TRAITEMENT du SIGNAL
COMPRESSION et CODAGE

Institut des Applications Avancées de l'Internet
Année 2001-02

Bruno Torrésani
Université de Provence
UFR Sciences de la Matière

CHAPITRE 1

Traitement du signal

1. Introduction

Le terme *signal* est généralement utilisé pour désigner un objet permettant la transmission d'information. Le support peut être un courant électrique (comme dans le cas des signaux transmis par des supports électroniques, mais aussi des signaux transmis au cerveau par le système visuel, le système auditif,...), une onde acoustique (signaux sonores) ou électromagnétique (signaux hertziens),... ou encore des séries de "trous" (comme les signaux audiophoniques stockés sur des CD ou des DVD), de pixels (points de niveaux de gris différents),...

Le traitement du signal est une discipline extrêmement vaste, qui inclut des tâches aussi différentes que la détection d'un signal plongé dans un bruit de fond, la transformation d'un signal d'un certain type (disons, un son) en signal d'un type différent (par exemple pour le stockage sur un CD), ou encore le codage et la compression (stocker une certaine quantité d'information dans un minimum de données, pour réduire le volume des données ou réduire le coût de la transmission) ou la cryptographie. On peut distinguer trois grandes familles de problématiques:

- Analyse et caractérisation de signaux, détection,...
- Transformation de signaux, débruitage,...
- Stockage, compression, codage, transmission.

Ces trois grandes familles de problèmes nécessitent une modélisation mathématique appropriée.

On distingue généralement deux classes de signaux: les signaux analogiques, qui sont généralement des signaux "physiques" (ondes acoustiques ou électromagnétiques, courants électriques faibles,...), et les signaux numériques (que ce soit sous forme de suites de nombres ou de suites de bits). Les opérations effectuées dans l'un des deux cadres ont quasiment toujours leur pendant dans l'autre cadre, le monde analogique étant généralement plus complexe que le monde numérique.

2. Signaux analogiques; filtrage analogique

2.1. Généralités. Les signaux analogiques sont modélisés comme fonctions d'une (ou plusieurs) variable(s) continue(s). La modélisation mathématique consiste à se donner un "espace" de signaux, auquel on pourra appliquer des traitements mathématiques classiques. Pour simplifier, on suppose que les signaux sont des fonctions définies sur l'axe réel.

On se limitera dans un premier temps à des signaux modélisés comme fonctions d'une seule variable (sons, courants électriques,...). Une notion importante est la notion d'énergie. Si un signal est représenté comme une fonction f d'une variable t (disons, le temps), l'énergie du signal est donnée par l'intégrale

$$E = \int_{-\infty}^{\infty} |f(t)|^2 \, dt$$

Cette notion se justifie en considérant l'exemple d'un circuit électrique, dont on mesure la tension aux bornes d'une résistance R. Si on note $i(t)$ l'intensité du courant dans la résistance, et $v(t)$ la tension aux bornes de cette dernière, l'énergie est donnée par

$$E = R\int_{-\infty}^{\infty} |i(t)|^2 dt = \frac{1}{R}\int_{-\infty}^{\infty} |v(t)|^2 dt \ .$$

Il est alors naturel de se limiter à la classe des signaux d'énergie finie, que l'on note $L^2(\mathbb{R})$, et qui possède des propriétés mathématiques utiles:

L'espace $L^2(\mathbb{R})$ des signaux d'énergie finie est un espace vectoriel (de dimension infinie): un signal est considéré comme un vecteur. Cet espace est muni d'un produit scalaire: si $f, g \in L^2(\mathbb{R})$, le produit scalaire de f et g, noté $\langle f, g\rangle$, est défini par

$$\langle f, g\rangle = \int_{-\infty}^{\infty} f(t)\overline{g}(t)\, dt \ .$$

Le produit scalaire permet également de définir une *norme*: si $f \in L^2(\mathbb{R})$,

$$||f|| = \sqrt{\langle f, f\rangle} = \sqrt{\int_{-\infty}^{\infty} |f(t)|^2\, dt} \ .$$

La norme $||f||$ de f représente en quelque sorte la "longueur" de f considéré comme vecteur. Dans ce langage, dire qu'un signal f est d'énergie finie signifie que la longueur du vecteur f est bien définie.

Similairement, on peut montrer que

$$|\langle f, g\rangle| \le ||f||\, ||g|| \ ;$$

ainsi, la quantité

$$\left|\frac{\langle f, g\rangle}{||f||\, ||g||}\right|$$

est inférieure à 1, et peut donc s'interpréter comme le cosinus de l'angle que font entre eux les vecteurs f et g.

L'opération fondamentale du traitement du signal est l'opération de filtrage.
Définition: un filtre analogique est une transformation $T : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ qui est invariant par translation, dans le sens suivant. Pour tout $f \in L^2(\mathbb{R})$, en considèrant la "copie translatée" g de f défini par $g(t) = f(t-\tau)$, on a

$$Tg(t) = Tf(t-\tau) \ .$$

En d'autres termes, une translation suivie d'un filtrage est identique au filtrage suivi de la translation.

Un exemple très concret de filtrage est donné par le circuit RC:

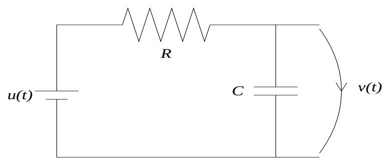

Fig. 1. *Le filtre RC*

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

$$RC\, v'(t) + v(t) = u(t)\ .$$

Pour résoudre cette dernière, il est utile d'introduire la fonction $w(t) = v(t)e^{t/RC}$. On a donc

$$w'(t) = \frac{1}{RC} e^{t/RC} u(t)$$

et la solution est

$$w(t) = \frac{1}{RC} \int_{-\infty}^{t} e^{s/RC} u(s) ds\ ,$$

soit

$$\begin{aligned} v(t) &= \frac{1}{RC} \int_{-\infty}^{t} e^{-(t-s)/RC} u(s) ds \\ &= \int_{-\infty}^{\infty} h(t-s) u(s) ds\ . \end{aligned}$$

où nous avons posé

$$h(t) = \Theta(t) e^{-t/RC}$$

$\Theta(t)$ étant la fonction d'Heaviside, qui vaut 1 pour $t \geq 0$ et 0 pour $t < 0$.

2.2. Filtres de convolution: Des exemples simples de filtres sont donnés par les *filtres de convolution*. Ces derniers sont définis par une fonction h, appelée *réponse impulsionnelle*, et on définit le filtre correspondant noté $T = K_h$ par: pour tout signal d'énergie finie $f \in L^2(\mathbb{R})$, son image $Tf = K_h f$ par le filtre est donné par l'intégrale

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

L'opération "$*$" représente le produit de convolution. Il s'agit d'une opération symétrique:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(s) g(t-s)\, ds = \int_{-\infty}^{\infty} g(u) f(t-u)\, du = (g * f)(t)\ .$$

On dit que le filtre de convolution K_h est *causal*, ou *réalisable*, si sa réponse impulsionnelle h est telle que

$$h(t) = 0 \text{ pour tout } t < 0\ .$$

La contrainte de causalité est une contrainte importante: seuls les filtres causaux sont susceptibles d'être mis en oeuvre en pratique.

Le filtre RC vu plus haut est un exemple de filtre de convolution causal. Sa réponse impulsionnelle (voir en FIG. 2 est donnée par

$$h(t) = \begin{cases} e^{-t/RC} & \text{si } t \geq 0 \\ 0 & \text{sinon}\ . \end{cases}$$

Les filtres de convolution ne sont pas les uniques exemples de filtres. Par exemple, le filtre le plus simple défini par $Tf = f$ pour tout f n'est pas un filtre de convolution. On verra plus loin une caractérisation des filtres analogiques utilisant la transformation de Fourier.

2.3. Transformation de Fourier et filtres analogiques.

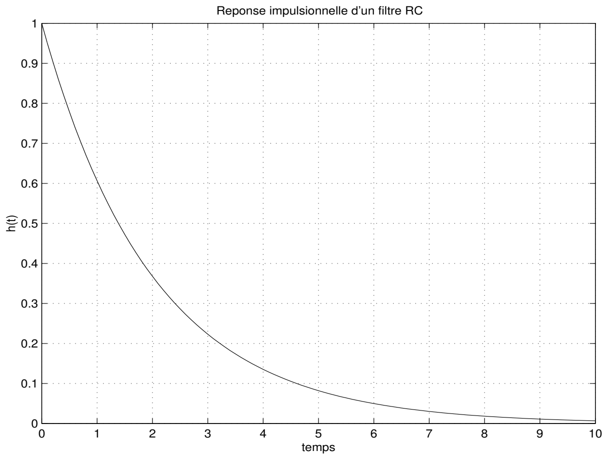

FIG. 2. *Réponse impulsionnelle d'un filtre RC*

2.3.1. *La transformation de Fourier intégrale.* La transformation de Fourier est l'un des outils mathématiques les plus fondamentaux, dans de nombreux domaines d'application. L'idée essentielle de l'analyse de Fourier est que sous certaines conditions, n'importe quelle fonction peu être représentée comme somme de fonctions oscillantes élémentaires, de la forme

$$t \to e^{i\omega t} = \cos(\omega t) + i\sin(\omega t) \ .$$

La variable ω, appelée *fréquence*, possède une interprétation "physique" importante. Mathématiquement, plus ω est élevé, plus la fonction trigonométrique correspondante est rapidement variable, comme on peut le voir sur la FIGURE 3. Cette interprétation en termes de "vitesse de variation" se décline differement dans differents contextes.

> Par exemple, dans le monde des signaux audiophoniques, si t repr ésente le temps, la fréquence correspond à la "hauteur" du son. Si t est exprimé en secondes, la fréquence s'exprime alors en Hertz. On peut voir dans la FIGURE 3 (les deux courbes du haut) les sinusoïdes (plus précisément, les parties réelles des fonctions exponentielles) correspondant à la fréquence $\omega = 400$ Hertz et $\omega = 800$ Hertz (donc, le *la* fondamental et le *la* une octave plus haut.)

Le résultat fondamental de Fourier est le suivant:

THÉORÈME: Tout signal d'énergie finie peut s'exprimer comme superposition de sinusoïdes: pour tout $f \in L^2(\mathbb{R})$, il existe une fonction notée $\hat{f}$, appelée *transformée de Fourier* de f, telle que l'on ait

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega t}\, d\omega \ .$$

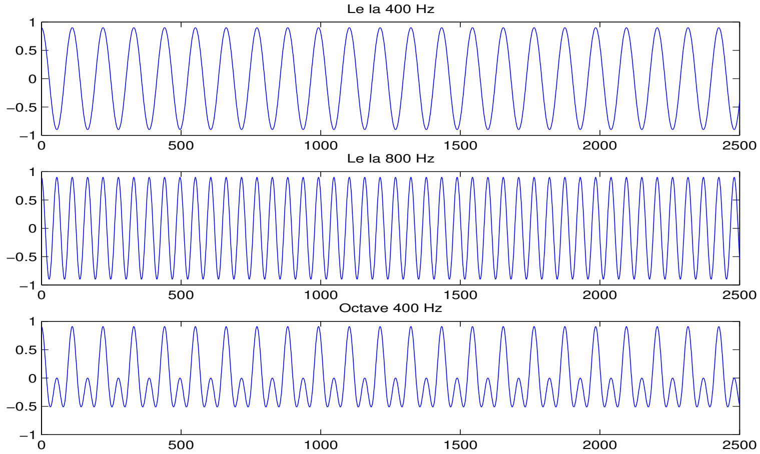

FIG. 3. *Tracés de trois signaux correspondant à un* la *fondamental, un* la *à l'octave, et de la somme des deux.*

La fonction $\hat{f}$, définie par

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}\, dt \ ,$$

est elle aussi d'énergie finie, et vérifie la *Formule de Plancherel*

$$\int_{-\infty}^{\infty} |\hat{f}(\omega)|^2\, d\omega = 2\pi \int_{-\infty}^{\infty} |f(t)|^2\, dt \ .$$

Plus généralement, pour tous $f, g \in L^2(\mathbb{R})$, on a

$$\langle f, g\rangle = \int_{-\infty}^{\infty} \hat{f}(\omega)\overline{\hat{g}}(\omega)\, d\omega = 2\pi \int_{-\infty}^{\infty} f(t)\overline{g}(t)\, dt = 2\pi\langle f, g\rangle \ .$$

On peut remarquer que la transformation de Fourier

$$\mathcal{F} : f \to \hat{f}$$

est une transformation linéaire.

Dans un certains sens, pour une fréquence ω donnée, le nombre (complexe) $\hat{f}(\omega)$ donne une mesure du "contenu" du signal f à la fréquence ω. On peut voir en FIGURE 4 les transformées de Fourier des trois signaux de la FIGURE 3. Comme on peut le voir la première figure montre que $\hat{f}(\omega) \neq 0$ uniquement pour $\omega = 400$Hz, alors que la deuxième correspond à $\hat{f}(\omega) \neq 0$ uniquement pour $\omega = 800$Hz. En ce qui concerne la troisième, elle correspond à un $\hat{f}$ égal à la somme des deux deux premiers. Puisque la transformation de Fourier est linéaire, ceci signifie que le f la troisième courbe de la FIGURE 3 est somme des deux autres, c'est à dire de deux *la*, ce qui est effectivement le cas.

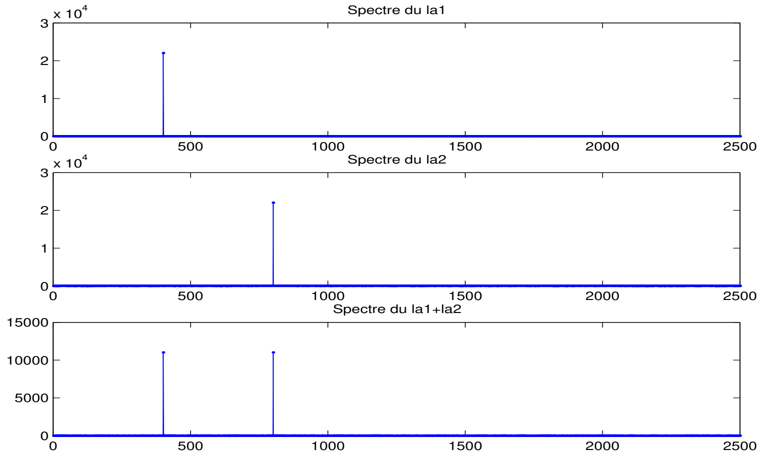

FIG. 4. *Transformées de Fourier des trois signaux de la* FIGURE *3*

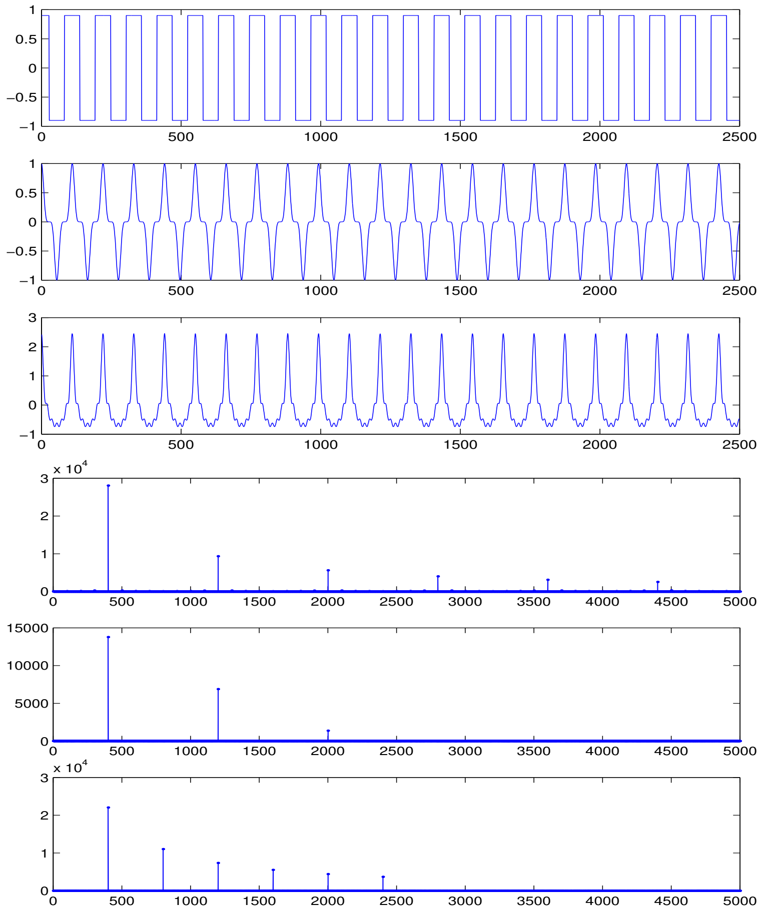

FIG. 5. *Trois exemples de signaux, et leurs transformées de Fourier. Les trois signaux (haut) ont la même périodicité, mais des contenus fréquentiels différents.*

La transformation de Fourier possède de multiples propriétés mathématiques importantes, notamment vis à vis du produit de convolution.

Etant donnés deux signaux $f, g \in L^2(\mathbb{R})$:

– La transformèe de Fourier du produit de convolution de f et g est

– La transformée de Fourier du produit de f et g est proportionnelle au produit de convolution de leurs transformées de Fourie

$$\widehat{fg}(\omega) = \frac{1}{2\pi}\,(\hat{f} * \hat{g})(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\eta)\hat{g}(\omega-\eta)\,d\eta\ .$$

2.3.2. *Transformation de Fourier et filtres analogiques.* Le résultat ci-dessus a d'importantes conséquences sur les filtres de convolution que nous avons vu plus haut. En effet, étant donné un filtre de convolution $T = K_h$, de réponse impulsionnelle h, il résulte de ce qui précède que l'on peut écrire pour tout signal $f \in L^2(\mathbb{R})$

$$\widehat{K_h f}(\omega) = \widehat{h * f}(\omega) = \hat{h}(\omega)\hat{f}(\omega)\ ,$$

autrement dit, après une transformation de Fourier inverse

$$K_h f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t}\hat{h}(\omega)\hat{f}(\omega)\,d\omega\ .$$

Ce résultat est en fait un cas particulier d'un résultat plus général: tout filtrage analogique (convolution ou pas) consiste en une multiplication par une *fonction de transfert* dans le domaine des fréquences. Plus précisément:

Théorème: Pour tout filtre analogique $T : L^2(\mathbb{R}) \to L^2(\mathbb{R})$, il existe une fonction (bornée)

$$m : \omega \to m(\omega)\ ,$$

appelée *fonction de transfert*, telle que pour tout signal $f \in L^2(\mathbb{R})$, on ait

$$\widehat{Tf}(\omega) = m(\omega)\hat{f}(\omega)\ ,$$

autrement dit

$$Tf(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega t} m(\omega)\hat{f}(\omega)\,d\omega\ .$$

Ceci donne un éclairage nouveau sur le filtrage: un filtrage analogique est en fait une opération qui permet de modifier le contenu fréquentiel des signaux. Les valeurs de ω pour lesquelles $m(\omega)$ est petit (en module) correspondent à des fréquences qui sont atténuées dans le signal filtré. Inversement, les valeurs de ω pour lesquelles $m(\omega)$ est grand (en module) correspondent à des fréquences qui sont renforcées dans le signal filtré.

Reprenant l'exemple du filtre RC vu plus haut, il est possible de voir que sa fonction de transfert est donnée par

$$m(\omega) = \int_{-\infty}^{\infty} h(t)e^{-i\omega t}\,dt = \int_{0}^{\infty} e^{-t/RC}e^{-i\omega t}\,dt = \frac{1}{i\omega - 1/RC}\ .$$

2.3.3. *Les filtres idéaux.* Les filtres "idéaux" sont des filtres qui permettent d'effectuer une atténuation "parfaite" de certaines fréquences des signaux. L'exemple classique est donné par le filtre "passe-bas" idéal, donné par une fonction de transfert de la forme

$$m(\omega) = \begin{cases} 1 & \text{si } |\omega| \le \omega_0 \\ 0 & \text{sinon}\ , \end{cases}$$

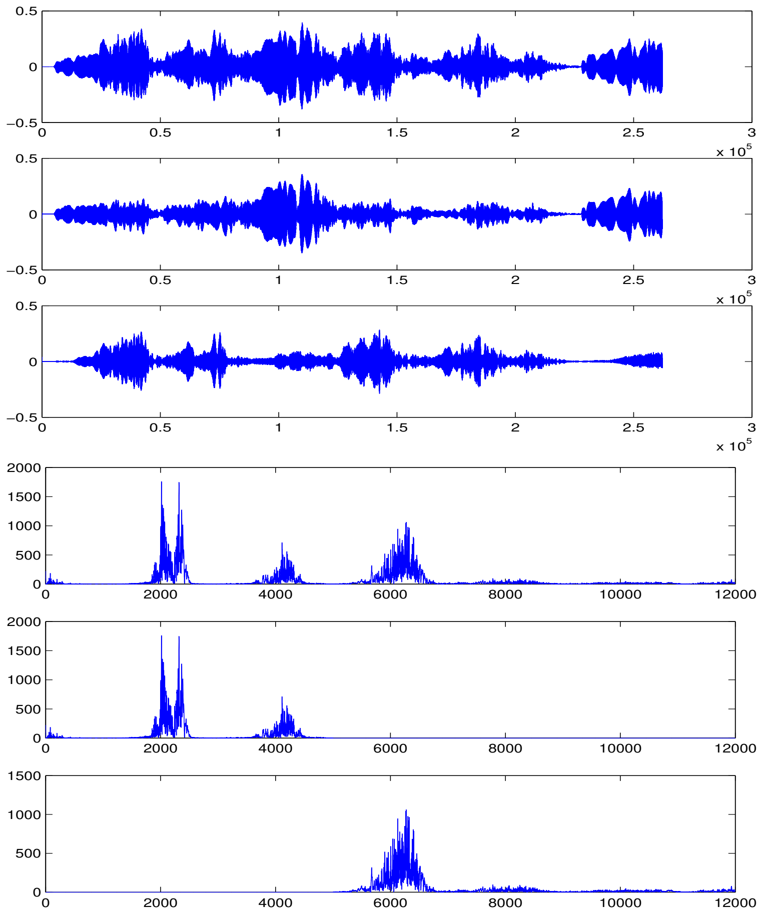

FIG. 6. *Chant de soprano, et le même filtré "passe bas" et "passe haut" (figures du haut); transformées de Fourier de ces trois signaux (figures du bas).*

où ω_0 est une fréquence fixée, appelée *fréquence de coupure*. L'effet d'un tel filtre est de supprimer dans les signaux toutes les composantes correspondant à des fréquences supérieures à ω_0.

On peut de la même façon effectuer des filtrages "passe haut" idéaux (c'est à dire en supprimant les fréquences inférieures à une certaine fréquence de coupure), on encore des filtrages "passe bande" idéaux (qui ne conservent que les fréquences incluses d'une "bande de fréquences" donnée).

Un exemple est présenté dans la FIGURE 6, qui représente un signal audio (un enregistrement d'une chanteuse soprano), ainsi que deux versions de ce signal filtrées passe bas et passe haut. On a également représenté les transformées de Fourier de ces signaux.

Il est possible de montrer que le filtre passe bas idéal est un filtre de convolution. Sa réponse impulsionnelle est donnée par la transformée de Fourier inverse de la fonction de transfert

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} m(\omega) e^{i\omega t}\, dt = \frac{1}{2\pi} \int_{-\omega_0}^{\omega_0} e^{i\omega t}\, dt = \frac{\omega_0}{\pi} \frac{\sin(\omega_0 t}{\omega_0 t} \ .$$

On voit immédiatement que ce filtre "idéal" ne l'est pas tant que cela, car il n'est pas causal: $h(t) \neq 0$ pour $t < 0$. La conséquence pratique est que ce filtre ne peut pas être mis en oeuvre en "hardware".

On peut aussi montrer que les filtres passe haut et passe bande idéaux ne sont pas réalisables non plus. On doit en pratique se contenter de filtres analogiques qui approximent les filtres idéaux.

2.3.4. *Transformation de Fourier, filtres analogiques et équations différentielles.* La transformation de Fourier possède une autre propriété fondamentale: elle transforme la dérivation en multiplication:

> La transformée de Fourier de la dérivée d'un signal f est donnée par
>
> $$\widehat{f'}(\omega) = i\omega \hat{f}(\omega) \ .$$
>
> Plus généralement, en notant
>
> $$f^{(n)}(t) = \frac{d^n f(t)}{dt^n}$$
>
> la dérivée n-ième de f, sa transformée de Fourier est donnée par
>
> $$\widehat{f^{(n)}}(\omega) = (i\omega)^n \hat{f}(\omega) \ .$$

Or, on sait que le comportement de nombreux systèmes physiques est gouvrné par des équations différentielles, qui peuvent en fait être résolues en utilisant la transformation de Fourier.

