

Traitement du signal

Stéphane Mallat, Eric Moulines, François Roueff

October 22, 2015

Table des Matières

I Analyse de Fourier et Filtrage	7
1 Transformée de Fourier: propriétés de base	9
1.1 Transformée de Fourier sur $L_1(\mathbb{R})$	10
1.2 Décroissance et dérivation	12
1.3 Un exemple remarquable	13
1.4 Quelques exemples classiques	14
1.5 Formulaire	15
1.6 L'espace de Schwartz	17
1.7 Formules d'inversion	18
2 Convolution et Filtrage analogique	21
2.1 Définition	22
2.2 Convolution dans $L_1(\mathbb{R})$	22
2.3 Convolution dans $L_p(\mathbb{R})$	24
2.4 Une première extension de la transformation de Fourier inverse	27
2.5 Convolution et transformée de Fourier dans $L_1(\mathbb{R})$	27
2.6 Applications aux filtres analogiques gouvernés par une équation différentielle	28
3 Transformée de Fourier-Plancherel	31
3.1 Espace des fonctions de carré intégrable	32
3.2 Transformée de Fourier sur $L_2(\mathbb{R})$	33
3.3 Application au calcul de transformées de Fourier	35
3.4 Principe d'incertitude : résolution en temps et en fréquence	35
3.5 Convolution et transformation de Fourier dans $L_2(\mathbb{R})$	35
4 Echantillonnage	37
II Bases de traitement du signal à temps discret	43
5 Transformée de Fourier discrète	45
5.1 La transformée de Fourier sur $\ell^1(\mathbb{Z})$	46
5.2 La transformée de Fourier sur $\ell^2(\mathbb{Z})$	48
5.3 Transformée de Fourier Discrète ou TFD	52

5.4	Transformée de Fourier à Court Terme (TFCT)	62
6	Transformée en z et filtrage	67
6.1	Transformée en z : définition	68
6.2	Transformée inverse	70
6.3	Décompositions en fractions simples	71
6.4	Propriétés de la transformée en z	73
6.5	Pôles et zéros	77
6.6	Fonction de transfert	78
6.7	Causalité et Stabilité	79
6.8	Équations aux différences	80
III	Bases de traitement du signal aléatoire	85
7	Introduction au signal aléatoire à temps-discret	87
7.1	Introduction	88
7.2	Définition et construction de la loi d'un processus aléatoire	89
7.3	Stationnarité stricte d'un processus à temps discret	95
7.4	Processus du second ordre	99
7.5	Covariance d'un processus stationnaire au second ordre	99
7.6	Mesure spectrale d'un processus stationnaire	106
8	Filtrage des signaux aléatoires à temps-discret	111
8.1	Filtrages linéaires de processus au second ordre	112
8.2	Processus ARMA	115
9	Prédiction des signaux aléatoires	129
9.1	Prédiction linéaire de processus stationnaires	130
9.2	Algorithme de Levinson-Durbin	134
9.3	Algorithme des innovations	138
IV	Information et codage de signaux et d'images	141
10	Éléments de théorie de l'information	143
10.1	Complexité et Entropie	144
10.2	Quantification scalaire	152
11	Application au codage de parole	157
12	Application au codage d'images	159

<i>TABLE DES MATIÈRES</i>	5
V Compléments mathématiques	163
A Bases d'analyse Hilbertienne	165
A.1 Définitions	166
A.2 Famille orthogonale et orthonormal	170
A.3 Séries de Fourier	174
A.4 Projection et principe d'orthogonalité	177
A.5 Isométries et isomorphismes d'espaces de Hilbert	181
B Une autre approche de la prédiction	183
C Transformée de Fourier rapide	185
VI Documents	187
VII Production du signal de parole	189

Part I

Analyse de Fourier et Filtrage

Chapitre 1

Transformée de Fourier: propriétés de base

1.1 Transformée de Fourier sur $L_1(\mathbb{R})$

Nous abordons dans cette partie la définition et les propriétés de la transformée de Fourier des fonctions.

Définition 1 (Transformée de Fourier). Soit $f \in L_1(\mathbb{R})$. On pose, pour tout $\xi \in \mathbb{R}$,

$$[\mathcal{F}(f)](\xi) = \hat{f}(\xi) = \int_{\mathbb{R}} e^{-i2\pi\xi x} f(x) dx \quad (1.1)$$

$$[\overline{\mathcal{F}}(f)](\xi) = \int_{\mathbb{R}} e^{i2\pi\xi x} f(x) dx \quad (1.2)$$

On appelle $\mathcal{F}(f)$ (noté aussi $\mathcal{F}f$) la Transformée de Fourier de f et $\overline{\mathcal{F}}(f)$ (noté aussi $\overline{\mathcal{F}}f$) la transformée de Fourier conjuguée de f .

Cette intégrale a un sens pour $f \in L_1(\mathbb{R})$, parce que $x \mapsto e^{-i2\pi\xi x} f(x)$ est alors aussi dans $L_1(\mathbb{R})$ pour tout $\xi \in \mathbb{R}$.

Exemple 2. Soit $f = \mathbb{1}_{[a,b]}(x)$, la fonction indicatrice de l'intervalle $[a,b]$. Un calcul immédiat montre que

$$\hat{f}(\xi) = \begin{cases} b-a & \xi = 0, \\ \frac{\sin \pi(b-a)\xi}{\pi\xi} e^{-i\pi(a+b)\xi} & \xi \neq 0. \end{cases}$$

On remarque que $\hat{f} \notin L_1(\mathbb{R})$. En revanche c'est une fonction continue bornée telle que $\lim_{|\xi| \rightarrow \infty} \hat{f}(\xi) = 0$.

En fait les propriétés décrites pour \hat{f} dans cet exemple sont des propriétés générales des fonctions de $\mathcal{F}(L^1(\mathbb{R}))$ comme le montre le résultat suivant.

Théorème 3 (Rieman-Lebesgue). Etant donné $f \in L_1(\mathbb{R})$ on a

1. $\mathcal{F}f$ est une fonction continue et bornée sur \mathbb{R} ,
2. \mathcal{F} est un opérateur linéaire et continu de $(L_1(\mathbb{R}), \|\cdot\|_1)$ dans $(C_\infty, \|\cdot\|)$ (l'espace des fonctions continues munie de la norme sup) et $\|\hat{f}\| \leq \|f\|_1$,
3. $\lim_{|\xi| \rightarrow \pm\infty} |\hat{f}(\xi)| = 0$.

Proof. 1. La fonction $\xi \mapsto e^{-i2\pi\xi x} f(x)$ est continue sur \mathbb{R} et majorée en module par $|f(x)|$ (qui ne dépend pas de ξ), qui est dans $L_1(\mathbb{R})$. On conclut en appliquant le théorème de convergence dominée.

2. La linéarité de \mathcal{F} découle directement de la linéarité de l'intégrale. Pour tout $\xi \in \mathbb{R}$, on a $|\hat{f}(\xi)| \leq \int |f(x)| dx = \|f\|_1$. On en déduit que \hat{f} est bornée par $\|f\|_1$ et que \mathcal{F} est continue.
3. Supposons tout d'abord que f est continue à support compact (il existe $M > 0$ tel que $f(x) = 0$ si $|x| > M$). Par changement de variable $x = t - 1/(2\xi)$ dans (1.1), on a, pour tout $\xi \neq 0$,

$$\hat{f}(\xi) = \int_{\mathbb{R}} e^{-i2\pi\xi t - i\pi} f(t + 1/(2\xi)) dt = - \int_{\mathbb{R}} e^{-i2\pi\xi t} f(t + 1/(2\xi)) dt.$$

D'où l'expression, en sommant cette équation avec (1.1), pour tout $\xi \neq 0$,

$$2\hat{f}(\xi) = \int_{\mathbb{R}} e^{-i2\pi\xi x} (f(x) - f(x + 1/(2\xi))) dx$$

Il s'en suit, par convergence dominée (en observant que $|f(x) - f(x + 1/(2\xi))|$ est majoré indépendamment de x et ξ et est nul pour $x \notin [-M-1, M+1]$ pour tout $|\xi| \geq 1$),

$$\lim_{|\xi| \rightarrow \pm\infty} |\hat{f}(\xi)| \leq \frac{1}{2} \lim_{|\xi| \rightarrow \pm\infty} \int_{\mathbb{R}} |f(x) - f(x + 1/(2\xi))| dx = 0.$$

Soit maintenant $f \in L_1(\mathbb{R})$. Il existe une suite $\{g_n\}$ de fonctions continues dans $L_1(\mathbb{R})$ telles que $\|f - g_n\|_1 \rightarrow 0$. Comme, $\|\hat{f} - g_n\|_\infty \leq \|f - g_n\|_1$ et $\lim_{\xi \rightarrow \pm\infty} g_n(\xi) = 0$, on en déduit aisément que $\lim_{\xi \rightarrow \pm\infty} \hat{f}(\xi) = 0$.

□

La propriété à la fois la plus immédiate et la plus fondamentale de la transformée de Fourier est son effet sur les translations.

Proposition 4 (Transformée de Fourier et translation). *Soit $f \in L_1(\mathbb{R})$. Alors, pour tout $t \in \mathbb{R}$, les fonctions $x \mapsto f(x - x_0)$ et $x \mapsto e^{i2\pi\xi_0 x} f(x)$ sont dans $L_1(\mathbb{R})$ et vérifient*

1. $[\mathcal{F}(x \mapsto f(x - x_0))](\xi) = e^{-i2\pi x_0} \hat{f}(\xi)$ pour tout $\xi \in \mathbb{R}$;
2. $[\mathcal{F}(x \mapsto e^{i2\pi\xi_0 x} f(x))](\xi) = \hat{f}(\xi - \xi_0)$ pour tout $\xi \in \mathbb{R}$.

Proof. La preuve élémentaire est laissée aux lecteurs. □

Une des propriétés remarquables de la transformée de Fourier est d'échanger la dérivation et la multiplication par un monôme

Proposition 5 (Transformée de Fourier et Dérivation). *Soit n un entier naturel.*

1. Si $x \mapsto x^k f(x)$ est dans $L_1(\mathbb{R})$ pour tout $k = 0, 1, \dots, n$, alors \hat{f} est n fois continûment dérivable et on a

$$\hat{f}^{(n)} = \mathcal{F}(x \mapsto (-2i\pi x)^n f(x))$$

2. Si f est n fois continûment dérivable avec $f^{(k)} \in L_1(\mathbb{R})$ pour tout $k = 0, 1, \dots, n$, alors

$$[\mathcal{F}(f^{(n)})](\xi) = (2i\pi\xi)^n \hat{f}(\xi) \quad \text{pour tout } \xi \in \mathbb{R}.$$

Proof. Dans les deux cas, il suffit de démontrer le résultat pour $n = 1$ puis d'appliquer une récurrence évidente.

1. La fonction $h : \xi \mapsto e^{-i2\pi\xi x} f(x)$ est continûment dérivable et $h'(\xi) = -2i\pi x e^{-i2\pi\xi x} f(x)$. De plus $|h'(\xi)| \leq 2\pi|x f(x)|$. Le résultat découle du théorème de dérivation sous le signe somme.

2. Comme $f' \in L_1(\mathbb{R})$, on peut calculer $\mathcal{F}(f')$ par la formule

$$[\mathcal{F}(f')](\xi) = \lim_{a \rightarrow \infty} \int_{-a}^a e^{-i2\pi\xi x} f'(x) dx, \quad \xi \in \mathbb{R}.$$

De plus, par intégration par parties, pour tout $\xi \in \mathbb{R}$ et tout $a > 0$,

$$\int_{-a}^{+a} e^{-i2\pi\xi x} f'(x) dx = [e^{-i2\pi\xi x} f(x)]_{-a}^a + \int_{-a}^a (2i\pi\xi) e^{-i2\pi\xi x} f(x) dx.$$

Comme $f' \in L_1(\mathbb{R})$ et $f(a) = f(0) + \int_0^a f'(t) dt$, $\lim_{a \rightarrow \infty} \int_0^a f'(t) dt$ existe et donc $\lim_{a \rightarrow \infty} f(a)$ existe. Cette limite est nécessairement nulle car $f \in L_1(\mathbb{R})$. De la même façon, $\lim_{a \rightarrow \infty} f(-a) = 0$. D'où le résultat.

□

La proposition suivante sera très utile pour établir des formules d'inversion de la transformée de Fourier.

Proposition 6. Soit f et g deux fonctions de $L_1(\mathbb{R})$. Alors $f\hat{g}$ et $\hat{f}g$ sont dans $L_1(\mathbb{R})$ et on a

$$\int f(x)\hat{g}(x) dx = \int \hat{f}(x)g(x) dx. \quad (1.3)$$

Proof. Comme $\hat{g} \in L_\infty(\mathbb{R})$, les fonctions $f\hat{g}$ et $\hat{f}g$ appartiennent à $L_1(\mathbb{R})$. Comme la fonction $(t, x) \mapsto e^{-i2\pi tx} f(t)g(x) \in L_1(\mathbb{R}^2)$, il résulte du théorème de Fubini que

$$\begin{aligned} \int f(t)\hat{g}(t) dt &= \int f(t) \left(\int e^{-i2\pi tx} g(x) dx \right) dt = \\ &\quad \int g(x) \left(\int e^{-i2\pi tx} f(t) dt \right) dx = \int g(x)\hat{f}(x) dx. \end{aligned}$$

□

1.2 Décroissance et dérivation

Nous avons observé dans la partie précédente dès le premier exemple de calcul de transformée de Fourier que $L_1(\mathbb{R})$ n'est pas stable sous l'effet de \mathcal{F} . Nous allons introduire un sous-espace de $L_1(\mathbb{R})$ stable par transformation de Fourier, dérivation et multiplication par un polynôme. Cet espace introduit par Laurent Schwartz et que l'on notera \mathcal{S} joue un rôle essentiel en analyse de Fourier.

Définition 7 (Fonction à décroissance rapide). Une fonction $f : \mathbb{R} \rightarrow \mathbb{C}$ est dite à décroissance rapide si, pour tout $p \in \mathbb{N}$, on a

$$\lim_{|x| \rightarrow \infty} |x|^p |f(x)| = 0.$$

C'est le cas par exemple de $f(x) = e^{-|x|}$. Mais on notera que contrairement à son nom, cette définition n'implique aucune monotonie pour f même dans un voisinage de l'infini (prendre par exemple $f(x) = e^{-|x|} \sin x$). Une propriété utile sur l'intégrabilité des fonctions à décroissance rapide est la suivante.

