frac{c}{f}}$$

8 Transmission sans-fil

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

8.1 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 :

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

$$A_{tt_{\text{dB}}} > 0$$

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

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

8.1.1 Avec γ

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi}\right)^2 \frac{1}{R^\gamma}$$

$$\underbrace{(P_r)_{\text{dB}} - (P_t)_{\text{dB}}}_{-Att_{\text{dB}}} = (G_t)_{\text{dB}} + (G_r)_{\text{dB}} +$$

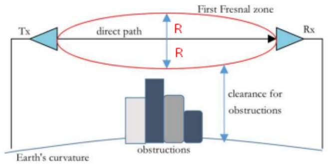
$$20\log_{10}\left(\frac{\lambda}{4\pi}\right) + 10\log_{10}\left(\frac{1}{R^\gamma}\right)$$

Valeurs de γ Espace libre 2, Environnement urbain 2.7 à 3.5, environnement urbain avec ombrage 3 à 5, dans un bâtiment avec vue de l'antenne 1.6 à 1.8, dans un bâtiment sans vue 4 à 6, dans une industrie 2 à 3

Si on rajoute γ après-coup, on utilise la formule suivante pour calculer la nouvelle atténuation Att_γ (on considère $\text{Att} > 0$) :

$$\text{Att}_\gamma = \text{Att} + 10\log_{10}(r^{\gamma-2})$$

8.2 zone de Fresnel



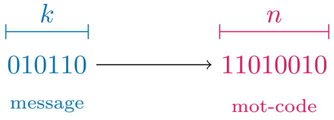
$$R = \frac{1}{2}\sqrt{\frac{cD}{f}}$$

Avec D la distance entre les antennes, c la vitesse de la lumière et f la fréquence

9 Codage de canal

$$(n, k, d)_q$$

Lorsque la base est $q = 2$ on ne l'affiche pas



k	Nombre de bits à transmettre (information)
n	Nombre de bits transmis $n \geq k$
d	Distance de Hamming

Message / information

$$x = \begin{pmatrix} x_0 & x_1 & \cdots & x_{n-1} \end{pmatrix} \in \mathbb{R}^{1 \times k}$$

$$x(X) = g_{k-1}X^{k-1} + g_{k-2}X^{k-2} + \cdots + g_1 X + g_0$$

Mot-code reçu

$$y = \begin{pmatrix} y_0 & y_1 & \cdots & y_{n-1} \end{pmatrix} \in \mathbb{R}^{1 \times k}$$

$$y(X) = g_{n-1}X^{n-1} + g_{n-2}X^{n-2} + \cdots + g_1 X + g_0$$

Distance minimale : d_{min} somme des différences entre deux codes. Pour déterminer la distance minimale on cherche

- "à l'oeil" entre la liste des mots-codes
- Le mot-code qui a le moins de 1 par rapport au mot-code nul (on compte les 1).

Capacité de détection d'erreurs : $d_{min} - 1$
Capacité de correction d'erreurs : $\left\lfloor \frac{d_{min}-1}{2} \right\rfloor$

9.1 Liste des codes

Faire une table avec chaque mot-code et message possible

X^3	X^2	X	1	X^6	X^5	X^4	X^3	X^2	X	1
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	0	1
	$\vdots$						$\vdots$			

Penser à réutiliser des codes (10 sera similaire à 1 mais décalé).

9.2 Matrice de génération

Pour déterminer tous les mots-codes, évaluer toutes les entrées possibles.

9.2.1 Matrice non-systématique

Pour passer à la matrice systématique on a le droit de

- Faire des combinaisons linéaires des lignes
- Permuter des lignes

On ne peut pas faire de permutations sur les colonnes.

9.2.2 Matrice systématique

La matrice identité se retrouve à gauche

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

Les x sont des 0 ou 1 en fonction de P (à déterminer pour chaque problème).
La forme systématique place les bits au début du mot-code, ce qui permet de décoder très facilement.
Pour construire la matrice systématique on peut simplement placer des mots-codes donnés sur les lignes de G_S afin d'avoir la matrice identité au début de G_S (fonctionne car 100 va donner la première ligne de G_S, donc la ligne doit être un mot-code).