C'est notamment le cas du circuit RC qu'on a vu plus haut, qui était décrit par l'équation différentielle

$$RC\, v'(t) + v(t) = u(t) \ .$$

Par une transformation de Fourier, on obtient une autre équation, qui relie les transformées de Fourier de u et v:

$$(1 + i\omega RC)\hat{v}(\omega) = \hat{u}(\omega) \ ,$$

dont la solution est

$$\hat{v}(\omega) = m(\omega)\hat{u}(\omega) \ ,$$

où la fonction de transfert est donnée par

$$m(\omega) = \frac{1}{1 + i\omega RC} \ ,$$

c'est à dire le même résultat que celui que nous avons obtenu plus tôt.

Dans le cas d'équations différentielles plus générales, on peut montrer un résultat similaire:

> Supposons que le signal d'entrée u et le signal de sortie v d'un filtre T soient reliés par une équation différentielle (à coefficients constante) de la

forme

$$a_0 v(t) + a_1 v'(t) + a_2 v''(t) + \cdots + a_N v^{(N)}(t) = \\ b_0 u(t) + b_1 u'(t) + b_2 u''(t) + \cdots + b_M u^{(M)}(t) \ ,$$

où $a_0 \dots a_N$ et $b_0 \dots b_M$ sont des constantes). Alors leurs transformées de Fourier sont reliées par l'équation

$$\hat{v}(\omega) = m(\omega)\hat{u}(\omega) \ ,$$

où la fonction de transfert m est le quotient de deux polynômes (fonctions de la variable complexe $p = i\omega$)

$$m(\omega) = \frac{b_0 + b_1 i\omega + b_2 (i\omega)^2 + \cdots + b_M (i\omega)^M}{a_0 + a_1 i\omega + a_2 (i\omega)^2 + \cdots + a_N (i\omega)^N} \ .$$

L'intérêt des polynômes est qu'ils sont "factorisables". Plus précisément, le dénominateur de la fraction ci-dessus est un polynôme de degré N; par conséquent, il s'annulle pour N valeurs de p, que l'on notera $p_1, p_2, \dots p_N$, et est proportionnel au produit $(p - p_1)(p - p_2)\dots(p - p_N)$. De même, le numérateur de la fraction ci-dessus est un polynôme de degré M; par conséquent, il s'annulle pour M valeurs de p, que l'on notera $q_1, q_2, \dots q_M$, et est proportionnel au produit $(p - q_1)(p - q_2)\dots(p - q_M)$. Finalement, la fonction de transfert est complètement caractérisées par les "racines" $p_1, \dots p_N$ du dénominateur, les "racines" $q_1, \dots q_M$ du numérateur et une constante C, et on peut écrire

$$m(\omega) = C\, \frac{(i\omega - q_1)(i\omega - q_2)\dots(i\omega - q_M)}{(i\omega - p_1)(i\omega - p_2)\dots(i\omega - p_N)} \ .$$

Il s'avère que si l'on impose que le filtre soit réalisable, cette contrainte impose de fortes contraintes sur les racines $p_1, \dots p_N$ du dénominateur. Plus précisément:

Théorème: Un filtre de fonction de transfert

$$\omega \to \frac{n(i\omega)}{d(i\omega)}$$

où le numérateur n et le dénominateur d sont deux polynômes est réalisable si et seulement si les racines $p_1, \dots p_N$ du dénominateur d ont une partie réelle négative:

$$\Re(p_1) < 0,\ \Re(p_1) < 0, \dots \Re(p_N) < 0$$

2.3.5. *Synthèse de filtres.* On a vu plus haut que les filtres idéaux ne sont pas réalisables. Le problème de la synthèse de filtres consiste à chercher des approximations réalisables de filtres idéaux (ou pas). La question est essentiellement la suivante: étant donnée une fonction de transfert m, ne correspondant pas à un filtre réalisable, comment construire une approximation de m par une fraction rationnelle comme plus haut?

Il s'avère que ce problème ne possède généralement pas de solution satisfaisante, et on se contente d'une version plus "faible": comment construire un filtre réalisable dont la fonction de transfert, de la forme

$$\omega \to \frac{n(i\omega)}{d(i\omega)}$$

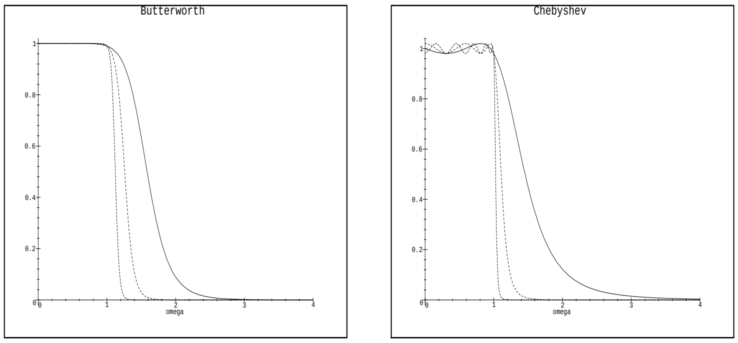

FIG. 7. *Modules carrés de fonctions de transfert de quelques filtres de Butterworth (à gauche) et de Chebyshev (à droite)*

(où le numérateur n et le dénominateur d sont deux polynômes) telle que la fonction $|n(i\omega)/d(i\omega)|^2$ fournisse une bonne approximation de $|m(\omega)|^2$?

En pratique, on commence par se donner un candidat pour $|n(i\omega)/d(i\omega)|^2$, sous la forme d'un quotient de deux polynômes

$$\omega \to \frac{\mathcal{N}(i\omega)}{\mathcal{D}(i\omega)}$$

Sous des hypothèses appropriées, il est possible de s'assurer qu'un tel quotient est en effet le module carré de la fonction de transfert d'un filtre réalisable. Plus précisément:

THÉORÈME: Soient $\mathcal{N}$ et $\mathcal{D}$ deux polynômes à coefficients réels positifs, pairs (c'est à dire tels que $\mathcal{N}(-p) = \mathcal{N}(p)$ et $\mathcal{D}(-p) = \mathcal{D}(p)$). Alors il existe deux polynômes n et d tels que l'on ait

$$\frac{\mathcal{N}(i\omega)}{\mathcal{D}(i\omega)} = \left|\frac{n(i\omega)}{d(i\omega)}\right|^2 .$$

De plus, d peut être choisi de sorte que le filtre de fonction de transfert

$$\omega \to \frac{n(i\omega)}{d(i\omega)}$$

soit réalisable.

Un exemple simple est donné par les *filtres de Butterworth* couramment utilisés pour approximer des filtres passe bas idéaux. Ils sont donnés par le choix

$$\frac{\mathcal{N}(i\omega)}{\mathcal{D}(i\omega)} = \frac{1}{1 + (\omega/\omega_0)^{2N}} ,$$

où N est un entier fixé. Les modules carrés des fonctions de transfert de quelques filtres de Butterworth sont représentées en FIGURE 7, avec les fonctions de transfert de filtres de Chebyshev (un autre choix classique)

2.4. Signaux bidimensionnels, images naturelles. Les signaux ne sont pas tous modélisables comme fonctions d'une seule variable. Par exemple, les images se représentent naturellement comme fonctions de deux variables, et se prètent à une modélisation similaire à celle des signaux unidimensionnels.

L'espace $L^2(\mathbb{R}^2)$ des signaux bidimensionnels d'énergie finie est un espace vectoriel (de dimension infinie), muni d'un produit scalaire: si $f, g \in L^2(\mathbb{R}^2)$, le produit scalaire de f et g est défini par

$$\langle f, g\rangle = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\overline{g}(x,y)\,dx\,dy\ .$$

Le produit scalaire permet également de définir une *norme*: si $f \in L^2(\mathbb{R}^2)$,

$$||f|| = \sqrt{\langle f, f\rangle} = \sqrt{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f(x,y)|^2\,dx\,dy}\ .$$

Les notions utilisées en 1D se transposent sans difficulté au cas 2D. On définit un filtre analogique de la même façon: un filtre analogique est une transformation linéaire $T : L^2(\mathbb{R}^2) \to L^2(\mathbb{R}^2)$, telle qu'un filtrage précédé d'une translation est équivalent à la translation précédée d'un filtrage. Par exemple, un filtre de convolution est défini par une réponse impulsionnelle h qui est cette fois une fonction de deux variables:

$$K_h f(x,y) = (h * f)(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(u,v)f(x-u, y-v)\,du\,dv\ .$$

Les fonctions de deux variables possèdent un développement de Fourier: elles peuvent se décomposer comme sommes de sinusoïdes 2D

$$(x,y) \to e^{i(\xi x + \eta y)}$$

Théorème: Pour tout $f \in L^2(\mathbb{R}^2)$, il existe une fonction notée $\hat{f}$, appelée *transformée de Fourier* de f, telle que l'on ait

$$f(x,y) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(\xi,\eta)e^{i(\xi x+\eta y)}\,d\xi\,d\eta\ .$$

La fonction $\hat{f}$, définie par

$$\hat{f}(\xi,\eta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i(\xi x+\eta y)}\,dx\,dy\ ,$$

est elle aussi d'énergie finie, et vérifie la *Formule de Plancherel*

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |\hat{f}(\xi,\eta)|^2\,d\xi\,d\eta = 4\pi^2\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} |f(x,y)|^2\,dx\,dy\ .$$

Plus généralement, pour tous $f, g \in L^2(\mathbb{R}^2)$, on a

$$\begin{aligned}\langle f, g\rangle &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(\xi,\eta)\overline{\hat{g}}(\xi,\eta)\,d\xi\,d\eta\\ &= 4\pi^2\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\overline{g}(x,y)\,dx\,dy\\ &= 4\pi^2\,\langle f, g\rangle\ .\end{aligned}$$

FIG. 8. *Exemples de filtrages sur une image: image originale (en haut), image filtrée passe-bas (à gauche), et image filtrée passe-haut (à droite).*

Enfin, les filtres bidimensionnels possèdent une caractérisation en termes de fonction de transfert: étant donné un filtre $T : L^2(\mathbb{R}^2) \to L^2(\mathbb{R}^2)$, il existe une fonction m, appelée fonction de transfert, telle que

$$Tf(x,y) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} m(\xi,\eta) \hat{f}(\xi,\eta) e^{i(\xi x + \eta y)} \, d\xi \, d\eta \ .$$

On peut tout comme précédemment introduire la notion de filtrage passe-bas et filtrage passe haut. Le filtrage passe-bas a pour objectif de réduire l'importance des hautes fréquences dans l'image filtrée; inversement, un filtrage passe haut conserve les composantes "haute fréquence" et réduit l'importance des basses fréquences. Les hautes fréquences correspondent dans ce cas à toutes les composantes rapidement variables de l'image (les détails), alors que les basses fréquences correspondent aux composantes lentement variables (les tendances). Des exemples d'images filtrées passe haut et passe bas se trouvent dans les FIGURES 8 et 9. Comme on peut le constater, les images filtrées passe bas sont plus "lisses", ou "floues" que les images originales. Par contre, les images filtrées passe haut sont quant à elles plus précises

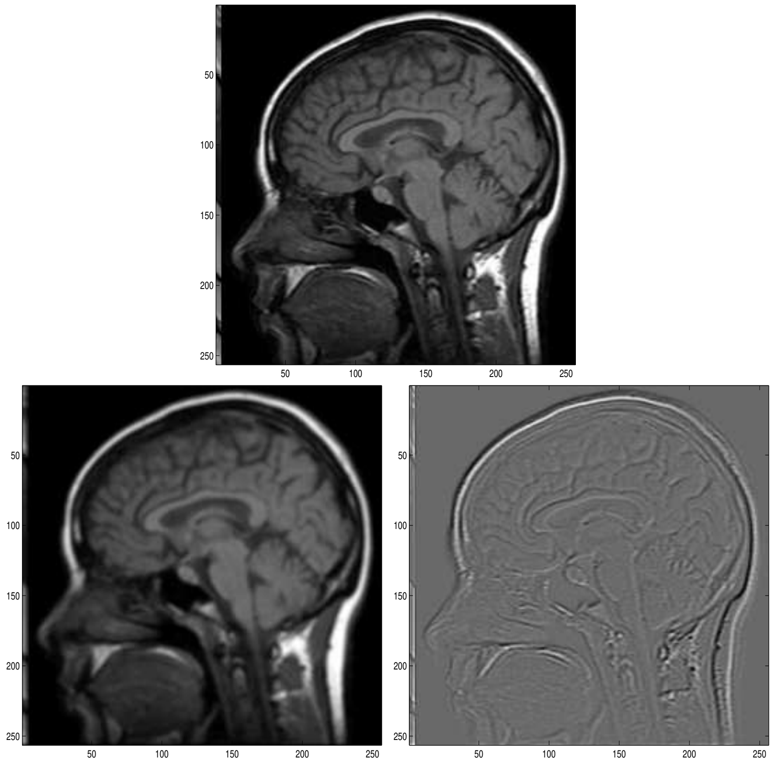

FIG. 9. *Exemples de filtrages sur une image: image originale (en haut), image filtrée passe-bas (à gauche), et image filtrée passe-haut (à droite).*

en ce qui concerne les détails de petite taille, mais les tendances les plus lentement variables n'y apparaissent plus.

Bien entendu, la proportion de détails ou de tendances qui apparaissent dans une image filtrée dépend du filtre utilisé, notamment de sa fréquence de coupure (dans le cas de filtres passe bas ou passe haut).

3. Signaux numériques

3.1. Signaux numériques; généralités. Bien que le monde des signaux numériques soit un monde différent du monde analogique (les "supports physiques" sont fondamentalement différents), il existe de grandes similarités entre le traitement du signal numérique et le traitement du signal analogique. En particulier, les opérations fondamentales du traitement du signal analogique (filtrage, transformation de Fourier,...) peuvent être adaptées au cas numérique.

Cependant, les contraintes qui pèsent sur les techniques du traitement du signal numériques sont différentes de celles qui sont imposées sur les techniques analogiques. En effet, dans le cas numérique, toutes les opérations doivent être effectuées numériquement, donc via un nombre fini d'opérations. Ceci impose des contraintes practiques fortes.

3.1.1. *Modèle mathématique pour les signaux numériques.* Le monde des signaux numériques est le monde des suites plutôt que des fonctions. Un signal numérique est une suite

$$n \in \mathbb{Z} \to f_n \in \mathbb{C} ,$$

par exemple obtenu par *échantillonnage* d'un signal analogique

$$u \in L^2(\mathbb{R}) \to f \in \ell^2(\mathbb{Z}) : \quad f_n = u(n/\eta) ,$$

où η est une constante positive.

Le pendant de l'espace $L^2(\mathbb{R})$ des signaux analogiques d'énergie finie est l'espace $\ell^2(\mathbb{Z})$ des signaux numériques d'énergie finie

> L'espace $\ell^2(\mathbb{Z})$ des signaux d'énergie finie est un espace vectoriel (de dimension infinie): muni d'un produit scalaire: si $f, g \in \ell^2(\mathbb{Z})$, le produit scalaire de f et g, noté $\langle f, g\rangle$, est défini par
>
> $$\langle f, g\rangle = \sum_{-\infty}^{\infty} f_n \overline{g}_n .$$
>
> Le produit scalaire permet également de définir une *norme*: si $f \in \ell^2(\mathbb{Z})$,
>
> $$||f|| = \sqrt{\langle f, f\rangle} = \sqrt{\sum_{-\infty}^{\infty} |f_n|^2} .$$

3.1.2. *Filtrage numérique.* un filtre numérique est défini de façon similaire au filtre analogique.

DÉFINITION: un filtre numérique est une transformation $T : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ qui est invariant par translation, dans le sens suivant. Pour tout $f \in \ell^2(\mathbb{Z})$, en considèrant la "copie translatée" g de f défini par $g_n = f_{n-k}$, on a

$$(Tg)_n = Tf_{n-k} .$$

En d'autres termes, une translation suivie d'un filtrage est identique au filtrage suivi de la translation.

De nouveau, les filtres de convolution fournissent de bons exemples: partant d'une suite $h = \{h_n, n \in \mathbb{Z}\}$, le filtre numérique associé K_h est défini par

$$K_h f = h * f ,$$

où $h * f$ représente le produit de convolution (discret) des suites h et f:

$$(h * f)_n = \sum_{k=-\infty}^{\infty} h_k f_{n-k} = \sum_{k=-\infty}^{\infty} f_k h_{n-k} .$$

Un exemple particulier est fourni par le "lissage" suivant:

$$(K_h f)_n = \frac{1}{2} (f_n + f_{n-1}) ,$$

qui correspond à $h_0 = h_1 = 1/2$, et $h_n = 0$ pour tout $n \neq 0, 1$. Cette opération remplace une valeur par la moyenne de cette valeur et de la valeur précédente. Le signal ainsi filtré est donc plus lentement variable, il s'agit d'un filtrage passe bas (non idéal). Un exemple se trouve en FIGURE 10.

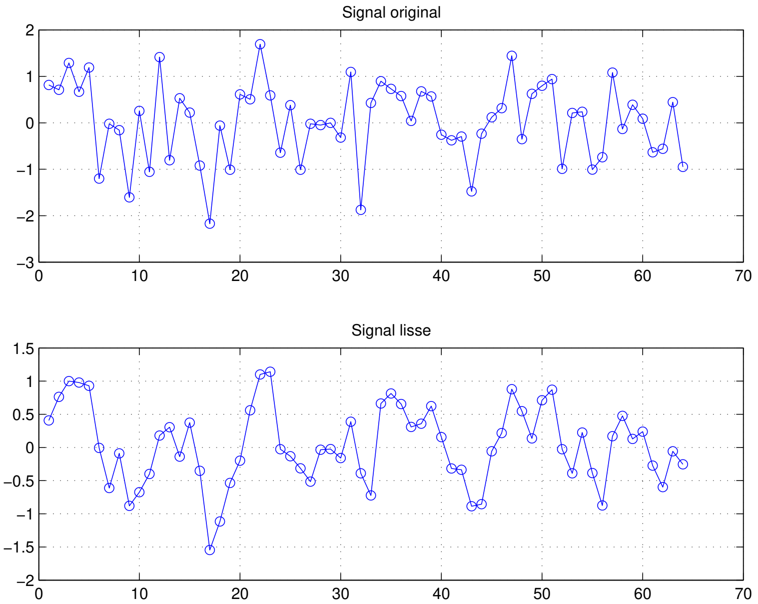

FIG. 10. *Exemple de lissage d'un signal numérique: signal original (en haut), signal filtré (en bas).*

3.2. Transformation de Fourier discrète. On a déjà vu une première version de la transformation de Fourier: la transformée de Fourier d'une fonction définie sur l'axe réel est une autre fonction, d'une autre variable (la fréquence) définie elle aussi sur l'axe réel. On a également vu que la notion de fréquence est étroitement associée à la vitesse de variations: la fonction

$$t \to e^{i\omega t}$$

est d'autant plus rapidement variable que la fréquence ω est élevée.

Le cas des suites (i.e. des signaux numériques) est similaire, à ceci près qu'une suite ne peut pas être arbitrairement rapidement variable: il existe une limite à la vitesse de variations, qui est fixé par la nature discrète de la suite. Cette remarque banale se reflète naturellement sur la version de la transformation de Fourier adaptée aux signaux numériques: la transformation de Fourier discrète (TFD): le domaine des fréquences "utiles" n'est plus l'axe réel entier, mais est cette fois l'intervalle $[-\pi, \pi]$:

> THÉORÈME: Tout signal numérique d'énergie finie peut s'exprimer comme superposition de sinusoïdes: pour tout $u \in \ell^2(\mathbb{Z})$, il existe une fonction notée $\hat{u}$, appelée *transformée de Fourier discrète* de u, telle que l'on ait
>
> $$u_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{u}(\omega) e^{in\omega}\, d\omega \ .$$

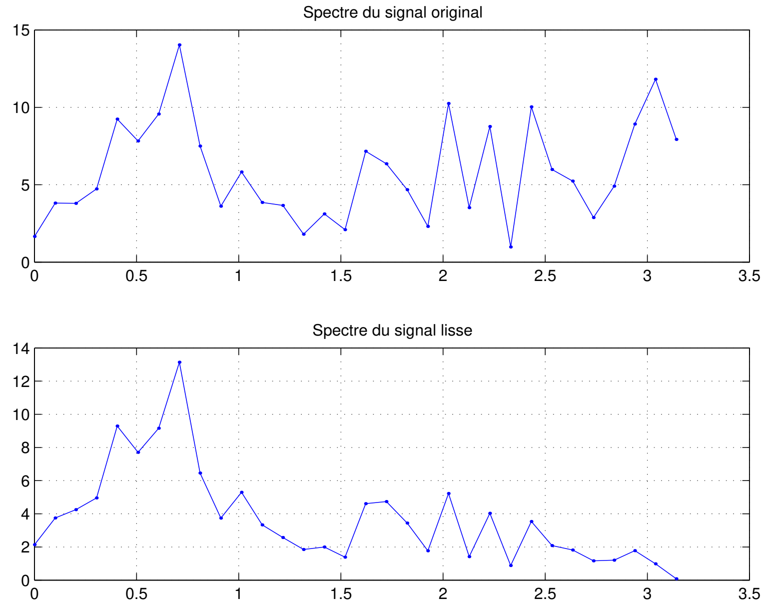

FIG. 11. *Spectre des signaux de la* FIGURE *10: signal original (en haut), signal filtré (en bas).*

La fonction $\hat{u}$, définie par

$$\hat{f}(\omega) = \sum_{-\infty}^{\infty} u_n e^{-in\omega} \ ,$$

est elle aussi d'énergie finie, et vérifie la *Formule de Parseval*

$$\int_{-\pi}^{\pi} |\hat{u}(\omega)|^2 \, d\omega = 2\pi \sum_{-\infty}^{\infty} |u_n|^2 \ .$$

Plus généralement, pour tous $u, v \in \ell^2(\mathbb{Z})$, on a

$$\langle u, v\rangle = \int_{-\pi}^{\pi} \hat{u}(\omega)\overline{\hat{v}}(\omega) \, d\omega = 2\pi \sum_{-\infty}^{\infty} u_n \overline{v}_n = 2\pi \langle u, v\rangle \ .$$

Comme on peut le voir, la transformée de Fourier discrète d'un signal numérique définit une fonction 2π-périodique, qui est donc complètement caractérisée par ses valeurs entre $-\pi$ et π.

Un exemple se trouve en FIGURE 11, où sont représentés les spectres (c'est à dire les modules carrés des transformées de Fourier) des deux signaux de la FIGURE 10 (seules les fréquences comprises entre 0 et π sont représentées). Comme on peut le voir, alors que les fréquences proches de zéro sont peu affectées par le filtrage, les “hautes fréquences” (donc les fréquences proches de π) sont très atténuées dans le signal lissé.

Le filtrage numérique s'exprime quant à lui de façon très simple dans l'espace des fréquences: un filtre numérique est complètement caractérisé par une fonction de transfert

$$m : \omega \in [-\pi, \pi] \to m(\omega) \ ,$$

via les relations suivantes:

$$\widehat{Tf}(\omega) = m(\omega)\hat{f}(\omega) \ ,$$

ou encore

$$(Tf)_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{in\omega} m(\omega)\hat{f}(\omega)\, d\omega \ .$$

De nouveau, dans le cas d'un filtre de convolution, la fonction de transfert n'est autre que la transformée de Fourier discrète de la réponse impulsionnelle:

$$m(\omega) = \hat{h}(\omega) = \sum_{-\infty}^{\infty} h_n e^{-in\omega}$$

Un cas particulier intéressant est fourni par les filtres idéaux; étant donné une fréquence de coupure $0 < \omega_0 < \pi]$, le filtre passe bas idéal correspondant est défini par

$$m(\omega) = \begin{cases} 1 & \text{si } |\omega| \leq \omega_0 \\ 0 & \text{sinon} \ , \end{cases}$$

Ce filtre est un filtre de convolution: un calcul simple (de transformation de Fourier discrète inverse) montre que sa réponse impulsionnelle est

$$h_n = \frac{\omega_0}{\pi} \frac{\sin(n\omega_0)}{n\omega_0} \ .$$

Le filtre n'est donc pas causal ($h_n \neq 0$ pour $n < 0$), mais cette propriété est moins grave que dans le cadre analogique: il est impossible de construire un filtre analogique qui ne soit pas causal (pour des raisons physiques: aucun système physique ne peut utiliser le futur pour calculer le présent), mais la non causalité ne se traduit dans le cas numérique que par un retard.

Cependant, un autre problème provient du fait que h_n décroît comme $1/n$ pour n grand, ce qui est très lent, et conduirait à des coûts en calcul prohibitifs pour bien des applications.

