Master Mathématiques et Applications,
première année
Aix-Marseille Université

Modélisation en Traitement du signal

B. Torrésani

Année 2015-16

Table des matières

Introduction

Le traitement du signal est la discipline qui développe et étudie les techniques de traitement (filtrage, amplification...), d'analyse et d'interprétation des signaux. Elle fait largement appel aux résultats de la théorie de l'information, des statistiques ainsi qu'à de nombreux autres domaines des mathématiques appliquées.
Les signaux à traiter peuvent provenir de sources très diverses, mais la plupart sont des signaux électriques ou devenus électriques à l'aide de capteurs et transducteurs (microphones, rétines, senseurs thermiques, optiques, de pression, de position, de vitesse, d'accélération et en général de toutes les grandeurs physiques et chimiques).
On distingue essentiellement les signaux analogiques qui sont produits par divers capteurs, amplificateurs, convertisseurs numérique-analogique ; les signaux numériques issus d'ordinateurs, de terminaux, de la lecture d'un support numérique ou d'une numérisation par un convertisseur analogique-numérique.
Le traitement peut être fait, sans numériser les signaux, par des circuits électroniques analogiques ou aussi des systèmes optiques (traitement du signal optique). Il est de plus en plus souvent réalisé par traitement numérique du signal, à l'aide d'ordinateurs, de microprocesseurs embarqués, de microprocesseurs spécialisés nommés DSP, de circuits reconfigurables ou de composants numériques dédiés.

Un aspect essentiel est celui de la représentation et de la modélisation de ces signaux. L'objectif de ce cours est de nous focaliser sur cette modélisation, en prenant comme problèmes de référence des problèmes liées aux télécommunications. On se concentrera plus particulièrement sur deux aspects : le codage des signaux et leur compression (et en particulier le passage des signaux analogiques aux signaux numériques), et la transmission.

Chapitre

1 Eléments de base

1.1 Analyse Hilbertienne

1.1.1 Rappels : Espaces préhilbertiens, espaces de Hilbert, orthogonalité

Il est utile de se rafraîchir la mémoire avec quelques notions de base. On rappelle qu'une forme sesquilinéaire sur un espace vectoriel complexe E est une application $\varphi : \ E \times E \longmapsto \mathbb{R}$, $(x, y) \longmapsto \varphi(x, y)$ telle que pour tout $x \in E$ l'application $y \to \varphi(x, y)$ est anti-linéaire et pour tout $y \in E$ l'application $x \to \varphi(x, y)$ est linéaire.
On dit que φ est :

- Hermitienne si $\varphi(x, y) = \overline{\varphi}(y, x)$ pour tous $x, y \in E$;
- Positive si $\varphi(x, x) \geq 0$ pour tout $x \in E$;
- Définie si pour $x \in E$ l'égalité $\varphi(x, x) = 0$ équivaut à $x = 0$.

Définition 1.1 *Un produit Hermitien sur un espace vectoriel complexe E est une forme sesquilinéaire hermitienne définie positive.*

On note en général $(x, y) \to \langle x, y\rangle$ un tel produit Hermitien (on parle aussi de produit scalaire).

Définition 1.2 *Un espace préhilbertien est un espace vectoriel muni d'un produit Hermitien. Un espace de Hilbert est un espace pré-hilbertien complet pour la norme $x \to \|x\| = \sqrt{\langle x, x\rangle}$.*

Dans un espace de Hilbert E, la norme définit le produit Hermitien via l'identité de polarisation : $\forall x, y \in E$,

$$\langle x, y\rangle = \frac{1}{4}\Big(\|x + y\|^2 - \|x - y\|^2 + i\|x + iy\|^2 - i\|x - iy\|^2\Big) \ . \tag{1.1}$$

Rappelons aussi l'inégalité de Cauchy-Schwarz : pour tous x, y,

$$|\langle x, y\rangle| \leq \|x\| \, \|y\| \ ,$$

de sorte qu'il existe un réel $\theta \in [-\pi, \pi]$ tel que

$$\cos(\theta) = \frac{|\langle x, y\rangle|}{\|x\| \, \|y\|} \ ,$$

c'est à dire θ est une mesure de l'angle entre x et y. Si $\theta = \pm\pi/2$, x et y sont orthogonaux.

Définition 1.3 *On dit que deux vecteurs $x, y \in E$ sont orthogonaux si $\langle x, y\rangle = 0$.*

Théorème 1.1 (Pythagore) *Les vecteurs x et y sont orthogonaux dans E si et seulement si $\|x + y\|^2 = \|x\|^2 + \|y\|^2$.*

Les notions importantes dont nous aurons besoin sont les notions d'orthogonal d'un sous-espace, et de projection orthogonale.

Définition 1.4 *L'orthogonal d'une partie non vide $X \subset E$ est l'ensemble :*

$$X^{\perp} = \{y \in E : \forall x \in X, \langle x, y\rangle = 0\} \ . \tag{1.2}$$

Il est facile de vérifier que $X^{\perp}$ est un sous espace vectoriel de E.

Définition 1.5 *On appelle famille orthogonale dans E toute famille $\{e_n,\ n = 0, 1, 2, \ldots\}$ de vecteurs de E telle que $\langle e_m, e_n\rangle = 0$ pour tous $m \neq n$. Si de plus $\|e_n\| = 1$ pour tout n, cette famille est orthonormée ou orthonormale.*

Définition 1.6 (Base Hilbertienne) *Soit E un espace pré-Hilbertien. Une famille orthonormale $\{e_n,\ n = 0, 1, 2, \ldots\}$ est une base Hilbertienne de E si elle est complète, dans le sens suivant : pour tout $x \in E$, il existe une famille de scalaires $\{a_n, n = 0, \ldots\}$ telle que*

$$\sum_n a_n e_n = x \ ,$$

où la sommabilité de la série dans le membre de droite est associée à la norme.

notons que dans cette définition, l'unicité de la famille de coefficients a_n est assurée par l'orthonormalité de la famille.

Théorème 1.2 *Une famille orthogonale de vecteurs non nuls de E est libre. Elle est une base orthogonale du sous-espace F de E qu'elle engendre.*

Dans ce cas, en notant comme plus haut $\{e_n,\ n = 0, 1, 2, \ldots\}$ la famille orthogonale, et en la supposant normée, on a aussi la formule de Parseval (généralisation du théorème de Pythagore) : pour tout $x \in F$

$$\|x\| = \sum_n |\langle x, e_n\rangle|^2 \ . \tag{1.3}$$

La notion centrale dont nous aurons besoin ici est la notion de projection orthogonale.

Théorème 1.3 (projection orthogonale) *Soit F un sous espace vectoriel de dimension finie de E. Pour tout $x \in E$, il existe un unique $y \in F$ à distance (induite par la norme) minimale de F, c'est à dire tel que*

$$\|x - y\| = d(x, F) = \inf_{z \in F} \|x - z\|$$

y est également l'unique vecteur de F tel que $x - y \in F^{\perp}$.

Nous verrons plus loin son expression lorsqu'une base ou une famille génératrice de F est connue.

1.1.2 Quelques exemples d'espaces de Hilbert utiles dans notre contexte

On va considérer ci-dessous quelques exemples d'espaces modèles, tous de même dimension, qui peuvent être utiles dans des situations de codage de signaux. On se placera dans le cadre général des espaces $L^2([a, b])$ définis par

$$L^2([a, b]) = \left\{ x : [a, b] \to \mathbb{C} : \ \|x\|^2 := \int_a^b |x(t)|^2 \, dt < \infty \right\} \ , \tag{1.4}$$

où $a < b$ sont deux réels, munis d'une structure d'espace de Hilbert grâce au produit Hermitien

$$\langle x, y\rangle = \int_a^b x(t)\overline{y}(t) \, dt \ . \tag{1.5}$$

— $P_N([0,1])$: espace des polynômes de degré inférieur ou égal à N. Il est bien connu que $P_{N-1}([0,1])$ est un espace linéaire de dimension N, qui hérite d'une structure d'espace de Hilbert grâce au produit Hermitien de $L^2([0,1])$. Toute fonction $x \in P_{N-1}([0,1])$ est caractérisée par N valeurs ponctuelles

$$x_n = x(t_n) \ ,$$

où les $t_n \in [0,1]$ sont N valeurs distinctes ; la reconstruction de x à partir des valeurs x_n peut s'effectuer en utilisant l'interpolation de Lagrange, qui est malheureusement connue pour être très instable quand N devient grand (pour en savoir plus, se documenter sur le *phénomène de Runge*).

— On considère l'espace $\mathcal{E}_0([0,1])$ des fonctions définies sur $[0,1]$, constantes sur les intervalles de la forme $[k/N,(k+1)/N[$ où N est un entier positif, et $k = 0,1,\dots N-1$. $\mathcal{E}_0$ est de dimension N, et il est facile de vérifier que toute fonction $x \in \mathcal{E}_0$ peut s'écrire sous la forme

$$x = \sum_{k=0}^{N-1} x\left(\frac{k}{N}\right) \mathbb{1}_k(t) \ ,$$

où $\mathbb{1}_k$ est l'indicatrice de l'intervalle $[k/N,(k+1)/N[$.

— On considère l'espace $\mathcal{E}_1([0,1])$ des fonctions définies sur $[0,1]$, continues et affines sur les intervalles de la forme $[k/N,(k+1)/N[$ où N est un entier positif, et $k = 0,1,\dots N-1$. Il est facile de voir que toute fonction de $\mathcal{E}_1$ est caractérisée par les valeurs qu'elle prend sur les *noeuds* k/N, $\mathcal{E}_1$ est donc un espace de dimension $N+1$, où N si on se limite aux fonctions x telles que $x(0) = x(1)$. On verra plus bas comment construire une base de cet espace.

— L'espace modèle le plus classique en traitement des signaux est l'espace des polynômes trigonométriques de degré inférieur ou égal à un certain degré donné :

$$\mathcal{P}_M([0,1]) = \{x : [0,1] \to \mathbb{C},\ c_m(x) = 0 \text{ si } |m| > M\} \ ,$$

où

$$c_m(x) = \int_0^1 x(t) e^{-2i\pi mt}\, dt \tag{1.6}$$

est le m-ième coefficient de Fourier de x. Il s'agit d'un sous-espace de dimension $2M+1$ de $L^2([0,1])$, et on sait d'après la théorie des séries de Fourier que la famille des fonctions oscillantes

$$\epsilon_m(t) = e^{2i\pi mt} \ , \quad m = -M, 1-M, \dots M \tag{1.7}$$

est une base orthonormée de $\mathcal{P}_M([0,1])$.

1.1.3 Approximation

Les signaux auxquels l'on s'intéresse n'ont aucune raison d'appartenir à ces espaces particuliers, on va donc les approximer par des éléments de ces espaces. Pour ce faire, le plus simple est de considérer la projection orthogonale sur ces sous-espaces modèle.
Le résultat qui suit est un corollaire du théorème de Gram-Schmidt, qui stipule qu'à toute famille libre de vecteurs dans un espace pré-Hilbertien, on peut associer une famille orthonormée qui engendre le même sous-espace, et en constitue donc une base orthonormée. La construction de cette base orthonormée est la procédure d'orthonormalisation de Gram-Schmidt.

Théorème 1.4 *Soit E un espace pré-Hilbertien, soit F un sous espace de E. Si F est de dimension finie, ou infinie-dénombrable, alors F admet une base orthonormée.*

Dans ces conditions, notons $\mathcal{B} = \{e_0, e_1, e_2, \ldots\}$ une telle base orthonormée. La meilleure approximation de $x \in E$ par un élément de F, au sens de la distance induite par la norme de E est donnée par

$$y = \Pi_F(x) = \sum_n \langle x, e_n \rangle e_n \ . \tag{1.8}$$

et l'erreur d'approximation est donnée par :

$$\|x - y\|^2 = \|x\|^2 - \|y\|^2 = \|x\|^2 - \sum_n |\langle x, e_n \rangle|^2 \ . \tag{1.9}$$

Plus généralement, il est possible de relaxer les hypothèses. On se limitera ici au cas de sous-espaces de dimension finie. Etant donnée une famille de vecteurs $f_0, f_1, \ldots f_{N-1}$, la *matrice de Gram* de cette famille est définie par

$$G = \{G_{mn},\ m, n = 0, \ldots N-1\} \ , \qquad G_{mn} = \langle f_n, f_m \rangle \ . \tag{1.10}$$

La matrice de Gram est un outil important d'algèbre linéaire, et a des propriétés importantes, par exemple

Lemme 1.1 *Soit $\mathcal{F} = \{f_0, \ldots f_{N-1}\}$ une famille d'éléments de E.*

1. *La matrice de Gram est auto-adjointe (c'est à dire telle que $G^* = G$), et semi-définie positive : pour tous $\alpha_0, \ldots \alpha_n \in \mathbb{C}$ et $m_0, \ldots m_n$,*

$$\sum_{j=0}^{n} \sum_{\ell=0}^{n} \alpha_j \overline{\alpha}_\ell G_{j\ell} \geq 0 \ .$$

2. *La matrice de Gram est inversible si et seulement si la famille $\{f_0, \ldots f_{N-1}\}$ est linéairement indépendante. Dans ce cas la matrice inverse $H = G^{-1}$ est elle aussi auto-adjointe.*

Donc en particulier, une famille finie de vecteurs est libre si et seulement le déterminant de la matrice de Gram (appelé déterminant de Gram) est non-nul. Notons aussi qu'en conséquence de ces propriétés, la matrice de Gram est diagonalisable, ses valeurs propres sont réelles, et positives ou nulles.

Plus généralement, on montre le résultat suivant

Proposition 1.1 *Soit F un sous-espace de E engendré par la famille de vecteurs $f_0, f_1, \ldots f_{N-1}$. Supposons que G soit inversible. Alors la projection orthogonale de E sur F s'écrit sous la forme*

$$E \ni x \to y = \Pi_F(x) = \sum_{n=0}^{N-1} a_n f_n \ , \tag{1.11}$$

où les coefficients a_n sont solutions du problème linéaire

$$\sum_{n=0}^{N-1} G_{mn} a_n = \langle x, f_m \rangle \ , \qquad m = 0, \ldots N-1 \ . \tag{1.12}$$

Ces coefficients s'écrivent également

$$a_n = \langle x, \tilde{f}_n \rangle \ , \tag{1.13}$$

où $\{\tilde{f}_0, \ldots \tilde{f}_{N-1}\}$ est la famille duale *de la famille des f_n, définie par*

$$\tilde{f}_n = \sum_{m=0}^{N-1} \overline{H}_{nm} f_m \ , \quad m = 0, \ldots N-1 \ , \tag{1.14}$$

et les H_{nm} sont les coefficients de la matrice $H = G^{-1}$. On a également

$$\Pi_F(x) = \sum_{m=0}^{N-1} \langle x, f_m \rangle \tilde{f}_m \ . \tag{1.15}$$

Preuve : Soit $x \in E$, et soit $\Pi_F(x) = \sum_{n=0}^{N-1} a_n f_n$ son projeté orthogonal sur F. On a alors $\langle x - \Pi_F(x), f_m\rangle = 0$ pour tout $m = 0, \dots N-1$, ce qui équivaut au système

$$\langle x, f_m\rangle = \sum_{n=0}^{N-1} a_n \langle f_n, f_m\rangle = \sum_{n=0}^{N-1} G_{mn} a_n \ .$$

Par conséquent on peut écrire

$$a_n = \sum_{m=0}^{N-1} H_{nm}\langle x, f_m\rangle = \left\langle x, \sum_{m=0}^{N-1} \overline{H}_{nm} f_m \right\rangle = \langle x, \tilde{f}_n\rangle \ ,$$

Finalement, écrivons

$$\Pi_F(x) = \sum_{n=0}^{N-1} \left(\sum_{m=0}^{N-1} H_{nm}\langle x, f_m\rangle \right) f_n = \sum_{m=0}^{N-1} \langle x, f_m\rangle \sum_{n=0}^{N-1} \overline{H}_{mn} f_n = \sum_{m=0}^{N-1} \langle x, f_m\rangle \tilde{f}_m$$

où on a utilisé le fait que $H = G^{-1}$ est auto-adjointe (donc $\overline{H}_{nm} = H_{mn}$). ♠

Remarque 1.1 Il est facile de démontrer que les familles $\{f_n\}$ et $\{\tilde{f}_n\}$ satisfont la relation de biorthogonalité

$$\langle f_n, \tilde{f}_m\rangle = \delta_{mn} \ , \tag{1.16}$$

et que la famille $\{\tilde{f}_n\}$ est une base de F. On dit aussi que la famille $\{\tilde{f}_n\}$ est la base biorthogonale de de $\{f_n\}$.

Remarque 1.2 Matriciellement, en notant A le vecteur colonne composé des coefficients a_n, et F le vecteur colonne composé des produits scalaires $\langle x, f_n\rangle$, on écrit donc

$$GA = F \ , \qquad A = HF \ .$$

Définition 1.7 (Opérateurs d'analyse et de synthèse) *Soit $\mathcal{F} = \{f_0, \dots f_{N-1}\}$ une famille de vecteurs d'un espace E, on lui, associe les opérateurs suivants :*

— **Opérateur d'analyse** *: l'opérateur linéaire*

$$U : x \in E \longmapsto Ux = \{\langle x, f_0\rangle, \dots \langle x, f_{N-1}\rangle\} \in \mathbb{C}^N \ .$$

— **Opérateur de synthèse** *: l'opérateur linéaire*

$$V : \alpha \in \mathbb{C}^N \longmapsto V\alpha = \sum_{m=0}^{N-1} \alpha_m \tilde{f}_m \ .$$

On a donc l'identité

$$\Pi_F = VU \ . \tag{1.17}$$

Par ailleurs, notons aussi qu'on peut également écrire, pour tout $\alpha \in \mathbb{C}^N$

$$V\alpha = \sum_{n=0}^{N-1} \alpha_n \tilde{\varphi}_n = \sum_{n=0}^{N-1} \alpha_n \sum_{m=0}^{N-1} H_{mn}\varphi_m = \sum_{m=0}^{N-1} \left(\sum_{n=0}^{N-1} H_{mn}\alpha_n \right) \varphi_m = U^* H\alpha \ ,$$

d'où on tire

Corollaire 1.1 *Avec les notations ci-dessus, le projecteur orthogonal sur F s'écrit sous la forme*

$$\Pi_F = U^* H U \ .$$

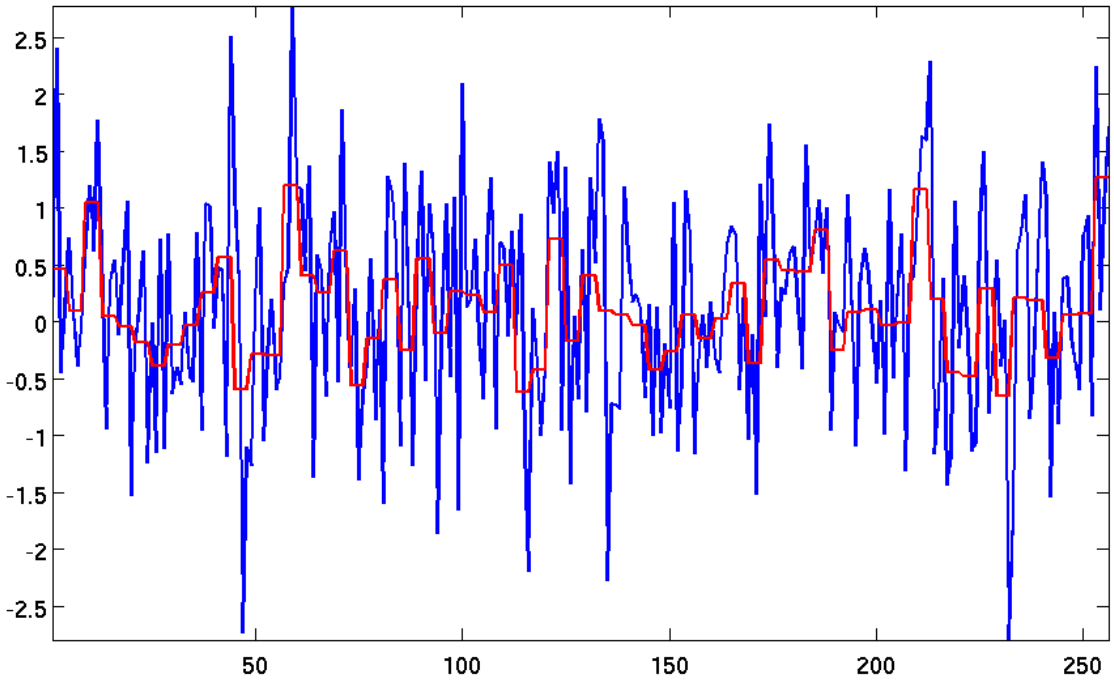

FIGURE 1.1 – Approximation constante par morceaux : signal original (bleu) et signal approché (rouge).

Exemple 1.1 (Approximation constante par morceaux) Dans $L^2([0,1])$ on considère les fonctions indicatrices normalisées des intervalles de la forme $[n/N, (n+1)/N[$, $n = 0, \dots N-1$

$$\mathbb{1}_n(t) = \begin{cases} \sqrt{N} & \text{si} \quad n/N \le t < (n+1)/N \\ 0 & \text{sinon} \end{cases}$$

Il est facile de voir que la famille des fonctions $\mathbb{1}_n$, $n = 0, \dots N-1$ est orthonormée, c'est donc une base orthonormée du sous-espace $\mathcal{E}_0 = \mathcal{E}_0^{(N)}$ de $L^2([0,1])$ qu'elle engendre (pour simplifier les notations on omettra généralement l'exposant (N) par la suite). La projection orthogonale correspondante $L^2([0,1]) \to \mathcal{E}_0^{(N)}$ s'écrit

$$x \to \Pi_{\mathcal{E}_0} x = \sum_{n=0}^{N-1} \alpha_n \mathbb{1}_n \ , \quad \text{avec} \quad \alpha_n = \langle x, \mathbb{1}_n \rangle = \sqrt{N} \int_{n/N}^{(n+1)/N} x(t)\, dt \ .$$

Un exemple d'approximation d'une fonction par une fonction constante par morceaux de $\mathcal{E}_0$ se trouve en Figure 1.1.