Lemme 8. *Si f est une fonction de $L_{1\text{loc}}(\mathbb{R})$ à décroissance rapide alors pour tout $p \in \mathbb{N}$, $x \mapsto x^p f(x)$ appartient à $L_1(\mathbb{R})$.*

Proof. L'indice “loc” signifie que la restriction de f à tout compact est dans $L_1(\mathbb{R})$. f étant à décroissance rapide, il existe $M > 0$ tel que pour tout $|x| \geq M$, on ait $|x|^{p+2}|f(x)| \leq 1$. D'où

$$\begin{aligned} \int |x^p f(x)| dx &\leq \int_{|x| \leq M} |x|^p |f(x)| dx + \int_{|x| > M} |x|^{-2} |x^{p+2} f(x)| dx \\ &\leq M^p \int_{|x| \leq M} |f(x)| dx + \int_{|x| > M} x^{-2} dx < \infty. \end{aligned}$$

□

On en déduit une propriété remarquable de la transformée de Fourier des fonctions à décroissance rapide.

Proposition 9. *Soit f une fonction de $L_1(\mathbb{R})$ à décroissance rapide. Alors \hat{f} est indéfiniment dérivable.*

Proof. D'après la proposition 5, \hat{f} est dans $C_\infty(\mathbb{R})$ dès que, pour tout $p \in \mathbb{N}$, $x^p f(x)$ est dans $L_1(\mathbb{R})$; ce qui est assuré par le lemme 8. □

Inversement si f est dans $C_\infty(\mathbb{R})$ quelles propriétés possède \hat{f} ? Le résultat suivant amène un élément de réponse.

Proposition 10. *Soit f une fonction de $C_\infty(\mathbb{R})$. Si pour tout $k \in \mathbb{N}$, $f^{(k)}$ est dans $L_1(\mathbb{R})$ alors \hat{f} est à décroissance rapide.*

Proof. D'après la proposition 5 on a, pour tout $k \in \mathbb{N}$, $\widehat{f^{(k)}}(\xi) = (2i\pi\xi)^k \hat{f}(\xi)$. En appliquant le théorème de Riemann-Lebesgue, il vient $\lim_{|\xi| \rightarrow \infty} |\xi|^k |\hat{f}(\xi)| = 0$. □

Autrement dit nous venons de voir que

1. plus f décroît rapidement à l'infini, plus \hat{f} est régulière;
2. plus f est régulière, plus \hat{f} décroît rapidement à l'infini. En particulier si $f \in C_\infty(\mathbb{R})$ et est à décroissance rapide, il en est de même pour \hat{f} .

1.3 Un exemple remarquable

Nous allons considérer une famille de fonctions qui reste stable par transformation de Fourier.

On introduit pour tout $\sigma > 0$ la fonction de *densité gaussienne*

$$g_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\xi^2/2\sigma^2}. \quad (1.4)$$

Lemme 11. La fonction $g_1(x) = 1/\sqrt{2\pi} \exp(-x^2/2)$ est la densité d'une probabilité sur \mathbb{R} , et sa transformée de Fourier est $\hat{g}_1(\xi) = e^{-2\pi^2\xi^2}$.

Proof. La fonction est positive et vérifie $\int_{\mathbb{R}} g_1(x)dx = 1$ (on peut le montrer en exercice en écrivant le carré de l'intégrale comme une double intégrale).

La fonction $x \mapsto xg_1(x)$ étant aussi dans $L_1(\mathbb{R})$, on peut appliquer la proposition 5(i), et on obtient, pour tout $\xi \in \mathbb{R}$,

$$\hat{g}_1'(\xi) = -2i\pi \int xg_1(x)e^{-i2\pi\xi x}dx.$$

Un calcul immédiat donne $g_1'(x) = -xg_1(x)$; une intégration par partie donne donc

$$\hat{g}_1'(\xi) = -2i\pi \int g_1'(x)(-i2\pi\xi)e^{-i2\pi\xi x}dx.$$

D'où l'on tire finalement que $\hat{g}'_1(\xi) = -4\pi^2\xi\hat{g}_1(\xi)$. La solution générale de l'équation différentielle à variables séparables $f'(u) = -4\pi^2uf(u)$ étant $f(u) = Ce^{-2\pi^2u^2}$, et comme on a $\hat{g}_1(0) = \int g_1(x)dx = 1$, on voit que nécessairement $\hat{g}_1(\xi) = e^{-2\pi^2\xi^2}$. \square

Par un changement de variable évident, ce résultat se généralise aisément à tout $\sigma > 0$. En particulier,

$$\hat{g}_\sigma(\xi) = e^{-2\pi^2(\xi\sigma)^2} = \frac{1}{\sigma\sqrt{2\pi}} g_{1/2\pi\sigma}(\xi), \quad \xi \in \mathbb{R}. \quad (1.5)$$

Comme annoncé plus haut, la famille $(g_\sigma)_{\sigma>0}$ est donc stable par transformé de Fourier.

1.4 Quelques exemples classiques

Rappelons que u est la fonction de Heaviside, définie par $u(x) = 1$ pour $x > 0$ et $u(x) = 0$ pour $x \leq 0$.

(i) $f_1(x) = e^{-ax}u(x)$, $\operatorname{Re}(a) > 0$.

$$\hat{f}_1(\xi) = \int_0^\infty e^{-2i\pi x\xi} e^{-ax} dx = \lim_{b \rightarrow +\infty} \left[\frac{-e^{-x(a+2i\pi\xi)}}{a+2i\pi\xi} \right]_0^b = \frac{1}{a+2i\pi\xi}.$$

(ii) $f_2(x) = e^{ax}u(-x)$, $\operatorname{Re}(a) > 0$.

$$\hat{f}_2(\xi) = \int_{-\infty}^0 e^{-2i\pi x\xi} e^{ax} dx = \frac{-1}{-a+2i\pi\xi}.$$

(iii) $f_3(x) = \frac{x^k}{k!} e^{-ax}u(x)$, $\operatorname{Re}(a) > 0$.

$f_3(x) = \frac{1}{(-2i\pi)^k} \frac{1}{k!} (-2i\pi x)^k f_1(x)$, and $\hat{f}_3(\xi) = \frac{1}{k!} \frac{1}{(-2i\pi)^k} \hat{f}_1^{(k)}(\xi)$. Comme $\hat{f}_1^{(k)}(\xi) = k!(a + 2i\pi\xi)^{-(k+1)} (-2i\pi)^k$,

$$\hat{f}_3(\xi) = \frac{1}{(a + 2i\pi\xi)^{k+1}}.$$

(iv) $f_4(x) = \frac{x^k}{k!} e^{ax} u(-x)$, $\operatorname{Re}(a) > 0$. Nous avons

$$\hat{f}_4(\xi) = \frac{-1}{(-a + 2i\pi\xi)^{k+1}}.$$

(v) $f_5(x) = e^{-a|x|}$, $\operatorname{Re}(a) > 0$. Nous déduisons des calculs précédents

$$\hat{f}_5(\xi) = \frac{2a}{a^2 + 4\pi^2\xi^2}.$$

(vi) $f_6(x) = \operatorname{sign}(x)e^{-a|x|}$, $\operatorname{Re}(a) > 0$. Nous avons

$$\hat{f}_6(\xi) = \frac{-4i\pi\xi}{a^2 + 4\pi^2\xi^2}.$$

1.5 Formulaire

(i)

$$\begin{aligned}\hat{f}^{(k)}(\xi) &= (\widehat{-2i\pi x})^k f(\xi) \\ \widehat{f^{(k)}}(\xi) &= (2i\pi\xi)^k \hat{f}(\xi)\end{aligned}$$

(ii)

$$\begin{aligned}f(x-a) &\xrightarrow{\mathcal{F}} e^{-2i\pi a\xi} \hat{f}(\xi) \\ e^{2i\pi ax} f(x) &\xrightarrow{\mathcal{F}} \hat{f}(\xi - a)\end{aligned}$$

(iii) $a \neq 0$.

$$f(ax) \xrightarrow{\mathcal{F}} \frac{1}{|\xi|} \hat{f}\left(\frac{\xi}{a}\right)$$

(iv) $a \in \mathbb{C}$, $\operatorname{Re}(a) > 0, k = 0, 1, 2, \dots$

$$\begin{aligned} \frac{x^k}{k!} e^{-ax} u(x) &\xrightarrow{\mathcal{F}} \frac{1}{(a + 2i\pi\xi)^{k+1}} \\ \frac{x^k}{k!} e^{ax} u(-x) &\xrightarrow{\mathcal{F}} \frac{-1}{(-a + 2i\pi\xi)^{k+1}} \\ e^{-a|x|} &\xrightarrow{\mathcal{F}} \frac{2a}{a^2 + 4\pi^2\xi^2} \\ \operatorname{sign}(x)e^{-a|x|} &\xrightarrow{\mathcal{F}} \frac{-4i\pi\xi}{a^2 + 4\pi^2\xi^2} \end{aligned}$$

(v) $a \in \mathbb{R}, a > 0$.

$$\begin{aligned} e^{-ax^2} &\xrightarrow{\mathcal{F}} \sqrt{\frac{\pi}{a}} e^{-\frac{\pi^2}{a}\xi^2} \\ 1_{[-a,+a]}(x) &\xrightarrow{\mathcal{F}} \frac{\sin 2a\pi\xi}{\pi\xi} \end{aligned}$$

1.6 L'espace de Schwartz

Nous avons vu comment les propriétés de décroissances se traduisent par des propriétés de dérivabilité sur la transformée de Fourier, et *vice versa*. Afin de construire un espace fonctionnel stable par transformation de Fourier, il est donc naturel d'introduire un espace contenant des fonctions comportant à la fois ces deux propriétés.

Définition 12. On désigne par $\mathcal{S}(\mathbb{R})$ ou tout simplement \mathcal{S} l'espace vectoriel des fonctions de \mathbb{R} dans \mathbb{C} qui vérifient les deux propriétés suivantes:

1. f est indéfiniment dérivable sur \mathbb{R} ;
2. f est à décroissance rapide, ainsi que toutes ses dérivées.

On appelle $\mathcal{S}(\mathbb{R})$ l'espace de Schwartz.

Donnons quelques exemples importants de fonctions appartenant à $\mathcal{S}(\mathbb{R})$.

Exemple 13. Il est facile de vérifier que la fonction f définie par $f(x) = e^{-x^2}$ appartient à \mathcal{S} . Par suite, il en est de même de la fonction g_σ définie par (1.4) quelque soit $\sigma > 0$.

Il faut travailler un peu plus pour trouver un exemple de fonction dans \mathcal{S} à support compact c'est-à-dire nulle en dehors d'un ensemble borné.

Exemple 14. On considère la fonction g définie par

$$g(x) = \begin{cases} e^{-1/x} & \text{si } x > 0 \\ 0 & \text{sinon.} \end{cases}$$

On montre facilement que f est \mathcal{C}^∞ . En revanche elle n'est pas à support compact puisque $g(x) \rightarrow 1$ quand $x \rightarrow \infty$. Cependant la fonction f définie par

$$f(x) = g(x) \times g(1-x), \quad x \in \mathbb{R},$$

est nulle en dehors de $[0, 1]$ et \mathcal{C}^∞ . C'est donc une fonction de \mathcal{S} à support compact.

Proposition 15. L'espace \mathcal{S} a les propriétés suivantes :

1. \mathcal{S} est stable pour la multiplication par un polynôme.
2. \mathcal{S} est stable par dérivation ($f \in \mathcal{S} \Rightarrow f' \in \mathcal{S}$).
3. $\mathcal{S} \subset L_1(\mathbb{R})$.

La démonstration de cette proposition est laissée en exercice. Le résultat essentiel, qui découle essentiellement de propriétés déjà démontrées, est contenu dans le théorème suivant.

Théorème 16. L'espace \mathcal{S} est stable par transformation de Fourier: si $f \in \mathcal{S}$ alors $\hat{f} \in \mathcal{S}$.

Proof. Soit $f \in \mathcal{S}$. Comme f est dans $L_1(\mathbb{R})$ et à décroissance rapide, $\hat{f} \in C_\infty(\mathbb{R})$ d'après la Proposition 9. Pour tout $k \in \mathbb{N}$, $f^{(k)}$ étant à décroissance rapide est intégrable d'après le lemme 8. On en déduit de la proposition 10 que \hat{f} est à décroissance rapide. Il reste à examiner la dérivation de f . Comme toutes les dérivées de f sont à décroissance rapide, les fonctions $x \mapsto (x^q f(x))^{(p)}$ sont dans $L_1(\mathbb{R})$ pour tout entiers p et q ; d'où, d'après la Proposition 5, pour tout $\xi \in \mathbb{R}$,

$$\xi^p \hat{f}^{(q)}(\xi) = \xi^p \mathcal{F}((-2i\pi)^q m_q \times f)(\xi) = \frac{1}{(i2\pi)^p} \mathcal{F}\left([(-i2\pi)^q m_q \times f]^{(p)}\right)(\xi), \quad (1.6)$$

où m_q est le monôme de degré q , $m_q(x) = x^q$. Le théorème de Riemann-Lebesgue montre que $\lim_{|\xi| \rightarrow \infty} |\xi^q \hat{f}^{(q)}(\xi)| = 0$. \square

Nous allons voir cette stabilité se traduit en fait par une propriété encore plus remarquable: \mathcal{F} est une bijection de \mathcal{S} dans \mathcal{S} . Pour cela, il faut montrer que \mathcal{F} admet une réciproque sur \mathcal{S} , ce sera le rôle de sa soeur jumelle $\bar{\mathcal{F}}$. Le procédé de régularisation sera fondamental pour arriver à cette fin.