Si on reprend l'exemple du lissage simple précédent, la fonction de transfert correspondante est facilement obtenue:

$$m(\omega) = \hat{h}(\omega) = \frac{1}{2}\left(1 + e^{-i\omega}\right) = e^{-i\omega/2} \cos\left(\frac{\omega}{2}\right) \ .$$

Comme on peut le voir, $|m(\omega)|$ est proche de 1 lorsque ω est proche de zéro, et tend vers 0 lorsque $\omega \to \pm\pi$. Ce filtre joue donc bien un rôle de filtre passe bas, dans la mesure où il atténue les hautes fréquences, et préserve les fréquences proches de 0; Il est cependant loin d'un filtre idéal.

3.3. Filtres FIR, IIR, récursifs. Rappelons que les seuls filtres susceptibles d'être implémentés pratiquement (sauf à utiliser numériquement une transformation de Fourier, ce sur quoi on reviendra un peu plus loin) sont des filtres faisant intervenir un nombre fini d'opérations.

3.3.1. *Filtres FIR et IIR.* Dans le cas de filtres de convolution

$$(Tu)_n = \sum h_k u_{n-k} \ ,$$

le fait de se limiter à un nombre fini d'opérations impose que la réponse impulsionnelle n'ait qu'un nombre fini de coefficients h_n non nuls. On parle alors de *filtres FIR* (pour "finite impulse response").

Par opposition, les *filtres IIR* (pour "infinite impulse response").ont une infinité de coefficients h_n non nuls. Pour implémenter pratiquement de tels filtres, il faut a priori les approximer par des filtres FIR en remplaçant pas des zéros tous les coefficients plus petits qu'une certaine limite de précision fixée à l'avance.

$$(Tu)_n = \sum_{-\infty}^{\infty} h_k u_{n-k} \approx \sum_{k=n_1}^{n_2} h_k u_{n-k} \ .$$

Il peut cependant arriver que le nombre de coefficients à retenir pour atteindre la précision voulue soit excessivement grand, et conduise à des coûts prohibitifs en temps de calcul. C'est en particulier le cas des filtres idéaux mentionnés plus haut. En effet, le fait que les coefficients décroissent comme $1/n$ implique que le nombre de coefficients à retenir pour atteindre une précision de l'ordre de 10^{-6} est de l'ordre du million. Donc, pour chaque n, le calcul de $(Tu)_n$ demande environ 1 million d'opérations. SI comme dans le cas de signaux de parole, le nombre de valeurs de $(Tu)_n$ est de l'ordre de 10 000 par seconde, on aboutit à 10 milliards d'opérations par seconde, ce qui est une charge trop lourde pour ce type d'application. On doit alors se rabattre à l'alternative que constituent les filtres récursifs.

3.3.2. *Les filtres récursifs.* Les filtres récursifs permettent d'implémenter des filtres IIR, en n'effectuant qu'un nombre fini (et limité) d'opérations. Etant donné un signal d'entrée u, l'idée est d'en déduire un signal de sortie v de la forme

$$v_n = \sum_{k=0}^{M} b_k u_{n-k} - \sum_{k=1}^{N} a_k v_{n-k} \ ,$$

c'est à dire tel que v_n soit obtenu par filtrage des u_m antérieurs à n et filtrage des v_m antérieurs à n, utilisant des filtres $\{b_0, \dots b_M\}$ et $\{a_1, \dots a_N\}$. Si cette équation admet une solution unique, alors on dit que la transformation linéaire $T : u \to v$ est un filtre récursif.

Pour étudier l'existence de solutions, posons pour simplifier $a_0 = 1$. L'équation ci-dessus est équivalent au système

$$\sum_{k=0}^{N} a_k v_{n-k} = \sum_{k=0}^{M} b_k u_{n-k} \ ,$$

qui après transformation de Fourier discrète devient

$$\hat{v}(\omega) \left(\sum_{k=0}^{N} a_k e^{-ik\omega} \right) = \hat{u}(\omega) \left(\sum_{k=0}^{M} b_k e^{-ik\omega} \right) \ .$$

La solution s'écrit (dans l'espace des fréquences)

$$\hat{v}(\omega) = m(\omega) \hat{u}(\omega) \ ,$$

avec

$$m(\omega) = \frac{\sum_{k=0}^{M} b_k e^{-ik\omega}}{\sum_{k=0}^{N} a_k e^{-ik\omega}} ,$$

et est bien définie dès que le dénominateur de $m(\omega)$ ne s'annule jamais (on reviendra sur cette condition un peu plus loin). Si tel est le cas, la solution s'écrit

$$v_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{in\omega} \frac{\sum_{k=0}^{M} b_k e^{-ik\omega}}{\sum_{k=0}^{N} a_k e^{-ik\omega}} \hat{u}(\omega)\, d\omega .$$

Dans l'immense majorité des cas, le filtre correspondant est un filtre IIR, qui est donc bien mis en oeuvre sous une forme ne faisant intervenir qu'un nombre fini d'opérations.

On a vu que l'existence d'un filtre récursif associé à des coefficients a_k et b_k dépend essentiellement de l'annulation possible du dénominateur de la fonction de transfert m. Ce dernier est un polynôme trigonométrique, que l'on peut mettre sous la forme

$$\sum_{k=0}^{N} a_k e^{-ik\omega} = \sum_{k=0}^{N} a_k z^{-k} ,$$

en introduisant la variable complexe $z = e^{i\omega}$. Il s'agit donc d'un polynôme de degré N en z^{-1}, qui peut donc être factorisé. On note $z_1, z_2, \dots z_N$ les valeurs (complexes) de z telles que ce polynôme s'annulle. Le résultat suivant résume les relations entre ces racines et les propriétés du filtre correspondant:

THÉORÈME:

1. La fonction de transfert $\omega \to m(\omega)$ est bornée si et seulement si aucune des racines $z_1, \dots z_N$ n'est de module égal à 1. Dans ce cas, le filtre récursif correspondant est bien défini.
2. Le filtre récursif est causal si et seulement si toutes les racines ont un module inférieur à 1:

$$|z_k| < 1 , \quad k = 1, 2, \dots N .$$

Ces remarques permettent de formuler le problème de synthèse de filtres, un problème similaire au problème rencontré dans le cas analogique. Etant donnée une fonction de transfert m telle que le filtre associé n'est pas causal, comment construire un filtre causal, tel que le module carré de sa fonction de transfert

$$\frac{\sum_{k=0}^{M} b_k e^{-ik\omega}}{\sum_{k=0}^{N} a_k e^{-ik\omega}} ,$$

soit une bonne approximation de $|m|^2$? Ce problème a reçu un certain nombre de réponses pratiques.

3.4. Signaux numériques en pratique: la FFT. En pratique, on ne manipule que des signaux de longueur finie. Les opérations de filtrage doivent être modifiées pour tenir compte de ce point. Il existe deux façons d'effectuer une telle adaptation:

- On peut considérer un signal de longueur finie comme un signal de longueur infinie ne possédant qu'un nombre fini de coefficients non nuls. Les opérations de filtrage peuvent alors être adaptées à ce point de vue.
- L'alternative consiste à considérer le signal fini comme un signal infini périodique, que l'on ne considère qu'à l'intérieur d'un domaine fini.

La transformation de Fourier discrète possède elle aussi une version "finie". Etant donnée une suite finie $u = \{u_n, n = 0 \dots N-1\}$ on lui associe la suite $\hat{u} = \{\hat{u}_k, k = 0, \dots N-1\}$, définie par

$$\hat{u}_k = \sum_{n=0}^{N-1} u_n e^{-2i\pi \frac{kn}{N}} \ .$$

Il est alors facile de montrer des propriétés analogues aux propriétés que nous avons déjà vues: formule de Parseval et inversion. De fait, on a

> THÉORÈME: Toute suite finie $u = \{u_n, n = 0 \dots N-1\}$ peut être développée comme superposition de fonctions exponentielles:
>
> $$u_n = \frac{1}{N} \sum_{k=0}^{N-1} \hat{u}_k e^{2i\pi \frac{kn}{N}} \ ,$$
>
> De plus, la formule de Parseval s'écrit
>
> $$\sum_{k=0}^{N-1} |\hat{u}_k|^2 = N \sum_{n=0}^{N-1} |u_n|^2 \ ,$$

La transformation de Fourier finie telle qu'elle est décrite ci-dessus est la seule version adaptée au calcul numérique[1] Il est important de savoir qu'il existe un algorithme rapide de calcul de la transformation de Fourier, généralement appelé FFT. La FFT permet d'implémenter une transformation de Fourier de longueur N avec un nombre d'opérations de l'ordre de $N \log N$ au lieu des N^2 opérations que nécessite une implémentation "standard".

4. Echantillonnage

Le problème d'échantillonnage consiste à associer à un signal analogique un signal numérique, en contrôlant la perte d'information. La solution la plus simple revient à considérer des valeurs ponctuelles $f(kT)$, régulièrement espacées, du signal f étudié. Cependant, ceci ne peut se faire sans précautions; il faut tout d'abord que les valeurs ponctuelles aient un sens, donc que f soit continue. Puis pour limiter la perte d'information, il faut que f varie "suffisamment lentement". Ceci conduit à poser le problème dans un cadre bien adapté. On se limitera ici au cadre de la théorie de l'échantillonnage classique, dans le cas des fonctions à bande limitée.

Le cadre naturel du théorème d'échantillonnage est l'espace des signaux à bande limitée, ou espace de Paley-Wiener. On dit qu'un signal analogique $f \in L^2(\mathbb{R})$ est

1. La transformation de Fourier intégrale pour les signaux analogiques peut quant à elle être mise en oeuvre via des dispositifs optiques.

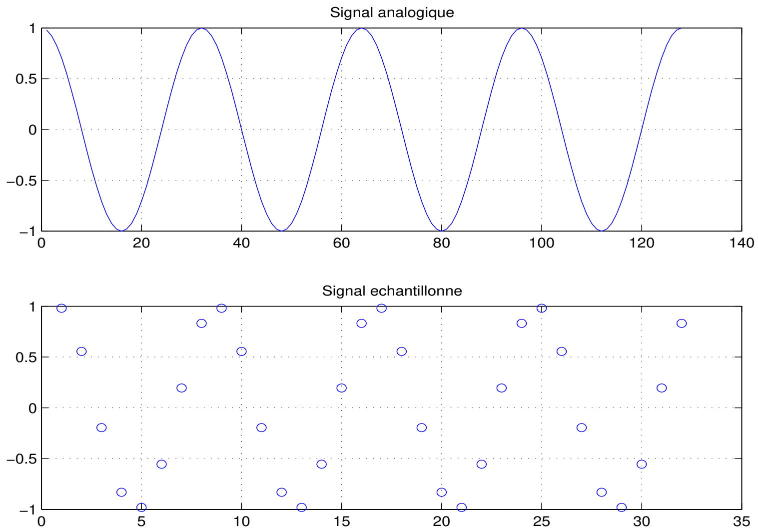

FIG. 12. *Exemple d'échantillonnage dans le cas où le signal varie suffisamment lentement pour que l'échantillonnage ne se traduise pas par une perte d'information.*

à bande limitée à l'intervalle $[-\omega_0, \omega_0]$ si sa transformée de Fourier est nulle à l'extérieur de cet intervalle.

$$BL_{\omega_0} = \left\{ f \in L^2(\mathbb{R}), \hat{f}(\omega) = 0 \text{ pour tout } \omega \notin [-\omega_0, \omega_0] \right\}$$

On peut montrer que BL_{ω_0} est un espace de fonctions continues, de sorte que pour tout signal à bande limitée, le nombre $f(t)$ est défini sans ambiguïté pour tout t.

On peut alors introduire l'opération d'échantillonnage E, associé à la frèquence d'échantillonnage η: si $f \in BL_{\omega_0}$, on note

$$(Ef)_n = f\left(\frac{n}{\eta}\right) \quad , \qquad n \in \mathbb{Z} \ .$$

La question est alors: sous quelle condition l'opération d'échantillonnage peut elle être effectuée sans perte d'information? Il est clair que la réponse doit tenir compte de la "vitesse" à laquelle varie le signal.

On peut voir sur la FIGURE 12 un exemple de signal assez lentement variable pour que l'échantillonnage ne se traduise pas par une perte d'information: le signal échantillonné "suit" les variations du signal original.

Par contre, dans le cas montré en FIGURE 13, les variations du signal original sont beaucoup plus rapide, et on peut voir que l'échantillonnage n'est pas suffisamment fin pour reproduire les variations les plus rapides.

Le théorème d'échantillonnage (dû à Shannon, Nyquist, Whittaker, Kotel'nikov et probablement d'autres) permet de donner un sens précis à ce problème,

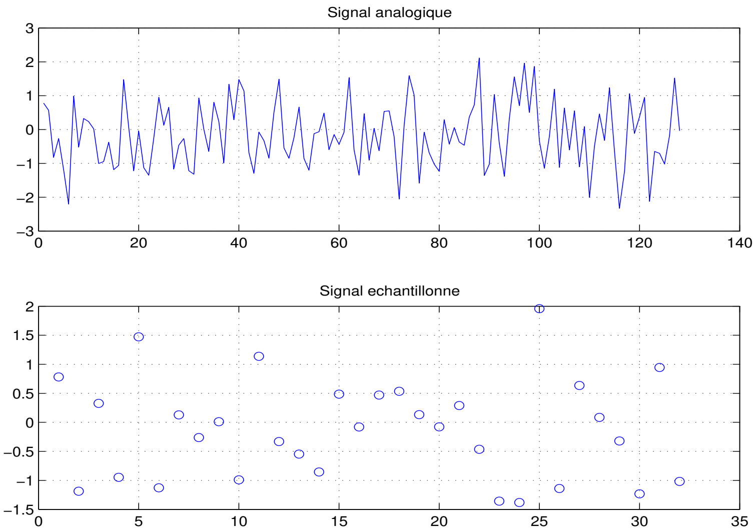

FIG. 13. *Exemple d'échantillonnage dans le cas où le signal varie suffisamment lentement pour que l'échantillonnage ne se traduise pas par une perte d'information.*

en spécifiant les rôles respectifs de la fréquence de coupure ω_0 et de la fréquence d'échantillonnage η.

THÉORÈME: Soit $f \in BL_{\omega_0}$ un signal analogique à bande limitée, et soit $\varphi = \{\varphi_n, n \in \mathbb{Z}\}$ le signal numérique obtenu par échantillonnage, à la fréquence d'échantillonnage η:

$$\varphi_n = f\left(\frac{n}{\eta}\right) \ .$$

1. Si $\eta < \omega_0/\pi$, il est impossible de retrouver f à partir de φ sans hypothèse supplémentaire.
2. Si $\eta = \omega_0/\pi$, le signal analogique peut être reconstruire à partir du signal échantillonné, via la relation

$$f(t) = \sum_{n=-\infty}^{\infty} f\left(\frac{n}{\eta}\right) \frac{\sin(\pi\eta(t - n/\eta))}{\pi\eta(t - n/\eta)} \ .$$

3. Si $\eta > \omega_0/\pi$, la formule de reconstruction précédente reste valable, mais n'est plus unique. Si γ est n'importe quelle fonction telle que

$$\hat{\gamma}(\omega) = \begin{cases} 1 & \text{si } \omega \in [-\omega_0, \omega_0] \\ 0 & \text{si } \omega \notin [-\pi\eta, \pi\eta] \end{cases}$$

on peut aussi reconstruire le signal original à partir du signal échantillonné, via la relation

$$f(t) = \sum_{n=-\infty}^{\infty} \frac{1}{\eta} f\left(\frac{n}{\eta}\right) \gamma\left(t - \frac{n}{\eta}\right) \ .$$

La fréquence critique qui définit la limite entre un "bon" et un "mauvais" échantillonnage

$$\eta_c = \frac{\omega_0}{\pi}$$

est appelée *fréquence de Nyquist.*

Sans entrer dans les détails de la démonstration, il est utile d'analyser les origines de ce résultat, et de comprendre quelles sont les erreurs qui s'introduisent lorsque l'échantillonnage n'est pas effectué correctement.

Tout d'abord, il est possible de montrer que la TFD du signal numérique φ est donnée par

$$\hat{\varphi}(\omega) = \eta \sum_{k=-\infty}^{\infty} \hat{f}(\eta(\omega + 2\pi k))$$

(remarquer que $\hat{f}$ est une transformée de Fourier intégrale, alors que $\hat{\varphi}$ est une transformée de Fourier discrète -donc périodique), c'est à dire une version "périodisée" de $\omega \to \eta\hat{f}(\eta\omega)$

Plaçons nous tout d'abord dans le troisième cas de figure. Dans cette situation, les différents "morceaux" de la somme infinie ci-dessus ont des supports disjoints. On voit alors qu'il suffit de multiplier $\hat{\varphi}$ par une fonction $\hat{\gamma}$ vérifiant la propriété indiquée dans l'énoncé du théorème pour retrouver la fonction $\omega \to \eta\hat{f}(\eta\omega)$, et de là la fonction f (par transformation de Fourier inverse. C'est ainsi que l'on retrouve la formule de reconstruction.

Dans le second cas de figure ($\omega_0 = \pi\eta$), le raisonnement est similaire, à ceci près que la seule fonction γ qui convienne est la fonction γ "critique" telle que

$$\hat{\gamma}_c(\omega) = \begin{cases} 1 & \text{si } \omega \in [-\omega_0, \omega_0] \\ 0 & \text{sinon ,} \end{cases}$$

ce qui fournit la seule formule d'inversion de l'échantillonnage qui fonctionne dans ce cas critique.

Finalement, dans le premier cas (non favorable), les supports des différents "morceaux" de la somme infinie

$$\sum_{k=-\infty}^{\infty} \hat{f}(\eta(\omega + 2\pi k))$$

se superposent, de sorte qu'il devient impossible de les séparer. C'est ce que l'on appelle le phénomène de "repliement de spectre" (aliasing en anglais).

Un exemple de repliement de spectre est montré en FIGURE 14, dans le cas d'un simple signal sinusoïdal. Dans ce cas, le sous-échantillonnage est tel que le pic visible dans le spectre du signal initial (aux fréquences ±175Hz) se trouve à l'extérieur de la bande de fréquences utile $[-\omega_0, \omega_0]$ du signal échantillonné. Le spectre du signal sous-échantillonné n'est pas seulement limité à un domaine de fréquences plus petit,

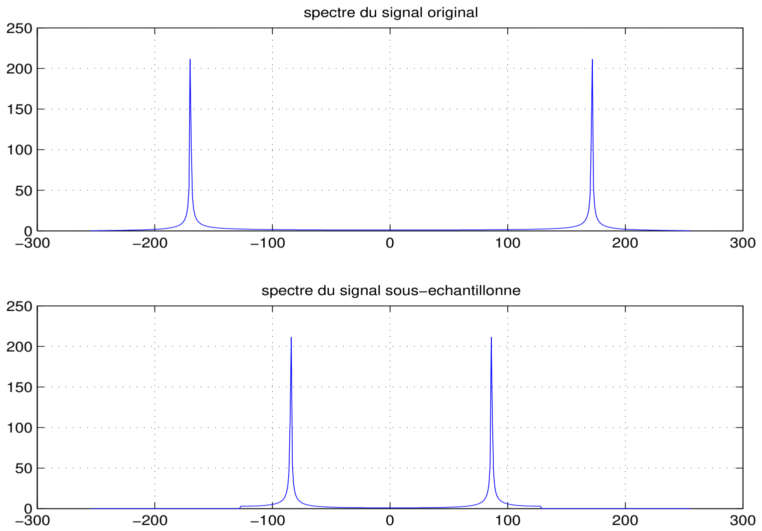

FIG. 14. *Exemple de repliement de spectre induit par un mauvais échantillonnage.*

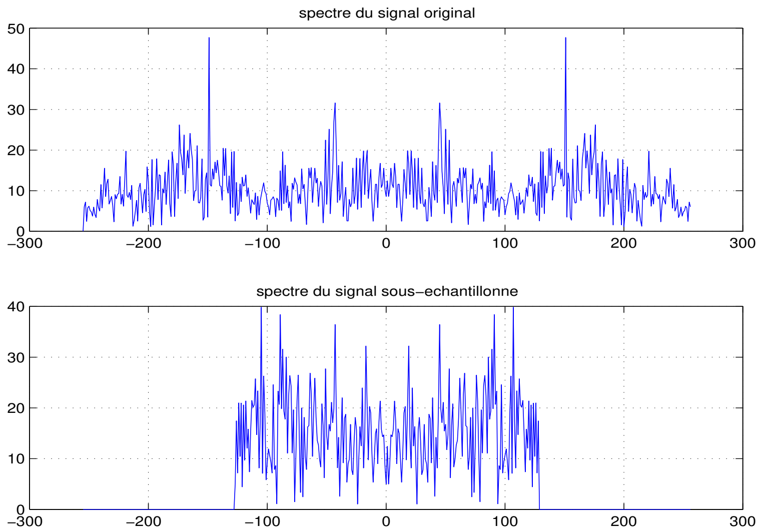

FIG. 15. *Exemple de repliement de spectre induit par un mauvais échantillonnage.*

mais aussi très perturbé à l'intérieur de son support: le pic réapparait aux alentours des fréquences ± 80Hz.

Un exemple similaire sur un signal réel se trouve en FIGURE 15. Là encore, on peut en particulier noter le pic présent aux alentours de 150Hz dans le spectre du signal initial, qui réapparait "replié" vers -100Hz.

CHAPITRE 2

Compression et codage de source

On décrit dans ce chapitre un certain nombre d'exemples de codeurs de signaux couramment utilisés dans divers "standards". On appellera *codeur* (ou codeur de source) un système permettant de générer un signal binaire partant d'un signal analogique. Le *décodeur* effectue le traitement inverse. L'ensemble "codeur + décodeur" est généralement appelé CoDec. Le prototype de ces codeurs est le codeur PCM (Pulse Code Modulation). Ce dernier est essentiellement constitué de plusieurs éléments, mis bout à bout:

1. Un filtre passe bas, aussi proche que possible d'un filtre idéal, dont le but est de limiter le contenu fréquentiel des signaux à une bande de fréquences fixée $[-\omega_0, \omega_0]$.
2. Un échantillonneur, c'est à dire un système associant à un signal analogique $f \in L^2(\mathbb{R})$ un signal numérique $\varphi = \{\varphi_n\}$, où $\varphi_n = f(n/\eta)$, la fréquence d'échantillonnage η étant bien évidemment prise inférieure à la fréquence critique ω_0/π.
3. Un quantificateur, qui associe à tout échantillon φ_n (un nombre réel quelconque) une valeur "quantifiée" $Q(\varphi_n)$, c'est à dire limitée à un ensemble fini de valeurs possibles $y_0, y_1, \dots y_{M-1}$.
4. Le codeur proprement dit, qui associe à chaque valeur quantifiée y_m un code binaire. L'exemple le plus simple est le code de longueur constante, qui "numérote" en binaire les syymboles $y_0, y_1, \dots y_{M-1}$.

1. Eléments de probabilités

La compression des signaux implique une étape de quantification, pour laquelle des notions élémentaires de probabilités sont nécessaires. En effet, les probabilités sont utilisées ici pour mod éliser la variabilité des signaux considérés. En particulier, un échantillon sera modélisé comme un nombre aléatoire auquel on sait associer une distribution de probabilités.

Sans entrer dans les détails (et en particulier, sans décrire le modèle probabiliste de Kolmogorov), on donne ici les éléments "minimaux" nécessaires notamment pour la formalisation de la quantification.

La notion essentielle pour ce qui nous concernera est la notion de variable aléatoire. Une variable aléatoire est un objet, que l'on notera X, qui peut prendre un certain nombre de valeurs possibles $x_1, x_2, \dots$, avec probabilités $p_1, p_2, \dots$ (dans le cas de variables aléatoires discrètes; on verra plus loin le cas des variables aléatoires continues). Une variable aléatoire peut être vue comme une fonction définie sur un espace de "hasard", à valeurs réelles. La théorie des probabilités associée permet de décrire le comportement de quantités qui dépendent de X.

On distinguera deux classes de variables aléatoires:

- les variables aléatoires discrètes, qui sont telles que leurs valeurs possibles forment un ensemble fini, ou infini dénombrable
- les variables aléatoires discrètes, qui sont telles que leurs valeurs possibles forment un ensemble infini continu.

Dans ce contexte, un quantificateur est une transformation (non linéaire) Q associant à une variable aléatoire continue X une variable aléatoire $Y = Q(X)$ prenant ses valeurs dans un ensemble fini $y_0, y_1, \dots y_{M-1}$.