Exemple 1.2 (Approximation affine par morceaux) Dans $L^2([0,1])$ on considère les fonctions triangle Λ_N, définies comme des copies translatées

$$\Lambda_n(t) = \Lambda_0 \left(\left(t - \frac{n}{N} \right) [\text{mod} 1] \right)$$

de la fonction Λ_0 ci dessous,

$$\Lambda_0(t) = \begin{cases} K\left(\frac{1}{N} - t\right) & \text{pour} \quad 0 \le t \le \frac{1}{N} \\ 0 & \text{pour} \quad \frac{1}{N} \le t \le 1 - \frac{1}{N} \\ K\left(t - 1 + \frac{1}{N}\right) & \text{pour} \quad 1 - \frac{1}{N} \le t \le 1 \end{cases}$$

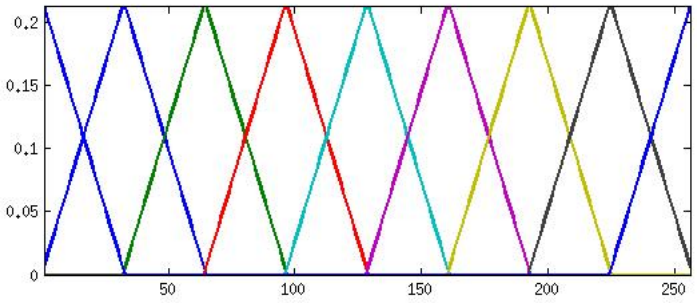

FIGURE 1.2 – Exemple de fonctions affines par morceaux engendrant l'espace $\mathcal{E}_1^{(8)}$. Les différentes fonctions sont représentées par des couleurs différentes.

où K est une constante de normalisation. Ces fonctions sont représentées (dans le cas $N = 8$) dans la Figure 1.2 ci-dessous.
Il est facile de voir que $\langle \Lambda_n, \Lambda_m \rangle = 0$ dès que $|(m-n)[\mathrm{mod}N]| \geq 2$, de sorte que pour calculer la matrice de Gram il suffit de calculer

$$\|\Lambda_n\|^2 = \|\Lambda_0\|^2 = 2K^2 \int_0^{1/N} \left(\frac{1}{N} - t\right)^2 dt = 2K^2 \int_0^{1/N} t^2\, dt = \frac{2}{3}\frac{K^2}{N^3} \ ,$$

et

$$\langle \Lambda_n, \Lambda_{n+1} \rangle = \langle \Lambda_0, \Lambda_1 \rangle = K^2 \int_0^{1/N} t\left(\frac{1}{N} - t\right) dt = \frac{K^2}{6N^3} \ .$$

La condition de normalisation $\|\Lambda_n\| = 1$ implique que $K^2 = 3N^3/2$, et donc pour tout N on a $\langle \Lambda_n, \Lambda_{n+1} \rangle = 1/4$. La matrice de Gram est donc de la forme

$$G = \begin{pmatrix} 1 & \frac{1}{4} & 0 & 0 & \dots & \frac{1}{4} \\ \frac{1}{4} & 1 & \frac{1}{4} & 0 & \dots & 0 \\ 0 & \frac{1}{4} & 1 & \frac{1}{4} & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ \frac{1}{4} & 0 & 0 & 0 & \dots & \frac{1}{4} \end{pmatrix}$$

Il s'agit d'une matrice que l'on appelle circulante [1], qui est diagonalisée par transformation de Fourier finie (voir le Lemme 1.2 ci-dessous). Ses valeurs propres sont données par la transformée de Fourier finie de sa première ligne, à savoir

$$\lambda_k = \sum_{n=0}^{N-1} g_n e^{-2i\pi kn/N} = 1 + \frac{1}{2}\cos(2\pi k/N) \ .$$

On voit facilement que ces valeurs sont toutes positives (en fait comprises entre 1/2 et 3/2), donc la matrice de Gram est inversible. En posant

$$\mu_k = \frac{1}{1 + \frac{1}{2}\cos(2\pi k/N)} \ ,$$

On a donc, en notant $D(\lambda)$ et $D(\mu)$ les matrices diagonales admettant respectivement les vecteurs λ et μ comme diagonale,

$$G = FD(\lambda)F^{-1} \ , \qquad \text{et} \qquad H = FD(\mu)F^{-1} \ ,$$

d'où on peut déduire la famille duale. La fonction Λ_0 et la fonction duale $\tilde{\Lambda}_0$ sont représentées en Figure 1.3, pour le cas $N = 8$.

1. On rappelle qu'une matrice carrée $A \in \mathcal{M}_N$ est circulante si ses éléments de matrice A_{mn} ne dépendent que de la différence $n - m$: il existe un vecteur $a \in \mathbb{C}^N$ tel que $A_{mn} = a_{n-m}$ ($n - m$ étant à prendre modulo N).

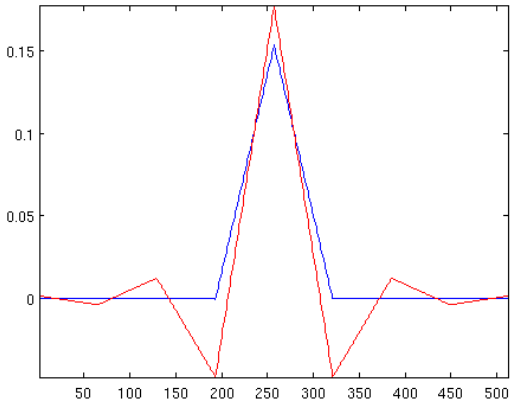

FIGURE 1.3 – La fonction Λ_0 (bleue) et sa fonction duale $\tilde{\Lambda}_0$ (rouge) dans le cas $N = 8$.

Un exemple d'approximation d'une fonction par une fonction affine par morceaux de $\mathcal{E}_1$ se trouve en Figure 1.4.

Exemple 1.3 (Base trigonométrique et signaux à bande limitée) Il résulte de la théorie L^2 des séries de Fourier que la famille des fonctions oscillantes $\{\epsilon_m,\, m \in \mathbb{Z}\}$ définies par

$$\epsilon_m(t) = e^{2i\pi mt}$$

est une base orthonormée de $L^2([0,1])$. Donc, la famille $\{\epsilon_m,\, m \in \mathbb{Z}, |m| \leq M\}$ est une base orthonormée de $\mathcal{P}_M([0,1])$, et la projection orthogonale de $L^2([0,1])$ sur $\mathcal{P}_M([0,1])$ s'écrit simplement

$$x \in L^2([0,1]) \longmapsto \Pi_{\mathcal{P}_M}(x) = \sum_{m=-M}^{M} c_m(x)\epsilon_m \ , \tag{1.18}$$

où les coefficients $c_m(x)$ sont les coefficients de Fourier de x définis en (1.6). La matrice de Gram est égale à la matrice identité, et la base duale est identique à la base des oscillations.
L'erreur causée par une telle approximation peut être mesurée en norme

$$\|x - \Pi_{\mathcal{P}_M}(x)\|^2 = \sum_{m>M} |c_m(x)|^2 \ .$$

Un exemple d'approximation d'une fonction par un polynôme trigonométrique de $\mathcal{P}_M$ se trouve en Figure 1.5.

1.1.4 Compléments

On a eu besoin ci-dessus de quelques résultats techniques liés à la transformation de Fourier finie.

Définition 1.8 (Transformation de Fourier finie (TFF)) *La transformation de Fourier finie associe à tout vecteur $x \in \mathbb{C}^N$ le vecteur $\hat{x} \in \mathbb{C}^N$ défini par*

$$\hat{x}_k = \sum_{n=0}^{N-1} x_n e^{-2i\pi kn/N} \ . \tag{1.19}$$

En version matricielle, en notant X et $\hat{X}$ les vecteurs colonne respectivement constitués des nombres x_n et $\hat{x}_k$, on peut écrire la TFF sous la forme suivante

$$\hat{X} = FX \ , \tag{1.20}$$

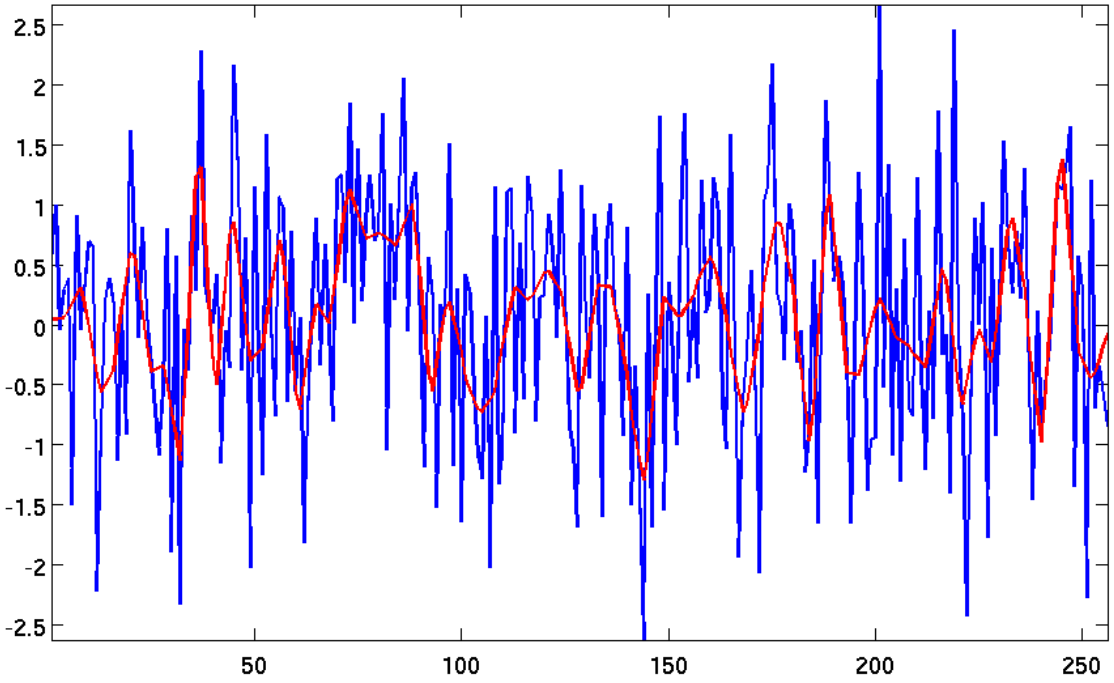

FIGURE 1.4 – Approximation affine par morceaux : signal original (bleu) et signal approché (rouge).

où la matrice de Fourier F est donnée par

$$F_{kn} = e^{-2i\pi kn/N} \ . \tag{1.21}$$

La Transformation de Fourier finie est multiple d'une isométrie, et est donc inversible. Plus précisément, on a :

Proposition 1.2 *La TFF est inversible, la transformation inverse est donnée par*

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}_k e^{2i\pi kn/N} \ . \tag{1.22}$$

De plus la TFF est multiple d'une transformation unitaire :

$$\|\hat{x}\| = \sqrt{N}\, \|x\| \ . \tag{1.23}$$

On peut noter que le vecteur $\hat{x}$ peut se voir comme le vecteur des produits scalaires de x par les vecteurs $\epsilon^{(k)}$ définis par

$$\epsilon_n^{(k)} = e^{2i\pi kn/N} \ , \qquad \text{et on a} \qquad \hat{x}_k = \langle x, \epsilon^{(k)} \rangle \ .$$

Matriciellement, on vérifie que ceci signifie que

$$F^{-1} = \frac{1}{N} F^* \ , \qquad \text{et que} \qquad FF^* = F^*F = NI_N \ , \tag{1.24}$$

où I_N est la matrice identité. La TFF permet de simplifier la situation lorsque l'on considère des matrices circulantes.

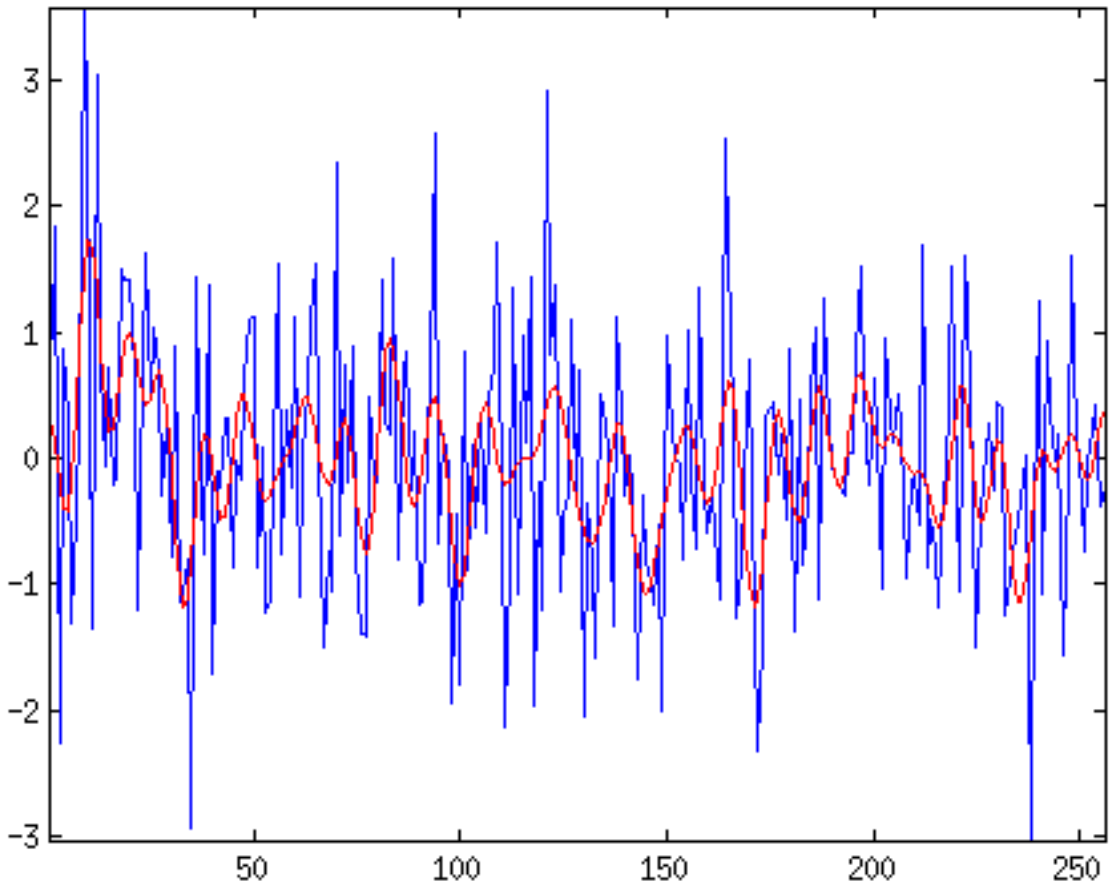

FIGURE 1.5 – Approximation par un polynôme trigonométrique : signal original (bleu) et signal approché (rouge).

Lemme 1.2 *Soit $A \in \mathcal{M}_N$ une matrice circulante, c'est à dire telle qu'il existe $a \in \mathbb{C}^N$ tel que $A_{mn} = a_{(n-m)[mod\ N]}$. Alors A est diagonalisée par la base des $\epsilon^{(k)}$. Plus précisément, pour tout $k = 0, \dots N-1$*

$$A\epsilon^{(k)} = \hat{a}_k \epsilon^{(k)} \ . \tag{1.25}$$

Matriciellement, on peut écrire

$$A = FD(\hat{a})F^{-1} = \frac{1}{N} FD(\hat{a})F^* \ . \tag{1.26}$$

Par conséquent, une matrice circulante est inversible si et seulement si le vecteur $\hat{a}$ ne s'annule pas. Dans ce cas, l'inverse de la matrice A est donné par

$$A^{-1} = FD(\hat{a})^{-1}F^{-1}$$

où $D(\hat{a})^{-1}$ est la matrice diagonale prenant sur sa diagonale les coefficients ${\hat{a}_k}^{-1}$.

1.2 Probabilités

1.2.1 Rappels

Nous aurons besoin ici de notions élémentaires de probabilités. On désignera par $(\mathcal{A}, \mathcal{F}, \mathbb{P})$ un espace probabilisé. On note par $\mathcal{L}^0(\mathcal{A}) = \mathcal{L}^0(\mathcal{A}, \mathbb{P})$ l'espace des variables aléatoires sur $(\mathcal{A}, \mathcal{F}, \mathbb{P})$, à valeurs réelles ou complexes. Etant données deux variables aléatoires $X, Y \in \mathcal{L}^0(\mathcal{A})$, on dit que $X \sim Y$ si $X = Y$ presque surement. Ceci définit une relation d'équivalence, et on note

$$L^0(\mathcal{A}) = \mathcal{L}^0(\mathcal{A}) / \sim$$

l'espace quotient, c'est à dire l'espace des variables aléatoires différentes presque sûrement. Etant donnée une variable aléatoire $X \in L^0$, on en notera $\mathbb{E}\{X\}$ l'espérance.

Définition 1.9 *Une variable aléatoire X est dite du second ordre si $\mathbb{E}\{|X|^2\} < \infty$. On notera $L^2(\mathcal{A}, d\mathbb{P})$ l'espace des variables aléatoires du second ordre sur l'espace probabilisé $(\mathcal{A}, \mathcal{F}, \mathbb{P})$.*

Il est intéressant de noter que $L^2(\mathcal{A}, d\mathbb{P})$ est en fait un espace de Hilbert, grâce au produit Hermitien défini par

$$(X|Y) = \int_{\mathcal{A}} X(a)\overline{Y}(a)\, d\mathbb{P}(a) \ , \qquad X, Y \in L^2(\mathcal{A}, d\mathbb{P}) \ . \tag{1.27}$$

Etant donnée une variable aléatoire du second ordre X, il résulte de l'inégalité de Cauchy-Schwarz que

$$\mathbb{E}\{|X|\} = \int |X(a)|\, d\mathbb{P}(a) \leq \sqrt{\int |X(a)|^2\, d\mathbb{P}(a)} < \infty \ .$$

Par conséquent, $\mathbb{E}\{X\}$ est aussi bien définie.

Définition 1.10 *Etant donnée une variable aléatoire réelle $X \in L^0(\mathcal{A}, \mathcal{F}, \mathbb{P})$, sa fonction de répartition est la fonction F_X définie par*

$$F_X(x) = \mathbb{P}\{X \leq x\} \ .$$

La fonction de répartition est une fonction bornée, plus précisément

$$0 \leq F_X(x) \leq 1 \qquad \forall x \ ,$$

et elle est monotone. A ce titre, F_X admet en tout point x une limite à gauche $F_X(x_-)$, égale ou non à $F_X(x)$ selon que F_X est continue en x ou non. F_X est une fonction *càdlàg* (continue à droite, admettant une limite à gauche).
La connaissance de la fonction de répartition permet de calculer la probabilité de tout intervalle

Proposition 1.3 *Soit X une variable aléatoire réelle sur un espace probabilisé $(\mathcal{A}, \mathcal{F}, \mathbb{P})$, soit F_X sa fonction de répartition. On a alors les propriétés suivantes :*

- $\mathbb{P}\{X \in]-\infty, x]\} = \mathbb{P}\{X \leq x\} = F_X(x)$,
- $\mathbb{P}\{X \in]x, +\infty[\} = \mathbb{P}\{X > x\} = 1 - F_X(x)$,
- $\mathbb{P}\{X \in]x, +\infty[\} = \mathbb{P}\{X > x\} = 1 - F_X(x)$,
- $\mathbb{P}\{X \in]x, y]\} = \mathbb{P}\{x < X \leq y\} = F_X(y) - F_X(x)$,
- $\mathbb{P}\{X \in]-\infty, x[\} = \mathbb{P}\{X < x\} = F_X(x_-)$,
- $\mathbb{P}\{X \in]x, y[\} = \mathbb{P}\{x < X < y\} = F_X(y_-) - F_X(x)$,
- $\mathbb{P}\{X \in [x, y[\} = \mathbb{P}\{x \leq X < y\} = F_X(y_-) - F_X(x_-)$,
- $\mathbb{P}\{X \in [x, y]\} = \mathbb{P}\{x \leq X \leq y\} = F_X(y) - F_X(x_-)$, *et*
- $\mathbb{P}\{X = x\} = F_X(x) - F_X(x_-)$.

Une variable aléatoire réelle X est dite à densité s'il existe une fonction $\rho_X \in L^1(\mathbb{R})$, définie et positive telle que pour tout intervalle $[a, b]$,

$$\mathbb{P}\{a \leq X \leq b\} = \int_a^b \rho_X(x)\, dx \ .$$

Plus généralement

Définition 1.11 *On appelle densité de probabilité d'une variable aléatoire X à valeur dans $\mathbb{R}^d$ une fonction ρ telle que pour toute partie borélienne $A \subset \mathbb{R}^d$,*

$$\mathbb{P}\{X \in A\} = \int_{\mathbb{R}^d} 1_A(u)\, \rho(u)\, du = \int_A \rho(u)\, du \ .$$

Définition 1.12 (Fonction caractéristique) *Soit X une variable aléatoire à valeurs dans $\mathbb{R}^d$. Sa fonction caractéristique est la fonction de d variables ϕ_X définie par*

$$\phi_X(u) = \mathbb{E}\left\{e^{-2i\pi u X}\right\} , \quad u \in \mathbb{R}^d \ .$$

Exemple 1.4 (Loi gaussienne)

1. Etant donnés deux réels μ et $\sigma > 0$, une variable aléatoire de loi Gaussienne $X \sim \mathcal{N}(\mu, \sigma)$ est définie par la densité de probabilités

$$\rho_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \ .$$

On vérifie facilement que $\mathbb{E}\{X\} = \mu$ et $\mathrm{Var}\{X\} = \sigma^2$. La fonction caractéristique de X est donnée par

$$\phi_X(u) = e^{-2i\pi u\mu} e^{-2\pi^2\sigma^2 u^2} , \quad u \in \mathbb{R} \ .$$

2. Soit $\mu \in \mathbb{R}^d$ et soit $\Sigma \in \mathcal{M}_d(\mathbb{R})$ une matrice symétrique semi-définie positive. Une variable aléatoire de loi gaussienne d-dimensionnelle $X \sim \mathcal{N}(\mu, \Sigma)$ est définie, lorsque Σ est inversible, par sa densité

$$\rho_X(x) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}}\ e^{-(x-\mu).\Sigma^{-1}(x-\mu)/2} \ .$$

De nouveau on a $\mathbb{E}\{X\} = \mu$, de plus $\mathbb{E}\{(X_k - \mu_k)(X_\ell - \mu_\ell)\} = \Sigma_{k\ell}$.

Exemple 1.5 (Mélange de gaussiennes)

1. Dans le cas univarié, une variable aléatoire X suit une loi de mélange de deux gaussiennes, notée

$$X \sim p\mathcal{N}(\mu, \sigma) + (1-p)\mathcal{N}(\mu', \sigma') \ ,$$

si X est distribuée suivant $\mathcal{N}(\mu, \sigma)$ avec probabilité p, et suivant une loi $\mathcal{N}(\mu', \sigma')$ avec probabilité $1 - p$. Cette définition s'étend sans difficulté à des mélanges de plus de deux lois gaussiennes.