1.7 Formules d'inversion

On est maintenant en mesure de montrer des résultats d'inversion de la transformée de Fourier dans $L_1(\mathbb{R})$. La preuve se base sur l'observation suivante. D'après (1.5), $\mathcal{F}(g_\sigma)$ est une fonction réelle paire de $\mathcal{L}_1(\mathbb{R})$ et $\mathcal{FF}(g_\sigma) = \bar{\mathcal{F}}\mathcal{F}(g_\sigma) = g_\sigma$. Le résultat suivant est une première généralisation de la formule d'inversion “ $\bar{\mathcal{F}}\mathcal{F}(f) = f$ ” qui sera par la suite étendue à des cadres bien plus généraux.

Proposition 17. Soit $f \in \mathcal{L}_1(\mathbb{R})$ et supposons que \hat{f} appartient aussi à $\mathcal{L}_1(\mathbb{R})$. Alors, en tout point x où f est continue, on a

$$[\bar{\mathcal{F}}\hat{f}](x) = f(x). \quad (1.7)$$

Proof. On a vu ci-dessus que la fonction $\hat{g}_\sigma(x) = e^{-2\pi^2\sigma^2x^2}$ a g_σ pour transformée de Fourier. Donc $[\mathcal{F}(x \mapsto \hat{g}_\sigma(x)e^{i2\pi tx})](\xi) = g_\sigma(\xi - t) = g_\sigma(t - \xi)$ par la proposition 4 puis par parité de g_σ . La proposition 6 appliquée avec $x \mapsto f(x)$ et $x \mapsto e^{i2\pi tx}\hat{g}_\sigma(x)$ donne donc, pour tout $t \in \mathbb{R}$,

$$\int \hat{f}(x)\hat{g}_\sigma(x)e^{i2\pi tx}dx = \int f(u)g_\sigma(t-u)du. \quad (1.8)$$

Lorsque $\sigma \rightarrow 0$, on peut passer à la limite dans l'intégrale de gauche, puisque l'on a, pour tout x , $\lim_{\sigma \rightarrow 0} \hat{g}_\sigma(x) = 1$ pour tout x et $|\hat{f}(x)\hat{g}_\sigma(x)e^{i2\pi tx}| \leq |\hat{f}(x)|$. Comme $\hat{f} \in L_1(\mathbb{R})$, on applique le théorème de convergence dominée, qui montre

$$\lim_{\sigma \rightarrow 0} \int \hat{f}(x)\hat{g}_\sigma(x)e^{i2\pi tx}dx = \int \hat{f}(x)e^{i2\pi tx}dx.$$

Le passage à la limite dans le membre de droite de (1.8) est lui une conséquence du lemme 18. On obtient bien le résultat annoncé si f est continue en t . \square

Lemme 18. Soit $f \in L_1(\mathbb{R})$. Soit K une fonction positive telle que $\int K(x) dx = 1$ et, pour une constante $C > 0$ et un exposant $\alpha > 2$, $K(x) \leq C(1 + |x|)^{-\alpha}$ pour tout $x \in \mathbb{R}$. Définissons, pour tout $\sigma > 0$, la fonction K_σ par

$$K_\sigma(x) = \frac{1}{\sigma} K(x/\sigma). \quad (1.9)$$

Alors, on a les propriétés suivantes.

(i) En tout point t où f est continue, on a

$$\lim_{\sigma \downarrow 0} \int f(u) K_\sigma(t-u) du = f(t).$$

(ii) Si f est continue à support compact, alors $t \mapsto \int f(u) K_\sigma(t-u) du$ converge uniformément vers f quand $\sigma \downarrow 0$.

Proof. Comme $\int K_\sigma(x) dx = 1$, par un changement de variable élémentaire,

$$\int f(t-u) K_\sigma(u) du - f(t) = \int \{f(t-u) - f(t)\} K_\sigma(u) du. \quad (1.10)$$

Montrons le point (i). Comme f est continue au point t , pour tout $\varepsilon > 0$, il existe η tel que $|u-t| \leq \eta$ implique que $|f(u) - f(t)| \leq \varepsilon$. On a pour tout $\sigma > 0$,

$$\int_{|u| \leq \eta} |f(t-u) - f(t)| K_\sigma(u) du \leq \varepsilon \int K_\sigma(u) du = \varepsilon. \quad (1.11)$$

De plus,

$$\int_{|u| \geq \eta} |f(t-u) - f(t)| K_\sigma(u) du \leq \|f\|_1 \sup_{|u| \geq \eta} K_\sigma(u) + |f(t)| \int_{|u| \geq \eta} K_\sigma(u) du.$$

D'autre part, lorsque $\sigma \downarrow 0$,

$$\sup_{|u| \geq \eta} K_\sigma(u) = \frac{1}{\sigma} \sup_{|v| \geq \eta/\sigma} K_\sigma(v) \leq \frac{C}{\sigma} (1 + \eta/\sigma)^{-\alpha}$$

et,

$$\int_{u \geq \eta} K_\sigma(u) du \leq \frac{C}{\sigma} \int_{|v| \geq \eta/\sigma} (1 + |v|)^{-\alpha} dv = O(\sigma^{\alpha-2}).$$

Par conséquent, en combinant ces majorations avec $\alpha > 2$,

$$\lim_{\sigma \downarrow 0} \int_{|u| \geq \eta} |f(t-u) - f(t)| K_\sigma(u) du = 0. \quad (1.12)$$

Le résultat suit donc avec (1.10).

Le point (ii) se montre de façon semblable. Comme f est continue à support compact, elle est uniformément continue. On peut donc choisir $\eta > 0$ tel que (1.11) soit valide pour tout t . De même la convergence (1.12) est uniforme en t en utilisant que f est borné. D'où le résultat. \square

Le résultat précédent conjugué avec le théorème 16 permet de définir une transformée inverse comme application réciproque de \mathcal{F} définie comme application de \mathcal{S} dans \mathcal{S} .

En effet, si f est un élément de \mathcal{S} , \hat{f} est dans \mathcal{S} et donc intégrable. f étant partout continue, la formule d'inversion (1.7) est valable pour tout $x \in \mathbb{R}$. Donc, pour tout $f \in \mathcal{S}$, $f = \overline{\mathcal{F}}(\mathcal{F}f)$. De la même façon, on a $f = \mathcal{F}(\overline{\mathcal{F}}f)$. \mathcal{F} est donc une bijection sur \mathcal{S} et son inverse est $\overline{\mathcal{F}}$.

Théorème 19. *La transformation de Fourier \mathcal{F} est une application linéaire bijective de \mathcal{S} sur \mathcal{S} . L'application inverse est $\mathcal{F}^{-1} = \overline{\mathcal{F}}$.*

Ayant montré le théorème 19, de nombreuses formules d'inversion peuvent être déduites par dualité. Nous verrons dans la section suivante comment appliquer ce principe dans un cadre hilbertien.

Chapitre 2

Convolution et Filtrage analogique

2.1 Définition

Définition 20. La convolution de deux fonctions f et g de \mathbb{R} à valeurs dans \mathbb{C} est une fonction $f * g$ définie par

$$f * g(x) = \int_{\mathbb{R}} f(x-t)g(t)dt = \int_{\mathbb{R}} f(u)g(x-u)du.$$

L'existence du produit de convolution requiert bien entendu des hypothèses que nous préciserons dans la suite.

Exemple 21. (i) Let $f = g = \mathbb{1}_{[0,1]}$. Alors

$$\int_{\mathbb{R}} f(x-t)g(t)dt = \int_0^1 \mathbb{1}_{[0,1]}(x-t)dt = \text{Leb}([0,1] \cap [x-1, x])$$

qui est la fonction "triangle"

$$f * g(x) = \begin{cases} 0 & x \leq 0, \\ x & 0 \leq x \leq 1, \\ 2-x & 1 \leq x \leq 2, \\ 0 & x \geq 2. \end{cases}$$

Le produit de convolution de deux fonctions discontinues est donc continue.

(ii) Soit $f \in L_1(\mathbb{R})$ and $g = (2h)^{-1} \mathbb{1}_{[1-h, +h]}$ où $h > 0$.

$$f * g(x) = \frac{1}{2h} \int_{-h}^{+h} f(x-t)dt = \frac{1}{2h} \int_{x-h}^{x+h} f(u)du,$$

qui est la moyenne de f sur l'intervalle $[x-h, x+h]$. Le lecteur montrera que la fonction $f * g$ est continue.

2.2 Convolution dans $L_1(\mathbb{R})$

Définition 22 (support d'une fonction mesurable). Soit $f : \mathbb{R} \rightarrow \mathbb{C}$ une fonction mesurable. Soit $O_i, i \in I$ la famille de tous les ouverts de \mathbb{R} tels que pour tout $i \in I$, $f = 0$ p.p. sur O_i . Soit $O = \bigcup_{i \in I} O_i$ et on définit le support de f , $\text{supp}(f)$, comme l'ensemble fermé $\mathbb{R} \setminus O$.

On vérifie aisément que si $f = g$ p.p. alors $\text{supp}(f) = \text{supp}(g)$. Dans l'exemple 21 $\text{supp}(f) = \text{supp}(g) = [0, 1]$, et la convolution $f * g$ a pour effet d'étendre le support $\text{supp}(f * g) = [0, 2]$. De façon générale, nous avons

Lemme 23. Soient f et g deux fonctions telles que $f * g$ existe. Alors

$$\text{supp}(f * g) \subset \overline{\text{supp}(f) + \text{supp}(g)}.$$

Proof. Notons $S = \mathbb{R} \setminus (\text{supp}(f) + \text{supp}(g))$ et S° l'intérieur de S . Soit $x \in S$. Pour tout $t \in \text{supp}(f)$ nous avons $(x-t) \notin \text{supp}(g)$ et par conséquent $\int_{\mathbb{R}} g(x-t)f(t)dt = 0$. Soit O_{f*g} le plus grand ensemble ouvert sur lequel $f * g = 0$ p.p. Nous avons montré que si $x \in S^\circ$ alors $x \in O_{f*g}$. Par conséquent $\mathbb{R} \setminus O_{f*g} = \text{supp}(f * g) \subset \mathbb{R} \setminus S^\circ$. La preuve découle de $\mathbb{R} \setminus S^\circ = \mathbb{R} \setminus S$. \square

Proposition 24. Si f et g sont des éléments $L_1(\mathbb{R})$, alors:

- (i) $f * g$ est défini presque-partout et $f * g \in L_1(\mathbb{R})$.
- (ii) La convolution est opérateur bilinéaire continu de $L_1(\mathbb{R}) \times L_1(\mathbb{R})$ à valeurs dans $L_1(\mathbb{R})$ et

$$\|f * g\|_1 \leq \|f\|_1 \|g\|_1. \quad (2.1)$$

Proof. (i) Comme $f, g \in L_1(\mathbb{R})$, le théorème de Fubini montre que la fonction $(y, z) \mapsto f(y)g(z)$ appartient à $L_1(\mathbb{R}^2)$. Le changement de variables $y = x - t$ et $z = t$ montre

$$\iint_{\mathbb{R} \times \mathbb{R}} f(y)g(z)dydz = \iint_{\mathbb{R} \times \mathbb{R}} f(x-t)g(t)dxdt.$$

La fonction $x \mapsto \int_{\mathbb{R}} f(x-t)g(t)dt$ est donc définie p.p. et appartient à $L_1(\mathbb{R})$, en appliquant encore le théorème de Fubini.

(ii) Pour établir l'inégalité (2.1), nous écrivons

$$|f * g(x)| \leq \int_{\mathbb{R}} |f(x-t)||g(t)|dt = |f * |g|(x)|.$$

Par conséquent, nous avons

$$\begin{aligned} \int_{\mathbb{R}} |f * g|(x)dx &\leq \int_{\mathbb{R}} |f * |g|(x)|dx = \int_{\mathbb{R}} dx \int_{\mathbb{R}} |f(x-t)||g(t)|dt \\ &= \int_{\mathbb{R}} |g(t)| \left(\int_{\mathbb{R}} |f(x-t)|dx \right) dt = \|g\|_1 \|f\|_1. \end{aligned}$$

\square

Proposition 25. Supposons que $f \in L_{1,\text{loc}}(\mathbb{R})$ et que $g \in L_1(\mathbb{R})$.

- (i) Si $\text{supp}(g)$ est borné, alors $f * g(x)$ est défini p.p. et $f * g \in L_{1,\text{loc}}(\mathbb{R})$.
- (ii) Si f est bornée alors $f * g(x)$ est défini pour tout x et $f * g \in L_\infty(\mathbb{R})$.

Proof. (i) $g = 0$ p.p. sur le complémentaire d'un intervalle $[-a, a]$. Soit $x \in [\alpha, \beta]$. Pour tout $t \in [-a, a]$ et $x \in [\alpha, \beta]$, $f(x-t)g(t) = \mathbb{1}_{[\alpha-a, \beta+a]}(x-t)f(x-t)g(t)$ et donc

$$f * g(x) = \int_{-a}^{+a} f(x-t)g(t)dt = (\mathbb{1}_{[\alpha-a, \beta+a]}f) * g(x).$$

$f * g$ coincide sur $[\alpha, \beta]$ avec la convolution de deux fonctions appartenant à $L_1(\mathbb{R})$. Proposition 24 montre que cette fonction est définie p.p. et est intégrable. Donc $f * g$ est définie p.p. et est intégrable sur tout ensemble compact.

(ii) Si $f \in L_\infty(\mathbb{R})$, alors

$$\left| \int_{\mathbb{R}} f(u)g(x-u)du \right| \leq \|f\|_\infty \int_{\mathbb{R}} |g(x-u)| du = \|f\|_\infty \|g\|_1.$$

pour tout x et $\|f*g\|_\infty \leq \|f\|_\infty \|g\|_1$.

□

2.3 Convolution dans $L_p(\mathbb{R})$

Proposition 26. Supposons que $f \in L_p(\mathbb{R})$ et $g \in L_q(\mathbb{R})$ où $p^{-1} + q^{-1} = 1$. Alors:

(i) $f*g$ est défini et continue sur \mathbb{R} .

(ii) $\|f*g\|_\infty \leq \|f\|_p \|g\|_q$.