1.1. Variables aléatoires discrètes. Comme on l'a vu, une variable aléatoires discrète prend un nombre fini, ou infini dénombrable, de valeurs possibles. On notera $x_1, x_2, \dots$ ces valeurs, et on notera $\mathbb{P}\{X = x_i\}$ la probabilité que X prenne la valeur x_i. Les nombres $p_i = \mathbb{P}\{X = x_i\}$ possèdent deux propriétés essentielles:

$$\mathbb{P}\{X = x_i\} \in [0,1]\ \forall i\ , \quad \text{et} \quad \sum_i \mathbb{P}\{X = x_i\} = 1\ . \tag{1}$$

Les valeurs possibles x_i et leurs probabilités $\mathbb{P}\{X = x_i\}$ déterminent la *distribution de probabilités* de X. De là, on peut par exemple calculer

$$\mathbb{P}\{X \in \{x_1, x_2\}\} = \mathbb{P}\{X = x_1\} + \mathbb{P}\{X = x_2\}\ ,$$

et plus généralement, étant donnés deux sous ensembles $A, B \subset \{x_1, x_2, \dots\}$ de l'ensemble des valeurs possibles, on a

$$\mathbb{P}\{X \in A \cup B\} = \mathbb{P}\{X \in A\} + \mathbb{P}\{X \in B\}$$

si A et B sont disjoints, et

$$\mathbb{P}\{X \in A \cup B\} = \mathbb{P}\{X \in A\} + \mathbb{P}\{X \in B\} - \mathbb{P}\{X \in A \cap B\} \tag{2}$$

dans le cas général.

Exemples:

1. *Variable aléatoire discrète uniforme:* l'ensemble des valeurs possibles est un ensemble fini $x_1, x_2, \dots x_M$, et ces valeurs ont toutes même probabilité
$$\mathbb{P}\{X = x_i\} = \frac{1}{M}\ .$$
2. *Variable aléatoire binômiale:* une variable aléatoire binômiale correspond à la situation suivante: une expérience, à deux résultats possibles (disons, *gagné* et *perdu*) est effectuée N fois. On suppose que les expériences sont indépendantes, et que les probabilités p et $q = 1 - p$ des deux résultats sont constantes. La variable aléatoire X décrit le nombre de fois où le résultat est *gagné*. L'ensemble des valeurs possibles est $X \in \{0, 1, 2, 3, \dots N\}$, et les probabilités sont données par
$$\mathbb{P}\{X = n\} = \frac{N!}{n!\,(N-n)!}\, p^n (1-p)^{N-n}\ .$$
Le fait que $\sum_{i=0}^N \mathbb{P}\{X = x_i\} = 1$ résulte de la formule du binôme de Newton ($\sum_0^N \frac{N!}{n!\,(N-n)!}\, p^n (1-p)^{N-n} = (p+q)^N = 1$).
3. *Variable aléatoire géométrique (loi de Pascal):* une variable aléatoire géométrique correspond à la situation suivante: une expérience, à deux résultats possibles (disons, *gagné* et *perdu* est effectuée autant de fois que nécessaire pour que le résultat *gagné* apparaisse. On suppose que les expériences sont indépendantes, et que les probabilités p et $q = 1 - p$ des deux résultats sont constantes. La variable aléatoire X décrit le nombre d'essais. L'ensemble des

valeurs possibles $X \in \{1, 2, 3, \dots\}$ est infini, et les probabilités sont données par

$$\mathbb{P}\{X = n\} = p(1-p)^{n-1} .$$

Le fait que $\sum_i \mathbb{P}\{X = x_i\} = 1$ résulte du résultat classique sur la somme de la série géométrique ($\sum_0^\infty a^n = 1/(1-a)$ si $|a| < 1$).

Une notion essentielle est la notion de moyenne probabiliste. On appelle *espérance mathématique* de la variable aléatoire discrète X la quantité

$$\mathbb{E}\{X\} = \sum_i x_i \, \mathbb{P}\{X = x_i\} . \tag{3}$$

On calcule de même l'espérance mathématique de variables aléatoires Y fonction de X par exemple

$$\mathbb{E}\{X^2\} = \sum_i x_i^2 \, \mathbb{P}\{X = x_i\} , \tag{4}$$

ou plus généralement, étant donnée une variable aléatoire $Y = \varphi(X)$

$$\mathbb{E}\{\varphi(X)\} = \sum_i \varphi(x_i) \, \mathbb{P}\{X = x_i\} . \tag{5}$$

En particulier, les quantités suivantes sont utiles: la variance

$$\mathrm{Var}\{X\} = \mathbb{E}\{(X - \mathbb{E}\{X\})^2\} = \mathbb{E}\{X^2\} - \mathbb{E}\{X\}^2 , \tag{6}$$

et l'écart-type

$$\sigma_X = \sqrt{\mathrm{Var}\{X\}} . \tag{7}$$

On utilisera également la notion d'*entropie* de Shannon

$$H(X) = \sum_i -\mathbb{P}\{X = x_i\} \log_2(\mathbb{P}\{X = x_i\}) , \tag{8}$$

qui permet de caractériser la quantité d'information portée par la variable aléatoire X.

1.2. Variables aléatoires continues. Dans le cas où la variable aléatoire X est susceptible de prendre n'importe quelle valeur dans un ensemble infini continu, les nombres $\mathbb{P}\{X = x\}$ n'ont plus de sens, et on doit recourir à une quantité intermédiaire pour caractériser la distribution de probabilités de X: la *densité de probabilités*: une fonction $x \to \rho_X(x)$, telle que

$$\rho_X(x) \geq 0 \ \ \forall x , \quad \text{et} \quad \int_{-\infty}^{\infty} \rho_X(x) \, dx = 1 . \tag{9}$$

On déduit de la densité de probabilités les valeurs

$$\mathbb{P}\{X \in [a, b]\} = \int_a^b \rho_X(x) \, dx . \tag{10}$$

La distribution de probabilités ainsi construire possède les mêmes propriétés que dans le cas discret. En particulier, la propriété (2) est toujours vérifiée.

Exemples:

1. *Variable aléatoire continue uniforme:* comme dans le cas discret, les valeurs possibles d'une variable aléatoire continue uniforme sont équiprobables. Une variable aléatoire continue uniforme, prenant ses valeurs dans un intervalle $[a, b]$ est ainsi caractérisée par une densité de probabilités de la forme

$$\rho_X(x) = \begin{cases} 1/(b-a) & \text{si } x \in [a,b] \\ 0 & \text{sinon.} \end{cases}$$

   On a donc, pour $[u, v] \subset [a, b]$

$$\mathbb{P}\{X \in [u,v]\} = \frac{v-u}{b-a} \ .$$

2. *Variable aléatoire Gaussienne:* une variable aléatoire Gaussienne est caractérisée par une densité de la forme

$$\rho_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \ ,$$

   où μ et σ sont deux nombres, représentant respectivement la moyenne et lécart type de X.
3. *Variable aléatoire Laplacienne:* une variable aléatoire Laplacienne est caractérisée par une densité de la forme

$$\rho_X(x) = \frac{\lambda}{2} \exp\{-\lambda|x-\mu|\} \ .$$

Les notions d'espérance mathématique vues plus haut dans le cas discret s'étendent au cas continu, essentiellement en remplaçant les sommes par des intégrales:

$$\mathbb{E}\{X\} = \int_{-\infty}^{\infty} x\, \rho_X(x)\, dx \ . \tag{11}$$

De même, en ce qui concernes des moyennes de fonctions de la variable aléatoire X

$$\mathbb{E}\{X^2\} = \int_{-\infty}^{\infty} x^2\, \rho_X(x)\, dx \ . \tag{12}$$

et

$$\mathbb{E}\{\varphi(X)\} = \int_{-\infty}^{\infty} \varphi(x)\, \rho_X(x)\, dx \ . \tag{13}$$

La variance et l'écart-type sont quant à eux toujours définis comme en (6) et (7).

La notion d'entropie (appelée dans ce cas *entropie différentielle*) se généralise au cas continu par

$$H(X) = -\int_{-\infty}^{\infty} \rho_X(x)\, \log_2(\rho_X(x))\, dx \ . \tag{14}$$

1.3. Plusieurs variables aléatoires. Considérons maintenant le cas, un peu plus complexe, où plusieurs variables aléatoires sont étudiées simultanément. Dans ce cas, le comportement d'ensemble de cette famille de variables aléatoires n'est généralement pas déterminé par les comportements individuels de chacune des variables aléatoires, à cause de possibles relations existant entre elles. Il est alors nécessaire d'introduire la notin de distribution de probabilités jointe de l'ensemble de variables aléatoires.

1.3.1. *Variables aléatoires discrètes.* Considérons tout d'abord le cas de paires de variables aléatoires. Si X et Y sont deux variables aléatoires, que l'on suppose discrètes dans un premier temps, leur *distribution de probabilités jointe* est caractérisée par les nombres de la forme

$$\mathbb{P}\left\{X = x_i \text{ et } Y = y_j\right\} = \mathbb{P}\left\{X = x_i, Y = y_j\right\} \ .$$

Les distributions de probabilités de X et Y, appelées distributions marginales, sont alors obtenues via

$$\mathbb{P}\left\{X = x_i\right\} = \sum_j \mathbb{P}\left\{X = x_i, Y = y_j\right\} \ ,$$

et une expression similaire pour $\mathbb{P}\left\{Y = y_j\right\}$.

Lorsque l'égalité

$$\mathbb{P}\left\{X = x_i, Y = y_j\right\} = \mathbb{P}\left\{X = x_i\right\}\mathbb{P}\left\{Y = y_j\right\}$$

est vérifiée pour tout x_i et y_j, X et Y sont dites *indépendantes*. Mais cette situation n'est pas le cas général, bien au contraire.

Une mesure simple de dépendance est donnée par les notions de *covariance* et de corrélation. Etant données deux variables aléatoires X et Y, on note

$$R_{XY} = \mathbb{E}\{XY\} \tag{15}$$

leur corrélation, et

$$C_{XY} = \mathbb{E}\{(X - \mathbb{E}\{X\})(Y - \mathbb{E}\{Y\})\} = R_{XY} - \mathbb{E}\{X\}\mathbb{E}\{Y\} \tag{16}$$

leur covariance. Si la covariance de X et Y est positive, ceci signifie que X et Y ont tendance à varier dans le même sens: $X - \mathbb{E}\{X\}$ et $Y - \mathbb{E}\{Y\}$ sont "en moyenne" de même signe. Inversement, si C_{XY} est négatif, $X - \mathbb{E}\{X\}$ et $Y - \mathbb{E}\{Y\}$ sont "en moyenne" de signe opposé. Par contre, la valeur absolue de la covariance C_{XY} est généralement difficile à interpréter (car elle dépend des valeurs absolues de X et de Y), et on utilise souvent le *coefficient de corrélation de Pearson*

$$r(X,Y) = \frac{C_{XY}}{\sigma_X \sigma_Y} \ . \tag{17}$$

THÉORÈME: *Le coefficient de corrélation est tel que*

$$-1 \leq r(X,Y) \leq 1 \ .$$

Si $r(X,Y) = 1$, il existe une constante positive a et une constante b telles que $Y = aX + b$. Inversement, si $r(X,Y) = -1$, il existe une constante négative a et une constante b telles que $Y = aX + b$.

Dans un certain sens, le coefficient $r(X,Y)$ fournit une mesure de l'écart moyen à une dépendance linéaire $Y = aX + b$. Lorsque $r(X,Y) = 0$, on dit que X et Y sont *décorrélées*. Notons que si X et Y sont indépendantes, elles sont automatiquement décorrélées. Par contre, la réciproque est fausse dans le cas général: la décorrélation n'implique pas l'indépendance. Lorsque l'on dispose de plusieurs variables aléatoires discrètes $X_1, X_2, \dots X_N$, leur comportement est caractérisé par leur distribution de probabilités jointe, c'est à dire l'ensemble des nombres

$$\mathbb{P}\{X_1 = x_1, X_2 = x_2, \dots X_N = x_N\} ,$$

desquels on pout toujours déduire les distributions marginales des X_n: par exemple

$$\mathbb{P}\{X_1 = x_1\} = \sum_{x_2}\sum_{x_3}\cdots\sum_{x_N} \mathbb{P}\{X_1 = x_1, X_2 = x_2, \dots X_N = x_N\} ,$$

et des expressions similaires pour les autres distributions marginales.

Une caractérisation exhaustive des relations de dépendance existant dans une famille de variables aléatoires est souvent difficile, et on se contente généralement d'étudier les relations de dépendance deux à deux. C'est ainsi que l'on peut introduire la *matrice de covariance* de la famille de variables aléatoires considérée: il s'agit de la matrice C, dont les éléments de matrice sont donnés par

$$C_{ij} = C_{X_iX_j} = \mathbb{E}\{X_iX_j\} - \mathbb{E}\{X_i\}\mathbb{E}\{X_j\} . \tag{18}$$

On peut voir immédiatement que la matrice de covariance est une matrice symétrique ($C_{ij} = C_{ji}$). Elle est donc diagonalisable, ce qui a d'intéressantes conséquences pratiques.

1.3.2. *Variables aléatoires continues.* Les variables aléatoires continues se prêtent à une analyse similaire. Etant données deux variables aléatoires continues X et Y, de densités de probabilités respectives ρ_X et ρ_Y, leur distribution de probabilités jointe est caractérisée par une fonction de deux variables, appelée *densité de probabilités jointe* et notée ρ_{XY}, telle que pour tout ensemble mesurable $U \subset \mathbb{R}^2$, on ait

$$\mathbb{P}\{(X,Y) \in U\} = \int\int_U \rho_{XY}(x,y)\, dx\, dy . \tag{19}$$

Comme dans le cas de variables aléatoires discrètes, les *densités marginales* ρ_X et ρ_Y s'obtiennent de façon simple à partir de la densité jointe: par exemple,

$$\rho_X(x) = \int_{-\infty}^{\infty} \rho_{XY}(x,y)\, dy ,$$

et de même pour ρ_Y. Notons qu'en général, $\rho_{XY}(x,y) \neq \rho_X(x)\rho_Y(y)$. Lorsque tel est le cas pour tout x, y, on dit que X et Y sont indépendantes.

Un exemple simple est fourni par la paire de variables aléatoires uniformément distribuées sur un disque de rayon r_0. La densité de probabilités jointe correspondante est de la forme

$$\rho_{XY}(x,y) = \begin{cases} 1/\pi r_0^2 & \text{si } \sqrt{x^2+y^2} \leq r_0 \\ 0 & \text{sinon} \end{cases}$$

Il est facile de voir que

$$\rho_{XY}(x,y) \neq \rho_X(x)\rho_Y(y) ,$$

de sorte que X et Y ne sont pas indépendantes.

Les notions vues plus haut dans le cas de variables aléatoires discrètes (corrélation, covariance...) s'étendent sans modification au cas de variables aléatoires continues.

1.4. Signaux numériques aléatoires. Les signaux numériques aléatoires représentent un cas particulier de familles (finies ou infinies) de variables aléatoires, dans lesquelles il existe un ordre (ou une chronologie) déterminé. On utilise généralement des modèles de signaux numériques aléatoires pour modéliser des classes de signaux échantillonnés. La notion de stationnarité que nous allons décrire est d'une grande importance dans ce contexte, dans la mesure où elle permet de décrire des signaux dont certaines caractéristiques statistiques sont invariantes dans le temps.

Considérons tout d'abord pour simplifier le cas de signaux numériques aléatoires infinis: soit $X = \{X_n, n \in \mathbb{Z}\}$ une collection de variables aléatoires, dont on suppose que $\mathbb{E}\{X_n\}$ et $\mathbb{E}\left\{X_n^2\right\}$ existent pour tout n.

DÉFINITION: *Un signal numérique aléatoire $X = \{X_n, n \in \mathbb{Z}\}$ est dis stationnaire en moyenne d'ordre deux si ses moments d'ordre un et deux sont invariants par translation:*

$$\mathbb{E}\{X_n\} = \mathbb{E}\{X_0\} \quad \forall n \in \mathbb{Z} \tag{20}$$

$$\mathbb{E}\{X_{n+\tau}X_{m+\tau}\} = \mathbb{E}\{X_n X_m\} \quad \forall m, n, \tau \in \mathbb{Z} \tag{21}$$

On peut tout de suite noter que si X est stationnaire en moyenne d'ordre deux, sa covariance ne dépend plus que d'un seul indice:

$$C_X(m,n) := C_{X_m X_n} = C_X(m-n, 0) := C_X(m-n) \ .$$

La suite C_X est appelée *suite d'autocovariance* (ou *suite de covariance* du signal X. Il est possible de démontrer qu'il existe une fonction positive ou nulle $\mathcal{E}$, appelée *spectre* de X (ou densité spectrale de X), telle que

$$C_X(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{E}(\omega) e^{in\omega}\, d\omega \ .$$

Cette dernière relation est appelée *théorème de Wiener-Khinchin*, ou *théorème de Wold.*

REMARQUE: Pour traiter de façon similaire le cas des signaux numériques de longueur finie, il est nécessaire d'apporter une petite modification à la définition pour donner un sens à la notion d'invariance par translation: les translations sont définies "modulo N. On dira qu'un signal numérique aléatoire de longueur finie $X = \{X_0, X_1, \dots X_{N-1}\}$ est stationnaire en moyenne d'ordre deux si

$$\begin{aligned} \mathbb{E}\{X_n\} &= \mathbb{E}\{X_0\} & \forall n = 0, 1, \dots N-1 \\ \mathbb{E}\left\{X_{(n+\tau)[\mathrm{mod}\ N]} X_{(m+\tau)[\mathrm{mod}\ N]}\right\} &= \mathbb{E}\{X_n X_m\} & \forall m, n, \tau = 0, 1, \dots N-1 \end{aligned}$$

Dans ce cas là, on a toujours

$$C_X(n,m) = C_X((n-m)[\mathrm{mod}\ N], 0) := C_X((n-m)[\mathrm{mod}\ N]) \ .$$

2. Quantification scalaire et PCM

PCM est le sigle désignant le *Pulse Code Modulation*, un standard (ou plutôt une famille de standards) adopté de façon assez universelle. Le système PCM est connu pour offrir des performances assez moyennes en termes de compression, mais aussi de robustesse du signal codé aux erreurs de transmission. Il présente toutefois l'avantage d'être simple, bien connu, et très peu coûteux en termes de complexité et mémoire. On décrit ici le PCM de façon assez sommaire, dans le but d'introduire quelques idées simples, notamment en ce qui concerne la quantification.

Le PCM consiste essentiellement en un échantillonneur, suivi d'un quantificateur (uniforme) appliqué aux échantillons, et enfin d'un système simple de codage. La phase d'échantillonnage a déjà été décrite plus haut. Le codage est basé sur le principe d'une attribution "démocratique" des bits: chaque valeur quantifiée sera codée sur un nombre de bits constant. On parle de code de longueur constante. On se concentre maintenant sur la quantification.

2.1. Quantification scalaire. Considérons donc des échantillons f_n d'un signal f. La base du PCM est de modéliser chacun de ces échantillons comme une variable aléatoire, sans se préoccuper des corrélations entre échantillons. Supposons que la variable aléatoire X soit une variable aléatoire continue, de densité de probabilités ρ_X.

On considère donc un quantificateur

$$Q : \mathbb{R} \to E_M = \{y_-, y_0, \dots y_{M-1}, y_+\} \ ,$$

et la variable aléatoire discrète $Y = Q(X)$, à valeurs dans l'ensemble fini E_M. Q est défini à partir d'intervalles de la forme $[x_k, x_{k+1}[$ par

$$Q(x) = \begin{cases} y_- & \text{si } x \le x_0 \\ y_k & \text{si } x \in [x_k, x_{k+1}[\ k = 0., \dots M-1 \\ y_+ & \text{si } x > x_M \end{cases} \tag{22}$$

La quantité qui nous intéresse est l'erreur de quantification, c'est à dire la variable aléatoire

$$Z = X - Y = X - Q(X) \ . \tag{23}$$

La quantité d'intérêt est ici la variance de l'erreur de quantification

$$\sigma_Z^2 = \mathbb{E}\{Z^2\} = \int (x - Q(x))^2 \rho_X(x) dx \tag{24}$$

$$= \sum_{k=0}^{M-1} \left(\int_{x_k}^{x_{k+1}} (x - y_k)^2 \rho_X(x) dx \right) + \int_{-\infty}^{x_0} (x - y_-)^2 \rho_X(x) dx + \int_{x_M}^{\infty} (x - y_+)^2 \rho_X(x) dx \ , \tag{25}$$

et c'est celle-ci que nous allons évaluer dans certaines situations simples. Les deux derniers termes 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_X a un support borné), le bruit de saturation est généralement évité.

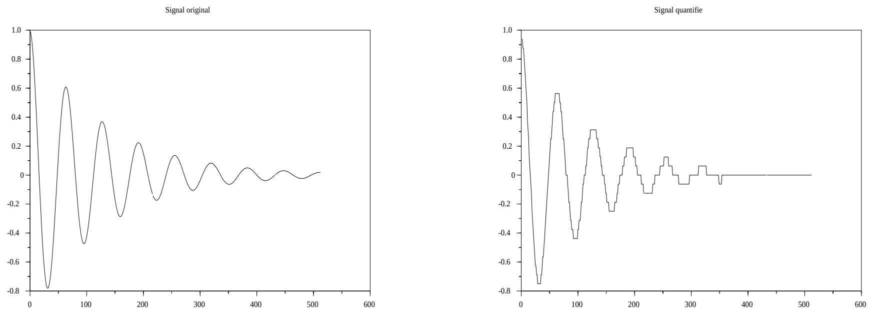

FIG. 1. *Exemple de quantification: un signal simple (à gauche) et le même signal après quantification de chaque échantillon sur 5 bits (32 niveaux de quantification).*

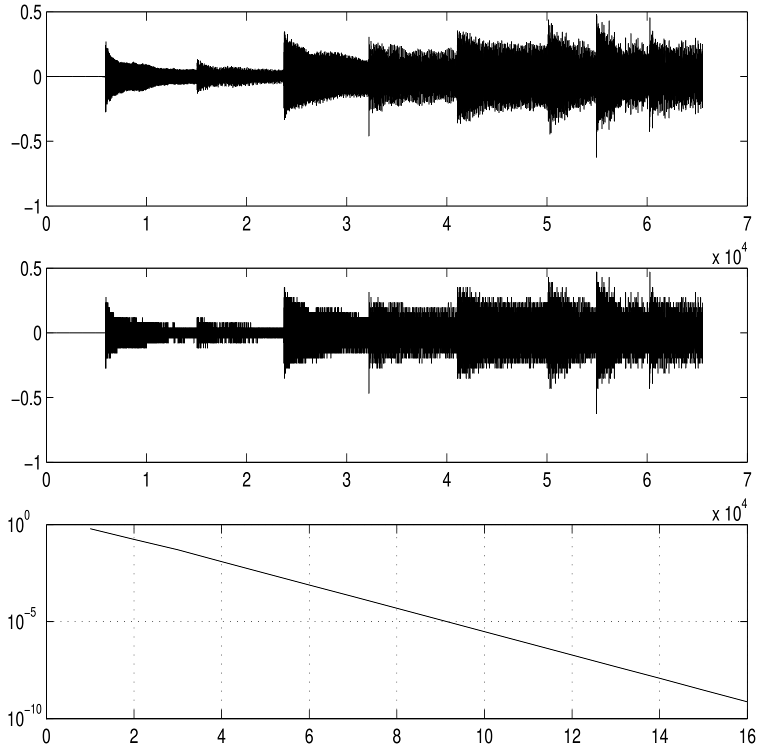

FIG. 2. *Quantification d'un signal audio: un signal "test" (le "carillon" test des codeurs MPEG audio, en haut), une version quantifiée sur 4 bits (milieu), et le logarithme* $\log_2(D)$ *de la distorsion en fonction du débit (en bas).*

Etant donné un quantificateur comme décrit ci-dessus, son *facteur de performance* est défini comme le quotient

$$\epsilon^2 = \frac{\sigma_X^2}{\sigma_Z^2} \ , \tag{26}$$

où σ_X^2 et σ_Z^2 sont les variances respectives du signal X et du bruit de quantification Z. Le *Rapport Signal à Bruit de Quantification* est quant à lui défini par

$$SNR_Q = 10\log_{10}(\epsilon^2) = 10\log_{10}\left(\frac{\sigma_X^2}{\sigma_Z^2}\right) \ . \tag{27}$$

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. On commencera par étudier le cas le plus simple, à savoir le cas de la quantification uniforme.