9.3 Matrice de vérification

$$H_s = \begin{pmatrix} P^T & I_{n-k} \end{pmatrix} \in \mathbb{R}^{n-k \times n}$$

9.3.1 Syndrome

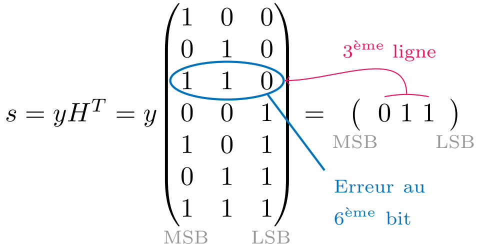
$$S = yH_S^T$$

Si le syndrome est différent de 0, sa valeur donne la position de l'erreur (si le nombre d'erreurs à corriger ne dépasse pas la limite de correction).

$$S = yH_S^T = \begin{pmatrix} \underset{MSB}{s_0} & s_1 & s_2 & \underset{LSB}{s_3} \end{pmatrix}$$

9.3.2 Matrice de Hamming ("nombres croissants")

La matrice de vérification de Hamming permet de déterminer directement la position d'une erreur. Cette matrice est construite avec des nombres binaires croissants (à partir de 1). ne pas lire le syndrome directement



$$s = yH^T = y \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0\,1\,1 \end{pmatrix}$$

Cette façon de faire permet d'éliminer les éventuelles erreurs due à l'ordre de construction de la matrice (LSB en haut ou LSB en bas).
Ceci fonctionne **uniquement** avec la matrice de Hamming (dont les lignes sont 1,2,3... en binaire). Si on regarde la matrice non-transposée ce sera les colonnes

9.4 Propriétés

$$G_S H_S^T = 0$$

9.5 Codes polynomiaux

1. Souvent linéaires (+ parfois cycliques)
2. Générateur rarement systématique (on peut utiliser un encodage systématique si le générateur ne l'est pas)

9.6 Codes linéaires

La somme de deux codes valides donne un nouveau code valide

9.7 Codes cycliques

Un décalage vers la gauche 0011 $\longrightarrow$ 0110 ou vers la droite 0011 $\longrightarrow$ 1001 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.
Si $g(x)$ est un facteur de $1+X^n$ alors le code est cyclique.

9.7.1 Générateur

Un générateur permet, par des décalages cycliques et des sommes d'obtenir tous les mots-code.
Si le générateur divise $(1+X^n)$ alors le code est cyclique

9.7.2 Encodage

On veut encoder un message x et obtenir le mot-code y

Encodage systématique :

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

Les bits du message se retrouvent au début du mot-code

Encodage "standard" :

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

9.7.3 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"

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

9.7.4 Correction d'erreurs

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

9.7.5 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" :

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

9.8 Générateur $\longrightarrow$ matrice

9.8.1 Non-systématique

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

$$\underset{k\times n}{\begin{pmatrix} g_{k-1} & g_{k-2} & \cdots & g_1 & g_0 & 0 & 0 \\ 0 & g_{k-1} & g_{k-2} & \cdots & g_1 & g_0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & g_{k-1} & g_{k-2} & \cdots & g_1 & g_0 \end{pmatrix}}$$

Fonctionne aussi si on mets le générateur dans l'autre sens.
A vérifier si valable en général (pas juste cyclique) mais en principe oui.
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 des combinaisons linéaires sur les lignes pour l'obtenir

9.8.2 Systématique

On prend des mots-codes et on les mets dans les lignes de la matrice. On choisi ces mots-codes pour qu'ils écrivent la matrice identité au début de G_S

$$G_S = \begin{pmatrix} \text{Mot-code qui commence par 1000} \\ \text{Mot-code qui commence par 0100} \\ \text{Mot-code qui commence par 0010} \\ \text{Mot-code qui commence par 0001} \end{pmatrix}$$