Proof. Nous allons établir ce résultat pour: $p = 1, q = +\infty$ and $p = 2, q = 2$. Considérons tout d'abord le cas $p = 1, q = +\infty$. Nous avons déjà établi que $f*g$ est défini partout et borné; seule la continuité reste à établir. Nous avons

$$\begin{aligned} |f*g(x) - f*g(y)| &\leq \int |f(x-t) - f(y-t)| |g(t)| dt \\ &\leq \|g\|_\infty \int |f(x-t) - f(y-t)| dt. \end{aligned}$$

Supposons tout d'abord que f est continue à support compact, $\text{supp}(f) \subset]-a, a[$. Pour $|x-y|$ suffisamment petit,

$$\begin{aligned} \int_{\mathbb{R}} |f(x-t) - f(y-t)| dt &= \int_{\mathbb{R}} |f(x-y+u) - f(u)| du \\ &= \int_{-a}^{+a} |f(x-y+u) - f(u)| du \leq 2a \sup_{|u| \leq a} |f(x-y+u) - f(u)|. \end{aligned}$$

Comme f est uniformément continue sur $[-a, a]$, $f*g$ est (uniformément) continu sur \mathbb{R} .

Les fonctions continues à support compact sont denses dans $L_1(\mathbb{R})$. Soit $\{f_n, n \in \mathbb{N}\}$ une suite de fonctions continues à support compact telles que $\lim_{n \rightarrow \infty} \|f_n - f\|_1 = 0$. Nous avons

$$|f*g(x) - f*g(y)| \leq |f*g(x) - f_n*g(x)| + |f_n*g(y) - f*g(y)| + |f_n*g(x) - f_n*g(y)|,$$

qui implique

$$|f*g(x) - f*g(y)| \leq 2\|g\|_\infty \|f-f_n\|_1 + |f_n*g(x) - f_n*g(y)|.$$

Le premier terme tend vers 0 lorsque $n \rightarrow \infty$, et le second terme est uniformément continu pour tout n ; par conséquent $f*g$ est uniformément continu sur \mathbb{R} .

Considérons le cas $p = 2$ et $q = 2$. L'inégalité de Cauchy-Schwarz montre que

$$|f * g(x)| \leq \int_{\mathbb{R}} |f(x-t)| |g(t)| dt \leq \left(\int_{\mathbb{R}} |f(x-t)|^2 dt \right)^{1/2} \left(\int_{\mathbb{R}} |g(t)|^2 dt \right)^{1/2}$$

et donc que $\|f * g\|_{\infty} \leq \|f\|_2 \|g\|_2$. La continuité découle de la densité de $C_c^0(\mathbb{R})$ (fonctions continues à support compact) dans $L_2(\mathbb{R})$.

La preuve pour $p \neq 1, 2$ est similaire en remplaçant l'inégalité de Cauchy-Schwarz par l'inégalité de Hölder et en utilisant la densité de $C_c^0(\mathbb{R})$ dans $L_p(\mathbb{R})$. \square

Nous considérons dans la proposition suivante la convolution d'une fonction de $L_1(\mathbb{R})$ et d'une fonction de $L_2(\mathbb{R})$.

Proposition 27. *Si $f \in L_1(\mathbb{R})$ et $g \in L_2(\mathbb{R})$, alors*

- (i) $f * g(x)$ est défini presque-partout.
- (ii) $f * g \in L_2(\mathbb{R})$ et $\|f * g\|_2 \leq \|f\|_1 \|g\|_2$.

Proof. (i) Nous avons

$$|f(u)g(x-u)| = (|f(u)||g(x-u)|^2)^{1/2}(|f(u)|)^{1/2} \quad (2.2)$$

Comme $|f| \in L_1(\mathbb{R})$ et $|g|^2 \in L_1(\mathbb{R})$, la fonction $u \mapsto |f(u)||g(x-u)|^2$ est intégrable pour presque-tout x . The right-hand term of (20.4), being the product of two square integrable functions, is integrable. Thus $f * g(x)$ is defined for almost all x .

(ii) En utilisant l'inégalité de Cauchy-Schwarz

$$\begin{aligned} |f * g(x)| &\leq \int_{\mathbb{R}} |f(u)| |g(x-u)| du \\ &\leq \left(\int_{\mathbb{R}} |f(u)| |g(x-u)|^2 du \right)^{1/2} \left(\int_{\mathbb{R}} |f(u)| du \right)^{1/2} \end{aligned}$$

et donc

$$|f * g(x)|^2 \leq (|f| * |g|^2(x)) \|f\|_1.$$

En intégrant les deux membres de la seconde inégalité, nous obtenons,

$$\begin{aligned} \int_{\mathbb{R}} |f * g(x)|^2 dx &\leq \|f\|_1 \int_{\mathbb{R}} |f| * |g|^2(x) dx \\ &\leq \|f\|_1 \|f\|_1 \|g^2\|_1, \end{aligned}$$

ce qui implique $\|f * g\|_2 \leq \|f\|_1 \|g\|_2$. \square

La version convoluée $f \star K$ de f adopte la régularité du noyau K . Ce principe est donné par le lemme suivant.

Lemme 28. *Soient $f, g \in L_1(\mathbb{R})$. Alors on a les propriétés suivantes*

(i) Si g est \mathcal{C}^k , $f \star g$ est \mathcal{C}^k et $(f \star g)^{(k)} = f \star g^{(k)}$.

(ii) On a pour tout entier $p \geq 0$,

$$\|(f \star g) \times m_p\|_\infty \leq \|f \times (1 + |m_p|)\|_\infty \|g \times (1 + |m_p|)\|_1, \quad (2.3)$$

où m_p désigne le monôme de degré p , $m_p(x) = x^p$.

(iii) Si f et g sont à décroissance rapide, il en est de même de $f \star g$.

(iv) Si f est à décroissance rapide et $g \in \mathcal{S}$, $f \star g \in \mathcal{S}$.

Proof. La propriété (i) est une simple application du lemme de dérivation sous le signe somme.

Montrons la propriété (ii). Pour tous $t, x \in \mathbb{R}$, on a $|x|^p \leq (|x-t| + |t|)^p \leq |x-t|^p + |t|^p$. D'où

$$|f \star g(x)| |x|^p \leq \int |f(x-t)| |x-t|^p |g(t)| dt + \int |f(x-t)| |t|^p |g(t)| dt.$$

On obtient donc (2.3).

La propriété (iii) est obtenue en appliquant de (ii) pour tout $p \geq 1$.

La propriété (iv) découle de (i) et (iii). \square

Remarque 29. On remarque facilement que l'on peut prendre dans les lemmes précédents $K = g_1$, c'est-à-dire $K_\sigma = g_\sigma$, où g_σ est définie en (1.4). On peut parler dans ce cas de régularisation gaussienne.

On peut prendre aussi

$$K(x) = \begin{cases} c^{-1} e^{-1/(1-x^2)} & |x| \leq 1 \\ 0 & \text{sinon} \end{cases} \quad \text{avec } c = \int_{-1}^{+1} e^{-(1-x^2)^{-1}} dx. \quad (2.4)$$

Théorème 30 (densité de \mathcal{S} dans $L_p(\mathbb{R})$). Soit $p \geq 1$ et $f \in L_p(\mathbb{R})$. Pour tout $\varepsilon > 0$, il existe $g_\varepsilon \in \mathcal{S}$ telle que $\|f - g_\varepsilon\|_p \leq \varepsilon$.

Proof. Pour tout $\varepsilon > 0$, il existe $f_\varepsilon \in C_c^0(\mathbb{R})$ telle que $\|f - f_\varepsilon\|_p \leq \varepsilon$. Pour $\sigma > 0$, on note $g_{\varepsilon,\sigma} = f_\varepsilon * K_\sigma$ où K est le noyau donné par (2.4). Lemme 18 montre que $g_{\varepsilon,\sigma}$ converge uniformément vers f_ε , i.e. $\lim_{\sigma \rightarrow 0} \|f_\varepsilon - g_{\varepsilon,\sigma}\|_\infty = 0$. Comme f_ε et K_σ sont à supports compacts, Lemme 23 montre que $\text{supp}(g_{\varepsilon,\sigma}) \subset [a, b]$ avec $-\infty < a < b < \infty$. On a donc aussi

$$\lim_{\sigma \rightarrow 0} \|f_\varepsilon - g_{\varepsilon,\sigma}\|_p = 0,$$

ce qui implique que, pour tout $\varepsilon > 0$,

$$\lim_{\sigma \rightarrow 0} \|f - g_{\varepsilon,\sigma}\|_p \leq \|f - f_\varepsilon\|_p \leq \varepsilon.$$

Lemme 28 montre que comme f est à décroissance rapide et $K_\sigma \in \mathcal{S}$, alors $g_{\varepsilon,\sigma} f * K_\sigma \in \mathcal{S}$. \square

2.4 Une première extension de la transformation de Fourier inverse

Ici nous l'appliquons dans le cadre $L_1(\mathbb{R})$ grâce au résultat suivant, qui nous permettra de compléter la proposition 17 par un théorème d'inversion.

Proposition 31. *Soient deux fonctions f et g dans $L_1(\mathbb{R})$. Si, pour toute fonction test ϕ dans \mathcal{S} , on a*

$$\int f(x) \phi(x) dx = \int g(x) \phi(x) dx,$$

alors $f = g$ (au sens $L_1(\mathbb{R})$).

Proof. En prenant la différence entre les deux membres de l'égalité de l'hypothèse, on voit qu'il suffit de montrer ce résultat pour $g = 0$. De plus, comme les fonctions continues sont denses dans l'ensemble des fonctions intégrables, on peut se contenter de prendre f continue, le cas général étant obtenu par passage à la limite. Or, pour f continue et $g = 0$, le résultat est immédiat par application du principe de régularisation en choisissant une fonction $K \in \mathcal{S}$ positive intégrant à 1 (par exemple g_1) puis en appliquant le lemme 18 en tout point de la droite réelle. \square

On en déduit le résultat annoncé qui complète la proposition 17.

Théorème 32. *Soit $f \in L_1(\mathbb{R})$ et supposons que \hat{f} appartient aussi à $L_1(\mathbb{R})$. Alors la fonction (continue) $\tilde{\mathcal{F}}\hat{f}$ est l'unique représentant continu de f .*

Proof. La continuité de $\tilde{\mathcal{F}}\hat{f}$ découle du théorème 3. Pour toute fonction test ϕ de \mathcal{S} , on a, d'après la proposition 6 et le théorème ??,

$$\int f(x) \phi(x) dx = \int \hat{f}(\xi) \overline{\mathcal{F}(\phi)}(\xi) d\xi.$$

Mais comme \hat{f} , on peut réappliquer l'équivalent de la proposition 6 mais pour la transformée inverse, ce qui donne alors directement

$$\int \hat{f}(\xi) \overline{\mathcal{F}(\phi)}(\xi) d\xi = \int \overline{\mathcal{F}(\hat{f})(x)} \phi(x) dx.$$

D'où le résultat en appliquant la proposition 31. \square

2.5 Convolution et transformée de Fourier dans $L_1(\mathbb{R})$

Proposition 33. *Soient $f, g \in L_1(\mathbb{R})$.*

(i) $\widehat{f * g}(\xi) = \hat{f}(\xi) \hat{g}(\xi)$ pour tout $\xi \in \mathbb{R}$,

(ii) Si de plus $\hat{f}, \hat{g} \in L_1(\mathbb{R})$, alors $\widehat{f \cdot g}(\xi) = \hat{f}(\xi) \hat{g}(\xi)$.

Proof. (i) Comme $f, g \in L_1(\mathbb{R})$, Proposition 24 shows that $f * g \in L_1(\mathbb{R})$. On peut donc calculer $\widehat{f * g}(\xi)$. En utilisant le théorème de Fubini, nous avons

$$\begin{aligned}\int e^{-i2\pi\xi x} f * g(x) dx &= \int e^{-i2\pi\xi x} \left(\int f(x-t)g(t) dt \right) dx \\ &= \int g(t) \left(\int e^{-i2\pi\xi x} f(x-t) dx \right) dt \\ &= \int g(t) e^{-i2\pi\xi t} \hat{f}(\xi) dt = \hat{g}(\xi) \cdot \hat{f}(\xi)\end{aligned}$$

(ii) On peut appliquer ((i)) avec $\overline{\mathcal{F}}$ en remplaçant i par $-i$. Comme \hat{f} et $\hat{g} \in L_1(\mathbb{R})$, on applique ((i)) à \hat{f} et \hat{g} ce qui montre

$$\overline{\mathcal{F}}(\hat{f} * \hat{g})(x) = \overline{\mathcal{F}}(\hat{f})(x) \cdot \overline{\mathcal{F}}(\hat{g})(x) = f(x) \cdot g(x), \text{ p.p.}$$

□

Corollaire 34. Soient $f, g \in L_1(\mathbb{R})$.

$$(i) \quad \widehat{f * g} = \hat{f} \cdot \hat{g}.$$

$$(ii) \quad \widehat{f \cdot g} = \hat{f} * \hat{g}.$$

2.6 Applications aux filtres analogiques gouvernés par une équation différentielle

La transformée de Fourier permet d'étudier les filtres définis par une équation différentielle ordinaire à coefficients constants.

$$\sum_{k=0}^q b_k g^{(k)} = \sum_{j=0}^p a_j f^{(j)}, \quad a_p \cdot b_q \neq 0, \quad (2.5)$$

où f est l'entrée et $g = A(f)$ est la sortie. Nous allons tout d'abord supposer que $f \in \mathcal{S}$. C'est un cas très particulier car il n'y a pas de raisons que l'entrée soit aussi régulière. Nous verrons dans la suite que l'étude de ce cas permet de considérer ensuite des situations plus générales.

Nous supposons que $f \in \mathcal{S}$ et nous recherchons des solutions $g \in \mathcal{S}$. Si une telle solution g existe, nous pouvons calculer la transformée de Fourier des deux membres de (2.5):

$$\sum_{k=0}^q b_k (2i\pi\lambda)^k \hat{g}(\lambda) = \sum_{j=0}^p a_j (2i\pi\lambda)^j \hat{f}(\lambda). \quad (2.6)$$

Considérons les deux polynômes: $P(X) = \sum_{j=0}^p a_j X^j$ and $Q(X) = \sum_{k=0}^q b_k X^k$ et supposons que la fraction rationnelle $P(z)/Q(z)$ ne possède pas de pôles sur l'axe imaginaire.