2.2. Quantification 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. Plus précisément, supposant que la densité ρ_X soit 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

$$x_0 = x_{min} \ , \quad x_m = x_0 + m\Delta \ , \tag{28}$$

avec

$$\Delta = |I|/M = |I|2^{-R} \ , \tag{29}$$

et

$$y_m = \frac{x_m + x_{m+1}}{2} \ . \tag{30}$$

Supposons que le quantificateur soit un quantificateur “haute résolution”, c'est à dire qu'à l'intérieur d'un intervalle $[x_k, x_{k+1}]$, la densité de probabilités $x \to \rho_X(x)$ soit lentement variable, et puisse être approximée par la valeur $\rho_k = \rho(y_k)$. On écrit alors

$$\int_{x_k}^{x_{k+1}} (x-y_k)^2 \rho_X(x)dx \approx \frac{\rho_k}{3}\left((x_{k+1}-y_k)^3 - (x_k-y_k)^3\right) \ . \tag{31}$$

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

$$\int_{x_k}^{x_{k+1}} (x-y_k)^2 \rho_X(x)dx \approx \rho_k \frac{\Delta^3}{12} \ . \tag{32}$$

Par ailleurs, on écrit également $1 = \int \rho_X(x)dx \approx \Delta \sum_k \rho_k$, d'où

$$\sum_k \rho_k \approx \frac{1}{\Delta} \ . \tag{33}$$

Finalement, en faisant le bilan, on aboutit à

$$\sigma_Z^2 \approx \frac{\Delta^3}{12} \sum_0^{M-1} \rho_k \approx \frac{\Delta^2}{12} \tag{34}$$

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

$$D = \sigma_Z^2 = C^{ste}\, 2^{-2R} \ .$$

Plus précisément, supposant que la densité de probabilités ρ_X est différentiable, on montre que

$$\sigma_Z^2 = \frac{\Delta^2}{12} + r \ , \tag{35}$$

avec

$$|r| \leq C^{ste}\, 2^{-3R} \sup_x |\rho_X'(x)| \ . \tag{36}$$

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

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

où la constante dépend de la loi de X. Ainsi, on aboutit à la règle empirique suivante (parfois appelée *règle des 6dB*):

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

Exemple: 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):

$$SNR_Q \approx 6,02\, R \ .$$

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

2.3. Compensation logarithmique. Une limitation de la quantification uniforme est que, tant que l'on reste dans le cadre de validité des approximations que nous avons faites, la variance σ_Z^2 du bruit de quantification (qui vaut $\Delta^2/12$) ne dépend pas de la variance du signal. Donc, le rapport signal à bruit de quantification décroît quand la variance du signal décroît. Or, celle-ci est souvent inconnue à l'avance, et peut aussi avoir tendance à varier (lentement) au cours du temps. Il peut donc être avantageux de "renforcer" les faibles valeurs du signal de façon "autoritaire". Pour ce faire, une technique classique consiste à effectuer sur les coefficients une transformation, généralement non-linéaire, afin de rendre la densité de probabilités plus "plate", plus prôche d'une densité de variable aléatoire uniforme. Il s'agit généralement d'une transformation logarithmique, modifiée à l'origine pour éviter la singularité.

La figure 3 reprend l'exemple de la figure 2, et montre l'effet de la compensation logarithmique sur la courbe débit-distorsion: la nouvelle courbe est toujours

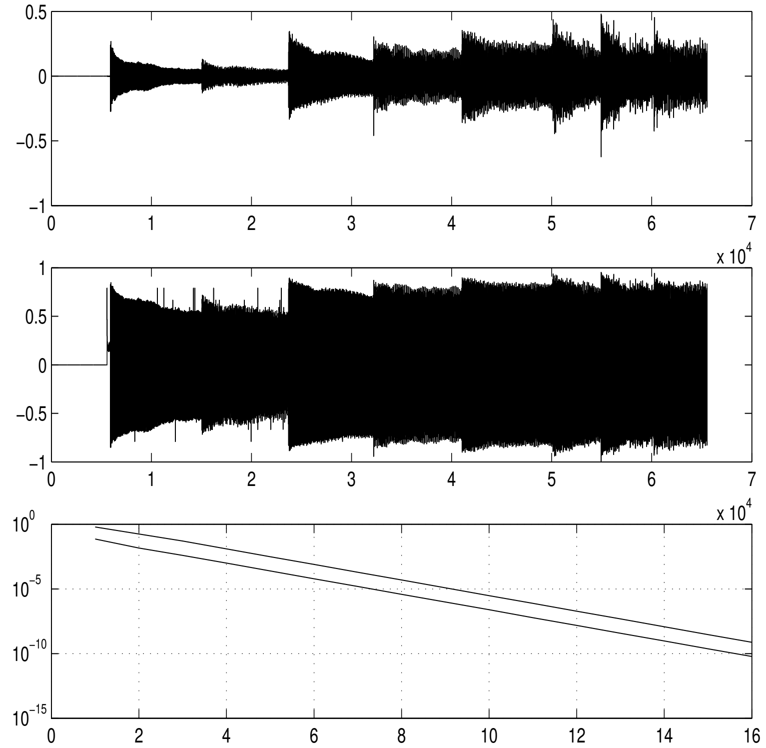

FIG. 3. *Quantification logarithmique d'un signal audio: le signal test "carillon" (en haut), une version corrigée par loi μ (milieu), et le logarithme* $\log_2(D)$ *de la distorsion en fonction du débit (en bas) pour le signal original et le signal compensé logarithmiquement (en bas).*

essentiellement linéaire, mais se situe en dessous de la précédente. La compensation logarithmique a donc bien amélioré les performances du quantificateur.

Deux exemples de telles transformations sont couramment utilisées, en téléphonie notamment: la *loi A*, qui correspond au standard européen, et la *loi* μ (standard nord-américain).

2.3.1. *La loi A.* Le premier exemple est la loi A, qui consiste en une modification linéaire pour les faibles valeurs du signal, et d'une compensation logarithmique pour les plus grandes valeurs. Plus précisément, la fonction de compensation est donnée par

$$c(x) = \begin{cases} \frac{A|x|}{1+\log A}\,\mathrm{sgn}(x) & \text{si } |x| \le \frac{x_{max}}{A} \\ x_{max}\,\frac{1+\log(A|x|/x_{max})}{1+\log A}\,\mathrm{sgn}(x) & \text{si } |x| > \frac{x_{max}}{A} \end{cases} \tag{37}$$

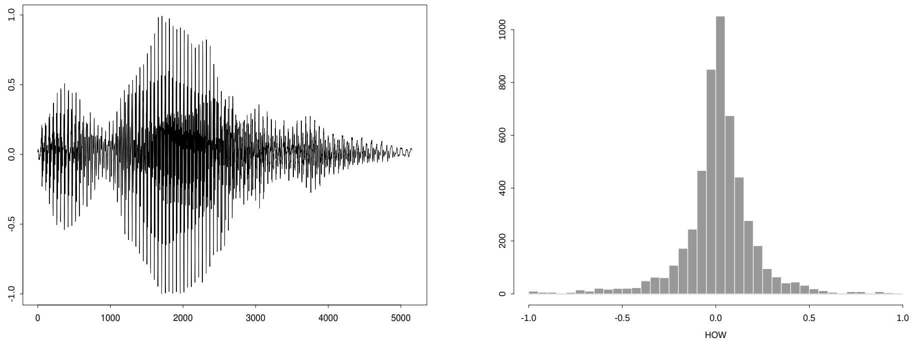

FIG. 4. *Signal de parole, et densité de probabilités empirique des valeurs du signal.*

On peut alors montrer que le rapport signal à bruit de quantification devient

(38)
$$SNR \approx \begin{cases} 6,02\,R + 4,77 - 20\log_{10}\left(1+\log A\right) & \text{pour les grandes valeurs du signal} \\ SNR_Q + 20\log_{10}\left(\frac{A}{1+\log A}\right) & \text{pour les faibles valeurs du signal .} \end{cases}$$

EXEMPLE: Valeur typique du paramètre, pour le codage du signal de parole: $A = 87.56$ (standard PCM européen). Le SNR correspondant, exprimé en décibels (dB), vaut

$$SNR_A \approx 6,02\,R - 9,99 \ .$$

2.3.2. *La loi μ.* La fonction de compensation est dans ce cas donnée par

$$c(x) = x_{max}\,\frac{\log(\mu|x|/x_{max})}{\log(1+\mu)}\,\mathrm{sgn}(x) \ . \tag{39}$$

On montre alors que le rapport signal à bruit de parole est approximativement donné par

(40)
$$SNR \approx \begin{cases} 6,02\,R + 4,77 - 20\log_{10}\left(\log(1+\mu)\right) & \text{pour les grandes valeurs du signal} \\ SNR_Q + 20\log_{10}\left(\frac{\mu}{\log(1+\mu)}\right) & \text{pour les faibles valeurs du signal .} \end{cases}$$

EXEMPLE: Valeur typique du paramètre: pour le signal de parole: $\mu = 255$ (standard PCM US). Le SNR correspondant, exprimé en décibels (dB), est de la forme

$$SNR_\mu \approx 6,02\,R - 10,1 \ .$$

La figure 5 représente le signal de parole de la figure 4 après compensation logarithmique, et la densité de probabilités correspondante, qui est beaucoup plus régulière que celle de la figure 4.

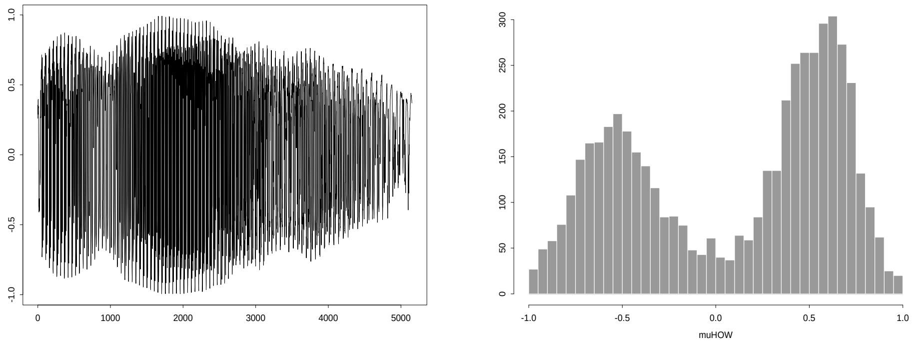

FIG. 5. *Signal de parole, et densité de probabilités empirique des valeurs du signal, après compensation logarithmique par loi μ.*

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. Pour obtenir le résultat, on commence par calculer

(41)
$$\sigma_Z^2 = \sum_{k=0}^{M-1} \left(\int_{x_k}^{x_{k+1}} (x - y_k)^2 \rho_X(x) dx \right) + \int_{-\infty}^{x_0} (x - y_-)^2 \rho_X(x) dx + \int_{x_M}^{\infty} (x - y_+)^2 \rho_X(x) dx \ ,$$

qu'il s'agit de minimiser par rapport aux variables x_k, y_k et y_+, y_-. Il s'agit d'un problème classique de minimisation de forme quadratique, dont la solution est fournie par les équations d'Euler-Lagrange. En égalant à zéro la dérivée par rapport à x_k, on obtient facilement les expressions suivantes

$$x_k = \frac{1}{2}(y_k + y_{k-1}) \ , \quad k = 1, \dots M - 1 \ , \tag{42}$$

$$x_0 = \frac{1}{2}(y_0 + y_-) \tag{43}$$

$$x_M = \frac{1}{2}(y_{M-1} + y_+) \tag{44}$$

De même, en égalant à zéro les dérivées par rapport aux y_k, à y_- et y_+, on obtient

$$y_k = \frac{\int_{x_k}^{x_{k+1}} x\rho_X(x)\,dx}{\int_{x_k}^{x_{k+1}} \rho_X(x)\,dx}\ , \tag{45}$$

$$y_- = \frac{\int_{-\infty}^{x_0} x\rho_X(x)\,dx}{\int_{-\infty}^{x_0} \rho_X(x)\,dx}\ , \tag{46}$$

$$y_+ = \frac{\int_{x_M}^{\infty} x\rho_X(x)\,dx}{\int_{X_M}^{\infty} \rho_X(x)\,dx}\ . \tag{47}$$

On obtient facilement des expressions similaires dans des situations où ρ_X est bornée. Ceci conduit naturellement à un algorithme itératif, dans lequel les variables x et y sont mises à jour récursivement. Naturellement, comme il s'agit d'un algorithme de type "algorithme de descente", 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.

3. Le PCM différentiel (DPCM) et ses produits dérivés

Le DPCM (*differential PCM*) a pour but d'exploiter les redondances souvent présentes dans un signal pour en améliorer le codage. L'idée essentielle est que si un signal présente une certaine redondance, il doit être prédictible dans un certain sens, et à un certain niveau de précision. L'idée est d'introduire un modèle de signal, et de ne coder que les écarts au modèle. On peut ainsi espérer dimiunuer le domaine de valeurs (en d'autre termes, la variance) du signal avant quantification, et par là même diminuer le bruit de quantification (à nombre de bits par échantillon fixé).

Prenons l'exemple le plus simple, celui des signaux numériques (une version analogique peut être développée aussi), plus simple à formaliser. Soit donc $x = \{x_n, n \in \mathbb{Z}\}$ un signal numérique, et supposons que l'on puisse écrire un modèle de la forme

$$x_n = f(x_{n-1}, x_{n-2}, x_{n-3}, \dots x_{n-N}) + w_n\ , \tag{48}$$

où f est une fonction de prédiction, et w est l'erreur de prédiction. Il suffirait alors de coder les w_n (et les N premières valeurs non nulles de x_n), via les formules:

$$\begin{aligned} \hat{x}_n &= f(x_{n-1}, x_{n-2}, x_{n-3}, \dots x_{n-N}) \\ w_n &= x_n - \hat{x}_n\ . \end{aligned}$$

En ce qui concerne le décodage, connaissant un coefficient w_n et les valeurs précédentes $x_{n-1}, \dots x_{n-N}$, on reconstitue tout d'abord la prédiction $\hat{x}_n$ puis la vraie valeur

$$x_n = w_n + f(x_{n-1}, x_{n-2}, \dots x_{n-N})\ .$$

En pratique, ce type d'approche présente le risque d'induire une propagation d'erreurs: en effet, entre le codage et le décodage, les coefficients w_n sont quantifiés, de sorte que les coefficients x_n sont reconstitués avec une erreur. Comme ces coefficients sont de plus utilisés pour calculer la prédiction des suivants (via la fonction de prédiction f), les erreurs introduites par la quantification se propagent bel et bien. On dit qu'un tel schéma n'est pas stable.

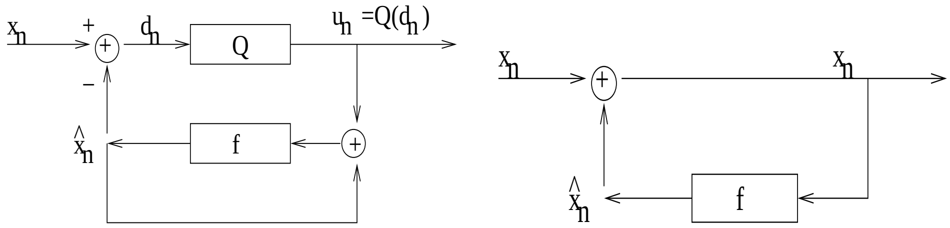

FIG. 6. *Le codeur DPCM: décomposition et reconstruction.*

Pour pallier cet inconvénient, il est plus efficace d'utiliser pour la prédiction les valeurs $\overline{x}_n$ qui seront disponibles à partir du signal décodé:

$$\hat{x}_n = f(\overline{x}_{n-1}, \overline{x}_{n-2}, \dots \overline{x}_{n-N}) \ . \tag{49}$$

On note alors

$$d_n = x_n - \hat{x}_n \tag{50}$$

l'erreur de prédiction, et

$$u_n = Q(d_n) \tag{51}$$

l'erreur de prédiction quantifiée. La valeur $\overline{x}_n$ est quant à elle donnée par

$$\overline{x}_n = \hat{x}_n + u_n \ , \tag{52}$$

et est également la valeur obtenue au décodage.

L'erreur commise au décodage est de la forme

$$z_n = \overline{x}_n = x_n = Q(d_n) - d_n \ , \tag{53}$$

c'est à dire coïncide avec l'erreur de quantification. Par conséquent, il n'y a effectivement pas de propagation d'erreur dans un tel schéma.

Comment évaluer l'amélioration apportée par le DPCM par rapport au PCM? Si on introduit le *gain de prédiction*

$$G_p = \frac{\sigma_d^2}{\sigma_x^2} \ , \tag{54}$$

qui évalue la diminution relative de variance du signal apportée par la prédiction, on obtient

$$SNR_{DPCM} = 10 \log_{10} \frac{\sigma_x^2}{\sigma_z^2} = 10 \log_{10} \frac{\sigma_d^2}{\sigma_z^2} - 10 \log_{10} G_p \ .$$

Or le premier terme n'est autre que la valeur de rapport signal à bruit que l'on aurait obtenue par codage PCM: on aboutit donc à

$$SNR_{DPCM} \approx SNR_{PCM} - 10 \log_{10} G_p \ . \tag{55}$$

On obtient bien ainsi l'effet recherché: une diminution de la variance du signal à quantifier, et donc une amélioration du rapport signal à bruit de quantification. A titre d'exemple, dans le cas du signal de parole, on peut ainsi obtenir une amélioration de l'ordre de 8dB en utilisant une prédiction d'ordre $N = 3$.

Reste à comprendre comment obtenir la fonction de prédiction f. Pour rester dans le cadre d'algorithmes simples, on se limite à des fonctions de prédiction linéaires.

3.1. Prédiction optimale, prédiction linéaire. Il est intéressant de formuler le problème dans un cadre plus large, qui permettra des généralisations à d'autres problèmes.

On considère deux vecteurs aléatoires $X = ((X_1, \dots X_K)^t$ et $Y = (Y_1, \dots Y_N)^t$ à valeurs dans $\mathbb{R}^K$ et $\mathbb{R}^N$ respectivement, et on se pose le problème de prédire les valeurs de Y connaissant les valeurs de X. Plus précisément, on recherche le prédicteur $\hat{Y}$ de Y qui est optimal dans le sens suivant: il minimise l'erreur en moyenne quadratique

$$\mathbb{E}\left\{ \|\hat{Y} - Y\|^2 \right\} .$$

Ce problème, formulé comme un problème c'optimisation, admet la solution classique suivante:

> THÉORÈME:*Etant donnés deux vecteurs aléatoires, le prédicteur optimal $\hat{Y}^*$ au sens de la moyenne quadratique de Y connaissant X est donné par*

$$\hat{Y}^* = \mathbb{E}\{Y|X\} . \tag{56}$$

Ce résultat est cependant difficile à exploiter dans un contexte de traitement du signal, car le calcul de l'espérance conditionnelle de Y demande de connaitre avec précision la distribution conditionnelle de Y sachant X, ce qui est très difficile. On doit donc se limiter à une sous-classe de prédicteurs, que l'on appelle les prédicteurs linéaires.

Ces derniers sont de la forme

$$\hat{Y} = AX \ , \tag{57}$$

et trouver le meilleur prédicteur linéaire au sens des moindres carrés revient à rechercher la matrice $N \times K$ A qui minimise l'erreur en moyenne quadratique. En notant génériquement $A_{k\ell}$ les éléments de la matrice A, l'équation normale

$$\frac{\partial}{\partial A_{k\ell}} \mathbb{E}\{ \|AX - Y\|^2 \} = 0$$

conduit à l'équation

$$\sum_{\ell'=1}^{K} A_{k\ell'} (R_X)_{\ell\ell'} = \mathbb{E}\{Y_k X_\ell\} . \tag{58}$$

En introduisant une notation matricielle, et en notant R_X la matrice de corrélation de X (définie par $(R_X)_{k\ell} = \mathbb{E}\{E_k X_\ell\}$), on aboutit ainsi au résultat suivant

> THÉORÈME:*Etant donnés deux vecteurs aléatoires, le prédicteur linéaire optimal $\hat{Y}^*$ au sens de la moyenne quadratique de Y connaissant X est donné par la matrice A solution de l'équation*

$$AR_X = \mathbb{E}\{YX^t\} . \tag{59}$$

Si de plus la matrice de covariance R_X de X est inversible, on a la solution

(60)
$$A = \mathbb{E}\{YX^t\}R_X^{-1} \ .$$

3.2. Application à la prédiction de signaux; prédicteurs linéaires de bas ordres. Revenons au cas de la prédiction de signaux unidimensionnels. Dans ce cas, il est nécessaire de supposer (au moins dans un premier temps) que le prédicteur reste valable pour tout le signal. Ceci est assuré dès que l'on suppose le signal stationnaire en moyenne d'ordre deux. On suppose aussi pour simplifier que le signal est centré. Alors, dans les notations précédentes, on a $\underline{Y} = X_n$ (donc $N = 1$), et $\underline{X} = (X_{n-1}, \dots X_{n-K})^t$. $\underline{\underline{A}}$ est donc une matrice $1 \times K$, c'est à dire un vecteur ligne, que l'on notera $\underline{h}^t = (h_1, \dots h_K)$. Le prédicteur linéaire est de la forme

(61)
$$\hat{X}_n = \sum_{k=1}^{K} h_k X_{n-k} \ .$$

Les coefficients h_k du prédicteur linéaire de rang K optimal sont solutions de l'équation

(62)
$$\underline{\underline{R}}_X \, \underline{h} = \underline{r} \ ,$$

où on a noté $\underline{r}$ le vecteur de coordonnées

$$r_k = R_X(k) = \mathbb{E}\{X_n X_{n-k}\} \ .$$

Cette équation est appelée *Equation de Yulle-Baxter*, ou *équation de Wiener-Hopf.*

Dans les cas les plus simples, la solution est explicite, et fait appel aux *coefficients de corrélation*

(63)
$$\rho_k = \frac{\mathbb{E}\{X_n X_{n-k}\}}{\mathbb{E}\{X_n^2\}} \ .$$

3.2.1. *Le cas $K = 1$.* Dans ce cas, $\underline{X}$ est lui aussi unidimensionnel, et on ne manipule que des nombres. $\underline{\underline{R}}_X = \sigma_X^2$, et $\underline{h}$ est un nombre

$$h = h_1 = \frac{\mathbb{E}\{X_n X_{n-1}\}}{R_X} = \rho_1 \ .$$

3.2.2. *Le cas $K = 2$.* Dans ce cas, $\underline{\underline{R}}_X$ est une matrice 2×2, de la forme

$$\underline{\underline{R}}_X = \sigma_X^2 \begin{pmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{pmatrix} \ .$$

De plus, on a

$$\mathbb{E}\{\underline{YX}^t\} = \sigma_X^2(\rho_1, \rho_2) \ ; \quad \underline{\underline{R}}_X{}^{-1} = \frac{1}{\sigma_X^2} \frac{1}{1-\rho_1^2} \begin{pmatrix} 1 & -\rho_1 \\ -\rho_1 & 1 \end{pmatrix} \ ,$$

d'où on déduit

$$\underline{h}^t = \frac{1}{1-\rho_1^2} \left(\rho_1 - \rho_1\rho_2^2, \rho_2 - \rho_1^2\right) \ .$$

3.2.3. *Le cas général.* Dans les cas où K est plus grand, il est nécessaire de revenir à l'équation de Yulle-Baxter (62), qui doit être résolue numériquement. Il existe des algorithmes classiques pour résoudre ce problème, comme par exemple l'algorithme de Levinson-Durbin. On pourra se rapporter à [**6**] pour plus de détails sur ces algorithmes. Il existe de nombreuses implémentations disponibles, dans des logiciels libres comme dans des produits commerciaux.

3.3. DPCM, ADPCM, LPC. L'utilisation de techniques de prédiction linéaire a conduit à de nombreuses stratégies de codage différentes. On mentionne quelques exemples ici.

1. DPCM: il s'agit du système de codage que nous avons décrit plus haut: un prédicteur "optimal" est choisi une bonne fois pour toutes, et le signal codé est alors caractérisé par les coefficients (quantifiés) u_n. Ceux-ci sont codés et transmis, et le décodeur reconstitue une approximation $\overline{x}$ du signal x à partir d'eux. Comme on l'a vu, plus le prédicteur est efficace, plus le nombre de bits requis pour la quantification est faible. Par conséquent, on s'attend à ce qu'une redondance accrue dans le signal après échantillonnage permette de diminuer le nombre de bits à allouer à chaque coefficient codé. C'est effectivement le cas, à tel point que certains codeurs se permettent de coder les coefficients u_n sur 1 bit seulement, au prix d'un échantillonnage extrêmement fin. C'est le cas des systèmes que l'on appelle Δ et $\Sigma\Delta$, implémentés notamment dans le nouveau standard SACD (Super Audio CD) proposé par Sony et Philips.
2. ADPCM: comme on l'a vu, améliorer la qualité de la prédiction permet de diminuer le nombre de bits à allouer à chaque nombre codé. Le principe du DPCM adaptatif (ADPCM) est de se baser sur une prédiction "locale", donc de meilleure qualité. Plus précisément, étant donné un signal échantillonné, il est tout d'abord segmenté, c'est à dire coupé en segments (ou trames) de longueur fixée. Un prédicteur linéaire est alors estimé à l'intérieur de chaque trame. Chaque trame est alors caractérisée par les coefficients u_n précédents, et les coefficients a_n du prédicteur, qui doivent être quantifiés et transmis eux aussi.
3. LPC: (Linear Predictive Coding). Le LPC est une variante du DPCM adaptatif dans laquelle seuls les coefficients du prédicteur (pour chaque trame) sont quantifiés et transmis. Le décodeur utilise alors des nombres pseudo-aléatoires pour remplacer les coefficients u_n.