9.9 Codage convolutionnel

Construire une table pour déterminer les états futurs (Avec 3 délais)

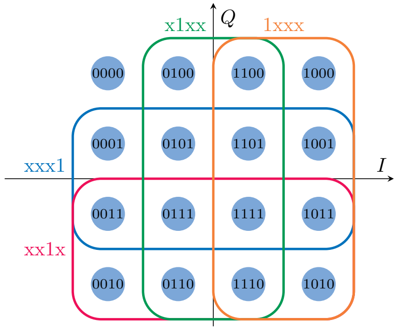
t	U	D_0	D_1	D_2	Y
0	u_{k-1}	0	0	0	0
1	u_{k-2}	u_{k-1}	$\cdots$		0
$\vdots$					$\vdots$
$k-2$	u_1				q_1
$k-1$	u_0				q_0
k	0	r_0	r_1	r_2	

Le reste se retrouve sur les mémoires lorsqu'on a terminé d'envoyer le message

10 Modulations

10.1 QAM

Exemple avec QAM16 :



On utilise le codage de Grey. Si il y a une petite perturbation sur le signal, il y a un minimum de bits modifiées et donc la correction d'erreur pourra récupérer le message d'origine.

11 Multiple access

Half-Duplex : Envoi puis réception (souvent limités par l'envoi et la réception avec des puissances radicalement différentes)

Full-Duplex : Envoi et réception simultanés

11.1 Méthode de duplex

Time Division Duplex TDD : Réception pendant un instant puis envoi pendant un autre

Frequency Division Duplex FDD : Envoi et réception sur des canaux différents (Utilisation de diplexeur, circulateur)

11.2 Méthodes de multiple access

Carrier Sense Multiple Access CSMA Aussi appelé "Listen Before Talk". Utilisé dans le Wifi.

Frequency Division Multiple Access FDMA : Un utilisateur par canal de fréquence (limité en nombre d'utilisateurs et bande passante. Pas possible de profiter de toute la bande si on est seul)

Time Division Multiple Access TDMA : Temps alloué à chaque utilisateur (sensible au temps, synchronisation critique).

Code Division Multiple Access CDMA : Code orthogonal pour chaque utilisateur (pas d'espace entre les canaux, hardware plus simplet et plus robuste aux interférences. Il faut toutefois gérer la puissance de chaque utilisateur vu qu'on reçoit certains beaucoup mieux que d'autres). Multiplication par le code en entrée, correlation à la sortie. Comme on augmente le spectre au TX, on peut diminuer l'amplitude pour fonctionner à énergie constante

Walsh-Hadamard : 2^n utilisateurs (et 2^n bits).

Orthogonal Frequency Division Multiple Access OFDMA : Canal OFDM séparé en 256 sous-canaux (répartis sur 9 blocks "RU" de 26 canaux). Chaque utilisateur utilise x canaux et il y a également une répartition sur le temps.

Space Division Multiple Access SDMA : Séparation des utilisateurs en zones géographiques (interférences entre les cellules, passage d'une à l'autre).

11.2.1 Input / Output

Single Input Single Output SISO : un récepteur, un émetteur $y_1 = h_1 x_1$

Single Input Multiple Output SIMO : un émetteur et deux récepteurs $y_1 = h_1 x_1$, $y_2 = h_2 x_2$. Pas de changement de capacité mais il y a de la diversité

Multiple Input Multiple Output MISO : Deux émetteurs et un récepteur. On doit différencier les signaux émis donc aucun gain $y_1 = h_1 x_1 + h_2 x_2$, $y_2 = h_1 x_2^* + h_2 x_1^*$

Multiple Input Multiple Output MIMO : Deux émetteurs et deux récepteurs, on peut faire la différence entre x_1 et x_2 $y_1 = h_1 x_1 + h_2 x_2$, $y_2 = h_3 x_1 + h_4 x_2$. La capacité du canal est doublée