Alors $P(2i\pi\lambda)/Q(2i\pi\lambda)$ est définie pour tout $\lambda \in \mathbb{R}$, et (2.6) est équivalent à

$$\hat{g}(\lambda) = \frac{P(2i\pi\lambda)}{Q(2i\pi\lambda)} \hat{f}(\lambda) = H(\lambda) \hat{f}(\lambda). \quad (2.7)$$

Cette identité détermine $g \in \mathcal{S}$ complètement, si cette solution existe, et prouve l'unicité de la solution de (2.5) dans \mathcal{S} . L'existence d'une solution découle aussi de (2.7), comme la fonction

$$G(\lambda) = \frac{P(2i\pi\lambda)}{Q(2i\pi\lambda)} \hat{f}(\lambda)$$

est un élément de \mathcal{S} dès que $f \in \mathcal{S}$. En appliquant ??, nous avons $g = \overline{\mathcal{F}}(G)$ est l'unique solution de (2.5) dans \mathcal{S} .

Proposition 35. *Si $P(x)/Q(x)$ n'a pas de pôles sur l'axe imaginaire et si $f \in \mathcal{S}$, alors (2.5) possède une unique solution $g \in \mathcal{S}$.*

Pour $f \in \mathcal{S}$, appelons $g = A(f)$ l'unique solution de (2.5) dans \mathcal{S} . Soient $f_1, f_2 \in \mathcal{S}$ et $\alpha_1, \alpha_2 \in \mathbb{C}$. Posons $f = \alpha_1 f_1 + \alpha_2 f_2$. On a clairement $f \in \mathcal{S}$ et par linéarité de la transformée de Fourier $\hat{f} = \alpha_1 \hat{f}_1 + \alpha_2 \hat{f}_2$. Comme

$$g = A(f) = \overline{\mathcal{F}}(H\hat{f}) = \alpha_1 \overline{\mathcal{F}}(H\hat{f}_1) + \alpha_2 \overline{\mathcal{F}}(H\hat{f}_2) = \alpha_1 A(f_1) + \alpha_2 A(f_2)$$

l'application $A : \mathcal{S} \rightarrow \mathcal{S}$ est un linéaire (c'est un endomorphisme de \mathcal{S}). En traitement du signal, c'est ce que l'on appelle le principe de *superposition*, la réponse du système à une combinaison linéaire des entrées est la combinaison linéaire (avec les mêmes poids) des sorties.

Pour $\tau \in \mathbb{R}$, appelons $L_\tau : \mathcal{S} \rightarrow \mathcal{S}$ l'opérateur de retard: $f_\tau = L_\tau(f)$. Il est immédiat de voir que pour tout $\tau \in \mathbb{R}$, L_τ est un opérateur linéaire de $\mathcal{S} \rightarrow \mathcal{S}$. Pour $f \in \mathcal{S}$, notons $f_\tau = L_\tau f$. Etudions maintenant l'image par A de f_τ . En utilisant le formulaire Section 1.5

$$\begin{aligned} A(f_\tau) &= \overline{\mathcal{F}}(H\hat{f}_\tau) = \overline{\mathcal{F}}((\xi \mapsto H(\xi)e^{-2i\pi\tau\xi})\hat{f}(\xi)) \\ &= L_\tau(Af). \end{aligned}$$

Par conséquent, pour tout $f \in \mathcal{S}$, nous avons $A \circ L_\tau(f) = L_\tau \circ A(f)$.

Définition 36. *Une application $A : \mathcal{S} \rightarrow \mathcal{S}$ est linéaire et invariante si*

- (i) *pour tout $f_1, f_2 \in \mathcal{S}$ et $\alpha_1, \alpha_2 \in \mathbb{C}$, $A(\alpha_1 f_1 + \alpha_2 f_2) = \alpha_1 A(f_1) + \alpha_2 A(f_2)$.*
- (ii) *pour tout $\tau \in \mathbb{R}$, $L_\tau \circ A = A \circ L_\tau$.*

Chapitre 3

Transformée de Fourier-Plancherel

3.1 Espace des fonctions de carré intégrable

Soit $\mathcal{L}_2(\mathbb{R})$ l'espace des fonctions définies sur \mathbb{R} et à valeurs complexes, $f : \mathbb{R} \rightarrow \mathbb{C}$, de carré sommable c'est-à-dire telles que:

$$\int |f(x)|^2 dx < \infty.$$

On note $L_2(\mathbb{R})$ l'espace des classes d'équivalence de $\mathcal{L}_2(\mathbb{R})$ pour la relation d'équivalence " $f = g$ p.p.". Pour I un sous ensemble borélien de \mathbb{R} (et en particulier, un intervalle), on peut définir de la même façon l'espace $\mathcal{L}_2(I)$ des fonctions de carré sommable sur I , $\int_I |f(x)|^2 dx < \infty$ et l'espace $L_2(I)$ des classes d'équivalence de $\mathcal{L}_2(I)$ par rapport à la relation d'équivalence d'égalité presque-partout.

Pour f et $g \in L_2(\mathbb{R})$, définissons.

$$\langle f, g \rangle_I = \int_I f(x) \bar{g}(x) dx \quad (3.1)$$

où, pour tout $z \in \mathbb{C}$, \bar{z} est le conjugué de z . Lorsque $I = \mathbb{R}$, nous omettons l'indice I . Cette intégrale est bien définie pour 2 représentants de f et g car $|f(x)\bar{g}(x)| \leq (|f(x)| + |\bar{g}(x)|)/2$ et sa valeur ne dépend évidemment pas du choix de ses représentant. D'autre part, $L_2(I)$ est bien le plus "gros" espace fonctionnel sur lequel ce produit scalaire est bien défini puisqu'il impose justement $\langle f, f \rangle_I < \infty$. Mentionnons aussi que, de même que pour $(L_1(\mathbb{R}), \|\cdot\|_1)$, $(L_2(\mathbb{R}), \|\cdot\|_2)$ est un espace de Banach (espace vectoriel normé complet), où la norme $\|\cdot\|_2$ est définie par

$$\|f\|_2 := \sqrt{\langle f, f \rangle} = \left(\int |f(x)|^2 dx \right)^{1/2}.$$

Cette norme étant une norme induite par un produit scalaire, on dit que $(L_2(\mathbb{R}), \langle \cdot, \cdot \rangle)$ est un espace de Hilbert.

Théorème 37. *L'ensemble des fonctions intégrables et de carré intégrable, $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ est un sous-espace vectoriel dense de $(L_2(\mathbb{R}), \|\cdot\|_2)$.*

Proof. Pour tout $f \in L_2(\mathbb{R})$, on note f_n la fonction égale à f sur $[-n, n]$ et nulle ailleurs. Alors $f_n \in L_1(\mathbb{R})$ pour tout n , et par convergence monotone, $\|f_n - f\|_2 \rightarrow 0$ quand $n \rightarrow \infty$ (on dit que f_n tend vers f au sens de $L_2(\mathbb{R})$). On en conclut que $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ est dense dans $(L_2(\mathbb{R}), \|\cdot\|_2)$. \square

On a immédiatement que la proposition 31 s'adapte à l'espace $L_2(\mathbb{R})$.

Corollaire 38. *Soient deux fonctions f et g dans $L_2(\mathbb{R})$. Si, pour toute fonction test ϕ dans \mathcal{S} , on a*

$$\int f(x) \phi(x) dx = \int g(x) \phi(x) dx,$$

alors $f = g$ (au sens $L_2(\mathbb{R})$).

Nous verrons que $L_2(\mathbb{R})$ pose un certain nombre de problèmes théoriques pour définir la transformée de Fourier qui ne se pose pas pour une fonction de $L_1(\mathbb{R})$. Or, en passant de $L_1(\mathbb{R})$ à $L_2(\mathbb{R})$, on impose à la fonction des conditions locales plus contraignantes (toute restriction d'une fonction de $L_2(\mathbb{R})$ à un compact est $L_1(\mathbb{R})$ mais l'inverse n'est pas vrai) et on autorise des comportements en $t = \pm\infty$ un peu plus généraux. Dès lors, on peut s'interroger sur l'intérêt d'étudier les fonctions de $L_2(\mathbb{R})$ plutôt que de $L_1(\mathbb{R})$, qui plus est quand, en pratique, une fonction n'est jamais observé sur un temps infini. La réponse à cette question est la suivante. Outre que les propriétés d'espace de Hilbert de $L_2(\mathbb{R})$ sont fondamentales dans la théorie, elles ont un lien physique évident dans les applications puisque le carré de la norme d'un signal dans $L_2(\mathbb{R})$ n'est rien d'autre que son énergie. Le fait qu'en pratique les "signaux" observés soient dans $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ explique que l'on peut en général ne pas se préoccuper des subtilités entre transformée de Fourier dans $L_1(\mathbb{R})$ et transformée de Fourier dans $L_2(\mathbb{R})$, mais, pour établir les résultats généraux que l'on utilise pour étudier les fonctions de carré sommable, il serait dommage de les énoncer dans le cas particulier $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ alors qu'ils sont valables dans $L_2(\mathbb{R})$, même si l'on doit pour cela donner des preuves qui peuvent apparaître plus abstraites.

3.2 Transformée de Fourier sur $L_2(\mathbb{R})$

L'idée de base de la construction consiste à étendre la transformée de Fourier de $L_1(\mathbb{R})$ à $L_2(\mathbb{R})$ par un argument de densité.

Proposition 39. Soit f et g dans \mathcal{S} . On a:

$$\begin{aligned} \int \hat{f}(\xi) \bar{\hat{g}}(\xi) d\xi &= \int f(x) \bar{g}(x) dx \\ \int |\hat{f}(\xi)|^2 d\xi &= \int |f(x)|^2 dx. \end{aligned}$$

Proof. Appliquons la formule d'échange (Proposition 6). On pose $h(\xi) = \bar{\hat{g}}(\xi)$. On a:

$$\int \hat{f}(\xi) h(\xi) d\xi = \int f(x) \hat{h}(x) dx.$$

Mais $\bar{\hat{g}}(\xi) = \overline{\mathcal{F}\bar{g}(\xi)}$, d'où $\hat{h} = \bar{g}$. □

Proposition 40. Soient E et F deux espaces vectoriels normés, F complet, et G un sous-espace vectoriel dense dans E . Si A est un opérateur linéaire continu de G dans F , alors il existe un prolongement unique \tilde{A} linéaire continu de E dans F et la norme de \tilde{A} est égale à la norme de A .

Proof. Soit $f \in E$. Comme G est dense dans E , il existe une suite f_n dans G telle que $\lim_{n \rightarrow \infty} \|f_n - f\| = 0$. La suite f_n étant convergente, elle est de Cauchy. A étant linéaire continu on a

$$\|Af_n - Af_m\| \leq \|A\| \|f_n - f_m\|.$$

On en déduit que Af_n est une suite de Cauchy de F qui est complet. La suite Af_n est donc convergente vers un élément g de F . On vérifie facilement que g ne dépend pas de la suite f_n et on pose donc $Af = g$. \tilde{A} est linéaire par construction et de plus on a

$$\|\tilde{A}f\| = \lim_{n \rightarrow \infty} \|Af_n\| \leq \lim_{n \rightarrow \infty} \|A\| \|f_n\| = \|A\| \|f\|,$$

ce qui prouve que $\|\tilde{A}\| \leq \|A\|$. Comme $\tilde{A}f = Af$ pour tout $f \in G$, on a $\|\tilde{A}\| = \|A\|$. Enfin, G étant dense dans E , il est clair que \tilde{A} est unique. \square

D'après la proposition 39, \mathcal{F} est une isométrie sur \mathcal{S} muni du produit scalaire $\langle \cdot, \cdot \rangle$. On applique le résultat précédent avec $E = F = L_2(\mathbb{R})$, $G = \mathcal{S}$ (voir Théorème 30). On obtient

Théorème 41. *La transformation de Fourier \mathcal{F} (respectivement la transformation inverse $\overline{\mathcal{F}}$) se prolonge en une isométrie de $L_2(\mathbb{R})$ sur $L_2(\mathbb{R})$. Désignons toujours par \mathcal{F} (resp. $\overline{\mathcal{F}}$) ce prolongement. On a en particulier*

1. (Inversion) pour tout $f \in L_2(\mathbb{R})$, $\mathcal{F}\overline{\mathcal{F}}f = \overline{\mathcal{F}}\mathcal{F}f = f$,
2. (Plancherel) pour tout $f, g \in L_2(\mathbb{R})$, $\langle f, g \rangle = \langle \mathcal{F}f, \mathcal{F}g \rangle$
3. (Parseval) pour tout $f \in L_2(\mathbb{R})$, $\|f\|_2 = \|\mathcal{F}f\|_2$.

Remarquons que l'égalité de Parseval peut se réécrire, pour tout f et g dans $L_2(\mathbb{R})$,

$$\langle \mathcal{F}f, g \rangle = \langle f, \mathcal{F}g \rangle \quad (3.2)$$

Proposition 42. *Le prolongement de \mathcal{F} sur \mathcal{S} par continuité à $(L_2(\mathbb{R}), \|\cdot\|_2)$ est compatible avec la définition de \mathcal{F} donnée précédemment sur $L_1(\mathbb{R})$. Plus précisément*

1. Pour tout $f \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$, $\mathcal{F}f$ défini par le théorème 41 admet un représentant $\hat{f} \in C_\infty$ vérifiant

$$\hat{f}(\xi) = \int_{\mathbb{R}} e^{-i2\pi\xi x} f(x) dx, \quad \xi \in \mathbb{R}.$$

2. Si $f \in L_2(\mathbb{R})$, $\mathcal{F}f$ est la limite dans $L_2(\mathbb{R})$ de la suite g_n , définie par $g_n(\xi) = \int_{-\infty}^{\infty} e^{-i2\pi\xi x} f(x) dx$.

Proof. Notons \hat{f} la transformée de Fourier sur $L_1(\mathbb{R})$ et $\mathcal{F}f$ celle sur $L_2(\mathbb{R})$. Prenons $f \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$. En appliquant la proposition 6 puis Parseval (voir (3.2)), on a pour tout $\psi \in \mathcal{S}$,

$$\int \psi \hat{f} = \int \hat{\psi} f = \int \mathcal{F}(\psi) f = \int \psi \mathcal{F}(f)$$

d'où $\int (\hat{f} - \mathcal{F}(f)) \psi = 0$ pour tout $\psi \in \mathcal{S}$. Le corollaire 38 fournit alors le premier résultat.