4. Quantification vectorielle

Le défaut essentiel du codeur PCM est son incapacité à prendre en compte les corrélations existant dans le signal codé. La quantification utilisée dans le PCM traite en effet chaque échantillon individuellement, et ne tient aucun compte de la "cohérence" du signal. Pour tenir compte de celle-ci, il faudrait a priori effectuer une quantification non plus sur des échantillons individuels, mais sur des familles, ou vecteurs, d'échantillons. C'est le principe de la quantification vectorielle. Cependant, alors que l'idée de la quantification vectorielle est somme toute assez simple, sa mise en œuvre effective peut être extrêmement complexe.

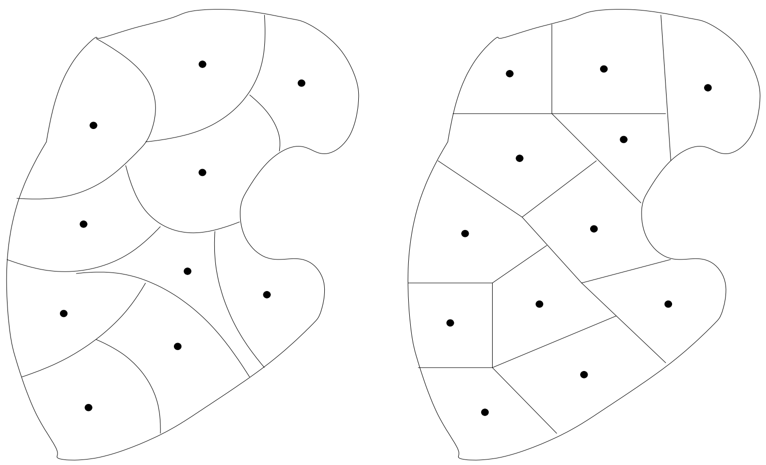

FIG. 7. *Quantification vectorielle: deux partitions d'un domaine de $\mathbb{R}^2$: quantifications régulière (à droite) et irrégulière (à gauche): boîtes de quantification et leurs centroïdes.*

Considérons un signal $\{x_0, x_1, \dots\}$, et commençons par le "découper" en segments de longueur fixée N. Chaque segment représente donc un vecteur $X \in \mathbb{R}^N$, et on se pose donc le problème de construire un quantificateur:

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

en essayant de minimiser l'erreur de quantification

$$D = \mathbb{E}\{|X - Q(X)|^2\} \ ,$$

où on a noté $|X|$ la norme Euclidienne de $X \in \mathbb{R}^N$.

DÉFINITION *L'ensemble des "vecteurs quantifiés" est appelé le* dictionnaire *du quantificateur (*codebook *en anglais).*

Le problème est le même que celui que nous avons vu dans le cas de la recherche du quantificateur scalaire optimal, mais la situation est cette fois bien plus complexe, à cause de la dimension supérieure dans laquelle on se place. En effet, dans le cas d'un quantificateur scalaire, le problème est essentiellement de "découper" un sous-domaine de l'axe réel, ou l'axe réel lui même, en un nombre fini de boîtes de quantification (c'est à dire d'intervalles). En dimension supérieure, il s'agit maintenant d'effectuer une partition d'une partie de $\mathbb{R}^N$ en sous-domaines. On doit pour ce faire effectuer de multiples choix, notamment en ce qui concerne la forme des frontières: dans le cas de frontières planes, on parlera de quantificateur régulier, dans le cas contraire de quantificateur irrégulier (voir la FIG. 7. La problèmatique de la quantification vectorielle présente des similarités certaines avec la problématique du "clustering".

Il existe de multiples façons différentes de construire un quantificateur vectoriel. On peut par exemple se référer à **[4]** pour une description relativement complète de l'état de l'art. Le problème de la recherche du quantificateur vectoriel optimal peut se formuler suivant les lignes ébauchées dans la section 2.4: à partir du moment où on s'est donné le nombre de sous-domaines recherchés pour la partition, il reste à optimiser la distorsion globale

$$D = \sum_{k=0}^{M-1} \int_{\Omega_k} (x - y_k)^2 \rho_X(x)\, dx \ ,$$

où on a noté $\Omega_0, \dots \Omega_{M-1}$ les sous-domaines considérés, et $y_0, \dots y_{M-1}$ les valeurs de quantification correspondantes. L'optimisation par rapport à y_k reste assez simple, et fournit la *condition de centroïde*

$$y_k = \frac{\int_{S_k} x \rho_X(x)\, dx}{\int_{S_k} \rho_X(x)\, dx} \ ,$$

mais l'optimisation par rapport aux sous-domaines S_k, généralement effectuée numériquement, peut s'avérer extrêmement complexe suivant les hypothèses que l'on fait sur la forme des sous-domaines.

Un exemple: les recommandations G729 et G723.1 pour le codage de la voix sur IP. Le codage du signal de parole sur internet représente un enjeu majeur pour les années qui viennent. Les protocoles de codage sont en cours de développement et de standardisation. La norme actuelle impose la présence du codeur G711, qui est essentiellement un codeur PCM (fréquence d'échantillonnage: 8 kHZ) couplé à une compensation logarithmique[1] (loi A ou μ). La norme recommande fortement la présence des codeurs G729 et G723.1, basés sur des schémas un peu plus sophistiqués.

G729 est un codeur de type CELP (Code Excited Linear Prediction), qui consiste essentiellement en un codeur LPC comme décrit plus haut, suivi d'une quantification vectorielle des coefficients du prédicteur. De façon assez remarquable, les formes d'onde correspondant aux vecteurs quantifiés peuvent être mises en correspondance avec les phonèmes, c'est à dire les formes d'ondes élémentaires de la parole.

5. Codage par transformation

On a vu quelques exemples de techniques visant à corriger le défaut essentiel du codeur PCM, c'est à dire son incapacité à prendre en compte les relations existant entre échantillons. On a vu en particulier que l'introduction d'un processus de prédiction permet de réduire le nombre de bits alloué à chaque coefficient à coder.

Bien qu'il poursuive le même but, le codage par transformation repose sur un principe assez différent: il s'agit plutôt de réduire le nombre de coefficients significatifs, grâce à une transformation linéaire effectuée sur les échantillons. L'idée est essentiellement la suivante. Prenons l'exemple d'une seconde de son, échantillonnée à 44,1 kHZ. Chaque seconde est caractérisée par 44100 échantillons, c'est à dire par un point dans un espace de dimension 44100. Or s'il existe des redondances systématiques dans les signaux échantillonnés (à cette fréquence d'échantillonnage là), on peut s'attendre à ce que l'ensemble des signaux ne "remplissent" pas tout l'espace de dimension 44100, mais un sous-espace de dimension bien inférieure. Il suffirait

1. La compensation logarithmique permet un nombre de bits par échantillon égal à 13 (loi A) ou 14 (loi μ), voire réduit jusquà 8 sans distorsion importante.

alors de projeter les signaux sur ce sous-espace pour obtenir une caractérisation de ceux ci avec beaucoup moins de données. L'idée du codage par transformation est d'utiliser des bases orthonormées particulières de l'espace des signaux pour effecter une telle projection. Etant donnée une base $\{e_1, e_2, e_3, \dots\}$ de l'espace de signaux considéré, tous les signaux sont caractérisés par les coefficients $\alpha_1, \alpha_2, \dots$ de leur développement sur cette base:

$$f = \sum_{k=1}^{\infty} \alpha_k e_k \ .$$

Si maintenant cette base a été choisie de sorte que de tous les coefficients α_k, tous sauf les N premiers soient systématiquement nuls ou très petits, on peut alors écrire

$$f \approx f_{app} = \sum_{k=1}^{N} \alpha_k e_k \ ,$$

et si nécessaire donner une estimation de l'erreur $f - f_{app}$.

La question qui se pose alors est la suivante: comment choisir une base qui possède de telles propriétés d'approximation, pour une classe de signaux bien déterminée? ce problème est un sujet de recherches très actif à l'heure actuelle. Néanmoins, on connaît quelques exemples de "bonnes bases", qui sont implémentées dans un certain nombre de codeurs "standardisés" (comme le codeux ATRAC des mini disques, ou encore les codeurs MPEG audio ou JPEG pour les images).

5.1. Bases trigonométriques. On a déjà vu, dans un contexte différent, des exemples de bases trigonométriques, notamment les bases trigonométriques pour les espaces $L^2_p([a, b])$ ou $\mathbb{C}^N$. De telles bases sont maintenant d'usage courant dans des schémas de codage par transformation, notamment en ce qui concerne le codage de signaux audiophoniques.

Pour simplifier, on se contentera ici de décrire le cas de bases adaptées aux signaux numériques, telles qu'elles sont employées notamment dans les codeurs audio MPEG. Partant d'un signal numérique, que l'on suppose pour simplifier de longueur infinie $f \in \ell^2(\mathbb{Z})$, la première étape consiste en une segmentation du signal: le signal est segmenté en "blocs" (aussi appelés *trames*, ou *segments*) de longueur fixée N; par exemple $\{f_0, f_1, \dots f_{N-1}\}$, puis $\{f_N, f_{N+1}, \dots f_{2N-1}\}$, $\{f_{2N}, f_{2N+1}, \dots f_{3N-1}\}$, et ainsi de suite. Puis chaque segment est décomposé sur une base trigonométrique. Pour un segment numéro K donné, on posera

$$g_n = f_{Nk+n} \ , \quad n = 0, \dots N-1 \ .$$

Il existe différents choix possibles de bases trigonométriques. On va en voir deux exemples

5.1.1. *Deux exemples de bases tgrigonométriques pour* $\mathbb{C}^N$. Le premier exemple est la base de Fourier finie classique, telle qu'elle est utilisée dans le cadre de la transformation de Fourier finie: les vecteurs (de longueur N) $e^0, e^1, \dots e^{N-1}$ définis par leurs composantes

$$e^k_n = \frac{1}{\sqrt{N}} \exp\{2i\pi \frac{kn}{N}\} \ , \quad n = 0, 1, \dots N-1 \tag{64}$$

forment une base orthonormée de $\mathbb{C}^N$. On en déduit les développements que nous avons déjà vus:

$$g_n = \frac{1}{N}\sum_{k=0}^{N} \hat{g}_k e^{2i\pi kn/N} \ , \tag{65}$$

où les coefficients de Fourier $\hat{g}_k$ sont donnés par

$$\hat{g}_k = \sum_{n=0}^{N} g_n e^{-2i\pi kn/N} \ . \tag{66}$$

Une variante consiste à utiliser une base de cosinus plutôt qu'une base trigonométrique complexe.

Théorème: les vecteurs (de longueur N) $c^0, c^1, \dots c^{N-1}$ définis par leurs composantes

$$c_n^0 = \frac{1}{\sqrt{N}} \ , \quad n = 0, 1, \dots N-1 \tag{67}$$

$$c_n^k = \sqrt{\frac{2}{N}}\cos\{\pi\frac{k(n+1/2)}{N}\} \ , \quad n = 0, 1, \dots N-1 \tag{68}$$

forment une base orthonormée de $\mathbb{C}^N$.

On en déduit des décompositions du type:

$$g_n = \sum_{k=0}^{N} \alpha_k c_n^k \ , \tag{69}$$

où les coefficients α_k sont donnés par

$$\alpha_k = \sum_{k=0}^{N} g_n c_n^k \ . \tag{70}$$

L'avantage de cette dernière base est d'une part que les coefficients α_k sont des nombres réels (et pas complexes comme dans le cas de la base précédente), mais surtout que les effets de bloc (dus à la segmentation) sont réduits. La transformation qui associe les coefficients α_k aux échantillons g_n est appelée DCT (Discrete Cosine Transform).

5.1.2. *Pourquoi des bases trigonométriques.* Comme on l'a déjà signalé, la transformation ne modifie pas le volume des données: par exemple, il y a autant de coefficients α_k que d'échantillons g_n. Par contre, pour certaines classes de signaux, il s'avère que l'immense majorité des coefficients α_k sont nuls ou en tous cas très proches de zéro, de sorte qu'il n'est pas nécessaire de les coder. Ceci peut donc conduire à une importante compression de données.

Reste à comprendre de façon plus précise les raisons de cette remarque. Un premier élément de réponse est fourni par le résultat suivant

Théorème: Soit X un signal numérique aléatoire fini. Si X est stationnaire en moyenne d'ordre deux, alors les coefficients α_k sont décorrélés:

$$\mathbb{E}\{\alpha_k\overline{\alpha}_\ell\} = \mathcal{E}_k\,\delta_{k\ell} \ . \tag{71}$$

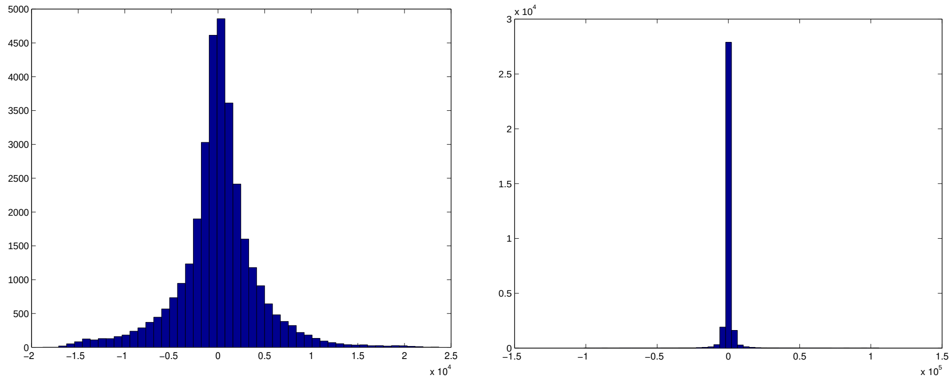

FIG. 8. *Densité de probabilités empirique des échantillons d'un signal audiophonique(à gauche), et des coefficients DCT α_k (à droite).*

Il s'avère que par exemple, les signaux audiophoniques peuvent être considérés comme des signaux aléatoires stationnaires, sur des intervalles de temps assez courts (de l'ordre de 10 à 20 millisecondes). Par conséquent, le théorème ci-dessus s'applique. De plus, la suite $\mathcal{E}_\ell$ est assez rapidement décroissante en fonction de ℓ, de sorte que l'on peut considérer que seuls les premiers coefficients α_k sont réellement significatifs. On peut donc se contenter d'approximer un segment de signal audiophonique en n'utilisant que ses premiers coefficients α_k, ce qui conduit généralement à un gain considérable en termes de volume de données.

Un exemple est donné en FIGURE 8, où on a représenté à gauche l'histogramme des échantillons d'un signal audio (une vieille chanson populaire Italienne), et à droite l'histogramme des coefficients MDCT α_k. Comme on peut le voir, la distribution des α_k est considérablement plus "piquée" en zéro que la distribution des échantillons, de sorte qu'un pourcentage écrasant (ici plus de 99 pour cent) des coefficients α_k sont négligeables, et seront quantifiés à zéro.

5.1.3. *Quelques exemples.* Les bases trigonométriques sont maintenant très populaires dans le monde du codage par transformation, et sont implémentées dans de nombreux codeurs.

1. Un premier exemple est fourni par le standard `JPEG` de compression des images. `JPEG` utilise une version bidimensionnelle des bases DCT, de la forme

$$C^{k\ell} : m, n = 0, 1, \dots N \to C^{l\ell}_{mn} = c^k_m \, c^\ell_n \ ,$$

   où les vecteurs c^k sont les vecteurs de base DCT unidimensionnels définis en (67). `JPEG` part du principe qu'une image fixe peut raisonnablement être considérée stationnaire à l'intérieur de blocs de taille 8×8. Ainsi, le codeur correspondant commence par segmenter l'image en blocs de taille 8×8, et représente les sous-images correspondantes par un certain nombre de coefficients DCT. Puis les coefficients sont quantifiés et codés comme d'habitude.
2. Les codeurs MPEG audio: les codeurs MPEG audio, et en particulier MPEG1 (qui conduit au fameux standard MP3) et MPEG 4 sont eux aussi basés

sur l'utilisation de bases trigonométriques. Il s'agit en l'occurrence de bases MDCT (Modulated Discrete Cosine Transform), qui sont une variante des bases DCT classiques. Un signal audio est tout d'abord segmenté en trames de 20 millisecondes de durée environ, puis chaque segment est codé par ses coefficients α_k les plus significatifs. Une étape supplémentaire est généralement ajoutée, qui vise à supprimer de l'information inaudible (on parle de codage perceptif: on se permet des erreurs, tant qu'elles ne s'entendent pas), puis viennent les étapes classiques de quantification et codage.

Les codeurs par transformation utilisant des bases trigonométriques sont généralement complétés par des stratégies relativement sophistiquées de quantification, et de codage binaire, que nous verrons dans le chapitre suivant.

5.2. Codage en sous bandes. Les bases trigonométriques sont généralement bien adaptées à des signaux qui présentent des oscillations régulières, comme par exemple la parole, ou les signaux audio. Par contre, elles introduisent une échelle de référence dans le problème, qui est la taille des fenêtres utilisées. L'introduction de ces fenêtres peut se traduire par l'apparition d'effets de "bloc", comme on peut en voir sur les images comprimées avec `JPEG` avec de forts taux de compression. Ces effets sont atténués lorsque l'on utilise des bases "adoucies" comme on vient de le voir, mais rarement supprimés. Le codage en sous bandes que nous allons maintenant voir (et son pendant, la transformation en ondelettes) constitue une alternative bien adaptée au codage des images, qui présente l'avantage de ne pas introduire d'effet de bloc.

Dans cette section, on étudiera tout d'abord le codage en sous bandes proprement dit, qui est la version adaptée aux signaux numériques. Puis on verra la transformation en ondelettes, qui est la version adaptée aux signaux analogiques, et on verra comment faire le lien entre les deux.

Le codage en sous bandes consiste essentiellement en une variation autour du thème de l'échantillonnage et du sous-échantillonnage. On se place dès le départ dans le cadre des signaux échantillonnés, c'est à dire dans $\ell^2(\mathbb{Z})$. On considère deux suites absolument sommables $\{h_n, n \in \mathbb{Z}\}$ et $\{g_n, n \in \mathbb{Z}\}$, similaires à la suite $\{h_n, n \in \mathbb{Z}\}$ précédente. Plus précisément, considérant les restrictions de leurs transformées de Fourier à $[-\pi, \pi]$, on impose que celle de $\{h_n, n \in \mathbb{Z}\}$ soit concentrée au voisinage de $[-\pi/2, \pi/2]$, et que celle de $\{g_n, n \in \mathbb{Z}\}$ soit concentrée au voisinage de $[-\pi, -\pi/2] \cup [\pi/2, \pi]$. On va alors procéder de manière similaire au chapitre précédent: la convolution de $\{s_n, n \in \mathbb{Z}\}$ par $\{h_n, n \in \mathbb{Z}\}$ ou $\{g_n, n \in \mathbb{Z}\}$ produit une suite dont la bande a été réduite d'un facteur 2 (à une erreur provenant d'un mauvais échantillonnage près). On sous-échantillonne alors chacun de ces deux produits de convolution d'un facteur 2. Ce faisant, on commet dans les deux cas une erreur (aliasing), et on va rechercher dans quels cas ces erreurs peuvent se compenser.

Pour cela, considérons en détail les opérations effectuées lors du schéma décomposition-reconstruction, et ce sur une seule étape tout d'abord.

5.2.1. *Sous-échantillonnage.* Commençons par évaluer l'effet du sous-échantillonnage. Soit $s = \{s_n, n \in \mathbb{Z}\}$ une suite de carré sommable, et considérons la suite

ρ définie par

$$\rho_n = \begin{cases} s_n & \text{si } n \text{ est pair} \\ 0 & \text{sinon.} \end{cases}$$

Un calcul immédiat montre que

$$\hat{\rho}(\omega) = \frac{1}{2}\left(\hat{s}(\omega) + \hat{s}(\omega+\pi)\right) \ .$$

On introduit maintenant la nouvelle suite sous-échantillonnée σ, définie par

$$\sigma_n = \rho_{2n} = s_{2n} \ .$$

On a alors

$$\hat{\sigma}(\omega) = \hat{\rho}\left(\frac{\omega}{2}\right) = \frac{1}{2}\left(\hat{s}\left(\frac{\omega}{2}\right) + \hat{s}\left(\frac{\omega}{2}+\pi\right)\right) \ . \tag{72}$$

5.2.2. *Filtres conjugués en quadrature.* On considère maintenant les deux filtres $h, g \in \ell^1(\mathbb{Z})$, et les opérateurs H et $G : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$, appelés *opérateurs de décomposition*, définis par

$$(Hf)_n = \textstyle\sum_k h_{2n-k} f_k = \displaystyle\sum_k h_k f_{2n-k} \ ; \tag{73}$$

$$(Gf)_n = \textstyle\sum_k g_{2n-k} f_k = \displaystyle\sum_k g_k f_{2n-k} \ . \tag{74}$$

Ces deux opérations sont essentiellement des convolutions, suivies d'un sous-échantillonnage. Par conséquent, il vient, en posant $s = Hf$ et $d = Gf$:

$$\hat{s}(\omega) = \frac{1}{2}\left[\hat{h}\left(\frac{\omega}{2}\right)\hat{f}\left(\frac{\omega}{2}\right) + \hat{h}\left(\frac{\omega}{2}+\pi\right)\hat{f}\left(\frac{\omega}{2}+\pi\right)\right] \tag{75}$$

$$\hat{d}(\omega) = \frac{1}{2}\left[\hat{g}\left(\frac{\omega}{2}\right)\hat{f}\left(\frac{\omega}{2}\right) + \hat{g}\left(\frac{\omega}{2}+\pi\right)\hat{f}\left(\frac{\omega}{2}+\pi\right)\right] \ . \tag{76}$$

On considère maintenant les opérateurs adjoints H^* et G^* de H et G, à savoir

$$(H^*f)_k = \sum_n \overline{h}_{2n-k} f_n \ ; \tag{77}$$

$$(G^*f)_k = \sum_n \overline{g}_{2n-k} f_n \ . \tag{78}$$

H^* et G^* sont appelés *opérateurs de reconstruction*, ou de *synthèse*. Un calcul direct montre que

$$\widehat{H^*f}(\omega) = \overline{\hat{h}}(\omega)\hat{f}(2\omega) \ ; \quad \widehat{G^*f}(\omega) = \overline{\hat{g}}(\omega)\hat{f}(2\omega) \ . \tag{79}$$

Considérons maintenant un système tel que décrit dans la figure 9. On impose que ce système soit à reconstruction parfaite, c'est à dire que

$$H^*H + G^*G = 1 \ . \tag{80}$$

En imposant une telle contrainte dans l'espace de Fourier, on aboutit à

$$\begin{aligned} \hat{f}(\omega) = \tfrac{1}{2}\Big[& \overline{\hat{h}}(\omega)\left(\hat{h}(\omega)\hat{f}(\omega) + \hat{h}(\omega+\pi)\hat{f}(\omega+\pi)\right) \\ & + \overline{\hat{g}}(\omega)\left(\hat{g}(\omega)\hat{f}(\omega) + \hat{g}(\omega+\pi)\hat{f}(\omega+\pi)\right)\Big] \ , \end{aligned} \tag{81}$$

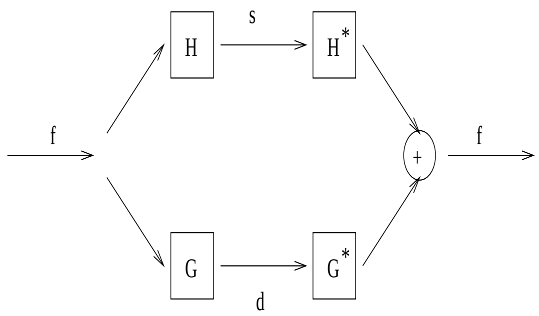

FIG. 9. *Un schéma "analyse-synthèse".*

ceci pour tout $f \in L^2([-\pi,\pi])$. On reconnait là les deux termes importants: le terme qui nous intéresse, proportionnel à $\hat{f}(\omega)$, et le terme de "repliement de spectre", proportionnel à $\hat{f}(\omega+\pi)$, que l'on souhaite annuler. La condition de reconstruction parfaite se met donc sous la forme suivante: pour tout ω,

$$|\hat{h}(\omega)|^2 + |\hat{g}(\omega)|^2 = 2 \tag{82}$$

$$\hat{h}(\omega)\overline{\hat{h}}(\omega+\pi) + \hat{g}(\omega)\overline{\hat{g}}(\omega+\pi) = 0\ , \tag{83}$$

ou encore, sous forme matricielle

$$\begin{pmatrix} \hat{h}(\omega) & \hat{g}(\omega) \\ \hat{h}(\omega+\pi) & \hat{g}(\omega+\pi) \end{pmatrix} \begin{pmatrix} \overline{\hat{h}}(\omega) \\ \overline{\hat{g}}(\omega) \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{84}$$

Soit $\Delta(\omega)$ le déterminant de cette matrice, que l'on suppose non nul pour presque tout ω. On a alors la caractérisation suivante des filtres permettant une reconstruction parfaite:

THÉORÈME

1. *Les filtres à reconstruction parfaite doivent satisfaire les équations de compatibilité*

$$\begin{pmatrix} \overline{\hat{h}}(\omega) \\ \overline{\hat{g}}(\omega) \end{pmatrix} = \frac{2}{\Delta(\omega)} \begin{pmatrix} \hat{g}(\omega+\pi) \\ -\hat{h}(\omega+\pi) \end{pmatrix} \tag{85}$$

et

$$|\hat{h}(\omega)|^2 + |\hat{h}(\omega+\pi)|^2 = 2\ . \tag{86}$$

2. *En supposant que les filtres h et g soient des suites finies, alors il existe $\varphi \in \mathbb{R}$ et $\ell \in \mathbb{Z}$ tels que*

$$\hat{g}(\omega) = e^{i\varphi} e^{i(2\ell+1)(\omega+\pi)}\, \overline{\hat{h}}(\omega+\pi)\ . \tag{87}$$

On peut alors poser la définition suivante:

DÉFINITION: *Des suites $h = \{h_n, n \in \mathbb{Z}\}$ et $g = \{g_n, n \in \mathbb{Z}\}$ telles que les conditions (86) et (87) soient satisfaites sont appelés* filtres miroirs conjugués.