2. Cette définition s'étend sans difficulté aussi au cas de lois multivariées, i.e. variables aléatoires vectorielles.

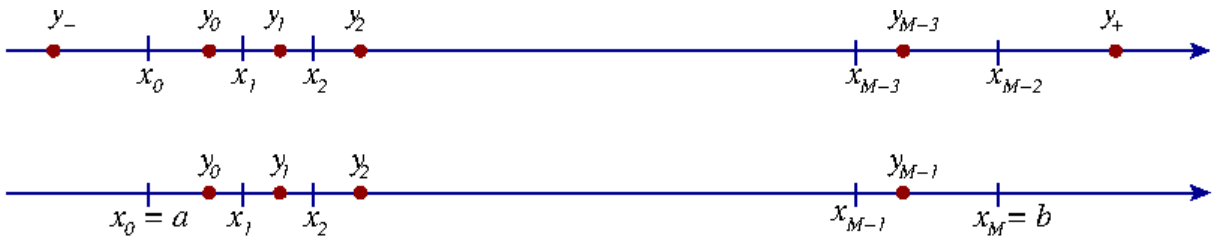

FIGURE 1.6 – Quantificateur scalaire : $Q : \mathbb{R} \to E_M$ (haut) et $Q : [a, b] \to E_m$ (bas)

1.2.2 Quantification scalaire

Définition 1.13 (Quantificateur scalaire) *Un quantificateur scalaire de taille M est une application non-linéaire*

$$Q : \mathbb{R} \to E_M = \{y_-, y_0, \dots y_{M-3}, y_+\} \ ,$$

définie à partir d'intervalles de la forme $[x_m, x_{m+1}[$ par

$$Q(x) = \begin{cases} y_- & si\ x \leq x_0 \\ y_m & si\ x \in [x_m, x_{m+1}[\ , \quad m = 0, \dots M-3 \\ y_+ & si\ x > x_{M-2} \end{cases} \tag{1.28}$$

Il s'agit donc d'une fonction constante par morceaux, telle que décrite dans la Figure 1.6 (haut).

Remarque 1.3 On peut également définir un quantificateur scalaire agissant sur un intervalle $[a, b[$, comme décrit dans la Figure 1.6 (bas) :

$$Q : [a, b] \to E_M = \{y_0, \dots y_{M-1}\} \ ,$$

On choisit alors $x_0 = a$ et $x_M = b$, et on définit

$$Q(x) = y_m \quad \text{si} \quad x \in [x_m, x_{m+1}[\ , \quad m = 0, \dots M-1 \ . \tag{1.29}$$

Dans ce qui suit on considèrera principalement le premier cas, mais le second s'étudie de façon tout à fait similaire. Notons que le choix d'un intervalle $[a, b[$ fermé à gauche et ouvert à droite est purement conventionnel, et sans aucune conséquence pratique.

On s'intéresse particulièrement à l'erreur commise par un tel quantificateur, appelée erreur de quantification. Pour obtenir une mesure de cette erreur, il est nécessaire d'aller plus avant dans la modélisation.

L'idée est de modéliser les échantillons ou coefficients comme des réalisations indépendantes d'une variable aléatoire X (en ignorant donc de possibles dépendances entre coefficients). Supposons que la variable aléatoire X soit une variable aléatoire continue du second ordre, de densité de probabilités notée ρ. La quantification lui associe la variable aléatoire $Y = Q(X)$, qui est une variable aléatoire discrète définie sur le même espace probabilisé. La précision du quantificateur se mesure alors en étudiant la variable aléatoire

$$Z = X - Y = X - Q(X) \ , \tag{1.30}$$

et on doit donc se donner une façon de mesurer cette quantité. Les quantités d'intérêt les plus simples sont ici la moyenne et la variance de l'erreur de quantification

Définition 1.14 *Soit $Q : \mathbb{R} \to E_M$ un quantificateur scalaire. Etant donnée une variable aléatoire X de densité de probabilités ρ, le biais et la distorsion associés au quantificateur Q sont les quantités*

$$B = \mathbb{E}\{X - Q(X)\} \ , \qquad D = \mathbb{E}\left\{(X - Q(X))^2\right\} \ . \tag{1.31}$$

Le quantificateur est non biaisé si $B = 0$.

Remarque 1.4 Nous avons utilisé ici une mesure quadratique d'erreur pour définir la distorsion. Il faut signaler qu'il est possible d'utiliser des mesures différentes. Dans ces cas là, la quantité D définie ci-dessus est appelée *distorsion quadratique* pour lever l'ambigüité.

Il est facile de vérifier, d'après la définition d'un quantificateur, que

$$B = \sum_{m=0}^{M-3} \left(\int_{x_m}^{x_{m+1}} (x - y_m)\rho(x)dx \right) + \int_{-\infty}^{x_0} (x - y_-)\rho(x)dx + \int_{x_{M-2}}^{\infty} (x - y_+)\rho(x)dx \; ; \tag{1.32}$$

par conséquent, une condition suffisante pour que le quantificateur soit non-biaisé lorsque l'on l'applique à une variable aléatoire de densité ρ est que

$$y_m = \frac{\int_{x_m}^{x_{m+1}} x\rho(x)\,dx}{\int_{x_m}^{x_{m+1}} \rho(x)\,dx} \; , \qquad y_- = \frac{\int_{-\infty}^{x_0} x\rho(x)\,dx}{\int_{-\infty}^{x_0} \rho(x)\,dx} \; , \qquad y_+ = \frac{\int_{x_{M-2}}^{\infty} x\rho(x)\,dx}{\int_{x_{M-2}}^{\infty} \rho_X(x)\,dx} \; . \tag{1.33}$$

Les conditions énoncées dans (1.33) portent le nom de *conditions de centroïde.*
De même, la distorsion s'écrit

$$D = \sum_{m=0}^{M-3} \left(\int_{x_m}^{x_{m+1}} (x - y_m)^2\rho(x)dx \right) + \int_{-\infty}^{x_0} (x - y_-)^2\rho(x)dx + \int_{x_{M-2}}^{\infty} (x - y_+)^2\rho(x)dx \; , \tag{1.34}$$

et c'est celle-ci que nous allons évaluer dans certaines situations simples. Les deux derniers termes de la distorsion forment le *bruit de saturation*, alors que les autres forment le *bruit granulaire.* Dans le cas d'un signal borné (c'est à dire quand ρ a un support borné), le bruit de saturation est généralement évité [2].

Définition 1.15 *On considère un quantificateur comme décrit ci-dessus. Le* facteur de performance *du quantificateur est le quotient*

$$\epsilon^2 = \frac{\sigma_X^2}{D} \; , \tag{1.35}$$

où $\sigma_X^2 = \mathbb{E}\{X^2\}$ *est la variance du signal, et* D *est la distorsion. Le* Rapport Signal à Bruit de Quantification *est quant à lui défini par.*

$$SNR_Q = 10\log_{10}(\epsilon^2) = 10\log_{10}\left(\frac{\sigma_X^2}{D}\right) \; . \tag{1.36}$$

Il est bien évident que l'objectif que l'on se fixe en développant un quantificateur est de maximiser le rapport signal à bruit, pour un débit R fixé. Ou, plus ambitieusement, on cherche à construire une *théorie Débit-Distorsion*, qui décrive l'évolution de la distorsion en fonction de R. On va voir que ceci est possible au prix d'approximations simplificatrices. On commencera par étudier le cas le plus simple, à savoir le cas de la quantification uniforme.

Remarque 1.5 On n'a considéré ici que le cas d'une variable aléatoire X dont la densité est à support non borné. On peut définir les mêmes quantités dans le cas d'une variable aléatoire à densité X dont la densité est à support borné, comme considéré dans la Remarque 1.3, disons $\mathsf{Supp}(\rho) \subset [a, b]$.
Le biais et la distorsion sont définis de façon similaire au cas précédent. Par exemple

$$D = \sum_{m=0}^{M-1} \left(\int_{x_m}^{x_{m+1}} (x - y_m)^2\rho(x)dx \right) \; .$$

2. Encore que ceci ne soit pas obligatoire ; on peut parfois se permettre une certaine quantité de bruit de saturation.

FIGURE 1.7 – Quantificateur scalaire uniforme

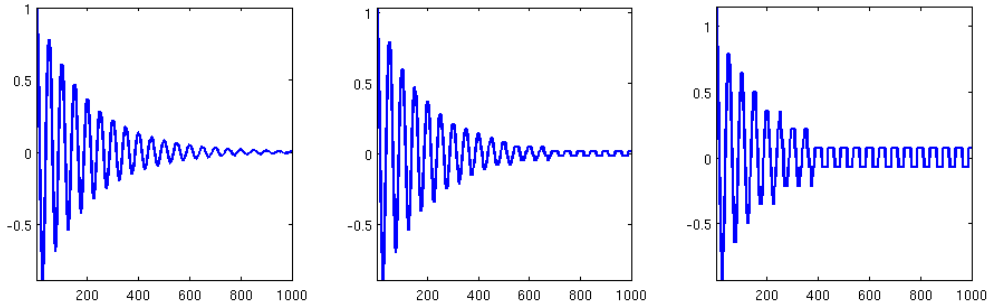

FIGURE 1.8 – Exemple de quantification : un signal simple (à gauche) et le même signal après quantification de chaque échantillon sur 6 bits (64 niveaux de quantification, milieu) et 4 bits (16 niveaux, droite).

1.2.3 Etude du quantificateur uniforme

On s'intéresse maintenant au cas le plus simple, à savoir le cas de la quantification uniforme.
Pour simplifier (en évitant d'avoir à considérer le bruit de saturation), on suppose pour cela que la variable aléatoire X est bornée, et prend ses valeurs dans un intervalle I. Une quantification uniforme consiste à découper I en $M = 2^R$ sous-intervalles de taille constante, notée Δ. Ceci est illustré en FIG. 1.7 (noter le changement de notations pour les indices). Plus précisément :

Définition 1.16 *Soit X une variable aléatoire dont la densité ρ_X est à support borné dans un intervalle $I = [x_{min}, x_{max}]$. Soit R un entier positif, et soit $M = 2^R$. Le quantificateur uniforme de débit R est donné par le choix*

$$x_0 = x_{min} \ , \quad x_m = x_0 + m\Delta \ , \tag{1.37}$$

avec

$$\Delta = |I|/M = |I|2^{-R} \ , \tag{1.38}$$

et

$$y_m = \frac{x_m + x_{m+1}}{2} \ . \tag{1.39}$$

L'effet de la quantification uniforme sur un signal est décrit en FIGS. 1.8 (signal synthétique) et 1.9 (signal audio).
Supposons que le quantificateur soit un quantificateur "haute résolution", c'est à dire qu'à l'intérieur d'un intervalle $[x_m, x_{m+1}]$, la densité de probabilités $x \to \rho(x)$ soit lentement variable, et puisse être approximée par la valeur

$$\rho_m = \rho(y_m) \ .$$

Sous ces conditions, on montre facilement que

$$\mathbb{E}\{X - Q(X)\} \approx 0 \ , \tag{1.40}$$

et on écrit alors

$$\int_{x_m}^{x_{m+1}} (x - y_m)^2 \rho(x) dx \approx \frac{\rho_m}{3} \left((x_{m+1} - y_m)^3 - (x_m - y_m)^3 \right) \ . \tag{1.41}$$

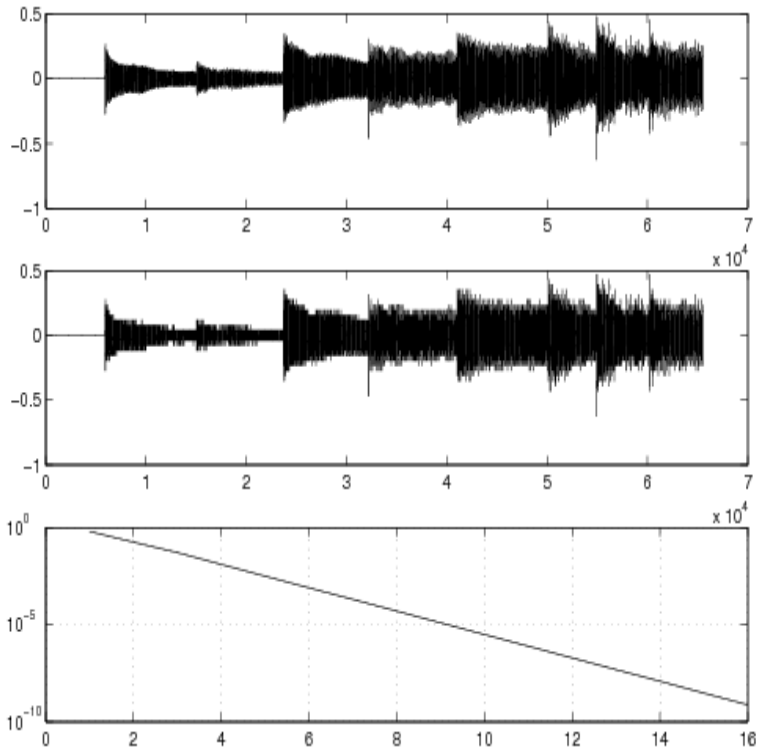

FIGURE 1.9 – Quantification d'un signal audio : un signal "test" (le "carillon" test des codeurs MPEG audio, en haut), une version quantifiée sur 4 bits (milieu), et le logarithme de la distorsion $D(R)$ en fonction du débit R (en bas).

Notons que pour un ρ_m donné, cette dernière quantité atteint son minimum (par rapport à y_m en $y_m = (x_m + x_{m+1})/2$, c'est à dire la valeur donnée en hypothèse, de sorte que $x_{m+1} - y_m = y_m - x_m = \Delta/2$. Par conséquent, on obtient

$$\int_{x_m}^{x_{m+1}} (x - y_m)^2 \rho(x) dx \approx \rho_m \frac{\Delta^3}{12} \ . \tag{1.42}$$

Par ailleurs, on écrit également $1 = \int \rho(x)dx \approx \Delta \sum_{m=0}^{M-1} \rho_m$, d'où

$$\sum_{m=0}^{M-1} \rho_m \approx \frac{1}{\Delta} \ . \tag{1.43}$$

Finalement, en faisant le bilan, on aboutit à

$$D \approx \frac{\Delta^3}{12} \sum_{m=0}^{M-1} \rho_m \approx \frac{\Delta^2}{12} \tag{1.44}$$

Remarque 1.6 Notons que d'après ces estimations, on obtient une estimation de la courbe débit-distortion fournie par la quantification uniforme :

$$D = K\, 2^{-2R} \ ,$$

où K est une constante.

Plus précisément, on montre que

Proposition 1.4 *Soit X une variable aléatoire bornée dans I, de densité ρ. Supposons en outre que $\rho \in C^1(\mathbb{R})$. Soit Q un quantificateur uniforme sur R bits par échantillon. Alors, on a*

$$D = \frac{\Delta^2}{12} + r = \frac{|I|^2}{12}\, 2^{-2R} + r \ , \tag{1.45}$$

avec

$$|r| \leq K\, 2^{-3R} \sup_x |\rho'(x)| \ . \tag{1.46}$$

Preuve : Il suffit de donner un sens plus précis à l'approximation (1.44). Par accroissements finis, on obtient

$$\int_{x_m}^{x_{m+1}} (x - y_m)^2 \rho(x)\, dx = \rho_m \frac{\Delta^3}{12} + \int_{x_m}^{x_{m+1}} (x - y_m)^3 \rho'(y)\, dx = \rho_m \frac{\Delta^3}{12} + r_m \ ,$$

pour un certain $y = y(x) \in [x_m, x_{m+1}]$. On a donc

$$|r_m| \leq \sup_{y \in [x_m, x_{m+1}]} |\rho'(y)| \int_{x_m}^{x_{m+1}} |x - y_m|^3\, dx \ .$$

Cette dernière intégrale vaut

$$2 \int_0^{\Delta/2} u^3 du = \frac{\Delta^4}{32} \ .$$

Donc,

$$|r| \leq \sum_{m=0}^{M-1} |r_m| \leq M \sup |\rho'| \frac{\Delta^4}{32} = \frac{|I|^4}{32} 2^{-3R} \sup |\rho'| \ .$$

L'estimation (1.43) se fait de façon similaire : calculons

$$1 = \sum_{m=0}^{M-1} \int_{x_m}^{x_{m+1}} \rho(x)\, dx = \sum_{m=0}^{M-1} \int_{x_m}^{x_{m+1}} [\rho_m + (x - y_m)\rho'(y)]\, dx = \Delta \sum_{m=0}^{M-1} \rho_m + \sum_m \int_{x_m}^{x_{m+1}} (x - y_m)\rho'(y)]\, dx \ ,$$

pour un certain $y = y(x) \in [x_m, x_{m+1}]$, d'où

$$\sum_{m=0}^{M-1} \rho_m = \frac{1}{\Delta} + r',$$

avec

$$|r'| = \frac{1}{\Delta} \left| \sum_{m=0}^{M-1} \int_{x_m}^{x_{m+1}} (x - y_m)\rho'(y)]\, dx \right| \leq \frac{2M}{\Delta} \sup |\rho'| \int_0^{\Delta/2} u\, du = \frac{|I|}{4} \sup |\rho'| \ .$$

En recollant les pièces du puzzle on aboutit à l'estimation désirée. ♠

Ces approximations permettent d'obtenir une première estimation pour l'évolution du SNR_Q en fonction du taux R. En effet, nous avons

$$SNR_Q = 10 \log_{10} \left(\frac{\sigma_X^2}{D} \right) = 20R \log_{10}(2) - 10 \log_{10} \left(\frac{|I|^2}{12\sigma_X^2} \right) \approx 6,02\, R + C^{ste}$$

où la constante dépend de la loi de X. Ainsi, on aboutit à la règle empirique suivante :

Ajouter un bit de quantification revient à
augmenter le rapport signal à bruit de quantification de 6dB environ.

Ceci est très bien illustré par la Fig. 1.9, qui représente un signal audio (un son de carillon, utilisé comme signal test par le consortium `MPEG`). La figure du haut représente le signal original, et la figure du milieu représente une version quantifiée sur 4 bits. Les distorsions sont assez visibles. La figure du bas représente quant à elle le rapport signal à bruit SNR_Q en fonction du débit R, pour un R variant de 1 à 16. On voit que comme attendu, le comportement est remarquablement proche d'un comportement linéaire.

Exemple 1.6 Prenons par exemple une variable aléatoire X, avec une loi uniforme sur un intervalle $I = [-x_0/2, x_0/2]$. On a alors

$$\sigma_X^2 = \frac{1}{x_0}\int_{-x_0/2}^{x_0/2} x^2 dx = \frac{x_0^2}{12}$$

de sorte que l'on obtient pour le rapport signal à bruit de quantification, exprimé en décibels (dB) :

$$SNR_Q = 20R\log_{10}(2) \approx 6,02\,R\ .$$

C'est le résultat standard que l'on obtient pour le codage des images par PCM.

1.2.4 Quantification scalaire optimale

Par définition, un quantificateur scalaire optimal est un quantificateur qui, pour un nombre de niveaux de quantification N fixé, minimise la distorsion. Il existe un algorithme, appelé *algorithme de Lloyd-Max*, qui permet d'atteindre l'optimum connaissant la densité de probabilités ρ_X de la variable aléatoire X à quantifier.
Supposons par exemple que nous ayions à quantifier une variable aléatoire X de densité ρ_X à support non borné.

Lemme 1.3 *Les paramètres d'un quantificateur optimal sont solutions du système d'équations*

$$\left\{\begin{array}{lcll} x_k &=& \frac{1}{2}(y_k + y_{k-1})\ , & k = 1, \dots M-3\ , \\ x_0 &=& \frac{1}{2}(y_0 + y_-) & \\ x_M &=& \frac{1}{2}(y_{M-1} + y_+) & \end{array}\right. \tag{1.47}$$

$$\left\{\begin{array}{lcll} y_k &=& \dfrac{\int_{x_k}^{x_{k+1}} x\rho_X(x)\,dx}{\int_{x_k}^{x_{k+1}} \rho_X(x)\,dx}\ , & k = 1, \dots M-3\ , \\ y_- &=& \dfrac{\int_{-\infty}^{x_0} x\rho_X(x)\,dx}{\int_{-\infty}^{x_0} \rho_X(x)\,dx}\ , & \\ y_+ &=& \dfrac{\int_{x_M}^{\infty} x\rho_X(x)\,dx}{\int_{x_M}^{\infty} \rho_X(x)\,dx}\ . & \end{array}\right. \tag{1.48}$$

Notons que les équations (1.48) ne sont autres que les conditions de centroïde données en (1.33) : les valeurs quantifiées sont les valeurs moyennes de la variable à quantifier à l'intérieur des intervalles de quantification.

Remarque 1.7 On obtient facilement des expressions similaires dans des situations où ρ_X est bornée. Dans ce cas, la condition de centroïde s'écrit

$$y_k = \frac{\int_{x_k}^{x_{k+1}} x\rho_X(x)\,dx}{\int_{x_k}^{x_{k+1}} \rho_X(x)\,dx}\ , \qquad k = 0, \dots M-1\ ,$$

et l'autre condition donne

$$x_k = \frac{y_k + y_{k-1}}{2}\ , \qquad k = 1, \dots M-1\ .$$

Remarque 1.8 (Mise en œuvre : algorithme de Lloyd-Max) La présence de deux systèmes d'équations dépendants suggère fortement de recourir à un algorithme itératif pour l'optimisation du quantifieur : étant données les bornes des intervalles on peut calculer les valeurs quantifiées et inversement, étant données les valeurs quantifiées on peut calculer les bornes des intervalles, et itérer ainsi jusqu'à convergence (ou arrêt forcé...).
Ceci dit, ces expressions supposent connue la densité de probabilités, ce qui n'est généralement pas le cas. On n'a généralement accès qu'à des échantillons. L'algorithme de Lloyd-Max fournit une alternative pratique très utilisée. L'idée est d'itérer deux étapes :

— Ré-estimation des valeurs quantifiées, via les conditions de centroïde, à partir d'échantillons affectés à des intervalles.
— Ré-affectation des échantillons à l'intervalle dont le centroïde est le plus proche.

Ceci conduit encore une fois à un algorithme itératif. Naturellement, rien ne garantit qu'il converge obligatoirement vers le minimum *global* de la distorsion. Lorsque tel n'est pas le cas (ce qui est en fait la situation la plus générale), on doit recourir à des méthodes plus sophistiquées.