Posons $f_n = f \mathbb{1}_{[-n,n]}$. Par convergence dominée, on a $\lim_n \|f_n - f\|_2^2 = 0$. Comme $f_n \in L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ on écrit $g_n = \hat{f}_n = \mathcal{F}(f_n)$ et par continuité il vient $\lim_{n \rightarrow \infty} \|\mathcal{F}f - g_n\|_2^2 = 0$. \square

3.3 Application au calcul de transformées de Fourier

Proposition 43. (i) Soit $f \in L_2(\mathbb{R})$. On a $\mathcal{F}\mathcal{F}f = f_\sigma$ p.p. où $f_\sigma(x) = f(-x)$

(ii) Si $f \in L_1(\mathbb{R}) \cap L^2(\mathbb{R})$, $\mathcal{F}(\hat{f}) = f$.

Proof. (i) Montrons que $\mathcal{F}(f) = \overline{\mathcal{F}(f)}_\sigma$. Soit $\{f_n, n \in \mathbb{N}\}$ une suite de fonctions de \mathcal{S} telle que $\lim_{n \rightarrow \infty} \|f - f_n\|_2 = 0$ (voir Théorème 30). On a $\mathcal{F}(f)_n = \mathcal{F}((f_n)_\sigma)$. En passant à la limite, on a donc $\mathcal{F}(f) = \overline{\mathcal{F}(f_\sigma)}$ (remarquons que $\|f_\sigma - (f_n)_\sigma\|_2 = 0$).

(ii) Résulte immédiatement du fait que $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$, $\mathcal{F}(f) = \hat{f}$. \square

Exemple 44. Considérons la fonction $f_\pm(x) = e^{\pm xax} u(\pm x)$ et $\Re(a) > 0$. On a $\hat{f}_\pm(\xi) = \pm 1/(\pm a + 2i\pi\xi)$. $\hat{f}_\pm \notin L_1(\mathbb{R})$ mais Proposition 43 montre que $\mathcal{F}(\hat{f}_\pm) = (f_\pm)_\sigma$. On a donc pour $a \in \mathbb{C}$, $\Re(a) > 0$,

$$(a + 2i\pi x)^{-1} \xrightarrow{\mathcal{F}} e^{a\xi} u(-\xi)$$

$$(a - 2i\pi x)^{-1} \xrightarrow{\mathcal{F}} e^{-a\xi} u(\xi).$$

En procédant de la même façon on obtient

$$\sin(x)/x \xrightarrow{\mathcal{F}} \pi \mathbb{1}_{[-(2\pi)^{-1}, (2\pi)^{-1}]}(\xi).$$

3.4 Principe d'incertitude : résolution en temps et en fréquence

3.5 Convolution et transformation de Fourier dans $L_2(\mathbb{R})$

Chapitre 4

Echantillonnage

Nous nous intéressons dans cette partie au sous espace de $L_2(\mathbb{R})$ des fonctions à bande limitée

Définition 45 (Bande Limitée). *Une fonction $f \in L_2(\mathbb{R})$ (à valeurs réelles) est dite à bande limitée s'il existe $B < \infty$ tel que: $\mathcal{F}f(\xi) = 0$ pour $\xi \notin [-B, +B]$. On note $BL(B)$ le sous espace vectoriel des fonctions $f \in L_2(\mathbb{R})$ telles que $\mathcal{F}f(\xi) = 0$ pour (presque tout) $\xi \notin [-B, +B]$.*

Pour $f \in L_2(\mathbb{R})$, la Transformée de Fourier définit un isomorphisme de $L_2(\mathbb{R})$ dans $L_2(\mathbb{R})$ d'inverse \mathcal{F} . Soit maintenant $f \in BL(B)$. Comme $\mathcal{F}f$ est à support compact et dans $L_2(\mathbb{R})$ par isométrie de \mathcal{F} , il est aussi dans $L_1(\mathbb{R})$. D'après la proposition 42, appliquée à la transformée de Fourier conjuguée de $\mathcal{F}f$ qui n'est autre que f , f admet donc un représentant continu. Par la suite on identifiera tout élément de $BL(B)$ à son représentant continu.

Soit $T \leq 1/(2B)$. Nous identifierons dans la suite T avec la *période d'échantillonnage* et $1/T$ avec la *fréquence d'échantillonnage*. Considérons la fonction $\xi \rightarrow F_T(\xi)$ obtenu en périodisant la fonction $\xi \rightarrow \mathcal{F}f(\xi)$ à la période $1/T$:

$$F_T(\xi) = \sum_{n \in \mathbb{Z}} [\mathcal{F}f] \left(\xi - \frac{n}{T} \right).$$

Par construction, la fonction $\xi \mapsto F_T(\xi)$ est une fonction périodique de période $1/T$, et sur chaque période $[(k-1/2)/T, (k+1/2)/T[$, la fonction F_T est égale à $\mathcal{F}f$ la translatée de k/T . Il est donc clair que $F_T \in L_2[-1/2T, 1/2T]$ et admet donc un développement en série de Fourier:

$$F_T(\xi) = \sum_{n \in \mathbb{Z}} c_n(F_T) e^{-i2\pi\xi n T}, \quad (4.1)$$

où $\{c_k(F_T)\}$ est la suite de des coefficients de Fourier de la fonction F_T , définis pour tout $k \in \mathbb{Z}$ par,

$$c_k(F_T) = T \int_{-1/(2T)}^{1/(2T)} F_T(\xi) e^{+i2\pi\xi k T} d\xi. \quad (4.2)$$

L'égalité dans (4.1) doit être comprise au sens de la convergence dans l'espace de Hilbert $L_2([-1/(2T), 1/(2T)])$: la série trigonométrique $F_{N,T}(t)(\xi)$, définie par

$$F_{N,T}(\xi) = \sum_{k=-N}^N c_k(F_T) e^{-i2\pi\xi k T}, \quad (4.3)$$

converge vers la fonction F_T au sens de la topologie induite par la norme $\|\cdot\|_2$, c'est-à-dire,

$$\lim_{N \rightarrow \infty} \int_{-1/(2T)}^{1/(2T)} |F_T(\xi) - F_{N,T}(\xi)|^2 d\xi = 0.$$

L'égalité de Parseval implique aussi que $\sum_{n \in \mathbb{Z}} |c_n(F_T)|^2 < \infty$. Comme nous avons supposé que $T \leq 1/(2B)$, nous avons donc $[-B, +B] \subset [-1/(2T), 1/(2T)]$, ce qui implique que, pour tout $\xi \in [-1/(2T), 1/(2T)]$, $F_T(\xi) = \mathcal{F}f(\xi)$, ce qui implique que les coefficients de Fourier $c_k(F_T)$ s'écrivent:

$$c_k(F_T) = T \int_{-B}^B \mathcal{F}f(\xi) e^{+i2\pi\xi k T} d\xi = T f(kT). \quad (4.4)$$

Ils correspondent donc aux échantillons de la fonction f prélevés aux instants régulièrement espacés kT (les instants d'échantillonage). La formule de Parseval pour les coefficients de Fourier implique en particulier que $\sum_{k \in \mathbb{Z}} |f(kT)|^2 < \infty$. (4.1) se réécrit donc

$$\sum_{n \in \mathbb{Z}} \mathcal{F}f\left(\xi - \frac{n}{T}\right) = T \sum_{n \in \mathbb{Z}} f(nT) e^{-i2\pi\xi nT}, \quad (4.5)$$

qui est appelée la *formule sommatoire de Poisson*. Cette formule, que nous avons démontré ici pour des fonctions à bande limitée, s'avèrent vérifiées sous des hypothèses beaucoup plus générales. En multipliant les deux membres de l'identité précédente par la fonction indicatrice de l'intervalle $[-1/(2T), +1/(2T)]$ et en utilisant $\mathbb{1}_{[-1/(2T), 1/(2T)]}(\xi) \mathcal{F}f(\xi) = \mathbb{1}_{[-1/(2T), 1/(2T)]}(\xi) F_T(\xi)$, nous obtenons donc l'identité,

$$\mathcal{F}f(\xi) = T \sum_{n \in \mathbb{Z}} f(nT) \mathbb{1}_{[-1/(2T), 1/(2T)]}(\xi) e^{-i2\pi\xi nT}.$$

qui doit être comprise au sens $L_2(\mathbb{R})$,

$$\lim_{N \rightarrow \infty} \int \left| \mathcal{F}f(\xi) - T \sum_{n=-N}^N f(nT) \mathbb{1}_{[-1/(2T), 1/(2T)]}(\xi) e^{-i2\pi\xi nT} \right|^2 d\xi = 0.$$

Comme l'application $\overline{\mathcal{F}}$ est continue de $L_2(\mathbb{R}) \rightarrow L_2(\mathbb{R})$ et que

$$\overline{\mathcal{F}}\left(\xi \rightarrow \mathbb{1}_{[-1/(2T), 1/(2T)]}(\xi) e^{-i2\pi\xi nT}\right)(x) = \frac{\sin\left(\frac{\pi}{T}(x - nT)\right)}{\pi(t - nT)}$$

(avec la convention $0/0 = 0$), on obtient la formule d'interpolation

$$f(t) = \sum_{n \in \mathbb{Z}} f(nT) s_T(x - nT), \quad (4.6)$$

où la fonction s_T , appelée *sinus-cardinal* est définie par

$$s_T(0) = 0 \quad \text{et} \quad s_T(x) = \frac{\sin\left(\frac{\pi}{T}x\right)}{\frac{\pi}{T}x} \quad \text{pour tout } x \neq 0. \quad (4.7)$$

La convergence de la série (4.6) a lieu dans $L_2(\mathbb{R})$. Si de plus on a

$$\sum_{k \in \mathbb{Z}} |f(kT)| < \infty,$$

la série (4.6) est uniformément (car normalement au sens de la norme sup) convergente vers une fonction g continue sur \mathbb{R} . Donc la série converge aussi dans $L_2(J)$, pour tout intervalle borné $J \subset \mathbb{R}$. On en déduit que $f(t) = g(t)$ presque-partout, et donc que $f(t) = g(t)$ pour tout t réel, puisque les fonctions f et g sont continues. Nous pouvons formuler le résultat important

Théorème 46. *Soit $f \in \text{BL}(B)$. Alors on pour tout $T \leq 1/(2B)$, on a*

$$\sum_{k=-\infty}^{\infty} |f(kT)|^2 < \infty,$$

et

$$f(t) = \sum_{n \in \mathbb{Z}} f(nT) \frac{\sin\left(\frac{\pi}{T}(t - nT)\right)}{\frac{\pi}{T}(t - nT)}.$$

La convergence de la série et l'égalité ont lieu au sens de la norme de $L_2(\mathbb{R})$. Elles ont lieu au sens de la convergence uniforme, et donc pour tout t réel si

$$\sum_{n \in \mathbb{Z}} |f(nT)| < \infty.$$

Il est intéressant de se poser la question de savoir ce qu'il advient du résultat précédent lorsque la condition sur la fréquence d'échantillonnage est violée. Nous supposons toujours que la fonction f est à bande limitée, $f \in \text{BL}(B)$, mais que la fréquence d'échantillonnage $1/T$ est inférieure à la bande $2B$ de la fonction. Comme F_T est périodique de période $1/T$ et $F_T \in L_2([-1/(2T), 1/(2T)])$, cette fonction est développable en série de Fourier,

$$F_T(\xi) = T \sum_{n=-\infty}^{\infty} \int_{-1/(2T)}^{1/(2T)} \sum_{n \in \mathbb{Z}} \mathcal{F}f(\xi - n/T) e^{i2\pi\xi nT} d\xi e^{i2\pi\xi nT}.$$

Un calcul élémentaire montre que

$$\begin{aligned} \int_{-1/(2T)}^{1/(2T)} \sum_{n \in \mathbb{Z}} \mathcal{F}f(\xi - n/T) e^{i2\pi\xi nT} d\xi &= \sum_{n \in \mathbb{Z}} \int_{-1/(2T)}^{1/(2T)} \mathcal{F}f(\xi - n/T) e^{i2\pi\xi nT} d\xi = \\ &= \sum_{n \in \mathbb{Z}} \int_{-1/(2T)-n/T}^{1/(2T)-n/T} \mathcal{F}f(\xi) e^{i2\pi\xi nT} d\xi = \int \mathcal{F}f(\xi) e^{i2\pi\xi nT} d\xi = f(kT). \end{aligned}$$

Par conséquent, nous avons encore $\sum_{k \in \mathbb{Z}} |f(kT)|^2 < \infty$ et la formule sommatoire de Poisson (4.5) reste valide. En appliquant \mathcal{F} aux deux membres de (4.5), nous obtenons donc

$$\left[\mathcal{F} \left(\sum_{n \in \mathbb{Z}} \mathcal{F}f(\xi - n/T) \mathbb{1}_{[-1/(2T), 1/(2T)]}(\xi) \right) \right] (t) = \sum_{n \in \mathbb{Z}} f(nT) \text{sinc}\left(\frac{\pi}{T}(t - nT)\right).$$

Par conséquent, si la condition sur la fréquence d'échantillonnage n'est pas respectée, la transformée de Fourier de la fonction interpolée sera égale à la transformée du signal "périodisé". On parle, pour qualifier ce phénomène de *repliement spectral*, ou *d'aliasing*.

Posons, pour tout entier k , $\phi_{T,k}(t) = s_T(t - kT)$. La fonction $\phi_{T,k}$ appartient à $L_2(\mathbb{R})$.