On note que la relation (87) se traduit, par TFD inverse, par

$$g_k = e^{i\varphi}(-1)^k \overline{h}_{-k-2\ell-1} \ . \tag{88}$$

Ainsi, il suffit de se donner une suite $h = \{h_n, n \in \mathbb{Z}\}$ telle que l'équation (86) soit satisfaite, et d'en déduire la suite $g = \{g_n, n \in \mathbb{Z}\}$ par (88) pour obtenir un schéma de codage en sous bandes.

REMARQUE: Il est en fait possible de généraliser encore la construction. En effet, rien n'oblige à utiliser dans le cadre du schéma de reconstruction, les mêmes filtres que ceux utilisés pour le schéma d'analyse. Il suffit de se donner deux autres filtres, de réponses impulsionnelles respectives $\tilde{h} = \{\tilde{h}_n, n \in \mathbb{Z}\}$ et $\tilde{g} = \{\tilde{g}_n, n \in \mathbb{Z}\}$, et les opérateurs correspondants

$$\begin{aligned} (\tilde{H}f)_k &= \sum_n \tilde{h}_{2n-k} f_n \ ; \\ (\tilde{G}f)_k &= \sum_n \tilde{g}_{2n-k} f_n \ . \end{aligned}$$

Alors si $h, g, \tilde{g}$ et $\tilde{g}$ sont tels que pour tout ω on ait

$$\hat{\tilde{h}}(\omega)\hat{h}(\omega) + \hat{\tilde{g}}(\omega)\hat{g}(\omega) = 2$$

on peut montrer facilement que

$$\tilde{H}H + \tilde{G}G = 1 \ ,$$

ce qui est suffisant pour construire une "boite" à reconstruction parfaite du type de celle décrite en FIGURE 9 (en remplaçant les H^* et G^* par des $\tilde{H}$ et $\tilde{G}$.

5.2.3. *Codage en sous bandes.* L'algorithme de codage en sous bandes repose sur l'utilisation récursive du schéma d'analyse-synthèse que nous venons de voir. Etant donnée une suite $f = \{f_n, n \in \mathbb{Z}\} \in \ell^2(\mathbb{Z})$, on pose

$$s^0 = f \ , \tag{89}$$

$$s^n = Hs^{n-1} \ , \tag{90}$$

$$d^n = Gs^{n-1} \ . \tag{91}$$

Ce schéma de décomposition est représenté en FIG. 10 (image de gauche).

On a alors de façon évidente

$$\begin{aligned} f &= H^*s^1 + G^*d^1 \\ &= H^*(H^*s^2 + G^*d^2) + G^*d^1 \\ &= H^*(H^*(H^*s^3 + G^*d^3) + G^*d^2) + G^*d^1 \\ &= \ \dots \ , \end{aligned}$$

ce qui conduit à un schéma simple de reconstruction de f à partir des suites $d^1, d^2, \dots, d^L, s^L$, où L est un nombre entier fixé. Ce schéma est représenté en FIG. 10, image de droite.

REMARQUES:

- Dans ce que nous avons vu, le codage en sous bandes s'effectue en divisant de façon récursive la bande de fréquences en deux, et en sous-échantillonnant en conséquence (d'un facteur 2). Bien que ce soit la solution la plus simple, rien n'oblige en fait à se limiter à

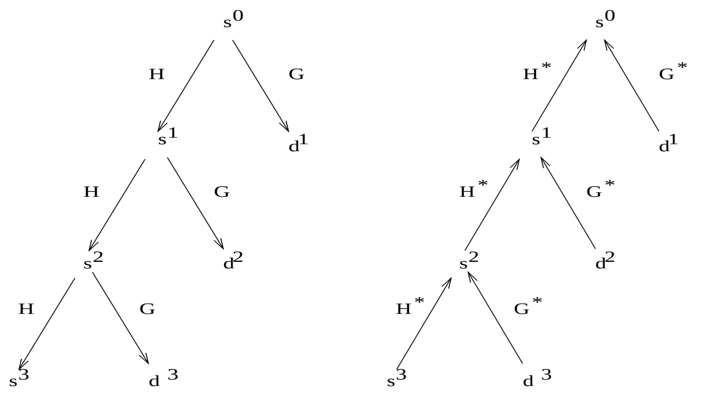

FIG. 10. *Schéma de décomposition en sous bandes (à gauche) et de reconstruction à partir des sous bandes (à droite).*

un facteur 2. On peut construire des schémas de codage en sous bandes associés à des sous-échantillonnages presque arbitraires. En pratique, le facteur 2 est souvent préféré.
- On généralise facilement le codage en sous bandes au contexte des images. Il suffit pour cela de considérér deux schémas de codage en sous bandes (généralement identiques) unidimensionnels, et de les appliquer aux deux directions des signaux 2D (voir plus loin).

5.2.4. *Application au codage des signaux.* Pour coder un signal, on peut donc se contenter de coder ses coefficients d (ainsi que les coefficients s à une échelle grossière). Un exemple est présenté en FIGURES 11, 12 et 13. Le signal original (un vieil enregistrement de Caruso) est montré en FIGURE 11, et ses coefficients d se trouvent en FIGURE 12 (les petites échelles sont vers le bas, et les grandes échelles vers le haut). Compte tenu du sous-échantillonnage inhérent au codage par sous bandes, il est difficile de comparer les sous bandes entre elles. C'est pour cela qu'on s'intéresse aussi à la représentation multirésolution montrée en FIGURE 13, dans laquelle on représente des reonstruction partielles obtenues à partir de coefficients d_{jk} avec une échelle j fixée. On voit apparaitre très nettement sur cette figure les différentes bandes de fréquence considérées: les courbes du haut de la figure varient bien plus lentement que celles du bas. Ceci confirme l'interprétation de la décomposition multirésolution comme un "banc de filtres" passe-bande, correspondant à des bandes de fréquence variables.

Il s'avère qu'après codage en sous bandes, les coefficients d obtenus sont souvent assez bien décorrélés, en tous cas bien plus que les échantillons. Ceci implique que la redondance ayant été réduite, beaucoup de ces coefficients sont très faibles numériquement. On peut en voir un exemple sur la FIGURE 14, où on a représenté un petit échantillon de signal audio, un histogramme de ses échantillons, et un histogramme des coefficients d d'un développement en sous bandes.

On voit très nettement que les coefficients d sont très majoritairement proches de 0, et pourront donc être négligés. Ceci suggère d'employer un schéma de codage dans lequel seuls les coefficients significatifs seront codés. Ceci contraint alors à

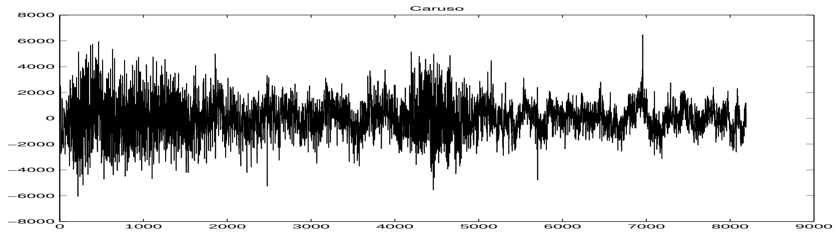

FIG. 11. *Un vieil enregistrement de Caruso*

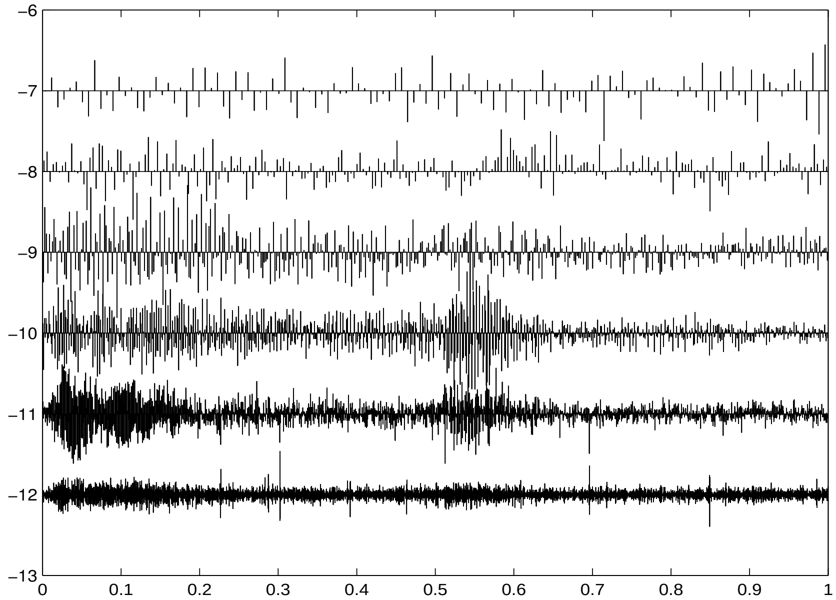

FIG. 12. *Un vieil enregistrement de Caruso: coefficients d'ondelettes*

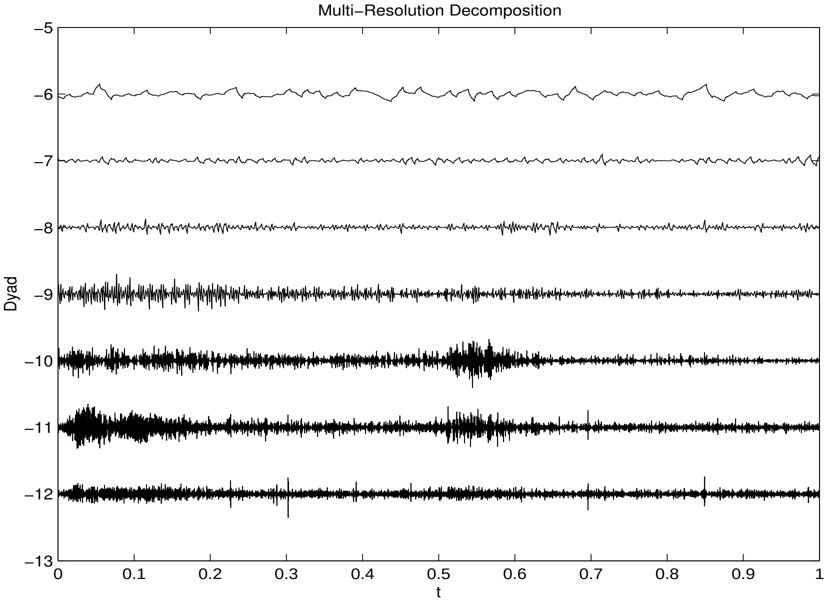

FIG. 13. *Un vieil enregistrement de Caruso: décomposition multirésolution*

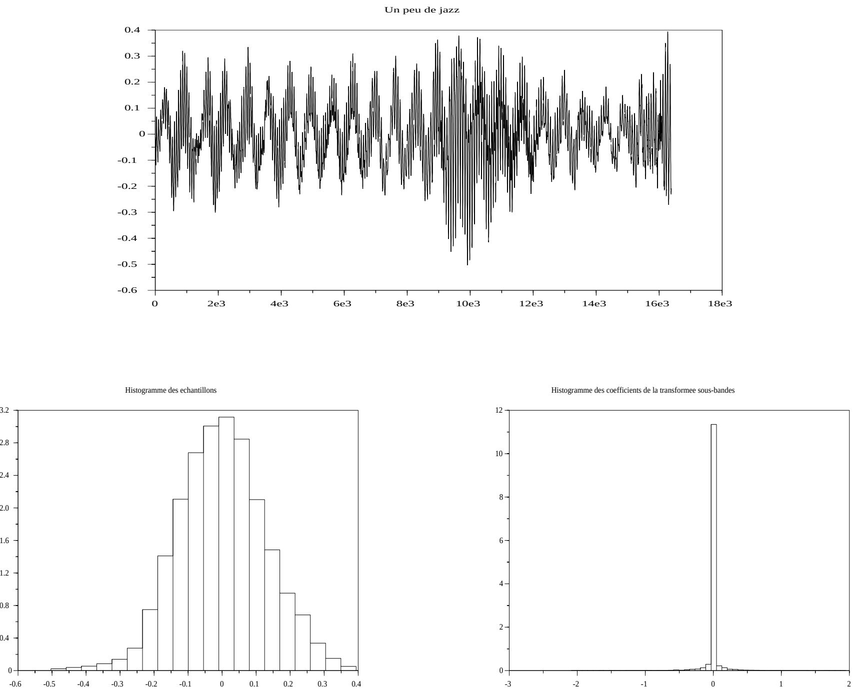

FIG. 14. *Signal audio (jazz), en haut; densité de probabilités empirique des échantillons (en bas à gauche), en des coefficients sous-bande (en bas à droite).*

coder l'adresse des coefficients significatifs, ce qui pose d'autres problèmes, qui seront abordés dans le chapitre suivant.

En termes de résultats, le codage par sous bandes s'avère être le principe de codage le plus efficace à l'heure actuelle pour ce qui est du codage des images. Il y a des raisons à cela, qui tiennent à la nature même des images. On verra dans une section suivante quelques éléments d'explication.

5.2.5. *Adaptation au cas des images.* Les images sont modélisées comme des signaux numériques à deux indices: $\{I_{mn}\}$. Il est possible d'adapter les techniques de codage en sous bandes unidimensionnelles au cas des images. On considère pour cela un schéma de codage en sous bandes unidimensionnel, caractérisé donc par deux filtres H et G. On construit alors quatre filtres $\mathcal{H}$, $\mathcal{G}_1$, $\mathcal{G}_2$, et $\mathcal{G}_3$ de la façon suivante: pour toute suite à deux indices $I = \{I_{mn}\}$, on pose

$$\left\{\begin{array}{ll} [\mathcal{H}I]_{mn} & = \sum_k \sum_\ell h_k h_\ell I_{2m-k,2n-\ell}\ , \\ [\mathcal{G}_1 I]_{mn} & = \sum_k \sum_\ell h_k g_\ell I_{2m-k,2n-\ell}\ , \\ [\mathcal{G}_2 I]_{mn} & = \sum_k \sum_\ell g_k h_\ell I_{2m-k,2n-\ell}\ , \\ [\mathcal{G}_3 I]_{mn} & = \sum_k \sum_\ell g_k g_\ell I_{2m-k,2n-\ell}\ . \end{array}\right. \tag{92}$$

On montre alors facilement que ces opérateurs satisfont une condition de reconstruction parfaite

$$\mathcal{H}^*\mathcal{H} + \mathcal{G}_1^*\mathcal{G}_1 + \mathcal{G}_2^*\mathcal{G}_2 + \mathcal{G}_3^*\mathcal{G}_3 = 1 \ . \tag{93}$$

Ainsi, on peut généraliser les schémas de codage en sous bandes: étant donn ée une image I_{mn}, on pose $s_0 = I$, et

$$s^1 = \mathcal{H}s^0 \quad d^{1,1} = \mathcal{G}_1 s^0 \ , \ d^{1,2} = \mathcal{G}_2 s^0 \ , \ d^{1,3} = \mathcal{G}_3 s^0 \ ,$$

et plus généralement

$$s^j = \mathcal{H}s^{j-1} \quad d^{j,1} = \mathcal{G}_1 s^{j-1} \ , \ d^{j,2} = \mathcal{G}_2 s^{j-1} \ , \ d^{j,3} = \mathcal{G}_3 s^{j-1} \ .$$

Comme dans le cas 1D, une image I est alors caractérisée par une approximation à une échelle grossière s^J, ainsi que par des “coefficients de détails” $d^{J,1}, d^{J,2}, d^{J,3}$ $d^{J-1,1}, d^{J-1,2}, d^{J-1,3}$... $d^{1,1}, d^{1,2}, d^{1,3}$.

Les algorithmes correspondants de décomposition et de reconstruction sont tout à fait similaires à ceux représentés en FIGURE 10, à ceci près qu'à chaque noeud de l'arbre de décomposition, il faut remplacer les deux branches (correspondant à H et G) par quatre branches (correspondant à $\mathcal{H}, \mathcal{G}_1, \mathcal{G}_2$ et $\mathcal{G}_2$).

5.2.6. *Bases d'ondelettes.* Jusqu'à présent, nous avons uniquement travaillé au niveau de signaux numériques, c'est à dire de suites. Or, le contexte dans lequel nous nous sommes placés depuis le début de ce chapitre est celui du codage par transformation, dans lequel la première étape (généralement aprés filtrage) est une décomposition du signal analogique par rapport à une base orthonormée. La question naturelle est alors la suivante: quel est le lien entre le codage en sous bandes et un schéma de codage par transformation? en d'autres termes, existe-t-il une base orthonormée sous-jacente dans un codage en sous bandes?

La réponse est positive, et est fournie par les bases orthonormées d'ondelettes. On peut en effet montrer le résultat suivant.

THÉORÈME:

1. *Il existe une fonction $\psi \in L^2(\mathbb{R})$, appelée* ondelette mère *telle que si l'on définit les fonctions (les ondelettes) ψ_{jk} par*

$$\psi_{jk}(t) = 2^{-j/2}\psi\big(2^{-j}t - k\big) \quad , \quad j,k \in \mathbb{Z} \tag{94}$$

la famille $\{\psi_{jk}, j,k \in \mathbb{Z}\}$ est une base orthonormée de $L^2(\mathbb{R})$.

2. *Par conséquent, pour tout $f \in L^2(\mathbb{R})$, il existe une unique décomposition*

$$f = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} d_k^j \psi_{jk} \ , \tag{95}$$

où les coefficients d_k^j sont donnés par

$$d_k^j = \langle f, \psi_{jk} \rangle = 2^{-j/2} \int_{-\infty}^{\infty} f(t)\overline{\psi}\left(2^{-j}t - k\right) \, dt \tag{96}$$

3. *Le calcul des coefficients d_k^j ci-dessus peut s'effectuer via un algorithme de codage en sous bandes.*

La preuve complète de ce résultat est assez longue et complexe, et repose sur la théorie des *Analyses multirésolution.* Nous renvoyons à [**1, 8, 14**] pour plus de détails.

5.2.7. *Les filtres à support borné de Daubechies.* La connection entre les bases d'ondelettes et le codage en sous bandes a été étudiée en détails par I. Daubechies [**1**], qui a construit une famille de filtres h et g de réponse impulsionnelle finie, associés à des bases orthonormées d'ondelettes. Les filtres h sont de longueur paire $N = 2L$, par défaut à support entre 0 et $N-1$, et les filtres g correspondants en sont déduits par une relation de type (88):

$$g_k = (-1)^k \overline{h}_{N-1-k} \ . \tag{97}$$

Des tables de coefficients peuvent être trouvées dans [**1**]. On montre que ces filtres sont effectivement associés à des bases d'ondelettes. Les fonctions ϕ et ψ correspondantes sont à support borné, et la taille du filtre N contrôle le support de ψ et ϕ, ainsi que son nombre de moments nuls: ψ est telle que

$$\int_{-\infty}^{\infty} t^m \psi(t)\, dt = 0 \ , \quad \forall m = 0, \dots L-1 \ . \tag{98}$$

REMARQUE: Cette dernière propriété a une grande importance en pratique, dans une perspective de compression de signaux: en effet, si ψ possède L moments nuls comme ci-dessus, et si un signal f se comporte comme un polynôme de degré inférieur ou égal à $L-1$ sur le support d'une ondelette ψ_{jk}, alors on a automatiquement $d_k^j = \langle f, \psi_{jk}\rangle = 0$. De telles ondelettes sont "aveugles" aux polynômes de degré inférieur à L, ce qui se traduit en pratique par le fait que beaucoup de coefficients d_k^j sont nuls. Dans ce cas, un codage par transformation sera extrêmement efficace. C'est en particulier le cas pour le codage des images: les images présentent des zones où elles varient extrêmement peu, et sont donc bien approximées par des polynômes de bas degré. Ces zones là seront caractérisées par un faible nombre de coefficients d_k^j significatifs.

Il existe également de nombreuses variantes, notamment basées sur des fonctions "splines".

5.2.8. *Quelques exemples de codeurs utilisant le codage en sous bandes.* Les techniques de codage en sous bandes ont rencontré un succès certain depuis leur invention, et sont mises en oeuvre dans un certain nombre de codecs actuels. On peut citer en particulier

1. Le codeur JPEG2000, successeur de JPEG, qui utilise un développement en sous bandes (bidimensionnel) de l'image à coder comme alternative au développement sur une base trigonométrique (après segmentation) utilisé dans la version précédente de JPEG.
2. Le codeur ATRAC, implémenté pour le codage des signaux audio sur les minidisques.
3. Le nouveau système développé par le FBI pour le stockage des empreintes digitales des sitoyens américains.

CHAPITRE 3

Codage entropique et codage de canal

Nous abordons maintenant le dernier volet de la majorité des schémas de codage et compression des signaux, à savoir les techniques qui associent un code binaire à une famille de symboles (par exemple, les échantillons d'un signal numérique dans le cas d'un codeur PCM, ou des coefficients d'un développement du signal sur une base orthonormée dans le cas d'un codeur par transformation). On parle parfois de *codes sans perte*, dans la mesure où contrairement aux deux étapes que nous avons déjà vues (échantillonnage et quantification), ce dernier volet n'induit a priori pas d'erreur.

Le principe du codage sans perte est d'associer un code binaire (c'est à dire une suite de zéros et de uns) à un "alphabet" de M symboles. Par exemple, étant donné un alphabet de 4 symboles $\{a, b, c, d\}$, on peut leur associer respectivement les quatre mots de longueur 2 suivants: 00, 01, 10 et 11. Ou pour un alphabet de $M = 2^R$ symboles, il est possible d'en coder les éléments en associant à chacun d'eux un mot de R bits. Il s'agit là de *codes de longueur constante*.

L'utilisation de tels codes se justifie dans de nombreux contextes. On se placera ici essentiellement dans les deux situations suivantes:

1. **Compression:** il s'agira, pour un alphabet $\mathcal{A} = \{a_0, \dots a_{M-1}\}$ donné, de trouver le code qui permette de représenter ces symboles de façon la plus compacte possible, en moyenne. Il faudra alors faire appel aux probabilités d'apparition de ces symboles.
2. **Détection et correction d'erreurs:** dans tout processus physique (comme par exemple des transmissions hertziennes ou de simples stockages sur disque), des erreurs s'introduisent. Il est possible, dans une certaine mesure, de construire des codes permettant de détecter de telles erreurs, ou mieux, de les corriger. Ceci présuppose que le code soit redondant, et se traduit donc par une augmentation du volume des données.