1.2.5 Quantification scalaire et décision

Les techniques de quantification jouent également un rôle dans un cadre de théorie de la décision, qu'on va aborder dans les problèmes de modulation-démodulation. Sans entrer dans les détails, un modulateur transforme des suites binaires en signaux analogiques, qui sont transmis, puis retransformés en suites binaires après transmission. Le problème peut se modéliser sous la forme suivante : un bit aléatoire, modélisé par une variable aléatoire binaire, est transmis, et après transmission on dispose de suites de réels, qu'on doit retransformer en suites binaires. C'est donc un problème de décision : quels sont les bits qui ont été émis... qu'on peut rapprocher d'un problème de quantification.
Commençons par illustrer la problématique sur le cas simple d'un quantificateur sur 1 bit (donc deux niveaux de quantification).

Exemple 1.7 Considérons le modèle simple suivant. Soit B une variable aléatoire de Bernoulli, prenant les valeurs 0 et 1 avec probabilités respectives p et $1-p$. Soient $\rho_0, \rho_1 \in C(\mathbb{R})$ deux densités de probabilités, de moyennes respectives μ_0 et μ_1, supposées continues. Sans perte de généralité, on suppose $\mu_0 \leq \mu_1$. On considère la variable aléatoire X définie par la loi conditionnelle

$$\rho_{X|B=i}(x) = \rho_i(x) \ , \quad i = 0, 1 \ .$$

Le problème de décision est de déterminer la valeur prise par B étant donnée une observation x, réalisation de X, en minimisant la probabilité d'erreur, mais aussi d'obtenir des estimations sur la précision.
Soit τ un réel, on lui associe la fonction de décision φ défini par

$$\varphi(x) = \begin{cases} 0 & \text{si } x \leq \tau \\ 1 & \text{sinon} \end{cases}$$

On introduit la variable aléatoire $B' = \varphi(X)$. Alors on a la probabilité d'erreur

$$\begin{aligned} \mathbb{P}\{B' \neq B\} &= \mathbb{P}\{B' = 1 | B = 0\}\mathbb{P}\{B = 0\} + \mathbb{P}\{B' = 0 | B = 1\}\mathbb{P}\{B = 1\} \\ &= pF_0(\tau) + (1-p)(1 - F_1(\tau)) \ , \end{aligned}$$

où F_0 et F_1 sont les fonctions de répartition associées respectivement à ρ_0 et ρ_1.
La probabilité d'erreur est extrêmale si la dérivée de cette quantité par rapport à τ s'annule, ce qui est obtenu quand

$$\frac{\rho_0(\tau)}{\rho_1(\tau)} = \frac{1-p}{p} \ .$$

Notons que dans ce cas, on cherche encore à optimiser un quantificateur, mais la mesure de qualité employée n'est plus la distorsion mais une probabilité d'erreur.
Plus généralement, considérons une variable aléatoire discrète U prenant $M = 2^R$ valeurs, disons $0, 1, \ldots M-1$, et soit X une variable aléatoire réelle à densité. On suppose connues les densités conditionnelles de X sachant U, notées

$$\rho_k = \rho_{X|U=k} \ .$$

Soit Q un quantificateur sur $M = 2^R$ niveaux, comme défini en (1.28), soit φ la fonction définie par

$$\varphi(x) = \begin{cases} 0 & \text{si} \quad x \leq x_0 \\ k+1 & \text{si} \quad x_k < x \leq x_{k+1} \\ M-1 & \text{si} \quad x > x_{M-2} \end{cases}$$

et soit

$$V = \varphi(X)$$

la variable aléatoire "décodée". On s'intéresse aux probabilités

$$\mathbb{P}\{V = U\} \qquad \text{et} \qquad \mathbb{P}\{V \neq U\} \ .$$

La première des deux est la plus facile à calculer, et conduit à l'expression

$$\mathbb{P}\{V = U\} = \sum_k p_k \mathbb{P}\{V = k | U = k\} = \sum_k p_k \mathbb{P}\{x_k < X \leq x_{k+1} | U = k\} = \sum_k p_k \int_{x_k}^{x_{k+1}} \rho_k(x) dx \ .$$

L'objectif étant de maximiser cette quantité, écrivons les équations normales. En dérivant par rapport à x_k, on obtient

$$p_{k-1}\rho_{k-1}(x_k) - p_k \rho_k(x_k) = 0 \ ,$$

d'où la condition

$$\frac{\rho_k(x_k)}{\rho_{k-1}(x_k)} = \frac{p_{k-1}}{p_k} \ . \tag{1.49}$$

Si l'on sait résoudre ces équations, alors on peut obtenir les valeurs optimales des bornes x_k des intervalles de décision.

Exemple 1.8 (Cas d'une loi Laplacienne) Supposons que les densités ρ_k sont de la forme

$$\rho_k(x) = \frac{\lambda_k}{2} e^{-\lambda_k |x - \mu_k|} \ ,$$

où $\mu_k \in \mathbb{R}$ est la moyenne de ρ_k et $\lambda_k \in \mathbb{R}^+$ contrôle la variance (qui vaut $\sigma_k^2 = 2/\lambda_k^2$). Dans ce cas, les expressions se simplifient et conduisent à une forme explicite pour les bornes des intervalles de décision :

$$x_k = \frac{\lambda_k \mu_k + \lambda_{k-1}\mu_{k-1}}{\lambda_k + \lambda_{k-1}} + \frac{1}{\lambda_k + \lambda_{k-1}} \ln\left(\frac{p_{k-1}\lambda_{k-1}}{p_k \lambda_k}\right) \ .$$

En particulier, cette forme devient très simple quand tous les paramètres λ sont égaux :

$$x_k = \frac{\mu_k + \mu_{k-1}}{2} + \frac{1}{2} \ln\left(\frac{p_{k-1}}{p_k}\right) \ ,$$

et les rôles des deux termes sont faciles à interpréter.

1.2.6 Quantification vectorielle

1.2.6.1 Définitions

Commençons par quelques notions, qui étendent la quantification scalaire au cas vectoriel.

Définition 1.17 *Un quantificateur vectoriel de dimension N et taille M est une application non-linéaire*

$$Q : \mathbb{R}^N \to E_M = \{y_0, \ldots y_{M-1}\}$$

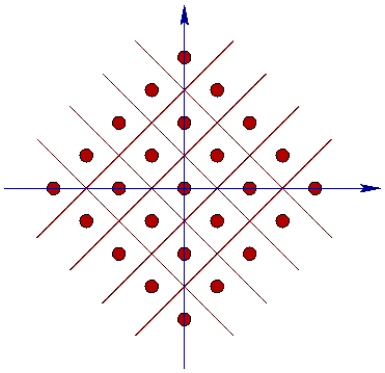

FIGURE 1.10 – Partition de Voronoi dans le plan, pour un ensemble E_M de cardinalité $M = 25$.

défini à partir d'une partition de $\mathbb{R}^N$ *en* M *cellules de quantification*

$$\mathbb{R}^N = \bigcup_{m=0}^{M-1} \mathcal{C}_m$$

par une règle d'association

$$Q(x) = y_m \quad si \quad x \in \mathcal{C}_m \ .$$

Le quantificateur est régulier si les cellules $\mathcal{C}_m$ *sont convexes, et si* $y_m \in \mathcal{C}_m$ *pour tout* $m = 0, \dots M-1$.

Exemple 1.9 (Partition de Voronoi) Dans $\mathbb{R}^N$, muni de la distance Euclidienne, on peut associer à un ensemble $E_M = \{y_0, \dots y_{M-1}\}$ de points de $\mathbb{R}^N$ une partition de $\mathbb{R}^N$, appelée partition de Voronoi, construite comme suit. Les cellules de Voronoi sont définies par

$$\mathcal{C}_m^{\mathsf{Vor}} = \{x \in \mathbb{R}^N \ / \ \forall n \in E_M, \ \|x - y_m\| \leq \|x - y_n\|\} \ .$$

— Il est assez facile de raisonner dans le cas $N = 2$. Dans ce cas, étant donnés deux points y_m, y_n, la frontière entre les cellules de Voronoi se trouve sur la médiatrice $\Pi(y_m, y_n)$ de ces deux points. En notant $H_m(n)$ le demi-plan de bord $\Pi(y_m, y_n)$ contenant y_m, la cellule $\mathcal{C}_m$ est l'intersection des demi-plans $H_m(n)$, $n \neq m$:

$$\mathcal{C}_m = \bigcap_{n \neq m} H_m(n) \ .$$

— Une fois bien compris le cas $N = 2$, le cas général s'en déduit simplement. Etant donnés deux points y_m, y_n, la frontière entre les cellules de Voronoi se trouve sur l'hyperplan affine $\Pi(y_m, y_n)$ constitué des points situés à égale distance de ces deux points. Si $H_m(n)$ est le demi-espace de bord $\Pi(y_m, y_n)$ contenant y_m, la cellule $\mathcal{C}_m$ est l'intersection des demi-espaces $H_m(n)$, $n \neq m$ donnée par l'expression ci-dessus. Un exemple est donné en Figure 1.10

Un quantificateur associé à une partition de Voronoi est régulier par construction.

1.2.6.2 Performances d'un quantificateur vectoriel

Les performances d'un quantificateur vectoriel s'étudient de façon assez similaire aux performances d'un quantificateur scalaire. En particulier, cette notion n'a de sens que lorsque l'on précise les caractéristiques des vecteurs que l'on doit quantifier.

On a généralement recours pour modéliser ces derniers à une variable aléatoire vectorielle $X \in \mathbb{R}^N$, de densité supposée connue ρ, et on mesure les performances via des mesures de la forme

$$B = \mathbb{E}\{X - Q(X)\} = \sum_{m=0}^{M-1} \int_{\mathcal{C}_m} (x - y_m)\rho(x)\, dx \ , \quad D = \mathbb{E}\{\phi(X, Q(X))\} = \sum_{m=0}^{M-1} \int_{\mathcal{C}_m} \phi(x, y_m)\rho(x)\, dx \ ,$$

où ϕ est une mesure de dissimilarité. Le choix le plus classique est de prendre pour ϕ le carré de la distance euclidienne

$$\phi(x, y) = \|x - y\|^2 \ ,$$

ce qui conduit à la définition usuelle de la distorsion, ou distorsion quadratique.

$$D = \mathbb{E}\left\{\phi(X, Q(X))\right\} = \sum_{m=0}^{M-1} \int_{\mathcal{C}_m} \|x - y_m\|^2 \rho(x)\, dx \ . \tag{1.50}$$

D'autres choix sont également considérés, par exemple $\phi(x, y) = \|x - y\|_1$.

Remarque 1.9 (Quantification vectorielle optimale) La problématique de l'optimisation de la quantification, pour une distribution de probabilités donnée, peut être posée comme dans le cas de la quantification scalaire. Etant donnée une densité de probabilités en dimension N, il s'agit de déterminer une partition de $\mathbb{R}^N$ en M cellules $\mathcal{C}_m$ et des représentants y_m de chaque cellule assurant que la distorsion définie en (1.50) soit minimale. Ceci conduit à des équations similaires aux équations obtenues dans la section 1.2.4. Par exemple, la condition de centroïde s'écrit

$$y_m = \frac{\int_{\mathcal{C}_m} x\rho(x)\, dx}{\int_{\mathcal{C}_m} \rho(x)\, dx}$$

alors que l'autre condition exprime le fait que la frontière entre deux cellules voisines se trouve sur la médiatrice du segment reliant les deux centres, ce qui correspond aux partitions de Voronoi.

Remarque 1.10 (Quantification vectorielle et décision) La quantification vectorielle peut également être adaptée aux problématiques de décision, telles que nous les avons vues dans la section 1.2.5. Il suffit d'adapter la discussion au cas d'un vecteur aléatoire X de dimension N, dont on connaît la loi conditionnellement à une variable aléatoire discrète B à valeurs dans $\{0, 1 \ldots M-1\}$ représentant les cellules d'une partition de $\mathbb{R}^N$. Le raisonnement est identique, et conduit à rechercher une partition optimale de $\mathbb{R}^N$ à partir de la distribution de X.

Chapitre

2 Codage-Décodage : théorie, exercices et projet

On s'intéresse ici à une simple chaîne de transmission de signaux, appelée CoDec (pour codage-décodage). Une telle chaîne consiste en une succession d'opérations, qui peuvent (ou pas) faire intervenir des pertes d'information (appelées distorsions). L'objectif est de contrôler et limiter autant que possible ces distorsions.
Les éléments de base d'un CoDec sont les opérations suivantes :

- La conversion "analogique → numérique" (ou *AD conversion* en anglais) associe à un signal analogique (donc une fonction d'une variable continue) une suite de 0 et 1. Elle consiste elle-même en trois sous opérations :
  1. l'échantillonnage, qui remplace la fonction par une suite de nombres réels,
  2. la quantification qui approxime les nombres réels par des nombres avec une précision finie,
  3. le codage binaire, qui transforme les nombres quantifiés en suites binaires.

  L'échantillonnage est souvent précédé d'une étape supplémentaire, appelée préfiltrage, dans laquelle le signal à coder est modifié afin que l'échantillonnage n'apporte pas de distorsion supplémentaire.
- Un codage correcteur d'erreurs, destiné à introduire de la robustesse dans le codage, quand celui-ci est suivi d'une opération "physique" (telle que la gravure d'un CD ou DVD, ou plus généralement l'écriture sur un disque, ou toute transmission à travers un *canal de transmission*), susceptible d'introduire des erreurs.
- La conversion "Numérique → Analogique", qui restaure un signal analogique que l'on essaie de rendre le plus proche possible de l'original.

Ceci est schématisé dans le diagramme de la figure 2.1.

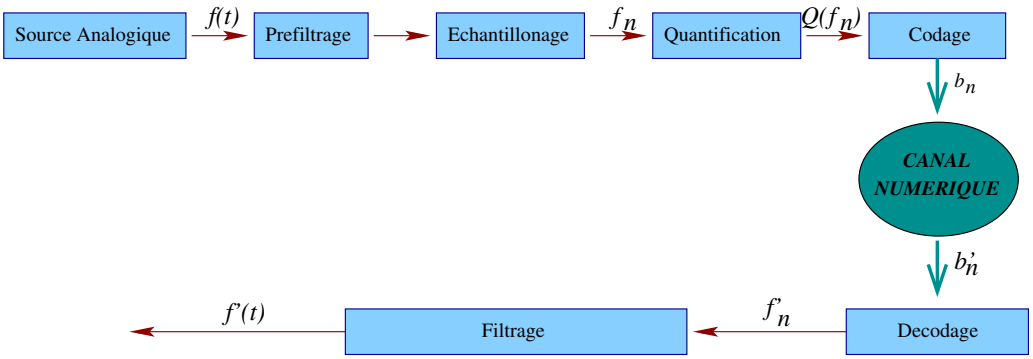

Figure 2.1 – Diagramme représentant un CoDec standard

L'objectif d'un CoDec est d'être le plus performant possible en termes de *débit-distorsion* : minimiser une mesure d'erreur entre le signal original et le signal restauré, tout en minimisant l'espace requis pour le stockage.
Avant d'entrer plus en détails dans l'architecture d'un CoDec, commençons par décrire un aspect qui n'a pas été traité dans le chapitre précédent : le codage binaire.

2.1 Eléments de théorie de l'information et codage

La théorie de l'information est une théorie probabiliste permettant de quantifier le contenu moyen en information d'un ensemble de messages, dont le codage informatique satisfait une distribution statistique précise. Ce domaine a été fondé par Claude Shannon en 1948, dans un contexte de théorie des communications.
Le paragraphe ci-dessous, qui décrit les fondements de la notion d'information, est copié de l'encyclopédie en ligne Wikipedia

Sur la notion d'information (Wikipedia)

Pour Shannon, l'information présente un caractère essentiellement aléatoire. Un événement aléatoire est par définition incertain. Cette incertitude est prise comme mesure de l'information. Une information sera donc uniquement définie par sa probabilité (I = - log p). Donc l'information est la mesure de l'incertitude calculée à partir de la probabilité de l'événement. Shannon a donc confondu la notion d'information et de mesure d'incertitude. Il faut remarquer que dans cette définition l'information est bien synonyme de mesure d'incertitude. Dans cet ordre d'idée, plus une information est incertaine, plus elle est intéressante, et un événement certain ne contient aucune information. En théorie de l'information de Shannon, il s'agit donc de raisonner en probabilité et non en logique pure.

L'information de Shannon se mesure en unités binaires dites bits. Le bit peut être défini comme un événement qui dénoue l'incertitude d'un récepteur placé devant une alternative dont les deux issues sont pour lui équiprobables. Plus les éventualités que peut envisager ce récepteur sont nombreuses, plus le message comporte d'événements informatifs, plus s'accroît la quantité de bits transmis. Il est clair que nul récepteur ne mesure en bits l'information obtenue dans un message. C'est seulement le constructeur d'un canal de télécommunication qui a besoin de la théorie, et mesure l'information en bits pour rendre la transmission de message la plus économique et la plus fiable.

La notion d'information d'après Shannon est nécessairement associée à la notion de « redondance » et à celle de « bruit ». Par exemple, en linguistique l'information n'est ni dans le mot, ni dans la syllabe, ni dans la lettre. Il y a des lettres voire des syllabes qui sont inutiles à la transmission de l'information que contient le mot : il y a dans une phrase, des mots inutiles à la transmission de l'information. La théorie de Shannon appelle redondance tout ce qui dans le message apparaît comme en surplus. Aussi est-il économique de ne pas transmettre la redondance.

L'information chemine à travers un canal matériel/énergétique : fil téléphonique, onde radio, etc. Or, dans son cheminement, l'information rencontre du bruit. Le bruit est constitué par les perturbations aléatoires de toutes sortes qui surgissent dans le canal de transmission et tendent à brouiller le message. Le problème de la dégradation de l'information par le bruit est donc un problème inhérent à sa communication. Ici, l'idée de redondance présente une face nouvelle ; alors qu'elle apparaît comme un surplus inutile sous l'angle économique, elle devient, sous l'angle de la fiabilité de la transmission un fortifiant contre le bruit, un préventif contre les risques d'ambiguïté et d'erreur à la réception.

2.1.1 Entropie

Dans leur article fondateur, Shannon et Weaver définissent l'information associée à une distribution de probabilités de la façon suivante. Etant donnée une variable aléatoire X prenant les valeurs $\{x_1, \dots x_N\}$ avec probabilités respectives $P = \{p_1, \dots p_N\}$, chaque valeur possible x_i porte une quantité d'information égale à $-\log_2(p_i)$. L'information portée par X est alors mesurée par la valeur moyenne $-\sum_{i=1}^{N} p_i \log_2(p_i)$ de ces quantités, et porte le nom d'entropie.

Définition 2.1 *Soit $P = \{p_n, n = 1, \dots N\}$ une distribution de probabilités de taille N. L'entropie de Shannon correspondante est définie par*

$$H(P) = -\sum_{n=1}^{N} p_n \log_2(p_n) \ . \tag{2.1}$$

L'entropie associée à une distribution de probabilités représente la quantité d'information reçue lorsqu'un évènement $n \in \{0, \dots N-1\}$ est observé. L'entropie possède nombre de propriétés simples. En particulier, on a

Lemme 2.1 *Soit $P = \{p_1, \dots p_N\}$ une distribution de probabilités. Alors*

$$0 \leq H(P) \leq \log_2(N) \ .$$

Preuve : L'entropie est positive par construction. Pour montrer la borne supérieure, il suffit de trouver la distribution de probabilités de taille N qui maximise H (avec évidemment la contrainte $\sum_{n=1}^{N} p_n = 1$). Cette contrainte peut être imposée en introduisant un multiplicateur de Lagrange λ, on se ramène alors à maximiser

$$\Phi(P, \lambda) = -\sum_{n=1}^{N} p_n \log_2(p_n) + \lambda \left(\sum_{n=1}^{N} p_n - 1 \right)$$

par rapport aux nombres p_n et au multiplicateur λ. On voit facilement qu'égaler à zéro la dérivée de Φ par rapport au multiplicateur λ revient à imposer la contrainte. Par contre, étant donné un n quelconque, il est facile de voir que si $p_n \neq 0$,

$$\frac{\partial}{\partial p_n} \Phi(p, \lambda) = -\log_2(p_n) - \frac{1}{\log(2)} + \lambda \ .$$

Ainsi, pour tout n tel que $p_n \neq 0$, on a $p_n = \mu$ où μ est une constante. En imposant la contrainte, et en notant m le nombre de p_n non nuls, on a $\sum_{n=1}^{N} p_n = n\mu = 1$, d'où $\mu = 1/m$. Finalement, l'entropie vaut alors

$$H(P) = -\sum_{n=1}^{N} p_n \log_2(p_n) = \log_2(m) \ .$$

Cette dernière expression est minimale pour $m = 1$, et vaut alors $H(p) = 0$, et maximale pour $n = N$, la solution étant alors donnée par $p_n = 1/N$, d'où le résultat. ♠

Ainsi, les faibles valeurs de l'entropie correspondent à des situations dans lesquelles un petit nombre de symboles sont très probables (donc pour lesquels $\log p_n$ est faible) et un grand nombre de symboles sont très probables (donc pour lesquels p_n est faible).

On a aussi le résultat important suivant

Lemme 2.2 (Inégalité de Gibbs) *Etant données deux distributions de probabilités P et Q de taille N, on a*

$$-\sum_{n=1}^{N} p_n \log_2(q_n) \geq H(P) \ . \tag{2.2}$$

avec égalité si et seulement si les deux distributions sont identiques.

Preuve : On va utiliser le fait que $\log(x) \leq x - 1$, avec égalité si et seulement si $x = 1$. Calculons

$$\begin{aligned} \sum_{n=1}^{N} p_n \log_2 \left(\frac{q_n}{p_n} \right) &= \frac{1}{\log(2)} \sum_{n=1}^{N} p_n \log \left(\frac{q_n}{p_n} \right) \\ &\leq \frac{1}{\log(2)} \sum_{n=1}^{N} p_n \left(\frac{q_n}{p_n} - 1 \right) \\ &= 0 \ , \end{aligned}$$

avec égalité si et seulement si $p_n = q_n$ pour tout $n = 1, \dots N$. Ceci prouve le lemme. ♠

2.1.2 Codage binaire

La dernière étape du codage d'un signal, postérieure à la quantification, est le codage binaire, qui vise à associer des suites binaires (c'est à dire des suites de 0 et de 1) à un ensemble $\mathcal{A}$ de symboles.
Le cas le plus simple est celui des codes de longueur constante (CLC) : tous les symboles sont codés sur un nombre R constant de bits, le code étant fixé une fois pour toutes. Cependant, il s'avère souvent plus efficace d'utiliser des codes de longueur variable, c'est à dire des codes dans lesquels on n'affecte pas nécessairement un code de la même longueur à différents symboles (dans notre cas, des coefficients quantifiés). On pourra ainsi affecter un code très court à des symboles très fréquent, et un code plus long à des symboles moins fréquents, afin d'optimiser le débit total. On rappelle que dans le cas d'un code de longueur constante, chaque symbole est codé sur R bits, ce qui correspond à $M = 2^R$ symboles. Les codes de longueur variable posent des problèmes différents, qu'on va étudier ci dessous.