Proposition 47. *La famille $\{\phi_{T,k}\}_{k \in \mathbb{Z}}$ est une base Hilbertienne de l'espace $\text{BL}(1/(2T))$.*

Proof. Montrons tout d'abord que les fonctions $\{\phi_{k,T}\}_{k \in \mathbb{Z}}$ forment une famille orthogonale. Nous avons, par application de la formule de Parseval,

$$\int \phi_{T,k} \phi_{T,l} = \int \mathcal{F}(\phi_{T,k}) \overline{\mathcal{F}(\phi_{T,l})}.$$

On a, par définition, $\mathcal{F}(\phi_{T,k}) = T \mathbb{1}_{[-1/(2T), 1/(2T)]}(\xi) e^{-i2\pi\xi k T}$. Par conséquent,

$$\int \phi_{T,k} \phi_{T,l} = T^2 \int_{-1/(2T)}^{1/(2T)} e^{-i2\pi\xi(k-l)T} d\xi = \begin{cases} T & k = l \\ 0 & k \neq l \end{cases}.$$

Montrons maintenant que la famille $\{\phi_{T,k}\}$ forment une base totale de l'ensemble $\text{BL}([-1/(2T), 1/(2T)])$. La formule de reconstruction (4.6) montre que pour tout N , et tout $f \in \text{BL}([-1/(2T), 1/(2T)])$, nous avons

$$\left\| f - \sum_{k=-N}^N f(kT) \phi_{T,k} \right\|_2^2 = T \sum_{N \leq |k| < \infty} |f(kT)|^2,$$

ce qui montre que l'ensemble de fonctions $\{\phi_{T,k}\}$ est dense dans l'ensemble $\text{BL}(1/(2T))$. \square

Soit $f \in L_2(\mathbb{R})$. Cette fonction n'est pas a priori à bande limitée, où, si elle est à bande limitée, cette bande n'est peut être pas compatible avec la fréquence d'échantillonnage de la fonction, $T \geq 1/(2B)$. On sait que si l'on applique la procédure d'échantillonnage décrite ci-dessus sans précaution particulière, le signal discréte et interpolé sera une version altérée du signal original (repliement de spectre). Une approche consiste, avant d'échantillonner la fonction, de la projeter sur l'espace $\text{BL}(1/(2T))$. Pour $f \in L_2(\mathbb{R})$, le calcul de cette projection est élémentaire. Considérons en effet la fonction \tilde{f} définie par:

$$\tilde{f}(t) = \int \mathcal{F}f(\xi) \mathbb{1}_{[-1/(2T), 1/(2T)]}(\xi) e^{+i2\pi\xi t} d\xi.$$

Par construction, nous avons $\mathcal{F}\tilde{f}(\xi) = \mathcal{F}f(\xi) \mathbb{1}_{[-1/(2T), 1/(2T)]}(\xi)$, et donc $\tilde{f} \in \text{BL}(1/(2T))$. Nous avons d'autre part, pour toute fonction $g \in \text{BL}(1/(2T))$, par l'identité de Plancherel

$$\|f - g\|_2^2 = \|\mathcal{F}f - \mathcal{F}g\|_2^2 \geq \int_{|\xi| \geq 1/(2T)} |\mathcal{F}f(\xi)|^2 d\xi = \|f - \tilde{f}\|_2^2.$$

Par conséquence \tilde{f} est la projection de f sur $\text{BL}(1/(2T))$.

Part II

Bases de traitement du signal à temps discret

Chapitre 5

Transformée de Fourier discrète

5.1 La transformée de Fourier sur $\ell^1(\mathbb{Z})$

5.1.1 Les principaux espaces de suites et les règles de calcul

Définition 48 (Espaces de suites). (i) $\ell^1(\mathbb{Z})$ l'espace des suites sommables c'est-à-dire des suites $u = \{u_n, n \in \mathbb{Z}\}$ (que nous noterons aussi parfois $u = \{u(n), n \in \mathbb{N}\}$) qui vérifient:

$$\sum_{n=-\infty}^{\infty} |u_n| < +\infty,$$

on note $\|u\|_1 = \sum_{n \in \mathbb{Z}} |u_n|$ qui est une norme sur l'espace des suites sommables.

(ii) On note $\ell^2(\mathbb{Z})$ l'espace des suites d'énergie finie c'est-à-dire des suites u qui vérifient:

$$\sum_{n \in \mathbb{Z}} |u_n|^2 < +\infty,$$

et on note $\|u\|_2 = (\sum_{n \in \mathbb{Z}} |u_n|^2)^{\frac{1}{2}}$ qui est une norme sur l'espace des suites d'énergie finie.

(iii) On note $\ell^\infty(\mathbb{Z})$ l'espace des suites bornées c'est-à-dire des suites u qui vérifient:

$$\sup_{n \in \mathbb{Z}} |u_n| < \infty,$$

et on note $\|u\|_\infty = \sup_{n \in \mathbb{Z}} \{|u_n|\}$ qui est une norme sur l'espace des suites bornées.

Lemme 49.

$$\ell^1(\mathbb{Z}) \subset \ell^2(\mathbb{Z}) \subset \ell^\infty(\mathbb{Z}).$$

Proof. Nous allons seulement prouver les inclusions:

1. $\ell^1(\mathbb{Z}) \subset \ell^2(\mathbb{Z})$: Soit $u \in \ell^1(\mathbb{Z})$. Soit $E \subset \mathbb{Z}$ défini par $E = \{n : |u_n| > 1\}$. L'ensemble E est forcément fini, sinon u aurait une somme infinie. Et on a

$$\sum_{n \in \mathbb{Z}} |u_n|^2 = \sum_{n \in E} |u_n|^2 + \sum_{n \in \mathbb{Z} \setminus E} |u_n|^2 \leq \sum_{n \in E} |u_n|^2 + \sum_{n \notin E} |u_n|$$

La dernière inégalité vient du fait que si $|x| \leq 1$ alors $|x|^2 \leq |x|$. Or le premier terme de la dernière somme est fini car E est fini. Le second est fini aussi car u est sommable. Donc $u \in \ell^2(\mathbb{Z})$.

2. $\ell^2(\mathbb{Z}) \subset \ell^\infty(\mathbb{Z})$: Il est clair que si u n'est pas bornée alors elle a, par exemple, une infinité de termes supérieurs à 1 en module. Les $|u_n|^2$ ne pourraient donc pas être sommables.

□

Définition 50 (Convolution discrète). Si u et v sont des suites, on appelle produit de convolution de u et v la suite définie par (si la somme a un sens):

$$(u * v)_n = \sum_{m \in \mathbb{Z}} u_m v_{n-m}.$$

Nous donnons dans la proposition suivante les règles qui disent, suivant les espaces auxquels appartiennent u et v , l'espace dont $u * v$ et $u \cdot v$ (produit terme à terme des deux suites) est élément.

Lemme 51. 1. Si $u \in \ell^1(\mathbb{Z})$ et $v \in \ell^\infty(\mathbb{Z})$, alors $u \cdot v \in \ell^1(\mathbb{Z})$ et

$$\|u \cdot v\|_1 \leq \|u\|_1 \|v\|_\infty$$

2. Si $u \in \ell^2(\mathbb{Z})$ et $v \in \ell^2(\mathbb{Z})$, alors $u \cdot v \in \ell^1(\mathbb{Z})$ et

$$\|u \cdot v\|_1 \leq \|u\|_2 \|v\|_2$$

Proof. La preuve élémentaire est laissée au lecteur. \square

Définition 52 (Opérateur de retard). *On note S l'opérateur de retard, l'application de $\mathbb{C}^\mathbb{Z} \mapsto \mathbb{C}^\mathbb{Z}$ défini pour $u = \{u_n, n \in \mathbb{Z}\} \in \mathbb{C}^\mathbb{Z}$ par*

$$(S u)_n = u_{n-1}, \quad \text{pour tout } n \in \mathbb{Z}. \quad (5.1)$$

Cet opérateur est inversible et son inverse est l'opérateur d'avance: $(S^{-1} u)_n = u_{n+1}$ pour tout $n \in \mathbb{N}$. Pour tout $m \in \mathbb{Z}$, $(S^m u)_n = u_{n+m}$. La proposition suivante donne les propriétés du produit de convolution

Proposition 53. Soient $u, v, w \in \ell^1(\mathbb{Z})$.

- (i) Le produit de convolution est commutatif: $u * v = v * u$.
- (ii) Le produit de convolution est associatif: $u * (v * w) = (u * v) * w$.
- (iii) Le produit de convolution est linéaire: $u * (\lambda v + w) = \lambda u * v + u * w$.
- (iv) Le produit de convolution est invariant par décalage: pour tout $m \in \mathbb{Z}$, $(S^m u) * v = S^m(u * v)$, où S est l'opérateur de retard.

Proof. La preuve est élémentaire et laissée au lecteur. \square

Nous obtenons ainsi les règles de convolution que nous résumons tableau suivant sauf comme suit, si u appartient à un espace index de ligne et v se trouve dans l'espace index de colonne, alors $u * v$ se trouve dans l'espace inscrit dans la case correspondante. Par exemple, si $u \in \ell^1(\mathbb{Z})$ et $v \in \ell^\infty(\mathbb{Z})$ alors $u * v \in \ell^\infty(\mathbb{Z})$. Si la case contient un tiret ($-$) alors l'opération est a priori impossible (la série peut ne pas avoir de sens). On a aussi à chaque fois, $\|u * v\|_\gamma \leq \|u\|_\alpha \|v\|_\beta$ où (α, β) sont les normes indexant les lignes et colonnes et γ la norme associée. Par exemple $\|u * v\|_\infty \leq \|u\|_1 \|v\|_\infty$.

*	$\ell^1(\mathbb{Z})$	$\ell^2(\mathbb{Z})$	$\ell^\infty(\mathbb{Z})$
$\ell^1(\mathbb{Z})$	$\ell^1(\mathbb{Z})$	$\ell^2(\mathbb{Z})$	$\ell^\infty(\mathbb{Z})$
$\ell^2(\mathbb{Z})$	$\ell^2(\mathbb{Z})$	$\ell^\infty(\mathbb{Z})$	
$\ell^\infty(\mathbb{Z})$	$\ell^\infty(\mathbb{Z})$		

Toutes les propriétés énoncées dans Proposition 53 s'étendent au cas de suite u, v à valeurs dans $\ell^2(\mathbb{Z})$ et $\ell^\infty(\mathbb{Z})$, à condition que les produits considérés soient définis.

5.1.2 La Transformée de Fourier à temps Discret (TFtD)

Définition 54. Si u est une suite sommable ($u \in \ell^1(\mathbb{Z})$) , on appelle Transformée de Fourier à temps Discret (TFtD en abrégé), la fonction définie sur l'intervalle $[-1/2, 1/2[$ et que l'on note soit \hat{u} soit $\mathcal{F}(u)$

$$\hat{u}(\nu) = \sum_{n \in \mathbb{Z}} u_n e^{-2i\pi n \nu}, \quad \nu \in [-1/2, 1/2[.$$

Quand $u \in \ell^1(\mathbb{Z})$, \hat{u} est une fonction continue et admet une limite en $1/2$ égale à sa valeur en $-1/2$ (elle est donc continue même si on la considère comme une fonction définie sur \mathbb{R})

Le fait que la définition ait un sens découle du fait que u est supposée sommable. Le fait que \hat{u} soit continue utilise le théorème de convergence dominée.

Proposition 55 (Propriétés de la TFtD). (i) $\mathcal{F}(S^m u)(\nu) = e^{-2im\nu} \mathcal{F}(u)(\nu)$ pour tout $\nu \in [-1/2, 1/2[$, où S est l'opérateur de décalage (voir Définition 52).

(ii) $\mathcal{F}(\delta)(\nu) = 1$ pour tout $\nu \in [-1/2, 1/2[$.

(iii) Si $u, v \in \ell^1(\mathbb{Z})$, $\mathcal{F}(u * v) = \mathcal{F}(u) \mathcal{F}(v)$

(iv) Si $u \in \ell^1(\mathbb{Z})$ et $v \in \ell^\infty(\mathbb{Z})$, $\mathcal{F}(u \cdot v) = \mathcal{F}(u) * \mathcal{F}(v)$

(v) Si $u \in \ell^1(\mathbb{Z})$, $v_0 \in [-1/2, 1/2[$, pour tout $\nu \in [-1/2, 1/2[$, $\mathcal{F}(u \cdot e_{v_0})(\nu) = \mathcal{F}(u)(\nu - v_0)$, où $e_{v_0}(n) = e^{-2i\pi v_0 n}$.

(vi) Si u est réelle, alors $\mathcal{F}(u)(-\nu) = \overline{\mathcal{F}(u)(\nu)}$.

Proof. Ces propriétés sont élémentaires et la preuve est laissée au lecteur. \square

5.2 La transformée de Fourier sur $\ell^2(\mathbb{Z})$

Nous avons défini la TFTD pour les suites sommables. Il est possible d'étendre cette définition aux suites $\ell^2(\mathbb{Z})$ (rappelons que $\ell^1(\mathbb{Z}) \subset \ell^2(\mathbb{Z})$).

Théorème 56 (Extension à $\ell^2(\mathbb{Z})$ et égalité de Parseval). Il existe un unique prolongement isométrique \mathcal{F} de $\ell^2(\mathbb{Z}) \mapsto L_2([-1/2, 1/2[)$ qui coïncide avec la TFTD sur $\ell^1(\mathbb{Z})$. Pour tout $u \in \ell^2(\mathbb{Z})$, $\|\hat{u}\|_2 = \|u\|_2$:

$$\int_{-1/2}^{1/2} |\mathcal{F}(u)(\nu)|^2 d\nu = \sum_{n \in \mathbb{Z}} |u_n|^2$$

Proof. Montrons tout d'abord que \mathcal{F} défini une isométrie sur $\ell^1(\mathbb{Z})$. Soit $u \in \ell^1(\mathbb{Z})$.