Massive MIMO : $N \times N$ antennes. En principe N fois la capacité mais il faut que les fading soient indépendants et non-corrélés

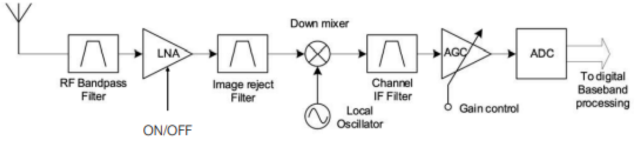
12 Acquisition et synchronisation

Il faut prendre en compte :

- Délais de propagation
- Effet doppler (émetteur / récepteur mobile)
- Fading
- Variations locales des oscillateurs (température, variations, etc...)

La fréquence est connue à peu près, la phase est inconnue. Les outils suivants permettent de palier à ces problèmes (en partie) :

Amplifier Gain Control AGC : Correction de l'amplitude sur l'amplificateur. Constitué d'un LNA (Low Noise Amplifier), d'un ADC et un multiplicateur digitals



PLL / FLL : Frequency locked loop / phase locked loop

Voltage Controlled Oscillator VCXO $f(t) = f_0 + K_0 v_{in}(t)$

12.1 Récupération de la porteuse

Réponse de la forme

$$H(s) = \frac{K\frac{G(s)}{s}}{1 + K\frac{G(s)}{s}} \qquad G(s) = \frac{1 + \tau_2 s}{1 + \tau_1 s}$$

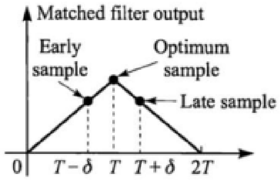
12.2 Récupération des symboles

Utilisation de :

- Beacon
- Préambule (USB)
- Horloge GPS / GNSS
- Récupération de l'horloge sur les symboles (récupération puis analyse après-coup : decision directed, probabilité max de chaque symbole : non decision directed)

12.2.1 Early-Late synchronizer

méthode non decision directed. Si $T - \delta$ est le même que $T - \delta$ on se trouve au bon endroit, sinon il faut bouger



12.3 Time equalization

Représentation puis application de la fonction de transfert du canal. Pas nécessaire avec l'OFDM

12.4 Frequency equalization

Utile pour des modulations à large bande comme CDMA ou OFDM (variation de fréquence le long du canal). Dans le cas de l'OFDM on utilise l'interference inter-porteuse (ICI)

$$\hat{S}[k] = \frac{R[k]}{H[k]}$$

12.4.1 Pilotes OFDM

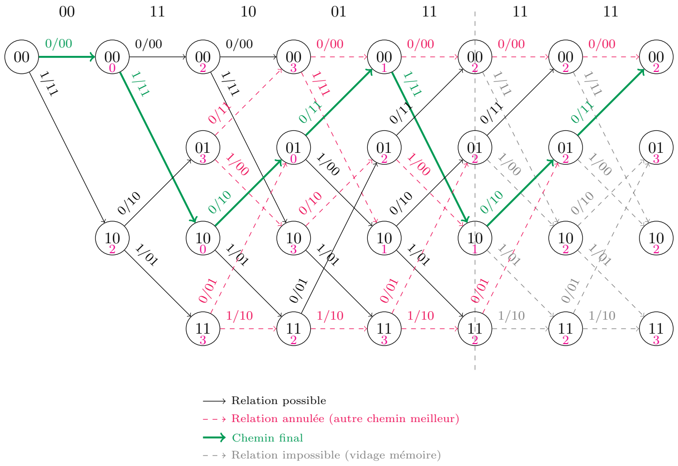
Continus ou intermittens mais prédéterminés. Modulés avec BPSK ou QPSK pour maximiser le SNR

13 Autres

$$1\,\mathrm{W} = 30\,\mathrm{dBm}$$

14 Algorithme de Viterbi (décodage)

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).