2.1.2.1 Codes de longueur variable (CLV)

Pour fixer les idées, on considère une suite de variables aléatoires X_n, prenant leurs valeurs dans un alphabet fini $A = \{a_0, a_1, \dots a_{M-1}\}$, avec des probabilités $\mathbb{P}\{X_n = a\} = p(a)$. Un codeur associera à chaque symbole $a \in A$ un mot binaire $\alpha(a)$, de longueur $\ell(a)$ (mesurée en bits). Le meilleur codeur sera celui qui minimise la longueur moyenne

$$\overline{\ell} = \sum_{a \in A} p(a)\ell(a) \ . \tag{2.3}$$

On rappelle que dans le cas d'un code de longueur constante, chaque symbole a_k est codé sur $\ell(a_k) = R$ bits, ce qui correspond à $M = 2^R$ symboles. Par conséquent, on a

$$\sum_{a \in A} 2^{-\ell(a)} = 1 \ . \tag{2.4}$$

On verra par la suite la signification de cette identité simple.

Définition 2.2 *Un code scalaire sans perte de longueur variable consiste en*

1. *Un codeur : une application α qui associe à tout symbole d'entrée $a \in A$ une suite binaire $\alpha(a)$, de longueur $\ell(a)$.*
2. *Un décodeur : une application β qui associe à toute suite binaire u un symbole $a = \beta(u) \in A$, tel que pour tout $a \in A$, $\beta(\alpha(a)) = a$.*

Le codeur est ensuite étendu aux suites de symboles (les mots) par concaténation : la suite binaire associée au mot $x_1x_2 \dots x_n$ est la concaténation

$$\alpha(x_1x_2 \dots x_n) = \alpha(x_1)\alpha(x_2) \dots \alpha(x_n) \ .$$

Par exemple, si $\alpha(x_1) = 0110$ et $\alpha(x_2) = 111$, le code binaire associé au mot x_1x_2 sera $\alpha(x_1x_2) = 0110111$.
La concaténation pose des problèmes si le code est un code de longueur variable. Dans le cas général, on ne sait pas où commencent et où s'arrêtent les mots. On introduit donc une sous-classe de codes, appelés codes uniquement décodables, définis comme suit.

Définition 2.3 *Le code est dit uniquement décodable si le codeur α est injectif : c'est à dire si toute suite binaire $\alpha(x_1x_2 \dots x_n)$ n'est l'image que d'un et un seul mot, à savoir $x_1x_2 \dots x_n$.*

Exemple 2.1 On considère les deux codeurs α_1 et α_2 et α_3 définis par les tableaux donnés dans la Table 2.1. On vérifie immédiatement que le code donné dans le tableau de gauche n'est pas uniquement décodable, alors que les deux autres le sont. Par contre, le code du tableau du milieu n'est pas *instantané*, au sens où il faut attendre le début du mot suivant pour savoir si un mot est terminé : 11 peut être suivi de 1, auquel cas il s'agit d'un a_4, ou d'un 0, auquel cas il s'agit d'un a_3. Le code présenté dans le tableau de droite est quant à lui un code uniquement décodable instantané.

input	code
a_0	0
a_1	10
a_2	101
a_3	0101

input	code
a_0	0
a_1	01
a_2	011
a_3	111

input	code
a_0	0
a_1	10
a_2	110
a_3	111

TABLE 2.1 – Trois exemples de codeurs : celui de gauche n'est pas uniquement décodable, les deux autres le sont. Le codeur du centre n'est pas instantané.

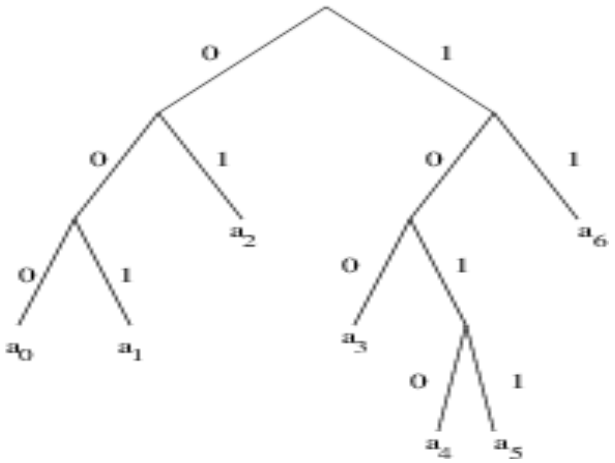

FIGURE 2.2 – Exemple d'arbre binaire associé à un alphabet.

2.1.2.2 Codes à préfixe, et codes associés à un arbre binaire

Il existe une façon simple d'assurer qu'un code est uniquement décodable et instantané. Il suffit d'imposer une condition, appelée *condition de préfixe*..

Définition 2.4 *Un codeur α satisfait la condition de préfixe si aucun mot binaire $\alpha(a), a \in A$ n'est préfixe d'un autre mot $\alpha(b), b \in A$. Un code satisfaisant la condition de préfixe est appelé* code à préfixe.

Il existe une construction générique de codes à préfixes. On peut associer un tel code à tout arbre binaire. Plus précisément, étant donné un alphabet $A = \{a_0, \dots a_{M-1}\}$ à M symboles, on considère un arbre binaire à M feuilles. On associe à chaque symbole $a \in A$ une des feuilles, et on numérote les branches de l'arbre par des bits : par exemple, les branches partant vers la gauche sont notées "0", et les branches partant vers la droite sont notées "1". Le code associé à chaque symbole est alors la concaténation des étiquettes des branches consécutives menant de la racine de l'arbre à la feuille considérée.

2.1.2.3 L'inégalité de Kraft

L'inégalité de Kraft montre que pour une distribution de probabilités de symboles donnée, les longueurs de mots binaires associés à ces symboles doivent, *dans une certaine moyenne*, être supérieures à une valeurs seuil.

input	a_0	a_1	a_2	a_3	a_4	a_5	a_6
code	000	001	01	100	1010	1011	11

TABLE 2.2 – Code associé à l'arbre binaire

Théorème 2.1 (Inégalité de Kraft) *Soit $A = \{a_0, \dots a_{M-1}\}$ un alphabet à M symboles. Le codeur α, qui associe à chaque $a \in A$ le mot $\alpha(a)$ de longueur $\ell(a)$, est uniquement décodable seulement si l'inégalité suivante, appelée* inégalité de Kraft *, est vérifiée :*

$$\sum_{a\in A} 2^{-\ell(a)} \leq 1 \ . \tag{2.5}$$

Preuve : Soit K un entier positif fixé. On considère une suite de K symboles $b = \{b_0, \dots, b_{K-1}\}$; la longueur totale de $\alpha(b_0 \dots b_{K-1})$ vaut

$$\ell(b) = \ell(b_0) + \cdots + \ell(b_{K-1}) \ .$$

On notera $N(L)$ le nombre de suites de K symboles ayant une longueur totale égale à L. On pose $\ell_{max} = \max_{a\in A} \ell(a)$. On a alors

$$\ell(b) \leq K\ell_{max} := L_{max} \ .$$

Le code étant uniquement décodable, les $N(L)$ suites ont un code différent. Comme il y a au plus 2^L codes de longueur L différents, on a

$$N(L) \leq 2^L \ .$$

Calculons alors

$$\begin{aligned}
\left[\sum_{m=0}^{M-1} 2^{-\ell(a_m)}\right]^K &= \left[\sum_{b\in A} 2^{-\ell(b)}\right]^K \\
&= \sum_{b_0\in A} 2^{-\ell(b_0)} \sum_{b_1\in A} 2^{-\ell(b_1)} \cdots \sum_{b_{K-1}\in A} 2^{-\ell(b_{K-1})} \\
&= \sum_{b_0\in A}\sum_{b_1\in A}\cdots\sum_{b_{K-1}\in A} 2^{-\ell(b_0)-\cdots-\ell(b_{K-1})} \\
&= \sum_{L=1}^{L_{max}} N(L)2^{-L} \\
&\leq L_{max} = K\ell_{max} \ .
\end{aligned}$$

Par conséquent, on a

$$\sum_{m=0}^{M-1} 2^{-\ell(a_m)} \leq (K\ell_{max})^{1/K} \ .$$

Ceci étant vrai pour tout K, on obtient bien l'inégalité de Kraft par passage à la limite $K \to \infty$. En effet,

$$\lim_{K\to\infty} \ln\left((K\ell_{max})^{1/K}\right) = \lim_{K\to\infty} \frac{\ln(K\ell_{max})}{K} = 0 \ .$$

Ceci conclut la preuve du théorème. ♠

Théorème 2.2 *Soit $A = \{a_0, \dots a_{M-1}\}$ un alphabet à M symboles, et soit $\{\ell_0, \dots, \ell_{M-1}\}$ une suite d'entiers positifs satisfaisant l'inégalité de Kraft (2.5). Alors il existe un codeur α uniquement décodable, tel que pour tout $k = 0, \dots M-1$, $\ell(a_k) = \ell_k$.*

2.1.2.4 Le théorème de Shannon-Fano

Théorème 2.3 (Inégalités de Shannon-Fano) *Soit $A = \{a_0, \dots, a_{M-1}\}$ un alphabet, et soit p la distribution de probabilités de ses symboles.*

1. *Soit α un codeur dont les longueurs de mots $\ell_0, \dots \ell_{M-1}$ satisfont à l'inégalité de Kraft. Alors on a*
$$\overline{\ell}(\alpha) \geq H(p) \ , \tag{2.6}$$
avec égalité si et seulement si $p_n = 2^{-\ell_n}$ pour tout $n = 0, \dots M-1$.
2. *Il existe un code uniquement décodable tel que*
$$\overline{\ell}(\alpha) < H(p) + 1 \ . \tag{2.7}$$

Preuve : La première partie est une conséquence directe de l'inégalité de Gibbs :

$$\overline{\ell}(\alpha) \geq -\sum_{n=0}^{M-1} p_n \log_2 q_n \geq H(p) \ ,$$

et on a égalité (i.e. les deux inégalités intervenant dans celle-ci sont des égalités) si et seulement si $p_n = q_n = 2^{-\ell_n}$ pour tout n. Pour la seconde partie, étant donnée la distribution p, soient les nombres ℓ_k définis par

$$2^{-\ell_k} \leq p_k < 2^{1-\ell_k} \ .$$

Il est évident que $\sum_k 2^{-\ell_k} \leq \sum_k p_k = 1$, donc l'inégalité de Kraft est satisfaite. Il existe donc un code uniquement décodable α associé aux longueurs de mots ℓ_k. De plus, on a

$$\ell_k < 1 - \log_2(p_k) \ ,$$

et donc

$$\overline{\ell}(\alpha) = \sum_k p_k \ell_k < \sum_k p_k (1 - \log_2(p_k)) = 1 + H(p) \ .$$

Le code correspondant est appelé code de Fano. Ceci conclut la démonstration. ♠

2.1.2.5 Le code de Huffman

Le code de Huffman est un algorithme récursif qui permet d'atteindre les performances optimales. Il est basé sur la règle d'agrégation suivante.
Partant d'un alphabet $A = \{a_0, a_1, \dots a_{M-1}\}$ dont les symboles ont probabilités $(p_0, p_1, \dots p_{M-1})$, on suppose (sans perte de généralité)que les symboles sont ordonnés par probabilités décroissantes : $p_0 \geq p_1 \geq \dots$ (si nécessaire, on peut les ré-ordonner). On construit un nouvel alphabet de longueur $M-1$ en agrégeant les symboles a_{M-1} et a_{M-2} en un nouveau symbole $\tilde{a}$ auquel on affecte la probabilité

$$p(\tilde{a}) = p(a_{M-1}) + p(a_{M-2}) \ .$$

Théorème 2.4 *Un arbre de préfixe optimal pour l'alphabet à M symboles ordonnés $A = \{a_0, a_1, \dots a_{M-1}\}$ s'obtient à partir d'un arbre de préfixe optimal pour $\{a_0, a_1, \dots a_{M-3}, \tilde{a}\}$, en adjoignant à ce dernier deux branches liant $\tilde{a}$ aux symboles a_{M-2} et a_{M-1}.*

L'algorithme de Huffman exploite ce résultat de la façon suivante : a_{M-1} et a_{M-2} sont les feuilles extrêmes de l'arbre. Les deux branches correspondantes sont caractérisées par un bit.
On est donc en présence d'un nouvel alphabet de $M-1$ symboles

$$\{a_0, a_1, \dots a_{M-3}, \tilde{a}\} \ .$$

Pour poursuivre l'algorithme, il suffit alors de réordonner ce nouvel alphabet par ordre de probabilités décroissantes, et d'itérer la procédure : prendre les deux (nouveaux) symboles de probabilités minimales, et leur associer deux nouvelles branches.
On en déduit

Corollaire 2.1 *Le code de Huffman est le code à préfixe optimal, et satisfait en particulier la borne (2.7).*

Exemple 2.2 Prenons l'exemple suivant d'un alphabet de 8 symboles, de probabilités données dans la table 2.3.

symbole	probabilité	code
a	0.01	110000
b	0.02	110001
c	0.05	11001
d	0.09	1101
e	0.18	111
f	0.2	00
g	0.2	01
h	0.25	10

TABLE 2.3 – Exemple de code de Huffman.

L'arbre de préfixes correspondant se trouve en FIGURE 2.3. On peut à partir de cet exemple estimer la longueur moyenne des mots :

$$\overline{\ell}(\alpha) = 0.25 \times 2 + 0.2 \times 2 + 0.2 \times 2 + 0.18 \times 3 + 0.09 \times 4 + 0.05 \times 5 + 0.02 \times 6 + 0.01 \times 6 = 2.63 \ ,$$

ce qui représente un gain par rapport à un codeur uniforme qui aurait nécessité 3 bits par symbole. A titre de comparaison, l'entropie de la distribution de probabilités vaut $H \approx 2.5821$, le code a donc une efficacité d'approximativement 98%, ce qui est excellent.

Remarque 2.1 Il arrive souvent que le code optimal ne soit pas unique : pour une distribution de probabilités donnée, l'algorithme de Huffman peut laisser une certaine liberté dans les mots binaires à associer aux symboles (typiquement, lorsque deux probabilités sont égales). Il est alors facile de voir que bien que les codes soient différents, la longueur moyenne des mots $\overline{\ell}$ est la même, ce qui est la seule chose réellement importante.

Remarque 2.2 Il existe des alternatives au codage de Huffman. On peut en particulier mentionner le *codage arithmétique*, qui permet d'associer, dans un certain sens, des "nombre de bits non entiers" aux symboles de A. On montre que ceci permet d'améliorer légèrement les performances d'un codeur.

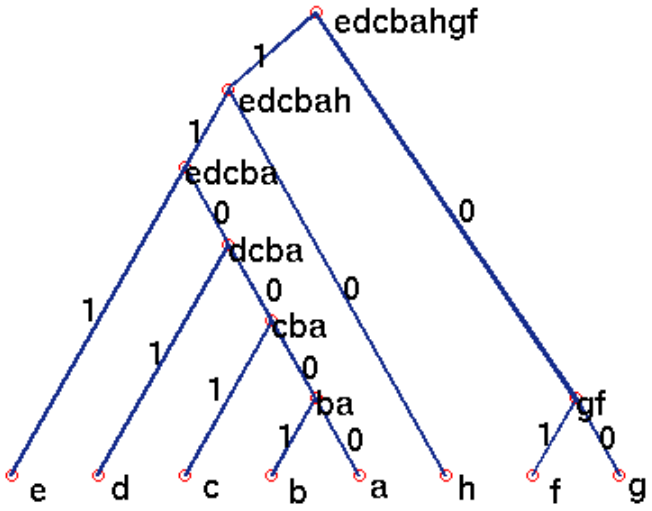

FIGURE 2.3 – Arbre de préfixes pour les symboles de la Table 2.3

2.2 Architecture d'un CoDec par transformation

Nous allons ici étudier une famille de CoDecs appelés **Codeurs par transformation**, dans lesquels les étapes de préfiltrage et échantillonnage sont composées par une projection sur un sous-espace d'approximation, comme vu dans le premier chapitre, les échantillons étant les coefficients de la projection.

2.2.1 Filtrage et échantillonnage

Le cadre général est celui de l'espace $E = L^2([0,1])$ (ou plus généralement d'un espace $E = L^2([0,T])$, le choix de l'origine étant purement conventionnel). Comme on l'a vu, si l'on fait le choix d'un sous-espace $F \subset E$ de dimension N, engendré par des fonctions $f_0, \dots f_{N-1} \in F$, on va représenter tout signal $x \in E$ par les coefficients a_n de la décomposition de son projeté orthogonal

$$\Pi_F(x) = \sum_{n=0}^{N-1} a_n f_n \ ,$$

coefficients calculés comme indiqué dans la Proposition 1.1.
Cette étape est ce que l'on appelle une étape à perte : en général, $\Pi_F(x) \neq x$, et les coefficients $a_0 \dots a_n$ ne permettront donc pas de reconstruire x exactement. Il est possible de montrer que

$$\|x - \Pi_F(x)\|^2 = \|x\|^2 - GA \cdot A \ , \tag{2.8}$$

où on a noté $A = (a_0, a_1, \dots a_{N-1})^T$ le vecteur colonne des coefficients de $\Pi_F(x)$, et où "$\cdot$" représente le produit Hermitien dans $\mathbb{C}^N$.
Cette étape associe donc à un signal $x \in E$ un vecteur de coefficients $A = (a_0, \dots a_{N-1})^T \in \mathbb{C}^N$.

2.2.2 Quantification

Les coefficients issus de la première étape sont des réels (ou des complexes) et doivent être quantifiés. On utilise pour cela un quantificateur $Q : \mathbb{R} \to \mathcal{A}$ associant à chaque réel u une valeur $Q(u)$ appartenant à un alphabet $\mathcal{A}$ de cardinal 2^R, R étant le nombre de bits caractérisant le quantificateur.
La quantification est elle aussi une étape à parte, dans le sens où le quantificateur introduit des erreurs d'approximation qui ne pourront pas être compensées. En sortie du quantificateur, en notant

$$q_n = Q(a_n) \ , \quad n = 0 \dots N-1 \ ,$$

on ne pourra donc reconstruire que l'approximation ci-dessous du signal

$$\tilde{x} = \sum_{n=0}^{N-1} q_n f_n \ , \tag{2.9}$$

et donc l'erreur globale commise par le CoDec sera donnée par $\|x - \tilde{x}\|$, et on peut montrer que

$$\|x - \tilde{x}\|^2 = \|x - \Pi_F(x)\|^2 + \|\Pi_F(x) - \tilde{x}\|^2 \ . \tag{2.10}$$

2.2.3 Codage binaire

Le codage binaire transforme les coefficients quantifiés en suites binaires, c'est à dire des suites de 0 et 1. On distingue deux types de codes binaires :

- Les codes de longueur constante (CLC), qui associent à chaque valeur quantifiée un code de même longueur. Si le quantificateur effectue une quantification sur R bits, le codeur binaire associe à chaque valeur y_m un mot $\alpha(y_m)$ de R bits.
- Les codes de longueur variable (CLV) qui essaient d'adapter la longueur des mots binaires à la probabilité d'apparition de la valeur à coder.

Dans tous les cas, étant donnée une suite de coefficients quantifiés $c_1, c_2, \dots c_K$, le codeur leur associe un mot binaire obtenu par concaténation des mots binaires associés aux c_k :

$$\alpha(c_1, c_2, \dots c_K) = \alpha(c_1)\alpha(c_2)\dots\alpha(c_K) \ .$$

2.3 Exercices

Exercice 2.1 (Estimations d'erreur)

1. Démontrer l'estimation d'erreur (2.8). On utilisera le développement de $\Pi_F(x)$ sur la base $\{f_0 \dots f_{N-1}\}$.
2. Démontrer l'égalité (2.10) et interpréter. On pourra utiliser le fait que $\Pi_F(x) - x$ est orthogonal à F.

Exercice 2.2 (Code de Huffman)

On considère l'alphabet A équipé de la distribution de probabilités donnée dans le tableau ci-dessous

Symbole	a	b	c	d	e	f	g	h	i	j	k	l	m
Probabilité	5/24	3/24	2/24	2/24	2/24	2/24	2/24	1/24	1/24	1/24	1/24	1/24	1/24

1. Appliquer l'algorithme de Huffman, et en déduire un code pour cet alphabet.
2. Calculer l'entropie de Shannon de la distribution de probabilités P, ainsi que la longueur moyenne des mots. Donner l'efficacité du code.

Exercice 2.3 (Code de Shannon-Fano)

Le code de Shannon-Fano, utilisé dans les années 50, est le premier code à avoir exploité la redondance d'une source. Le principe est basé sur une décomposition récursive de l'alphabet en deux parties telles que leur probabilité soit sensiblement égale.
On considère un alphabet $A = \{a\}$, et une distribution de probabilités $P = \{p(a), a \in A\}$. L'algorithme de Shannon-Fano comporte les étapes suivantes :

— Les probabilités d'apparition de chaque symbole sont placées dans un tableau trié par ordre décroissant de probabilités .
— L'alphabet est coupé en deux groupes de symboles A_0 et A_1 dont la somme des probabilités de chaque groupe avoisine 0.5.
— Le groupe A_0 est codé par un "0" et A_1 par un "1".
— Si un groupe A_i n'a qu'un seul élement, c'est une feuille terminale, sinon la procédure reprend récursivement à l'étape 2 sur le groupe A_i.

1. Appliquer l'algorithme de Shannon-Fano ci-dessus à l'alphabet de l'exercice 2.2, et en déduire un code pour cet alphabet.
2. Calculer l'entropie de Shannon de la distribution de probabilités P, ainsi que la longueur moyenne des mots. Donner l'efficacité du code. Comparer les résultats aux résultats obtenus avec le code de Huffman

2.4 Projet

Le but de ce projet est de construire quelques exemples de CoDecs, et de les tester sur des signaux synthétiques puis réels. Dans tous les cas, les deux aspects principaux sont les suivants : partant d'un signal de longueur L donnée

- **Codeur :** calculer une approximation (sous forme de projection sur un sous-espace de dimension N donnée), puis quantifier les coefficients de cette approximation sur un nombre de bits R donné. En sortie du codeur, on dispose donc des coefficients quantifiés.
- **Décodeur :** reconstituer une approximation du signal à partir des coefficients quantifiés.