1. Compression et codage entropique

1.1. Généralités. On a déjà vu plus tôt des exemples de codes de longueur constante: partant d'un alphabet à $M = 2^R$ symboles, il est possible de caractériser chacun d'eux par une suite de R bits (zéros ou uns), et ce code est bijectif.

De là, le code peut être étendu aux suites de symboles: à une suite de symboles, disons $abcd$, on associe la suite de leurs codes binaires, concaténés. Là encore, ce code est décodable, dans la mesure où tous les mots binaires ayant même longueur, il est facile de voir que les R premiers bits correspondent au a, les R bits suivants correspondent au b, et ainsi de suite.

Dans un cadre plus général, on pose la définition suivante:

DÉFINITION: 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 $x_1 x_2 \dots x_n$ est la concaténation

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

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'arrètent les mots. On introduit donc une sous-classe de codes, appelés codes uniquement décodables, définis comme suit.

DÉFINITION: Le code est dit uniquement décodable si le codeur α est injectif: c'est à dire si toute suite binaire $\alpha(x_1 x_2 \dots x_n)$ n'est l'image que d'un et un seul mot, à savoir $x_1 x_2 \dots x_n$.

EXEMPLE 3.1. On considère les deux codeurs α_1 et α_2 et α_3 définis par les tableaux donnés dans la TABLE 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

TAB. 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.*

1.2. 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: 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.*

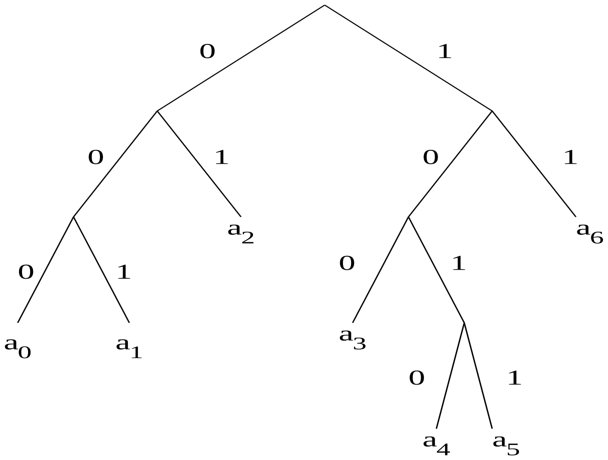

FIG. 1. *Exemple d'arbre binaire associé à un alphabet.*

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

TAB. 2. *Code associé à l'arbre binaire*

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.

Il est utile d'avoir des informations sur les longueurs des mots qui sont possibles pour un alphabet donné. L'inégalité de Kraft donne de telles informations. Dans le cas de codes de longueur constante, si l'alphabet possède $M = 2^R$ symboles, chaque mot $\alpha(a)$ est de longueur $\ell(a) = R$, de sorte que

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

Cette relation est en fait une borne pour les codes de longueur variable uniquement décodables, comme l'exprime le résultat suivant

THÉORÈME: 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:

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

Ce premier théorème fournit donc une condition nécessaire pour l'unique décodabilité d'un code. Cette condition est en fait également suffisante dans un certain sens, comme l'exprime le théorème suivant.

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

1.3. L'entropie. Pour optimiser les longueurs des mots binaires associés aux symboles de l'alphabet, il est nécessaire de tenir compte des probabilités d'occurrence de ces symboles. Partant d'un alphabet

$$\mathcal{A} = \{a_0, a_1, \ldots a_{M-1}\} \ ,$$

on notera $p(a)$ la probabilité d'occurrence du symbole a.

Il est aussi utile d'introduire l'entropie de la distribution de probabilités p:

(2) $$H(p) = -\sum_{a \in A} p(a) \log_2(p(a)) \ .$$

L'entropie représente la quantité d'information reçue lorsqu'une valeur de X est observée. Mathématiquement, on a

$$0 \leq H(p) \leq \log_2 (\mathrm{Card}(A)) \ ;$$

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

De là, on considère la longueur moyenne $\overline{\ell}(\alpha)$ des mots codés par un codeur α,

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

Le résultat suivant permet d'obtenir des bornes sur les longueurs moyennes des mots, tenant compte des probabilités des symboles.

> THÉORÈME: Soit $A = \{a_0, \ldots, a_{M-1}\}$ un alphabet, et soit p la distribution de probabilités de ses symboles.
>
> 1. Soit α un codeur dont les longueurs de mots satisfont à l'inégalité de Kraft. Alors on a
>
> (4) $$\overline{\ell}(\alpha) \geq H(p) \ ,$$
>
> avec égalité si et seulement si $p(a) = 2^{-\ell(a)}$ pour tout $a \in A$.
>
> 2. Il existe un code uniquement décodable tel que
>
> (5) $$\overline{\ell}(\alpha) < H(p) + 1 \ .$$

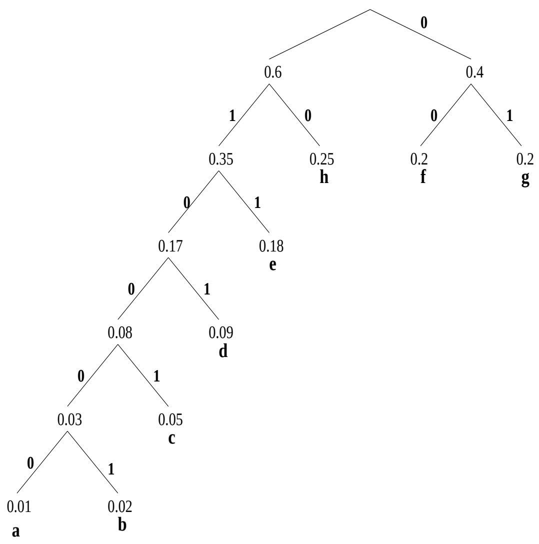

FIG. 2. *Arbre de préfixes pour les symboles de la Table 3*

1.4. 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 $(p(a_0), p(a_1), \dots p(a_{M-1}))$, on suppose (sans perte de généralité)que les symboles sont ordonnés par probabilités décroissantes: $p(a_0) \geq p(a_1) \geq \dots$ (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 $\overline{a}$ auquel on affecte la probabilité

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

> THÉORÈME: 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 $\{a_0, a_1, \dots a_{M-3,\overline{a}}\}$, en adjoignant à ce dernier deux branches liant $\overline{a}$ 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

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

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 que le code de Huffman est le code à préfixe optimal, et satisfait en particulier la borne (5).

EXEMPLE 3.2. Prenons l'exemple suivant d'un alphabet de 8 symboles, de probabilités données dans la table 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

TAB. 3. *Exemple de code de Huffman.*

L'arbre de préfixes correspondant se trouve en FIGURE 2.

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.

2. Codage de canal, codes détecteurs/correcteurs d'erreurs

Les systèmes de codage que nous avons rencontrés jusquà présent sont des systèmes de *codage de source*, et ne prennent pas en compte les notions de *canal de transmission*. Un canal de transmission, qu'il s'agisse d'un canal Hertzien ou de la gravure sur un CD, génère nécessairement des erreurs. Le *codage de canal* introduit explicitement ces risques d'erreurs dans le système de codage, et donne la possibilité de détecter, voire de corriger, une certaine quantité d'erreurs.

Bien évidemment, il est possible de détecter ou de corriger des erreurs uniquement dans les cas où il existe des restrictions sur les messages possibles, donc si le message comporte une certaine quantité de redondance. Le codage de canal consiste à se donner un modèle probabiliste pour les erreurs, et optimiser un codeur en fonction de ce modèle.

2.1. Canal bruité sans mémoire. On modélise un canal de la façon suivante: étant donnés un alphabet d'entrée de M symboles $\mathcal{A}_X = \{a_0, a_1, \dots a_{M-1}\}$, dont les probabilités $\mathbb{P}\{X = a\}$, $a \in \mathcal{A}_X$ sont connues et un alphabet de sortie

$\mathcal{A}_Y = \{b_0, b_1, \dots b_{M-1}\}$, un canal bruité sans mémoire est caractérisé par l'ensemble de *probabilités conditionnelles*

$$Q_{j|i} = \mathbb{P}\{Y = b_j | X = a_i\} \tag{6}$$

(lire : probabilité que la variable aléatoire Y prenne la valeur b_j, sachant que la variable aléatoire X a pris la valeur a_i), aussi appelées *probabilités de transition.*

Il est important de rappeler deux résultats essentiels concernant les probabilités conditionnelles:

$$\sum_j \mathbb{P}\{Y = b_j | X = a_i\} = 1 \ , \qquad \text{pour tout } i \ , \tag{7}$$

et la formule de Bayes: pour tous $x \in \mathcal{A}_X, y \in \mathcal{A}_Y$,

$$\mathbb{P}\{X = x \text{ et } Y = y\} = \mathbb{P}\{X = x | Y = y\}\mathbb{P}\{Y = y\} \tag{8}$$

<u>EXEMPLES:</u>

- Canal binaire symétrique: on se donne $\mathcal{A}_X = \mathcal{A}_Y = \{0, 1\}$, et un nombre $q \in [0, 1]$, qui représente la probabilité d'erreur. On suppose que le canal commet une erreur avec probabilité q. On a donc $\mathbb{P}\{Y = 0|X = 0\} = \mathbb{P}\{Y = 1|X = 1\} = 1 - q$ et $\mathbb{P}\{Y = 0|X = 1\} = \mathbb{P}\{Y = 1|X = 0\} = q$.
- La machine à écrire détraquée: $\mathcal{A}_X = \mathcal{A}_Y = \{a, b, \dots z\}$ représentent les lettres de l'alphabet, et on suppose que lorsqu'une touche est frappée, la bonne lettre est écrite avec probabilité $1/3$, et ses deux voisines sont écrites avec probabilité $1/3$ chacune. Par exemple,

$$\mathbb{P}\{Y = a|X = b\} = \mathbb{P}\{Y = b|X = b\} = \mathbb{P}\{Y = c|X = b\} = 1/3 \ .$$

- Plein d'autres...

Le problème du décodage est le suivant: étant donné un symbole reçu y, comment deviner celui qui a été émis ?

La technique du *maximum a posteriori* permet de donner une réponse acceptable à cette question. En effet, supposons que le canal (c'est à dire l'ensemble des probabilités de transition) soit connu, ainsi que la distribution de probabilités des symboles d'entrée $a \in \mathcal{A}_X$. Alors, d'après la formule de Bayes, on peut écrire

$$\begin{aligned}
\mathbb{P}\{X = x|Y = y\} &= \frac{\mathbb{P}\{X = x \text{ et } Y = y\}}{\mathbb{P}\{Y = y\}} \\
&= \frac{\mathbb{P}\{Y = y|X = x\}\mathbb{P}\{X = x\}}{\mathbb{P}\{Y = y\}} \\
&= \frac{\mathbb{P}\{Y = y|X = x\}\mathbb{P}\{X = x\}}{\sum_{y'} \mathbb{P}\{Y = y'|X = x\}\mathbb{P}\{X = x\}} \ .
\end{aligned}$$

Tous les termes du membre de droite de la dernière équation sont complètement calculables si on s'est donné les probabilités de transition et les probabilités $\mathbb{P}\{X = a\}, a \in \mathcal{A}_X$. Ainsi, pour une observation y donnée, on peut en déduire l'entrée $x \in \mathcal{A}_X$ dont la probabilité *sachant l'observation* $\mathbb{P}\{X = x|Y = y\}$ est maximale.

Prenons l'exemple d'un canal binaire symétrique, associé à la probabilité $q = 0,15$, et des symboles d'entrée 0 et 1, de probabilités $\mathbb{P}\{X = 0\} = 0,9$ et $\mathbb{P}\{X = 1\} =$

$0,1$. Supposons qu'après transmission, on reçoive le symbole $Y=1$, comment inférer quel est le symbole émis? Il faut pour cela calculer la probabilité *a postériori*

$$\begin{aligned}\mathbb{P}\{X=1|Y=1\} &= \frac{\mathbb{P}\{Y=1|X=1\}\mathbb{P}\{X=1\}}{\sum_{y'}\mathbb{P}\{Y=y'|X=1\}\mathbb{P}\{X=1\}}\\ &= \frac{0,85\times 0,1}{(0,85\times 0,1)+(0,15\times 0,9)}\\ &= 0,39\ .\end{aligned}$$

Ainsi, bien que le signal reçu soit $Y=1$, il est tout de même plus probable que ce soit $X=0$ qui ait été émis, et le décodeur décidera de choisir la solution $X=0$.

2.2. Information, Information mutuelle, et le grand théorème de Shannon.

2.2.1. *Questions d'information.* On a déjà rencontré plus tôt la notion d'entropie, introduite en tant que mesure d'information: étant donné un symbole aléatoire X, prenant ses valeurs dans un alphabet $\mathcal{A}_X=\{a_0,a_1,\dots a_{M-1}\}$, avec probabilités $\mathbb{P}\{X=a\}$, $a\in\mathcal{A}_X$, son entropie est donnée par

$$H(X)=-\sum_{i=0}^{M-1}\mathbb{P}\{X=a_i\}\log_2(\mathbb{P}\{X=a_i\})\ , \tag{9}$$

et caractérise la quantité d'information portée par X (en d'autres termes, le nombre de bits nécessaire pour coder X. De même, étant donnés deux symboles aléatoires X et Y à valeurs dans les alphabets $\mathcal{A}_X=\{a_0,a_1,\dots a_{M-1}\}$ et $\mathcal{A}_Y=\{b_0,b_1,\dots b_{M-1}\}$, et leur distribution jointe donnée par les nombres $\mathbb{P}\{X=a_i,Y=b_j\}$, on peut associer à cette dernière l'entropie correspondante, notée

$$H(X,Y)=-\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\mathbb{P}\{X=a_i,Y=b_j\}\log_2(\mathbb{P}\{X=a_i,Y=b_j\})\ , \tag{10}$$

et qui caractérise l'information contenue dans le couple (X,Y).

Pour caractériser la quantité d'information portée par X, sachant Y, on a recours à la notion d'*entropie conditionnelle*:

$$H(X|Y)=-\sum_i\sum_j\mathbb{P}\{X=a_i,Y=b_j\}\log_2(\mathbb{P}\{X=a_i|Y=b_j\})\ . \tag{11}$$

Il est facile de voir, en utilisanty la formule de Bayes, que l'entropie conditionnelle est aussi donnée par

$$H(X|Y)=-\sum_j\mathbb{P}\{Y=b_j\}\sum_i\mathbb{P}\{X=a_i|Y=b_j\}\log_2(\mathbb{P}\{X=a_i|Y=b_j\})$$

(notons que l'entropie conditionnelle n'est pas symétrique: $H(X|Y)\neq H(Y|X)$). Etant donné un canal bruité, on peut remarquer que plus l'entropie conditionnelle $H(Y|X)$ est faible, moins la perte d'information à travers le canal est importante. En fait, pour quantifier la déperdition d'information à travers un canal, on a recours à la notion d'*information mutuelle*:

$$I(X,Y)=H(X)-H(X|Y)=H(Y)-H(Y|X) \tag{12}$$

(on note que l'information mutuelle est quant à elle symétrique: $I(X,Y) = I(Y,X)$).

> DÉFINITION: étant donné un canal bruité sans mémoire, caractérisé par ses probabilités de transition $\mathbb{P}\{Y = y|X = x\}$, la capacité du canal est donné par la valeurs maximale de l'information mutuelle, le maximum étant pris sur toutes les distributions de probabilités possibles pour X

$$C = \max_X I(X,Y) \ . \tag{13}$$

Comme on va le voir avec le théorème de Shannon, la capacité d'un canal mesure la quantité maximale d'information (c'est à dire le nombre de bits) que l'on peut transmettre sans erreur par le canal par unité de temps.

Reprenons l'exemple précédent du canal binaire symétrique, avec $q = 0,15$, et $\mathbb{P}\{X = 0\} = 0,9$. Un calcul explicite donne une information mutuelle égale à $I = 0,15$. Plus généralement, en posant $p = \mathbb{P}\{X = 0\}$, et en maximisant $I(X,Y)$ par rapport à p, on peut montrer que le maximum de $I(X,Y)$ est atteint pour $p = 0,5$, ce qui fournit une information mutuelle $C = \max_X I(X,Y) = 0{,}39$.

2.2.2. *Codes par blocs, et le théorème de Shannon.* On se pose maintenant le problème de construire des schémas de codage capables de corriger des erreurs. On a vu que cette possibilité est directement liée à l'introduction de redondance dans le message transmis. L'exemple le plus simple est celui des codes à répétition: chaque bit est transmis plusieurs fois.

Prenons l'exemple d'un *code par triplication*: supposons qu'on utilise un canal binaire symétrique, de probabilité d'erreur q. A chaque bit $X = b \in \{0,1\}$ on associe le mot de 3 bits bbb, qu'on transmet, éventuellement avec erreurs. Le décodeur quant à lui, à partir du mot de 3 bits reçu Z lui associe le bit majoritaire (c'est à dire représenté 2 ou 3 fois). La probabilité d'erreur de ce schéma, qui correspond au cas 2 ou 3 bits faux sur les 3, vaut alors

$$q^3 + 3q^2(1-q) \ ,$$

ce qui si q est assez petit, est inférieur à q.

Il existe des façons plus subtiles et efficaces d'introduire de la redondance, comme on le verra. On utilise pour cela la notion de code par blocs: un code par blocs associe à des mots binaires de longueur donnée 2^K d'autres mots sensiblement plus longs, tels que leur redondance permette la correction d'une certaine quantité d'erreurs.

> DÉFINITION: Etant donné un alphabet $\mathcal{A}_X$ de longueur $M = 2^K$, un code par blocs (N,K) associe à tout $a \in \mathcal{A}_X$ un mot binaire de longueur constante N:

$$a \in \mathcal{A}_X \longrightarrow \alpha(a) \in \{0,1\}^N \tag{14}$$

Le débit du code est donné par

$$R = \frac{K}{N} \ . \tag{15}$$

Un décodeur associe à tout mot binaire b de longueur N un élément $\beta(b)$ de $\mathcal{A}_X$.

La probabilité d'erreur de bloc est donnée par

$$p_B = \sum_{a \in \mathcal{A}_X} \mathbb{P}\{X = a\} \mathbb{P}\{\beta(\alpha(X)) \neq a | X = a\} \ , \tag{16}$$

et la probabilité d'erreur de bloc maximale est

$$p_{BM} = \max_{a \in \mathcal{A}_X} \mathbb{P}\{X = a\} \mathbb{P}\{\beta(\alpha(X)) \neq a | X = a\} \ , \tag{17}$$

Une sous classe particulièrement intéressante est celle des *codes par blocs linéaires*, pour lesquels tous les mots binaires $\alpha(a)$ appartiennent à un sous-espace linéaire de $\{0,1\}^N$. Supposons pour simplifier que $\mathcal{A}_x = \{0,1\}^K$. En représentant un élément de $\mathcal{A}_X$ par un vecteur colonne $\underline{s}$ dont les K éléments sont des zéros ou des uns, et un élément de $\{0,1\}^N$ par un vecteur colonne $\underline{s}'$ de longueur N, on peut alors utiliser une forme matricielle:

$$\underline{s}' = \mathbf{G}\underline{s} \, [\mathrm{mod}\ 2] \ , \tag{18}$$

où $\mathbf{G}$ est une matrice $N \times K$, et le produit matrice-vecteur est effectué “modulo 2”.

Le prototype des codes correcteurs d'erreurs est donné par les codes de Hamming: un mot binaire de longueur K est codé sur $N > K$ bits, les bits supplémentaires étant des bits de vérification de parité. Un exemple simple est donné par le code $(7,4)$ de Hamming, qui code donc des mots de longueur 4 par des mots de longueur 7, grace à 3 bits de parité: la matrice $\mathbf{G}$ correspondante est

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Par exemple, l'image $\underline{s}' = \mathbf{G}\underline{s}$ par $\mathbf{G}$ du mot $\underline{s} = 1010$ est le mot $\underline{s}' = 1010010$.

Le théorème de Shannon démontre que pour tout canal bruité sans mémoire, il existe une limite naturelle au débit atteignable par un code

Théorème: On considère un canal bruité sans mémoire, dont on note C la capacité. Alors pour tout $\epsilon > 0$, pour tout $R < C$ et pour tout N assez grand, il existe un code de longueur N et de débit inférieur ou égal à R, et un décodeur correspondant, tels que l'erreur de bloc soit inférieure ou égale à ϵ.

Malheureusement, la preuve du théorème de Shannon n'est pas constructive, de sorte que le code optimal dont Shannon prouve l'existence est généralement inconnu. De plus, même lorsque le codeur et le décodeur sont inconnus, il n'existe pas d'algorithme efficace pour effectuer le décodage.

2.2.3. *Codes correcteurs courants.* Comme on l'a vu, les codes de Hamming constituent les prototypes des codes correcteurs ou détecteurs. Un code de Hamming est capable de détecter deux erreurs, ou d'en corriger une seule. Pour cela, le nombre de vérifications de parité doit être suffisamment grand. La règle est la suivante: en notant K le nombre de bits à coder, et $P = N - K$ le nombre de bits de parité, on doit avoir

$$N + 1 = K + P + 1 \leq 2^P \ .$$

L'idée est que P bits de parité permettent de coder 2^P nombres, qui peuvent représenter les positions de l'erreur potentielle dans les $N = K + P$ bits. Il suffit alors de bien choisir les vérifications de parité pour que la détection des erreurs soit sans ambiguité. Ceci est effectué de la façon suivante: on construit une matrice $\mathbf{H}$, qui est une matrice $P \times N$, qui permet de localiser l'erreur, de la façon suivante: si aucune erreur n'a été commise, le vecteur colonne (à P éléments) $\mathbf{H}\underline{u}$ est le vecteur nul. Si une seule erreur a été commise (c'est à dire que si le vecteur de longueur $\underline{s}' = \mathbf{G}\underline{s}$ a été remplacé par un vecteur $\underline{u}$ qui diffère d'un ou zéro bit), le vecteur colonne $\mathbf{H}\underline{u}$ coïncide avec la colonne de $\mathbf{H}$ dont la position est identique à celle de l'erreur.

Revenant par exemple au code (7,4) vu plus haut, considérons la matrice

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

dont les colonnes représentent les sept premiers entiers non nuls. Considérons le mot binaire $\underline{s} = 1010$, qui est donc codé comme $\underline{s}' = 1010010$. En appliquant $\mathbf{H}$ à $\underline{s}'$, on obtient $\mathbf{H}\underline{s}' = 000$, qui indique qu'aucune erreur n'a été commise.

Par contre, supposons qu'une erreur ait été effectuée sur le deuxième bit (en partant de la gauche), c'est à dire considérons $\underline{u} = 1110010$. Alors, on obtient $\mathbf{H}\underline{u} = 110$, qui est est la deuxième colonne de $\mathbf{H}$; ceci signifie que l'erreur est en deuxième position, et que le mot initial était 1010. De même, prenant $\underline{v} = 1010000$, on obtient $\mathbf{H}\underline{v} = 010$, qui est la sixième colonne de $\mathbf{H}$, et indique que le sixième bit est erroné. Donc $\underline{v}$ doit être remplacé par 1010010, qui correspond à 1010.

La plupart des codes couramment utilisés sont des codes linéaires. Ceci signifie que l'étape de codage est simple. Ceci ne signifie pas pour autant que le décodage soit simple: il s'agit en général d'un problème NP-complet, de sorte que la solution demande probablement un temps exponentiel. C'est pour cela qu'on se limite généralement aux codes pour lesquels il existe un algorithme rapide de décodage.

On appelle code (N,K,T) un code qui code K bits dans des mots de longueur N, et tels qu'il existe un algorithme de décodage qui peut corriger jusquà T erreurs.

Les codes de Reed-Muller sont des généralisations des codes de Hamming. Par exemple, considérons la matrice 16×11

$$\mathbf{G}_{16} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \end{pmatrix}$$

En se limitant à la première colonne $\mathbf{G}_1$ de la matrice, on voit facilement qu'on obtient un code par répétition (16,1,7). En se limitant à la matrice $\mathbf{G}_5$ formée des 5 premières colonnes, on obtient un code (16,5,3). En utilisant les 11 colonnes, on obtient un code (16,11,1). Le décodeur correspondant utilise certaines propriétés de périodicité des lignes de cette matrice, et se ramène en fait à un algorithme de la famille des algorithmes de FFT.

Les codes les plus couramment utilisés sont les codes de Reed-Solomon, qui sont des généralisations des codes de Hamming. Bien qu'étant toujours assez éloignés des performances optimales telles que formulées dans le théorème de Shannon, ils permettent d'atteindre des performances respectables. Par exemple, les codes de Reed-Solomon utilisés pour les compact disques sont capables de corriger des paquets de bits erronés jusquà 4000 bits consécutifs.

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: *Les distributions*
[4] A. Gersho et D. Gray (1992): Vector quantization
[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 rélle 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.