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

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

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

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

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

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

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

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

1.4.1 Propriétés

Linéarité

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

Décalage temporel

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

1.5 Équation aux différences

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

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

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

2 Algèbre linéaire

3 Aléatoire

4 Systèmes de communication

5 Codage de source

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 :

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

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

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

5.2 Lempel Ziv

Entrée : AABABBBABAABABBBABBABB
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

6 AWGN et Shannon

7 Wireless

8 Transmission sans-fil

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 :

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

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

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

8.2 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

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

9 Codage de canal

$$\overbrace{010110}^{k} \longrightarrow \overbrace{11010010}^{n}$$

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

Information (bits à transmettre)

$$x = \begin{pmatrix} x_{k-1} & x_{k-2} & \cdots & x_1 & x_0 \end{pmatrix} \in \mathbb{R}^{1\times k}$$

Matrice systématique (identité à gauche).

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

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

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

9.1 Propriétés

$$G_S H_S^T = 0$$

9.2 Codes polynomiaux

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

9.3 Codes linéaires

La somme de deux codes valides donne un nouveau code valide

9.4 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

9.4.1 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

9.4.2 Encodage

Encodage systématique :

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

Encodage "standard" :

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

9.4.3 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"

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

9.4.4 Correction d'erreurs

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

9.4.5 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" :

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

9.5 Générateur $\leftrightarrow$ matrice

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

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

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.

9.6 Codage convolutionnel

10 Modulations

11 Multiple access

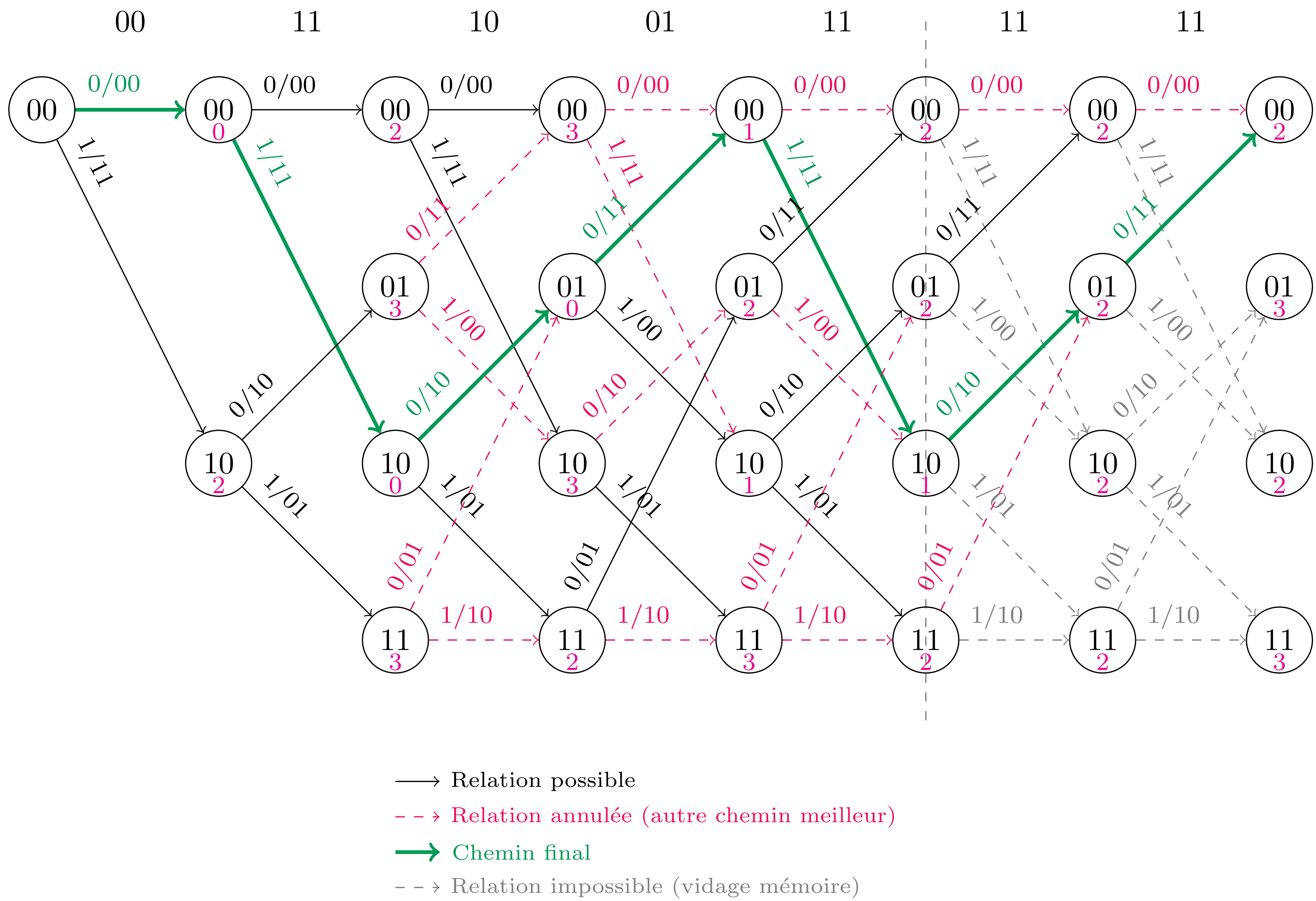
12 Acquisition et synchronisation

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).