Deux approximations sont ainsi réalisées, et le but est d'étudier l'influence de N et R sur la précision de ces approximations.

Dans un second temps, on pourra finaliser le CoDec en réalisant un codage de Huffman à la sortie du quantificateur. Alternativement, on pourra utiliser un code de longueur constante, plus simple.

Les étudiants peuvent travailler soit seuls, soit par binômes (conseillé). Chaque étudiant ou binôme doit rendre en fin de projet :

- Un compte rendu, décrivant le travail effectué, les principales étapes de celui-ci (en particulier les programmes développés), et les résultats obtenus. Les résultats pourront être présentés sous forme de graphiques et de tableaux ; chaque graphique, chaque tableau devra être commenté et interprété.

- Une archive (au format `.zip`, ou `.tar.gz`) contenant les programmes développés, ainsi qu'un fichier `README.txt` donnant la liste des programmes et fonctions, avec quelques explications succintes (par exemple le prototype de la fonction).

Remarques :

- Les programmes devront être opérationnels, au sens où un utilisateur devra pouvoir les exécuter en se basant sur les indications fournies dans le compte-rendu.

- Les programmes devront être commentés à l'intérieur du code, suffisamment pour qu'un utilisateur puisse comprendre facilement son fonctionnement.

Le projet donne lieu à une note de projet, qui se base sur le compte rendu et les programmes, et une soutenance orale en janvier 2016.

2.4.1 Segmentation et approximation

Les signaux réels sont généralement des signaux longs, trop longs pour pouvoir être traités dans leur intégralité. Il faut effectuer une opération appelée "segmentation", qui consiste à couper les signaux en segments de longueur constante

Partie 2.1 *Ecrire une fonction* `segmentation.m` *prenant en entrée un signal (a priori long) x et la longueur N des segments, et retournant en sortie une séquence de signaux de longueur N. On pourra organiser ces séquences sous forme matricielle.*
Remarque : *L n'étant pas nécessairement un multiple de N, il peut être nécessaire de rallonger le signal en lui adjoignant autant de valeurs nulles que nécessaire pour que la longueur du signal étendu soit égale au multiple de N le plus proche (par valeurs supérieures). Cette opération porte le nom de zero-padding en anglais.*

Partie 2.2 *En combinant* `segmentation.m` *et* `ConstSplineApprox.m` *(resp.* `AffSplineApprox.m`*), écrire une fonction* `ConstSplineApprox_Long.m` *(resp.* `AffSplineApprox_Long.m`*) prenant en entrée un signal (long), effectuant la segmentation et l'approximation, et retournant en sortie les coefficients de l'approximation.*

2.4.2 Introduction de la quantification

Partie 2.3 *Ecrire une fonction* `simplecodec.m` *prenant en entrée un (long) signal* `x` *et effectuant les opérations suivantes :*

- *Segmentation en segments de longueur fixée*
- *Projection de chaque segment sur un sous-espace d'approximation (constante ou affine par morceaux) de dimension donnée, et calcul des coefficients du développement de celle-ci sur une base du sous-espace*
- *Quantification uniforme des coefficients, avec un nombre* `R` *de bits par coefficient fixé.*
- *Reconstruction du signal décodé à partir des coefficients quantifiés.*

Syntaxe possible : `[y,alpha,alphaq] = simplecodec(x,L,N,R);`
où `x` *est le signal d'entrée,* `L` *la longueur des segments,* `N` *la dimension du sous-espace d'approximation,* `R` *le débit,* `y` *le signal reconstruit,* `alpha` *le vecteur de coefficients de l'approximation, et* `alphaq` *le vecteur de coefficients quantifiés.*
Remarque : *il faudra être attentif au choix des paramètres du quantificateur.*

Partie 2.4 *Utiliser un code de Huffman pour finaliser avec un codage binaire. A défaut, on pourra écrire une fonction effectuant un codage de longueur constante.*
Remarque : *le codage binaire est un codage sans perte, il n'introduit pas d'erreur d'approximation.*

2.4.3 Analyse des performances

Comme mentionné plus haut, les performances du codeur développé s'évaluent sur la base de l'évolution de l'erreur commise par le codeur en fonction des deux paramètres N et R. On pourra, pour un signal donné, le coder pour différentes valeurs de N et R, et tracer le rapport signal à bruit SNR en fonction de ces deux valeurs.

On pourra pour cela

- Soit tracer des courbes en fixant l'un des deux paramètres et en faisant varier l'autre
- Soit représenter le SNR sous forme de fonction de deux variables, en utilisant soit la fonction `imagesc`, soit la fonction `surf` (se renseigner en utilisant `help` ou `doc`).

On comparera les résultats obtenus sur des exemples de signaux réels (ou sons), avec l'approximation constante par morceaux et l'approximation affine par morceaux.

Partie 2.5 (Optionnel) *On pourra également essayer un codeur basé sur l'approximation par polynômes trigonométriques, ou des bases de bosses. Dans ce dernier cas il faudra bien choisir les paramètres de la base, et s'assurer du bon conditionnemment de la matrice de Gram.*

Chapitre

3 Modulation-Démodulation : théorie, exercices et projet

On considère ici un autre aspect d'une chaîne de transmission, à savoir la modulation, qu'on traite sous sa forme numérique. En télécommunications, le signal transportant une information doit passer par un moyen de transmission entre un émetteur et un récepteur. Le signal est rarement adapté à la transmission directe par le canal de communication choisi, hertzien, filaire, ou optique. La modulation peut être définie comme le processus par lequel le signal est transformé de sa forme originale en une forme adaptée au canal de transmission, par exemple en faisant varier les paramètres d'amplitude et d'argument (phase/fréquence) d'une onde sinusoïdale appelée porteuse. Le dispositif qui effectue cette modulation, en général électronique, est un modulateur (voir modem). L'opération inverse permettant d'extraire le signal de la porteuse est la démodulation. Le couple Modulateur-Démodulateur est généralement désigné par l'acronyme MoDem [1].

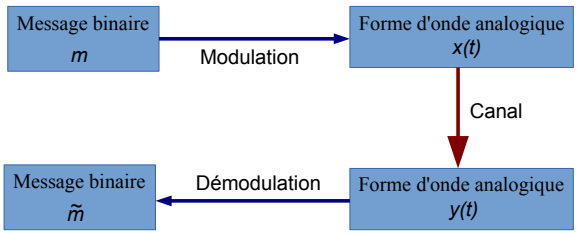

Figure 3.1 – Modulation-Démodulation

La modulation a pour objectif d'adapter le signal à émettre au canal de transmission. Cette opération consiste à modifier un ou plusieurs paramètres d'une onde porteuse $S(t) = A\cos(\omega_0.t + \varphi_0)$ centrée sur la bande de fréquence du canal.

3.1 Eléments d'analyse de Fourier et modulation analogique

Le concept fondamental en modulation est le concept de fréquence, lequel est intrinsèquement associé à la transformation de Fourier. Commençons par quelques rappels sur les séries de Fourier. En quelques mots, étant donnée une fonction $x : [a, b] \to \mathbb{C}$, elle peut sous des hypothèses appropriées être caractérisée par ses coefficients de Fourier

$$c_k(x) = \frac{1}{b-a}\int_a^b x(t)e^{-2i\pi kt/(b-a)}dt \ ,$$

1. Voir par exemple `http://fr.wikipedia.org/wiki/Modulation_du_signal`

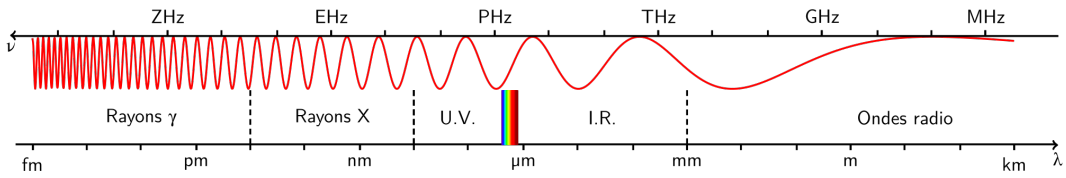

FIGURE 3.2 – Spectre électromagnétique (source : wikipedia)

au sens où elle s'exprime sous forme de combinaison linéaire de fonctions oscillantes $t \to e^{2i\pi kt/(b-a)}$ suivant l'expression

$$x(t) = \sum_{k=-\infty}^{\infty} c_k(x) e^{2i\pi kt/(b-a)} \ .$$

Cette expression est formelle, le sens précis à donner à l'égalité ainsi qu'à la convergence de la série étant à préciser (en fonction d'hypothèses faites sur x). On y voit apparaître une variable discrète

$$\nu_k = \frac{k}{b-a}$$

appelée fréquence, qui possède une signification physique bien établie : elle caractérise la vitesse de variation de la sinusoïde $t \to e^{2i\pi\nu_k t}$. Par exemple, si $x(t)$ représente un signal sonore, les sinusoïde correspondant à de faibles valeurs de la fréquence ν_k sont perçues par l'oreille comme des sons graves, alors que les sinusoïdes "haute fréquence" sont perçues comme des sons aigüs. Dans un tout autre domaine, correspondant aux signaux utilisés en télécommunications, quand $x(t)$ représente une onde électromagnétique, la variable de fréquence est associée au spectre électromagnétique, dont les différentes régions sont souvent associées (à tort) à des phénomènes différents : spectre visible, micro-ondes, domaine hertzien... alors qu'il s'agit en fait du même phénomène. L'ensemble du spectre électromagnétique est schématisé dans la figure 3.2.
En télécommunications, différentes parties de ce spectre sont utilisées pour différents protocoles (radio et télévision hertzienne, radio et télévision numérique, téléphonie mobile, communications nautiques et aéronautiques,...). Dans chaque cas, le signal à transmettre est "embarqué" dans une onde électromagnétique de fréquence proche de la fréquence réservée. Cette opération est appelée *modulation.*

On travaillera (parce que c'est plus simple) avec des signaux analogiques à support infini. On décrit ci-dessous les éléments de base de la théorie de Fourier associée.

3.1.1 Transformation de Fourier et propriétés simples

3.1.1.1 Transformation de Fourier

Définition 3.1 *Etant donnée une fonction x, sa transformée de Fourier est la fonction d'une variable réelle $\nu \to \hat{x}(\nu)$, définie par*

$$\hat{x}(\nu) = \int_{-\infty}^{\infty} x(t) e^{-2i\pi\nu t}\, dt \ , \tag{3.1}$$

pour tout ν tel que l'intégrale soit convergente. On note $\mathcal{F}$ l'opérateur linéaire défini par $\hat{x} = \mathcal{F}x$. La variable ν porte le nom de fréquence.

On voit facilement que si $x \in L^1(\mathbb{R})$, $\hat{x}$ est bornée. Le résultat suivant, donné sans démonstration, précise les choses.

Théorème 3.1 (Riemann-Lebesgue) *Soit $x \in L^1(\mathbb{R})$. Alors l'équation* (3.1) *définit une fonction $\hat{x}$ bornée, uniformément continue : pour tout $\epsilon > 0$, il existe $\delta > 0$ tel que pour tout ν,*

$$|\hat{x}(\nu + \delta) - \hat{x}(\nu)| \leq \epsilon \tag{3.2}$$

De plus,

$$\lim_{\nu \to \pm\infty} \hat{x}(\nu) = 0 \ .$$

La transformation de Fourier possède des propriétés simples, faciles à vérifier. En supposant que $x \in L^1(\mathbb{R})$ pour simplifier (ce qui assure l'existence de $\hat{x}(\nu)$ pour tout ν), on vérifie facilement les propriétés suivantes :

1. *Comportement vis à vis des translations et des modulations.* Considérons maintenant une fonction intégrable $x \in L^1(\mathbb{R})$. On définit la *translatée* de x par la quantité $b \in \mathbb{R}$ comme la fonction $y : t \to y(t) = x(t-b)$. On a alors,

$$\hat{y}(\nu) = e^{2i\pi\nu b}\hat{x}(\nu) \ . \tag{3.3}$$

   On dit que $\hat{y}$ est une version *modulée* de $\hat{x}$. Similairement, si $y \in L^1(\mathbb{R})$, définie par $z(t) = e^{2i\pi\eta t}x(t)$ (où $\eta \in \mathbb{R}^*$) est une version modulée de la fonction intégrable x, alors on a

$$\hat{y}(\nu) = \hat{x}(\nu - \eta) \ , \tag{3.4}$$

   de sorte que la transformée de Fourier de y est une version translatée de $\hat{x}$.

2. *Comportement vis à vis des dilatations.* Soit $x \in L^1(\mathbb{R})$, et soit $\hat{x}$ sa transformée de Fourier. Si a est une constante réelle, on considère une fonction x_a, dilatée de x du facteur a, définie par $x_a(t) = x(t/a)$. Alors, on a, par un changement de variable $u = t/a$,

$$\hat{x}_a(\nu) = a\hat{x}(a\nu) \ . \tag{3.5}$$

   Ainsi, la transformée de Fourier de la copie dilatée d'une fonction n'est autre qu'une copie dilatée (d'un rapport inverse) de la transformée de Fourier de la fonction originale.

3.1.1.2 Inversion dans $L^1(\mathbb{R})$

Le problème de l'inversion de la transformation de Fourier est un délicat problème. La transformation de Fourier inverse est définie par :

$$\check{x}(t) = \int_{-\infty}^{\infty} x(\nu)e^{2i\pi\nu t}\, d\nu \ .$$

et on note $\overline{\mathcal{F}}$ l'opérateur linéaire défini par $\check{x} = \overline{\mathcal{F}}x$. Le problème qui se pose est de donner un sens à $\check{x}$, mais aussi de définir dans quelles conditions et en quel sens $\overline{\mathcal{F}}$ est effectivement la transformation inverse de la transformation de Fourier $\mathcal{F}$.
Un premier résultat d'inversion est donné par le théorème suivant, donné lui aussi sans démonstration (il est possible de montrer des résultats plus généraux, qui évitent l'hypothèse de continuité)

Théorème 3.2 (Dirichlet) *Soit $x \in L^1(\mathbb{R})$, telle que $\hat{x} \in L^1(\mathbb{R})$. Si x est continue en $t = t_0$, alors*

$$x(t_0) = \int_{-\infty}^{\infty} \hat{x}(\nu)e^{2i\pi\nu t_0}\, d\nu \ . \tag{3.6}$$

3.1.1.3 La théorie $L^2(\mathbb{R})$

L'espace $L^1(\mathbb{R})$ ne donne pas un cadre suffisant pour la théorie de Fourier, et le cadre le plus naturel est $L^2(\mathbb{R})$ [2]. Cependant, la définition de la transformation de Fourier dans $L^2(\mathbb{R})$ n'est pas facile (pour une fonction $x \in L^2(\mathbb{R})$, $\hat{x}(\nu)$ n'est pas nécessairement défini pour tout ν). Sans entrer dans les détails, il est possible de montrer que la transformation de Fourier, qui est bien définie dans $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ peut s'étendre à $L^2(\mathbb{R})$ et y avoir des propriétés intéressantes et utiles.

2. L'espace $L^2(\mathbb{R})$ est également le plus agréable car c'est un espace de Hilbert.

Théorème 3.3 (Transformation de Fourier sur $L^2(\mathbb{R})$) *La transformation de Fourier sur $L^1(\mathbb{R})\cap L^2(\mathbb{R})$ se prolonge en une transformation unitaire de $L^2(\mathbb{R})$ sur $L^2(\mathbb{R})$. De plus, on a la formule de Plancherel : $\forall x, y \in L^2(\mathbb{R})$,*

$$\int_{-\infty}^{\infty} x(t)\overline{y(t)}dt = \int_{-\infty}^{\infty} \hat{x}(\nu)\overline{\hat{y}(\nu)}d\nu \ . \tag{3.7}$$

Il est complété par la proposition suivante, qui prouve que la transformée de Fourier d'une fonction de $L^2(\mathbb{R})$ s'obtient (aux points oú elle est bien définie) via le calcul usuel.

Proposition 3.1 *Si $x \in L^2(\mathbb{R})$, alors en tout point où $\hat{x}(\nu)$ est bien défini $\hat{x}(\nu)$ peut s'obtenir comme*

$$\hat{x}(\nu) = \lim_{T\to\infty} \int_{-T}^{T} x(t)e^{-2i\pi\nu t}dt \ . \tag{3.8}$$

3.1.2 Filtres linéaires

Le filtrage est l'une des opérations fondamentales du signal. Un filtre (linéaire) est par définition un opérateur linéaire invariant par translation.

Définition 3.2 *Un filtre linéaire est un opérateur $T : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ qui commute avec les translations : pour tout $x \in L^2(\mathbb{R})$ et $\tau \in \mathbb{R}$, en définissant $y \in L^2(\mathbb{R})$ par $y(t) = x(t-\tau)$, on a*

$$Ty(t) = Tx(t-\tau) \ .$$

Les exemples les plus simples de filtres linéaires sont construits via un produit de convolution.

Définition 3.3 *Soient x, y deux fonctions d'une variable réelle. Le produit de convolution est la fonction notée $z = x * y$, donnée par*

$$z(t) = \int_{-\infty}^{\infty} x(s)y(t-s)\,ds$$

lorsque l'intégrale est bien définie.

Soit $h \in L^1(\mathbb{R})$, et soit K_h l'opérateur défini par

$$K_h x(t) = \int_{-\infty}^{\infty} h(s)x(t-s)\,ds \ . \tag{3.9}$$

Lorsqu'il est bien défini, le produit de convolution possède des propriétés simples, faciles à vérifier :
- *Commutativité : $x * y = y * x$.*
- *Associativité : $(x * y) * z = x * (y * z)$.*
- *Distributivité par rapport à l'addition : $x * (y + z) = x * y + x * z$.*

Plus généralement, il existe des situations dans lesquelles l'existence du produit de convolution (dans un certain sens) est garantie.

Proposition 3.2 (Inégalités d'Young) *Soient $x \in L^p(\mathbb{R})$ et $y \in L^q(\mathbb{R})$. Alors,*

$$x * y \in L^r(\mathbb{R}) \ , \quad où \quad 1 + \frac{1}{r} = \frac{1}{p} + \frac{1}{q} \ , \quad et \quad ||x * y||_r \leq ||x||_p \, ||y||_q \ . \tag{3.10}$$

Si de plus p et q sont conjugués, c'est à dire si $1/p + 1/q = 1$, alors $x * y$ est bornée et continue.

En particulier, si $p = 1$ alors $r = q$.

Un filtre de convolution K_h est un opérateur linéaire défini par un produit de convolution : étant donnée une fonction $h \in L^1(\mathbb{R})$ (appelée *réponse impulsionnelle du filtre*, le filtre correspondant est défini par

$$K_h x = h * x \ . \tag{3.11}$$

Si $h \in L^1(\mathbb{R})$, il est immédiat d'après l'inégalité d'Young correspondante que K_h est un opérateur borné $L^2(\mathbb{R}) \to L^2(\mathbb{R})$, et que c'est un filtre linéaire. Un calcul explicite montre aussi

$$\widehat{K_h x}(\nu) = \hat{h}(\nu)\hat{x}(\nu) \ ,$$

de sorte qu'un filtrage de convolution se ramène à un produit simple dans l'espace de Fourier.

Remarque 3.1 *On considère aussi souvent des filtres linéaires de la forme K_h, où $h \in L^1_{loc}(\mathbb{R})$. Dans ce cas, la fonction $\hat{h}$ n'est plus nécessairement bornée, mais l'équation ci-dessus reste valide en tout point où ses deux membres sont définis.*

Plus généralement, on a le résultat suivant, que nous donnons sans démonstration :

Théorème 3.4 *Soit T un filtre linéaire. Alors il existe $m \in L^\infty(\mathbb{R})$ telle que pour tout $x \in L^2(\mathbb{R})$,*

$$\widehat{Tx}(\nu) = m(\nu)\hat{x}(\nu) \ . \tag{3.12}$$

m est appelée fonction de transfert *du filtre. On a donc, au sens de la convergence dans $L^2(\mathbb{R})$*

$$Tx(t) = \int_{-\infty}^{\infty} m(\nu)\hat{x}(\nu)e^{2i\pi\nu t}d\nu \ .$$

Remarque 3.2 En introduisant le *spectre d'énergie* $\mathcal{S}_x = |\hat{x}|^2$, on obtient

$$\mathcal{S}_{Tx}(\nu) = |m(\nu)|^2\mathcal{S}_x(\nu) \ . \tag{3.13}$$

On dit que le filtrage modifie le contenu fréquentiel du signal. Par exemple, si $m(\nu) \to 0$ quand $\nu \to \infty$, la décroissance à l'infini de $\widehat{Tx}$ est plus rapide que celle de $\hat{x}$, et l'on s'attend donc à ce que Tx soit une fonction plus "régulière" que x. Nous verrons des conséquences importantes de cette remarque plus loin.

Définition 3.4
1. *Le filtre K_h est dit* stable *si il est borné de $L^\infty(\mathbb{R})$ sur $L^\infty(\mathbb{R})$: il existe une constante $C > 0$ telle que pour tout $x \in L^\infty(\mathbb{R})$, $\|K_h x\|_\infty \leq C\|x\|_\infty$.*
2. *Le filtre K_h est dit* réalisable *(ou* causal*) si $h(t) = 0$ pour tout $t \leq 0$.*
3. *Le filtre K_h est dit* dynamique *si il est stable et réalisable.*

La fonction h est appelée réponse impulsionnelle *du filtre. Sa transformée de Fourier (quand elle est définie) coïncide avec la fonction de transfert m.*

La contrainte de causalité est importante quand il s'agit de donner une implémentation "physique" du filtre (par un circuit électrique par exemple). En effet, si $h(t) \neq 0$ sur un ensemble de mesure non-nulle de $\mathbb{R}^-$, le calcul de $K_h x(t)$ pour un temps t donné

$$K_h x(t) = \int_{-\infty}^{\infty} h(s)x(t-s)\,ds$$

utilise des valeurs $x(x-s)$ antérieures à t, ce qui est irréalisable (ou tout du moins incompatible avec ce que nous savons de la physique).

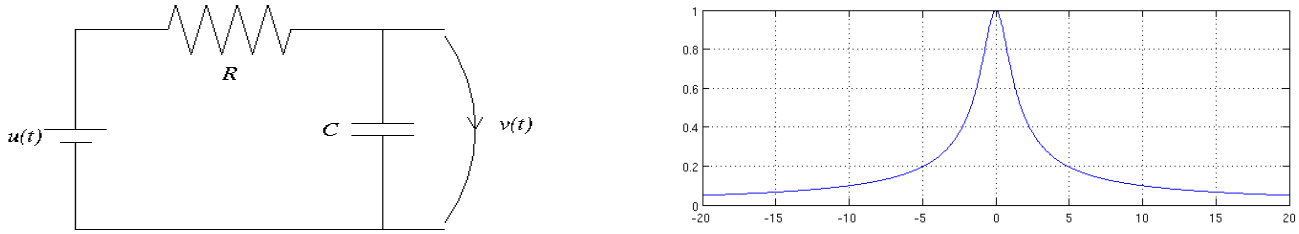

FIGURE 3.3 – Le filtre RC : circuit (gauche) et fonction de transfert (droite)

3.1.2.1 Filtres idéaux

1. Étant donnée une fréquence ν_0 (appelée *fréquence de coupure*), le **filtre passe-bas idéal** est défini par
$$m(\nu) = 1_{[-\nu_0,\nu_0]}(\nu) \ ,$$
où 1_I est la fonction indicatrice d'un intervalle I. Il est immédiat de montrer que la réponse impulsionnelle $t \to h(t)$ du filtre est de la forme
$$h(t) = 2\nu_0 \, \frac{\sin(2\pi\nu_0 t)}{2\pi\nu_0 t} \ .$$
Donc, $h \in L^1_{loc}(\mathbb{R})$, mais $h \notin L^1(\mathbb{R})$, et le filtre n'est pas stable. Il n'est pas réalisable non plus de façon évidente.
2. Étant donnée une *fréquence de coupure* ν_0, le **filtre passe-haut idéal** est défini par
$$m(\nu) = 1 - 1_{[-\nu_0,\nu_0]}(\nu) \ .$$
On vérifie immédiatement qu'il n'est ni stable ni réalisable.
3. On définit également des **filtres passe-bande** idéaux (définis par $m(\nu) = 1_{[a,b]}(\nu)$) et des **filtres coupe-bande** (définis par $m(\nu) = 1 - 1_{[a,b]}(\nu)$), qui ne sont eux aussi ni stables ni réalisables.

3.1.2.2 Circuits analogiques

Une façon simple de construire des filtres réalisables est d'utiliser des circuits électriques. Par exemple, un circuit du type de celui de la Fig. 3.3.
En notant $v(t) = Q(t)/C$ la tension aux bornes du condensateur, la loi d'Ohm s'écrit $Ri(t)+v(t) = u(t)$, ce qui entraîne, puisque $i(t) = Q'(t) = Cv'(t)$, que la tension $v(t)$ satisfait à l'équation différentielle ordinaire
$$RC\, v'(t) + v(t) = u(t) \ , \qquad t \in \mathbb{R}^+ \ .$$
Il est facile de montrer que la solution est de la forme
$$v(t) = \frac{1}{RC} \int_{-\infty}^{t} e^{-(t-s)/RC} u(s) ds = \int_{-\infty}^{\infty} h(t-s) u(s) ds \ ,$$
où nous avons posé
$$h(t) = \Theta(t) \, \frac{1}{RC} \, e^{-t/RC}$$
$\Theta(t)$ étant la fonction d'Heaviside, qui vaut 1 pour $t \geq 0$ et 0 pour $t < 0$. Nous sommes bien en présence d'un filtre réalisable et stable. La fonction de transfert $m = \hat{h}$ de ce filtre est facilement obtenue par un calcul explicite :
$$m(\nu) = \frac{1}{RC} \int_0^{\infty} e^{-t/RC} e^{-2i\pi\nu t} dt = \frac{1}{1 + 2i\pi\nu RC} \ .$$

m est à valeurs complexes, le comportement du filtre peut être mieux compris en étudiant le module $|m|$ de la fonction de transfert, donné par

$$|m(\nu)| = \frac{1}{\sqrt{1+(2\pi\nu RC)^2}} \ ,$$

où on peut voir (voir Figure 3.3, droite) que $|m(\nu)|$ est proche de 1 pour $\nu \approx 0$ et décroît de façon monotone comme $1/|\nu|$ quand $\nu \to \pm\infty$. Le filtre RC se comporte donc comme un filtre passe-bas (préserve les basses fréquences et atténue les hautes fréquences), très éloigné cependant du filtre idéal (et impossible !).

Des circuits électroniques plus complexes conduisent quant à eux à des équations différentielles plus complexes aussi, et produisent d'autres types de filtres.

3.1.3 Modulation analogique

Comment transmettre un signal analogique ? il doit être en quelque sorte "embarqué" dans un signal de référence, la *porteuse*, que l'on est capable de transmettre. La porteuse peut être une onde électromagnétique dans le cas de la téléphonie mobile, ou autre. La transmission est possible/permise à l'intérieur d'un *canal de transmission*, qui correspond à une bande de fréquence donnée.

3.1.3.1 Modulation d'amplitude

Dans la modulation d'amplitude (aussi appelée modulation AM, voir figure 3.4), la porteuse $t \to x_0(t)$ est modulée multiplicativement par le signal à transmettre (appelé signal modulant ou tout simplement message). On distingue deux techniques : la modulation sans porteuse (aussi appelée modulation double bande sans porteuse, DBSP)

$$x \to y_{sp} = x.x_0 \ : \qquad y_{sp}(t) = x(t)x_0(t) \ . \tag{3.14}$$

et la modulation avec porteuse (aussi appelée modulation double bande avec porteuse, DBAP)

$$x \to y_{ap} = (1+kx).x_0 \ : \qquad y_{ap}(t) = (1+kx(t))x_0(t) \ , \tag{3.15}$$

où k est un réel positif, appelé indice de modulation choisi de sorte que $1+kx$ soit à valeurs positives (dans le cas contraire, on parle de surmodulation)
La porteuse est généralement un signal sinusoïdal de fréquence fixée et connue ν_0 :

$$x_0(t) = \cos(2\pi\nu_0 t) \ .$$

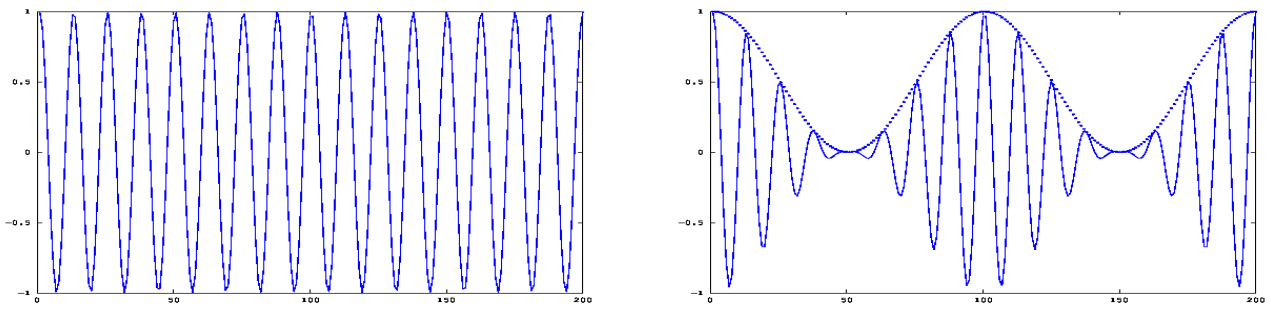

FIGURE 3.4 – Porteuse (gauche) et signaux modulant (droite, pointillés) et modulé (droite, plein)

Pour pouvoir récupérer l'information utile, il faut démoduler le signal, c'est à dire extraire x à partir du signal modulé y. Plusieurs techniques sont utilisées dans la pratique, dont la plupart sont liées à l'étude de la transformée de Fourier du signal modulé.

Signal analytique, transformation de Hilbert, détection d'enveloppe

Si le signal modulant est à bande limitée, c'est à dire tel que le support de sa transformée de Fourier est borné, $\mathsf{Supp}(\hat{x}) \subset [-\eta, \eta]$ et si la fréquence ν_0 de la porteuse est supérieure à η, x peut être retrouvé à partir de y de la façon suivante. On introduit la transformation de Hilbert

Définition 3.5 (Transformation de Hilbert) *La transformation de Hilbert est l'opérateur linéaire $H : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ défini par*

$$\widehat{Hx}(\nu) = -i\mathsf{sgn}(\nu)\hat{x}(\nu) \ , \qquad x \in L^2(\mathbb{R})$$

où $\mathsf{sgn}(\nu)$ est le signe de $\nu \in \mathbb{R}$. Le signal analytique associé à x est le signal $Z_x \in L^2(\mathbb{R})$ défini par

$$Z_x = x + iHx \ .$$

La reconstruction repose sur le résultat suivant.

Lemme 3.1 *Soit $x \in L^2(\mathbb{R})$, tel que $\mathsf{Supp}(\hat{x}) \subset [-\eta, \eta]$, où η est un réel positif. Soit ν_0 un réel positif, et soit $y_{sp} \in L^2(\mathbb{R})$ un signal modulé de la forme $y_{sp}(t) = x(t)\cos(2\pi\nu_0 t)$ (DBSP). Si $\nu_0 > \eta$, la transformée de Hilbert Hy_{sp} du signal modulé $y_{sp} = x.x_0$ est donnée par*

$$Hy_{sp}(t) = x(t)\sin(2\pi\nu_0 t) \ .$$

Preuve : Un calcul direct montre que

$$\widehat{y}_{sp}(\nu) = \frac{1}{2}\left[\hat{x}(\nu - \nu_0) + \hat{x}(\nu + \nu_0)\right] \ .$$

Sous les conditions données, c'est à dire si $\mathsf{Supp}(\hat{x}) \subset [-\eta, \eta]$, les supports respectifs des fonctions $\nu \to \hat{x}(\nu - \nu_0)$ et $\nu \to \hat{x}(\nu + \nu_0)$ sont les intervalles $[-\eta + \nu_0, \eta + \nu_0]$ et $[-\eta - \nu_0, \eta - \nu_0]$. Si $\nu_0 > \eta$, ces intervalles sont disjoints, et inclus respectivement dans les demi axes positif et négatif. Par conséquent, la transformation de Hilbert donne

$$\widehat{Hy}_{sp}(\nu) = \frac{-i}{2}\left[\hat{x}(\nu - \nu_0) - \hat{x}(\nu + \nu_0)\right]$$

ce qui prouve le résultat. ♠

Ainsi, si le signal modulant x est à valeurs positives, il peut s'écrire sous la forme du module du signal analytique associé au signal modulé $Z_{y_{sp}} = y_{sp} + iHy_{sp}$, c'est à dire

$$x = |Z_{y_{sp}}| = |y_{sp} + iHy_{sp}| \ .$$

Si x n'est pas à valeurs positives, cette approche ne fonctionne plus. Par contre, elle peut être adaptée pour traiter la modulation DBAP, en définissant

$$y_{ap} = (1 + kx)x_0 \ ,$$

où k est un réel positif (l'indice de modulation) choisi de sorte que $1 + kx$ soit à valeurs positives. Alors, le même calcul montre que

$$x = \frac{1}{k}\left[|Z_{y_{sp}}| - 1\right] \ .$$

Remarque 3.3 L'ensemble des fonctions $x \in L^2(\mathbb{R})$ dont la transformée de Fourier est à support borné dans un intervalle $[-\eta, \eta]$ est un sous-espace de Hilbert, appelé *espace de Paley-Wiener* de fréquence de coupure η, noté

$$PW_\eta = \left\{x \in L^2(\mathbb{R}) : \mathsf{Supp}(\hat{x}) \subset [-\eta, \eta]\right\} \ .$$

Il est possible de montrer que les fonctions de PW_η sont continues (ou plus exactement, admettent un représentant continu, puisque $L^2(\mathbb{R})$ est un espace de classes d'équivalence de fonctions).

Remarque 3.4 Une difficulté majeure de cette technique est liée à la structure même de la transformation de Hilbert, qui est une opération non-causale : elle peut s'écrire dans le domaine temporel sous la forme d'un filtrage de convolution

$$Hx = k * x \ ,$$

où k est une distribution, dont le support est l'axe réel tout entier. Par conséquent, le filtre n'est pas causal car le calcul de $Hx(t)$ demande la connaissance de toutes les valeurs postérieures $x(s)$, $s \geq t$ de x, ce qui ne peut pas être réalisé par quelque appareil physique que ce soit.

Démodulation par détection synchrone

Comme mentionné plus haut, la démodulation par détection d'enveloppe ne fonctionne pas lorsque la modulation est de type DBSP. Dans ce cas on se troune vers une autre technique. Un calcul simple montre qu'en modulant de nouveau le signal modulé y_{sp}, sous la forme

$$z(t) = y_{sp}(t)\cos(2\pi\nu_0 t) \ ,$$

on obtient dans le domaine fréquentiel

$$\hat{z}(\nu) = \frac{1}{4}\left[\hat{x}(\nu - 2\nu_0) + 2\hat{x}(\nu) + \hat{x}(\nu + 2\nu_0)\right] \ .$$

Par conséquent, si $\hat{z}$ est à support borné, tel que $\mathsf{Supp}(\hat{z}) \subset [-\eta, \eta]$, et si $\eta < \nu_0$, alors $\hat{x}$ peut être retrouvé en multipliant $\hat{z}$ par n'importe quelle fonction m égale à 2 dans l'intervalle $[-\eta, \eta]$ et à zéro à l'extérieur de l'intervalle $[-\nu_0, \nu_0]$, de sorte que x s'obtient par le filtre passe-bas correspondant.
Ainsi, un démodulateur est obtenu en appliquant une nouvelle modulation DBSP, suivie d'un filtrage passe-bas.

Rectification

La rectification est l'opération qui associe à un signal (c'est à dire une fonction) sa valeur absolue, prise point par point :

$$Rx(t) = |x(t)| \ .$$

La rectification a pour effet d'accélérer les oscillations. Sous certaines conditions, le signal modulant x peut être recouvré par filtrage passe-bas, c'est à dire en supprimant les variations les plus rapides. Un exemple se trouve en Figure 3.5 ci-dessous.

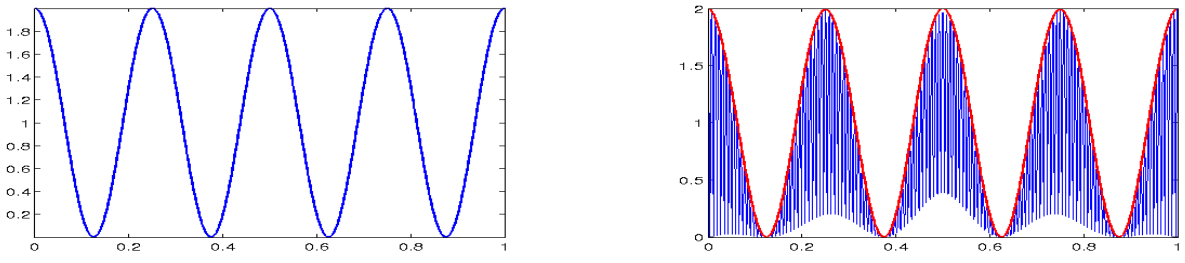

FIGURE 3.5 – Modulation d'amplitude et rectification : signal modulant (gauche), signal modulé rectifié (droite, bleu), et signal modulant estimé par filtrage passe-bas (droite, rouge).

Là encore, cette technique suppose que le signal modulant soit à valeurs positives, et est donc adapté à la démodulation de signaux DBAP.

3.1.3.2 Modulation de phase, modulation de fréquence

Dans le cas de la modulation de phase, le signal modulant est embarqué dans la phase de la porteuse :

$$y(t) = \cos(2\pi\nu_0 t + mx(t)) \ . \tag{3.16}$$

Pour la modulation de fréquence (figure 3.6), le signal modulant est embarqué dans la fréquence de la porteuse :

$$y(t) = \cos\left(2\pi\left(\nu_0 t + m\int_0^t x(\tau)d\tau\right)\right) \ . \tag{3.17}$$

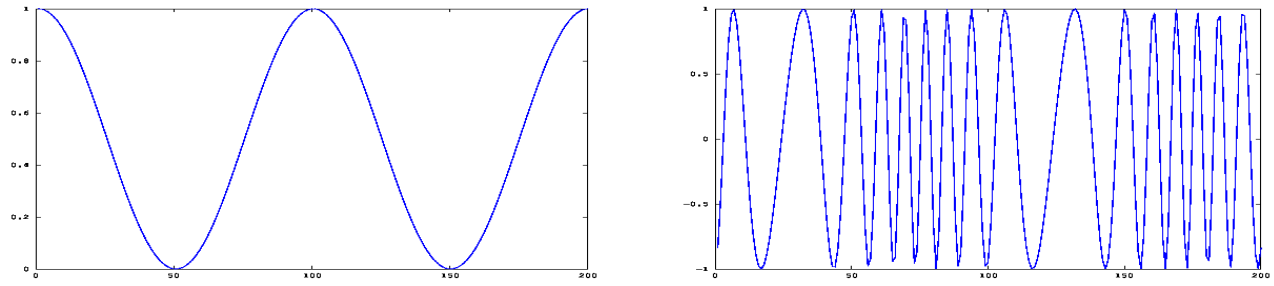

FIGURE 3.6 – Modulation de fréquence : signal modulant (gauche) et signal modulé en fréquence (droite)

Le principe de la démodulation est de pouvoir extraire la *fréquence instantanée* $2\pi\nu_0 + mx(t)$ et d'en soustraire la porteuse. Il existe une multitude de techniques (plus ou moins solides mathématiquement) permettant d'effectuer une telle démodulation. Sans entrer dans les détails, on peut en lister un certain nombre, dont certaines sont résumées très brièvement ci dessous :

— Utilisation des passages par zéro du signal modulé : il existe des dispositifs permettant de détecter les passages par zéro du signal modulé, c'est à dire les valeurs t_k telles que $y(t_k) = 0$. Supposons que les instants de deux passages par zéro consécutifs t_k et t_{k+1} aient ainsi été détectés. On sait alors que

$$\nu_0 t_{k+1} + m\int_0^{t_{k+1}} x(\tau)d\tau - \nu_0 t_k + m\int_0^{t_k} x(\tau)d\tau = \frac{1}{2} \ ,$$

soit

$$\nu_0(t_{k+1} - t_k) + m\int_{t_k}^{t_{k+1}} x(\tau)d\tau = \frac{1}{2} \ ,$$

ce qui fournit une estimation de la valeur moyenne du message x dans l'intervalle $[t_k, t_{k+1}]$:

$$\overline{x}_k := \frac{1}{t_{k+1} - t_k}\int_{t_k}^{t_{k+1}} x(\tau)d\tau = \frac{1}{2(t_{k+1} - t_k)} - \nu_0 \ .$$

De là, des techniques d'interpolation peuvent être utilisées pour reconstituer une estimée du message x.

— Utilisation de la transformation de Hilbert : sous certaines hypothèses, il est possible de démontrer que le signal analytique associé au signal modulé en fréquence est de la forme

$$Z_y(t) = e^{2i\pi\left(\nu_0 t + m\int_0^t x(\tau)d\tau\right)}$$

Ainsi, tout dispositif analogique permettant d'évaluer l'argument de cette fonction et d'en calculer la dérivée temporelle donnera une estimation du message $x(t)$.

3.1.3.3 Canal, bruit,...

Le signal modulé est transmis (via un courant électrique, une onde électromagnétique,...). Il est illusoire de penser que cette transmission peut s'effectuer de façon parfaite. Dans les faits, le signal reçu par un récepteur peut être très différent du signal modulé émis y, de sorte que le signal initial x peut être complexe à reconstituer. On modélise souvent le canal de transmission sous la forme d'un opérateur linéaire T, qui se conjugue à l'ajout d'une perturbation additive, appelée bruit ε, sous la forme

$$t \longmapsto Ty + \varepsilon \ .$$

Le problème est alors à la fois de corriger ces perturbations, et de construire un modulateur/démodulateur aussi robuste que possible.

3.2 Modulation numérique

3.2.1 Principe de la modulation numérique

Le principe de la modulation numérique est d'embarquer une suite binaire (une suite de 0 et de 1) dans un signal analogique, correspondant à une bande de fréquence donnée. On va se limiter ici à des classes de modulateurs très simples.
Plus précisément, le modulateur fait intervenir deux opérations fondamentales suivantes : la traduction des bits en symboles, et l'utilisation des symboles pour constituer un signal analogique.
Soit R un entier positif, soit $M = 2^R$. Dans ce qui suit, on appellera message un mot binaire de R bits. Il y a donc M messages différents.

— La première phase associe à tout message $m = b_0 b_1 \dots b_{R-1}$ un **symbole** $s \in \mathcal{C}$, où $\mathcal{C} \subset \mathbb{R}$ ou $\mathbb{C}$ est un ensemble fini appelé **constellation**

$$\mathcal{C} = \left\{ S^{(0)}, S^{(1)}, \dots S^{(M-1)} \right\} \ .$$

Il y a donc M symboles possibles.

— Soit $N \in \mathbb{N}$, soit T un réel positif, soit $\varphi \in L^2([0, NT])$ une fonction telle que $\mathsf{Supp}(\varphi) \subset [0, T]$, et que $\|\varphi\| = 1$. Étant donnés N symboles $s_0, \dots s_{N-1}$ (correspondant donc à N messages, donc NR bits) on associe la signal modulé $x : t \in [0, NT] \to x(t)$ défini par

$$x(t) = \sum_{n=0}^{N-1} s_n \varphi_n(t) \ , \quad \text{où} \quad \varphi_n(t) = \varphi\left(t - \frac{n}{T}\right) \ . \tag{3.18}$$

φ est appelé **porteuse**, T est la durée de la porteuse (mesurée en seconde), et $1/T$ est le débit de symboles (le signal modulé transmet $1/T$ symboles par seconde), qui se mesure en `Bauds` [3]. R/T est le débit de bits (R/T bits par seconde)

Définition 3.6 *Soient $T \in \mathbb{R}^+$ et $\varphi \in L^2(\mathbb{R})$, telle que $\mathsf{Supp}(\varphi) \subset [0, T]$ et $\|\varphi\| = 1$. Soit $\mathcal{C}$ une constellation de $M = 2^R$ éléments. Le modulateur associé à $T, \varphi, \mathcal{C}$ est la fonction $\mathcal{M} = \mathcal{M}_{T,\varphi,\mathcal{C}}$ associant à toute suite binaire $b_0 \dots b_{K-1}$ la fonction x*

$$\mathcal{M} : b_0 \dots b_{K-1} \in \{0,1\}^K \to x = \sum_{n=0}^{N-1} s_n \varphi_n \ ,$$

où $N = K/R$ est le nombre de symboles transmis, où la suite $\{s_0, \dots s_{N-1}\}$ est la suite de symboles associés aux bits $\{b_0, \dots b_{K-1}\}$ et où on a posé $\varphi_n(t) = \varphi(t - nT)$.

3. En hommage à Émile Baudot, inventeur du code de Baudot pour la télégraphie

Des contraintes pratiques et physiques s'imposent à ces dispositifs.

— Contrainte de faisabilité : ils doivent être réalisables pratiquement (en *hardware*).
— Contrainte énergétique : toutes ces opérations sont coûteuses en énergie, il importe de minimiser le coût énergétique du modulateur.
— Contrainte de détectabilité : le message, après passage par le canal de transmission, doit être démodulable : il faut pouvoir reconstituer le message.

Remarque 3.5 1. On vérifie facilement que $\langle \varphi_n, \varphi_m \rangle = \delta_{mn}$, puisque les supports de φ_n et φ_m sont disjoints si $n \neq m$, et $\|\varphi_n\| = 1$ puisque $\mathsf{Supp}(\varphi_n) \subset [nT, (n+1)T]$ et

$$\|\varphi_n\|^2 = \int_{nT}^{(n+1)T} \left|\varphi\left(t - \frac{n}{T}\right)\right|^2 dt = \int_0^T |\varphi(t)|^2 dt = 1 \ .$$

2. Etant donnés N symboles à transmettre, on leur associe donc la fonction x définie en (3.18). On voit facilement que $\mathsf{Supp}(x) \subset [0, NT]$ et que $x \in L^2([0, NT])$. Par ailleurs, les symboles peuvent être obtenus à partir de x via

$$s_n = \langle x, \varphi_n \rangle \ , \quad n = 0, \dots N-1 \ . \tag{3.19}$$

Ainsi la correspondance entre mot binaire et signal modulé est bijective.

3. L'énergie consommée par le modulateur est identifiée à la norme du signal modulé, et la puissance consommée est l'énergie par unité de temps. On écrit donc

$$E = \|x\|^2 \ = \sum_{n=0}^{N-1} |s_n|^2 \ , \quad P = \frac{E}{NT} \ . \tag{3.20}$$

Exemple 3.1 (Modulation en bande de base, ou ASK) Supposons $A = 1$, soit $M = 2$. On définit les symboles par $S^{(0)} = -A$ et $S^{(1)} = A$, où A est un réel positif, et

$$\varphi = \frac{1}{\sqrt{T}} 1_{[0,T]}$$

est multiple de l'indicatrice de l'intervalle $[0, T]$.
Par exemple, le message 10010 sera modulé sous la forme donnée en Figure 3.7. L'énergie consommée vaut $E = 5A^2$ et la puissance consommée vaut $P = A^2/T$.

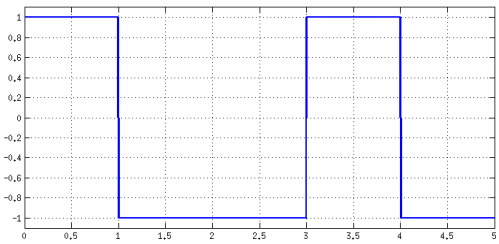

FIGURE 3.7 – Modulation en bande de base (ASK) du message "10010" (on a pris $A = 1$)

La démodulation, c'est à dire l'identification des symboles, pourra alors se faire en comparant le signal à une valeur seuil.

Exemple 3.2 (Modulation de phase binaire, ou BPSK) Supposons encore $R = 1$, soit $M = 2$. On définit encore les symboles par $S^{(0)} = -A$ et $S^{(1)} = A$, où A est un réel positif, et

$$\varphi = \sqrt{\frac{2}{T}} \cos(2\pi\nu_0 t/T) 1_{[0,T]} \ ,$$

où ν_0 est un entier. Par exemple, le message 10010 sera modulé sous la forme donnée en Figure 3.8.

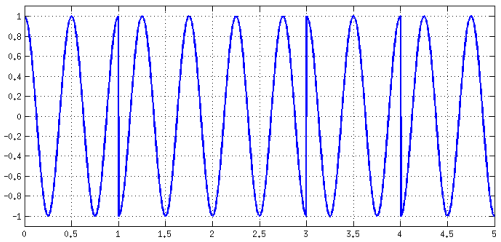

FIGURE 3.8 – Modulation de phase (bpsk) du message “10010” (on a pris $A = 1$).

3.2.2 Modélisation

Pour analyser les performances d'un MoDem, on développe le modèle suivant. On suppose que les messages (et donc les symboles) sont distribués aléatoirement avec une loi connue : les symboles sont des réalisations d'une variable aléatoire discrète S prenant ses valeurs dans la constellation $\mathcal{C}$, et on note

$$p_k = \mathbb{P}\Big\{S = S^{(k)}\Big\} \ . \tag{3.21}$$

Le signal modulé est donc un signal aléatoire

$$X(t) = \sum_{n=0}^{N-1} s_n \varphi_n \ , \tag{3.22}$$

où les s_n sont des réalisations indépendantes de la variable aléatoire S.

Définition 3.7 *Soit $\mathcal{C} = \{S^{(0)}, \dots S^{(M-1)}\}$ une constellation, soit $P = \{p_0, \dots p_{M-1}\}$ la distribution de probabilités associée. On considère le modulateur $\mathcal{M} = \mathcal{M}_{T,\varphi,\mathcal{C}}$ associé à la forme d'onde φ de durée T et la constellation $\mathcal{C}$. L'énergie moyenne E_m et la puissance moyenne P_m consommées par le modulateur $\mathcal{M}$ sont données par*

$$E_m = \mathbb{E}\left\{\|X\|^2\right\} \ , \qquad P_m = \frac{E}{NT} \ .$$

On montre facilement que, compte tenu de l'orthonormalité des fonctions φ_n, l'énergie moyenne s'écrit

$$E_m = \mathbb{E}\left\{\|x\|^2\right\} = N\mathbb{E}\left\{|S|^2\right\} = N\sum_{m=1}^{M} p_m |S^{(m)}|^2 \ . \tag{3.23}$$

et la puissance moyenne s'écrit quant à elle

$$P_m = \frac{E}{NT} = \frac{\mathbb{E}\left\{|S|^2\right\}}{T} \ . \tag{3.24}$$

L'autre quantité d'intérêt est l'énergie consommée par bit transmis, qui vaut

$$\mathcal{E} = \frac{E}{NR} = \frac{\mathbb{E}\left\{|S|^2\right\}}{R} \ .$$

Exemple 3.3 Considérons le modulateur ASK défini par la constellation $\mathcal{C} = \{-A, A\}$. On voit immédiatement que $E_m = NA^2$ et $P_m = A^2/T$.

3.2.3 Démodulation/détection

Le signal modulé (3.22) est transmis, se propage à travers un canal de transmission qui le détériore et ajoute du bruit, et le démodulateur reçoit finalement un signal Y. La question posée est d'identifier les symboles à partir de Y.

3.2.3.1 Cas d'un canal sans bruit

Supposons tout d'abord que le canal n'introduise ni distorsion ni bruit. Alors $y = x$, et les symboles $s_0, s_1, \dots s_{N-1}$ correspondants s'obtiennent simplement via (3.19) : la forme

$$d_n = \langle x, \varphi_n \rangle \ ,$$

et on a alors $d_n = s_n$. On voit facilement aussi que

$$d_n = \int x(t)\varphi(t - n/T)dt = (x * \tilde{\varphi})(n/N)$$

où on a noté $\tilde{\varphi}(t) = \overline{\varphi}(-t)$. d_n s'exprime donc comme produit de convolution, opération facile à implémenter en *hardware*.

3.2.3.2 Cas d'un canal bruité

Supposons maintenant que le canal se limite à l'ajout d'un bruit de fond (ou plutôt que les distorsions ont été corrigées, de sorte que seul subsiste le bruit de fond), de la forme

$$\epsilon(t) = \sum_{n=1}^{N} v_n \varphi_n \ .$$

On modélise, comme souvent en traitement du signal, le bruit de fond comme un processus aléatoire, ce qui revient ici à supposer que les coefficients v_n sont des réalisations indépendantes et identiquement distribuées d'une variable aléatoire V, que l'on supposera centrée ($\mathbb{E}\{V\} = 0$) et dont on supposera connue la densité ρ. Dans ces conditions, le détecteur ci-dessus produit une estimation bruitée du symbole

$$d_n = \langle x + \epsilon, \varphi_n \rangle = s_n + v_n \ , \tag{3.25}$$

et peut être modélisé comme une réalisation d'une variable aléatoire $D = s_n + V$. On rappelle

Lemme 3.2 *Soit ρ la densité de la variable aléatoire V. Alors la densité conditionnelle de D sachant S est donnée par*

$$\rho_s(t) = \rho_{D|S=s}(t) = \rho(t - s) \ , \qquad s \in \mathcal{C} \ .$$

Le problème est donc de déterminer s_n connaissant d_n. Il s'agit d'un problème de décision identique aux problèmes rencontrés dans la section 1.2.5 : étant donnée une valeur observée d_n d'une variable aléatoire D dont on connaît les lois conditionnelles à S, estimer la valeur correspondante de S.

3.2.3.3 Cas plus complexe

Lorsque la distorsion due au canal ne peut plus être ignorée, le détecteur doit être modifié pour prendre en compte les distorsions. On modélise généralement les distorsions sous la forme d'un opérateur linéaire $A : L^2([0,T]) \to L^2([0,T])$, qui conduirait à des observations

$$d_n = \langle Ax + \epsilon, \varphi_n \rangle = \langle Ax, \varphi_n \rangle + V_n \ .$$

Il est en fait souvent mieux de modifier le processus d'observation pour corriger l'effet de la distorsion... mais c'est une autre histoire qui dépasse le cadre de ce cours.

3.3 Exercices

Exercice 3.1

Démontrer les égalités (3.3), (3.4) et (3.5) portant sur la transformée de Fourier de copies respectivement translatées, modulées et dilatées d'une fonction $f \in L^1(\mathbb{R})$.

Exercice 3.2 (Modulation et démodulation AM (DBSP))

Soit $T \in \mathbb{R}^+$. Calculer $\hat{\varphi}(\nu)$ pour les fonctions $\varphi \in L^1(\mathbb{R})$ suivantes :

$$\varphi(t) = \frac{1}{\sqrt{T}} 1_{[0,T]}(t) \ , \quad \varphi(t) = \sqrt{\frac{1}{T}} \cos\left(2\pi\nu_0 \frac{t}{T}\right) 1_{[0,T]}(t) \ , \quad \varphi(t) = \frac{1}{\sqrt{T}} e^{2i\pi\nu_0 t/T} 1_{[0,T]}(t) \ ,$$

où $1_{[0,T]}$ est l'indicatrice de $[0,T]$ et où $\nu_0 \in \mathbb{N}^*$.
Que peut on dire de l'allure de φ et $\hat{\varphi}$ lorsque T varie ?

Exercice 3.3 (Modulation et démodulation AM (DBSP))

1. Dans la modulation AM DBSP (double bande sans porteuse), la porteuse $t \to x_0(t)$ est multipliée par le signal à transmettre (signal modulant) :

   $$x \to y = x.x_0 \ : \qquad y(t) = x(t)x_0(t) \ .$$

   La porteuse est généralement un signal sinusoïdal de fréquence fixée et connue ν_0 : $x_0(t) = \cos(2\pi\nu_0 t)$.

   a) Exprimer la transformée de Fourier $\hat{y}$ du signal modulé en fonction de la transformée de Fourier $\hat{x}$ du signal modulant.

   b) On suppose que le signal modulant est *à bande limitée*, c'est à dire tel que $\text{supp}(\hat{x}) \subset [-\eta, \eta]$, η étant un réel positif. Déterminer le support de $\hat{y}$.

2. Dans le cadre $L^2(\mathbb{R})$, les signaux à bande limitée avec fréquence limite η donnée forment un sous-espace de Hilbert, appelé *espace de Paley-Wiener*

   $$PW_\eta = \left\{x \in L^2(\mathbb{R}),\ \text{supp}(\hat{x}) \subset [-\eta, \eta]\right\} \ .$$

   Cet espace a des propriétés importantes, dans de nombreux contextes tels que l'échantillonnage des signaux analogiques à support infini.
   En utilisant la transformation de Fourier inverse, et le fait que $L^2([-\eta,\eta]) \subset L^1([-\eta,\eta])$ (qu'on pourra démontrer au passage), démontrer que toute fonction $x \in PW_\eta$ est continue (ou plus précisément, $L^2(\mathbb{R})$ étant un espace de classe d'équivalence de fonctions, définies à un ensemble de mesure nulle près : toute fonction $x \in PW_\eta$ admet un représentant continu).

3. La démodulation synchrone est une technique de démodulation est effectuée de façon similaire à la modulation, en introduisant le produit point par point

   $$z = x_0.y \ : \qquad z(t) = x_0(t)y(t) \ .$$

   a) Soit $x \in PW_\eta$. Exprimer la transformée de Fourier $\hat{z}$ de z en fonction de $\hat{x}$, et déterminer le support de $\hat{z}$.

   b) Montrer que si η est suffisamment petit, x peut être obtenu à partir de z par un filtrage "passe bas", c'est à dire en appliquant un filtre dont la fonction de transfert m est à support borné, et est uniformément égale à 1 dans un intervalle bien choisi (ne pas hésiter à faire un dessin...).

3.4 Projet

Le projet porte sur la modulation numérique. L'objectif est de construire

— Un **modulateur numérique** : une transformation associant à une suite de bits $b_0 b_1 \dots b_{K-1}$ un signal analogique $x(t)$ transportant l'information contenue dans la suite binaire :

$$b_0 b_1 \dots b_{N-1} \to x = \sum_n s_n \varphi_n : \quad x(t) = \sum_n s_n \varphi_n(t)$$

Ce modulateur consomme une énergie égale à $E = \|x\|^2 = \int |x(t)|^2 dt$.

— Un **démodulateur numérique** : une transformation permettant de reconstruire le message à partir du signal analogique, ou d'une version dégradée de celui-ci.

On s'intéressera aussi aux performances des MoDems ainsi développés, en termes de puissance émise et robustesse au bruit.

Les étudiants peuvent travailler soit seuls, soit par binômes (conseillé). Chaque étudiant ou binôme doit rendre en fin de projet :

— Un compte rendu, décrivant le travail effectué, les principales étapes de celui-ci (en particulier les programmes développés), et les résultats obtenus. Les résultats pourront être présentés sous forme de graphiques et de tableaux ; chaque graphique, chaque tableau devra être commenté et interprété.

— Une archive (au format `.zip`, ou `.tar.gz`) contenant les programmes développés, ainsi qu'un fichier `README.txt` donnant la liste des programmes et fonctions, avec quelques explications succintes (par exemple le prototype de la fonction).

Remarques :

— Les programmes devront être opérationnels, au sens où un utilisateur devra pouvoir les exécuter en se basant sur les indications fournies dans le compte-rendu.

— Les programmes devront être commentés à l'intérieur du code, suffisamment pour qu'un utilisateur puisse comprendre facilement son fonctionnement.

Le projet donne lieu à une note de projet, qui se base sur le compte rendu et les programmes, et une soutenance orale en janvier 2016.

3.4.1 Modulation ASK

Dans la modulation ASK, la porteuse φ est l'indicatrice d'un intervalle $[0, T]$: $\varphi(t) = K 1_{[0,T]}(t)$, où K est choisi de sorte que $\|\varphi\| = 1$. La séquence d'instructions

```
> L=fs*T;
> phi=ones(L,1);
> phi=phi/norm(phi);
```

permet de générer une telle forme d'onde. Ici `T` est la durée de la forme d'onde (en secondes), et `fs` est la fréquence d'échantillonnage, c'est à dire le nombre de valeurs par seconde. `L=fs*T` est donc le nombre de valeurs de la forme d'onde.

On considère tout d'abord le cas où la constellation est l'ensemble $\{-A, A\}$ où A est un réel positif. On associe donc A au bit 1, et $-A$ au bit 0.

Partie 3.1 (Reprise du TP2) 1. Ecrire une fonction `ask_modul.m` prenant comme variables d'entrée un message binaire `B` de longueur `N` bits, une amplitude `a`, une fréquence d'échantillonnage `fs` et une durée de forme d'onde `T` , et retournant le signal modulé `x` (qui doit donc être un vecteur de longueur `N*fs*T`), ainsi que l'énergie dépensée `E`.
Syntaxe possible : `[x,E] = ask_modul(B,A,fs,T)`

2. Ecrire une fonction `ask_demodul.m`, prenant en entrée un signal modulé `x`, la fréquence d'échantillonnage `fs` et la durée de la forme d'onde `T`, et retournant un message binaire démodulé. Pour chaque bit n la décision sera basée sur une comparaison de la moyenne de `x` sur l'intervalle `[(n-1)*T,n*T]`.
   Syntaxe possible : `B = ask_demodul(x,A,fs,T)`

On considère maintenant des constellations de la forme $-(M-1)A, -(M-3)A, \cdots -A, A, \ldots, (M-1)A$, où $M = 2^R$, R étant le nombre de bits encodé par la constellation. Par exemple, dans le cas $R = 2$, la constellation a 4 éléments, et les symboles sont respectivement $-3A$ associé à 00, $-A$ associé à 01, A associé à 10 et $3A$ associé à 11.

Partie 3.2
1. Ecrire une fonction `ask_modul2.m` prenant comme variables d'entrée un message binaire `B` de longueur `N` bits, l'amplitude `A`, le nombre de bits par symbole `R`, une fréquence d'échantillonnage `fs` et la durée `T` , et retournant le signal modulé `x` (qui doit donc être un vecteur de longueur `N*fs*T/R`), ainsi que l'énergie consommée.
2. Ecrire une fonction `ask_demodul2.m`, prenant en entrée un signal modulé `x`, la fréquence d'échantillonnage `fs` et le débit `T`, et retournant un message binaire démodulé.

Remarque : Le nombre total de bits transmis K devra être multiple de R. En cas de difficulté, on pourra se contenter du cas $M = 2$.

3.4.2 Modulation PSK

Dans le cas de la modulation psk (modulation de phase), la porteuse (complexe) est de la forme $\varphi(t) = Ke^{2i\pi\nu_0 t}1_{[0,\tau]}(t)$, où ν_0 est choisi de sorte que τ soit multiple de la période $1/\nu_0$, et K est choisi de sorte que $\|\varphi\| = 1$. La séquence d'instructions

```
> L=fs*T;
> phi=exp(2*pi*i*f0*[0:(L-1)]*T/L)';
> phi=phi/norm(phi);
```

permet de générer une telle forme d'onde. Ici `T` est la durée de la forme d'onde (en secondes), et `fs` est la fréquence d'échantillonnage, c'est à dire le nombre de valeurs par seconde et `f0` est la fréquence ν_0. Là encore, `L=fs*T` est le nombre de valeurs de la forme d'onde.
La constellation est constituée de 4 nombres complexes de la forme $Ae^{i(2k+1)\pi/4}$.

Partie 3.3 Ecrire une fonction `psk_modul.m` prenant comme variables d'entrée un message binaire `B` de longueur `N` bits, une amplitude `A`, une fréquence porteuse `f0`, une fréquence d'échantillonnage `fs` et une durée `tau`, et retournant le signal modulé `x` (qui doit donc être un vecteur de longueur `N*fs*tau`), ainsi que l'énergie dépensée `E`.
Syntaxe possible : `[x,E] = psk_modul(B,A,f0,fs,T)`
Ecrire une fonction `psk_demodul.m`, prenant en entrée un signal modulé `x`, la fréquence porteuse `f0`, la fréquence d'échantillonnage `fs` et la durée `T`, et retournant un message binaire démodulé. Pour chaque bit `n` la décision sera basée sur le produit scalaire de `x` avec la forme d'onde `phi` décalée de `n*T`.

3.4.3 Analyse de performances

Pour comparer les performances de ces MoDems, on utilisera la procédure suivante :

1. Génération de messages binaires (aléatoires) $B = b_0b_1b_2\ldots b_{K-1}$
2. Modulation $B \to x = \sum s_n\varphi_n$, suivie d'ajout d'un bruit aléatoire $x \to \tilde{x} = x + \epsilon$ (on utilisera la fonction `randn`).
3. Démodulation $\tilde{x} \to B' = b'_0b'_1\ldots b'_{K-1}$, mesure du taux d'erreur. On pourra par exemple utiliser la distance de Hamming $d(B, B') = \sum_{n=0}^{K-1} |b'_n - b_n|$.

On pourra (entre autres) étudier et tracer l'évolution de l'erreur $d(B, B')$ en fonction de l'amplitude du modulateur, pour un niveau de bruit fixé.

Index

Notations

$L^2([a,b])$, 8
PW_η, 47
$P_N([0,1])$, 9
SNR_Q, 20
$\mathcal{E}_0([0,1])$, 9
$\mathcal{E}_1([0,1])$, 9
$\mathcal{P}_M([0,1])$, 9

CLC, 30
CLV, 30

DBAP, 46
DBSP, 45

H(P), 29

TFF, 14

Bibliographie

[1] I. Daubechies (1992) : *Ten Lectures on Wavelets*. Vol. 61, CBMS-NFS Regional Series in Applied Mathematics.

[2] C. Gasquet et P. Witomski (1990) : *Analyse de Fourier et Applications*, Editions Masson, Paris.

[3] I.M. Gelfand et G.E. Shilov : *Théorie des distributions*, Dunod.

[4] A. Gersho et D. Gray (1992) : *Vector quantization*, The Springer International Series in Engineering and Computer Science, Springer (1992).

[5] B.B. Hubbard (1996) : *Ondelettes : la saga d'un outil mathématique*, Editions *Pour la Science*.

[6] N.S. Jayant et P. Noll (1984) : *The digital coding of waveforms*, Prentice Hall.

[7] J. Lamperti (1977) : *Stochastic Processes : a survey of the Mathematical Theory*. Applied Mathematical Sciences **23**, Springer Verlag.

[8] S. Mallat (1998) : *A Wavelet Tour of Signal Processing*. Academic Press, New York, N.Y.

[9] A. Papoulis *Signal Processing*. McGraw & Hill, New York.

[10] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T Wetterling (1986) : *Numerical Recipes*. Cambridge Univ. Press, Cambridge, UK.

[11] F. Riesz et B. Nagy (1955) : *Leçons d'Analyse Fonctionnelle*, Gauthier-Villars.

[12] W. Rudin : *Analyse réelle et complexe.*, McGraw et Hill.

[13] C. Soize (1993) : *Méthodes mathématiques en traitement du signal*. Masson, Paris.

[14] M. Vetterli and J. Kovacevic (1996) : *Wavelets and SubBand Coding*, Prentice Hall, Englewood Cliffs, NJ.

[15] M.V. Wickerhauser (1994) : *Adapted Wavelet Analysis, from Theory to Software*. A.K. Peters Publ.