Nous avons

$$\begin{aligned} \int_{-1/2}^{1/2} |\hat{u}(v)|^2 e^v dv &= \int_{-1/2}^{1/2} \left| \sum_{k \in \mathbb{Z}} u_k e^{-2i\pi k v} \right|^2 dv \\ &= \int_{-1/2}^{1/2} \sum_{k,l=-\infty}^{\infty} u_k \bar{u}_l e^{-2i\pi(k-l)v} dv \\ &= \sum_{k,l \in \mathbb{Z}} u_k \bar{u}_l \int_{-1/2}^{1/2} e^{-2i\pi(k-l)v} dv \\ &= \sum_{k=-\infty}^{\infty} |u_k|^2. \end{aligned}$$

La permutation des intégrales et des sommes est justifiée par application du théorème de Fubini car

$$\int_{-1/2}^{1/2} \sum_{k,l \in \mathbb{Z}} |u_k| |u_l| dv < \infty.$$

D'autre part, $\ell^1(\mathbb{Z})$ est dense dans $\ell^2(\mathbb{Z})$: si $u \in \ell^2(\mathbb{Z})$ alors pour tout $N \in \mathbb{N}$, la suite u^N définie par $u_k^N = u_k$ si $|k| \leq N$ et $u_k^N = 0$ sinon est élément de $L_1(\mathbb{Z})$ et $\|u - u^N\|_2^2 = \sum_{|k|>N} |u_k|^2 \rightarrow 0$. La preuve découle de Théorème 205 \square

Les propriétés de Proposition 55 restent encore vraies en prenant u et v dans $\ell^2(\mathbb{Z})$, à des détails près : il faut prendre $u \in \ell^2(\mathbb{Z})$ et $v \in \ell^1(\mathbb{Z})$, sinon la convolution de u et v n'est a priori ni dans $\ell^1(\mathbb{Z})$ ni dans $\ell^2(\mathbb{Z})$

Théorème 57 (Inversion de la TFtd). *Si $u \in \ell^2(\mathbb{Z})$ est une suite d'énergie finie et $\mathcal{F}(u) = \hat{u}$ sa TFtD alors on a, pour tout $n \in \mathbb{Z}$,*

$$u_n = \int_{-\frac{1}{2}}^{\frac{1}{2}} \hat{u}(v) e^{2i\pi nv} dv$$

où, de façon plus concise $\overline{\mathcal{F}}(\mathcal{F}(u)) = u$.

Proof. Considérons tout d'abord $u \in \ell^1(\mathbb{Z})$. Nous avons

$$\begin{aligned} \int_{-1/2}^{1/2} \hat{u}(v) e^{2i\pi nv} dv &= \int_{-1/2}^{1/2} \left(\sum_{k \in \mathbb{Z}} u_k e^{-2i\pi k v} \right) e^{2i\pi nv} dv \\ &= \sum_{k \in \mathbb{Z}} u_k \int_{-1/2}^{1/2} e^{-2i\pi(k-n)v} dv = u_n. \end{aligned}$$

La permutation de l'intégrale et du signe somme est justifié car

$$\int_{-1/2}^{1/2} \sum_{k \in \mathbb{Z}} |u_k| dv < \infty.$$

Soit maintenant $u \in \ell^2(\mathbb{Z})$ et u^N la suite tronquée, $u_k^N = u_k$ pour $|k| \leq N$, $u_k^N = 0$ sinon. En notant $v \mapsto e_n(v) = e^{-2i\pi nv}$, nous avons

$$\begin{aligned} \int_{-1/2}^{1/2} \hat{u}(v) e^{2i\pi nv} &= \langle u, e_n \rangle \\ &= \lim_{N \rightarrow \infty} \langle u^N, e_n \rangle = \lim_{N \rightarrow \infty} u_n^N = u_n. \end{aligned}$$

□

On a vu que la réponse fréquentielle d'un SLI est la TFtD de sa réponse impulsionale. Le théorème d'inversion nous permet de retrouver la réponse impulsionale à partir de sa réponse fréquentielle. Ainsi, pour définir un SLI, il suffit de donner soit sa réponse impulsionale (ce que nous savions déjà) ou sa réponse fréquentielle (qui permet de retrouver la réponse impulsionale grâce au théorème d'inversion).

5.2.1 Décroissance à l'infini et régularité

Nous savons déjà que si u est sommable alors \hat{u} est continue. La sommabilité est une forme de décroissance à l'infini (pour les suites, cela implique même que u_n tende vers 0 à l'infini). Le théorème suivant dit que plus la suite u décroît rapidement à l'infini, plus sa TFtD est régulière.

Théorème 58 (Décroissance à l'infini et régularité de la TFtD). *Soit $k \geq 0$ un entier, on a :*

(i) *Si $\sum_{n \in \mathbb{Z}} |n|^k |u_n| < \infty$ alors \hat{u} est k fois continuement dérivable.*

(ii) *Notons $v^{(k)}$ la suite de terme général*

$$v_n^{(k)} = (-2i\pi n)^k u_n.$$

Si $v^{(k)} \in \ell^2(\mathbb{Z})$, la TFtD de $v^{(k)}$ est $\mathcal{F}(v^{(k)}) = \mathcal{F}(u)^{(k)}$ (la dérivée k -ième de $\mathcal{F}(u)$)

Proof. La preuve est élémentaire et laissée au lecteur □

5.2.2 Système linéaire invariant (SLI)

Définition 59. *On dit qu'un opérateur T sur $\ell^p(\mathbb{Z})$ avec $p = 1, 2, \infty$ est linéaire et invariant si*

(i) *T est un opérateur linéaire continu sur $\ell^p(\mathbb{Z})$.*

(ii) *pour tout $u \in \ell^p(\mathbb{Z}) \cap \ell^\infty(\mathbb{Z})$, $Tu \in \ell^p(\mathbb{Z}) \cap \ell^\infty(\mathbb{Z})$ (si $\|u\|_\infty < \infty$ alors $\|Tu\|_\infty < \infty$).*

(iii) *T commute avec l'opérateur de retard S (voir Définition 52): $S \circ T = T \circ S$.*

Remarquons que Définition 59-(iii) implique que pour tout $m \in \mathbb{Z}$, $S^m \circ T = T \circ S^m$. Notons $\delta = \{\delta_k, k \in \mathbb{Z}\}$ la suite impulsionale

$$\delta_k = \begin{cases} 1 & k = 0 \\ 0 & \text{sinon} \end{cases} \quad (5.2)$$

Toute suite $u \in \ell^p(\mathbb{Z})$ ($p = 1, 2, \infty$) peut s'écrire

$$u = \sum_{n \in \mathbb{Z}} u_n S^n \delta, \quad (5.3)$$

cette série étant absolument convergente car, pour tout $N \in \mathbb{N}$,

$$\left\| \sum_{|n| \geq N} u_n S^n \delta \right\|_p \leq \sum_{|n| \geq N} |u_n| \|S^n \delta\|_p = \sum_{|n| \geq N} |u_n| \rightarrow_{N \rightarrow \infty} 0.$$

L'opérateur T étant linéaire et continu sur $\ell^p(\mathbb{Z})$, nous avons donc

$$T(u) = \sum_{n \in \mathbb{Z}} u_n T(S^n \delta). \quad (5.4)$$

En effet, en appelant $u^N = \sum_{|n| \leq N} u_n S^n \delta$ la suite tronquée, la linéarité de l'opérateur T implique que

$$T(u^N) = \sum_{|n| \leq N} u_n T(S^n \delta).$$

D'autre part, la continuité de T implique que

$$\|T(u) - T(u^N)\|_p \leq \|T\|_p \|u - u^N\|_p \rightarrow_{N \rightarrow \infty} 0,$$

où $\|T\|_p$ est la norme opérateur de T . Comme T est invariant par translation, nous avons $T \circ S^n = S^n \circ T$, ce qui implique que

$$T(u) = \sum_{n \in \mathbb{Z}} u_n S^n T(\delta). \quad (5.5)$$

Nous en déduisons le résultat important suivant

Théorème 60. *Un SLI T est caractérisé par $T(\delta)$, appelée sa réponse impulsionale, i.e. pour tout $u \in \ell^p(\mathbb{Z})$, la suite $v = T(u)$ est donnée pour tout $k \in \mathbb{Z}$ par*

$$v_k = \sum_{n \in \mathbb{Z}} u_n h_{k-n}, \quad \text{où } h_k = [T(\delta)]_k, k \in \mathbb{Z},$$

où de façon plus concise

$$v_k = (u * h)_k.$$

Lorsque $p = 1, 2$, la réponse impulsionale du SLI T est $h = T(\delta) \in \ell^p(\mathbb{Z})$; on peut donc calculer la TFID de la réponse impulsionale h , qui est appelée la *fonction de transfert*. Dans le domaine de Fourier, l'action du SLI T revient à multiplier la TFID de l'entrée u par la fonction de transfert $\hat{h} = \mathcal{F}(h)$:

$$\hat{v} = \hat{h}\hat{u}.$$

5.3 Transformée de Fourier Discrète ou TFD

La Transformée de Fourier discrète est la transformation de Fourier pour les signaux définis sur un ensemble $\{0, \dots, N-1\}$. Les suites définies sur cet ensemble sont toutes sommables, bornées et d'énergie finie (car elles sont à support fini).

Définition 61 (Transformée de Fourier Discrète). *Si $\mathbf{u} = [u_0, \dots, u_{N-1}]'$ est une suite finie définie sur $\{0, \dots, N-1\}$, on note $\mathbf{U} = [U_0, \dots, U_{N-1}]'$ sa Transformée de Fourier Discrète (TFD en abrégé) définie, elle aussi sur $\{0, \dots, N-1\}$, pour tout $k \in \{0, \dots, N-1\}$*

$$U_k = \sum_{n=0}^{N-1} u_n e^{-2i\pi \frac{k}{N} n}$$

Notons, pour $N \in \mathbb{N}$

$$W_N = e^{2i\pi/N} \quad (5.6)$$

et pour $k \in \{0, \dots, N-1\}$, le vecteur $E_k = [W_N^{k \cdot 0}, \dots, W_N^{k \cdot (N-1)}]'$. Nous avons pour tout $k, \ell \in \{0, \dots, N-1\}$, $E_k^H E_\ell = N \delta_{k,\ell}$ où $\delta_{k,\ell}$ est le symbole de Kronecker. Les vecteurs $(E_0, E_1, \dots, E_{N-1})$ définissent une base orthogonale de l'espace vectoriel \mathbb{C}^N . Nous avons donc, en notant $\mathbf{u} = [u_0, \dots, u_{N-1}]'$, pour tout $k \in \{0, \dots, N-1\}$

$$U_k = E_k^H \mathbf{u}.$$

Théorème 62 (Inversion de la TFD). *Soit \mathbf{u} une suite définie sur $\{0, \dots, N-1\}$ et \mathbf{U} sa TFD on a, pour tout $n \in \{0, \dots, N-1\}$*

$$u_n = \frac{1}{N} \sum_{k=0}^{N-1} U_k W_N^{nk}.$$

De plus,

$$\sum_{k=0}^{N-1} |u_k|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |U_k|^2.$$

Proof. Nous avons, comme les vecteurs $(E_k)_{k=0}^{N-1}$ sont orthogonaux.

$$\mathbf{u} = \sum_{k=0}^{N-1} \frac{E_k^H \mathbf{U}}{E_k^H E_k} E_k = \frac{1}{N} \sum_{k=0}^{N-1} \hat{u}_k E_k.$$

D'autre part,

$$\sum_{k=0}^{N-1} |u_k|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |\hat{u}_k|^2 E_k^H E_k = \frac{1}{N} \sum_{k=0}^{N-1} |U_k|^2.$$

□

Clairement, la TFD est une application linéaire de \mathbb{C}^N à valeurs dans \mathbb{C}^N . Les autres propriétés de la TFD doivent être considérés avec un peu de précaution. Remarquons tout d'abord la suite finie $\mathbf{u} = \{u_0, \dots, u_{N-1}\}$ peut être prolongée en une suite périodique de période N ,

$$\tilde{u}_n = u_n \pmod{N} \quad (5.7)$$

Soit maintenant $m \in \mathbb{Z}$. Il est légitime de considérer, pour $m \in \mathbb{Z}$,

$$S^m \tilde{u}_n = \tilde{u}_{n-m} = u_{(n-m) \pmod N}$$

Lorsque l'on applique l'opérateur décalage S^m à la suite périodisée, nous obtenons de nouveau une suite périodique de période N . On peut donc calculer la TFD de cette suite, pour tout $k \in \{0, \dots, N-1\}$, nous avons

$$\sum_{n=0}^{N-1} u_{(n-m) \pmod N} W_N^{nk} = \sum_{n=0}^{N-1} \tilde{u}_{(n-m) \pmod N} W_N^{nk} = \sum_{\ell=-m}^{N-m-1} \tilde{u}_\ell W_N^{(\ell+m)k}.$$

Comme $W_N^{(\ell-m)k} = W_N^{mk} W_N^{\ell k}$, nous avons

$$\sum_{n=0}^{N-1} u_{(n-m) \pmod N} W_N^{nk} = W_N^{mk} \sum_{\ell=-m}^{N-m-1} \tilde{u}_\ell W_N^{\ell k} = W_N^{mk} \hat{u}_k,$$

où nous avons utilisé que la suite $v_\ell = \tilde{u}_\ell W_N^{\ell k}$ est N -périodique et pour tout suite w N -périodique et tout $m \in \mathbb{Z}$, $\sum_{\ell=-m}^{N-m-1} w_\ell = \sum_{\ell=0}^{N-1} w_\ell$. Les propriétés de la TFD vis-à-vis des translations et des convolutions doivent donc être adaptées.

5.3.1 Lien entre TFD et TFtD

La TFD est la seule transformée de Fourier calculable sur ordinateur. La TFD s'intéresse à des signaux définis sur un espace fini et discret. Dans cette partie nous allons voir comment une TFD peut permettre d'analyser un signal défini sur \mathbb{Z} , réalisant ainsi un passage de l'infini au fini.

Evidemment, une TFD ne peut capturer toutes les caractéristiques d'un signal quelconque, il nous faut restreindre notre étude à des cas particuliers. Nous verrons :

1. Cas d'une suite à support fini.
2. Détermination de la fréquence d'une exponentielle complexe.
3. Séparation de la somme de deux exponentielle complexe (qui nous permettra de comprendre l'importance du fenêtrage)

Cas d'une suite à support fini

Soit une suite u définie sur \mathbb{Z} à support fini. Sans perte de généralité, quitte à la translater, on peut supposer qu'il existe un entier N tel que, pour tout $n \in \mathbb{Z}$, $u_n = 0$ si $n < 0$ ou $n \geq N$

Pour tout entier $M \geq N$ on considère la suite finie v définie sur $\{0, \dots, M-1\}$, pour tout $n \in \{0, \dots, M-1\}$,

$$v_n = u_n$$

La suite finie v est simplement la restriction de u à l'ensemble $\{0, \dots, M-1\}$. On appelle parfois v la suite zéro-padding à l'ordre M de u . On ajoute des zéros à la

Figure 5.1: Un signal à support fini

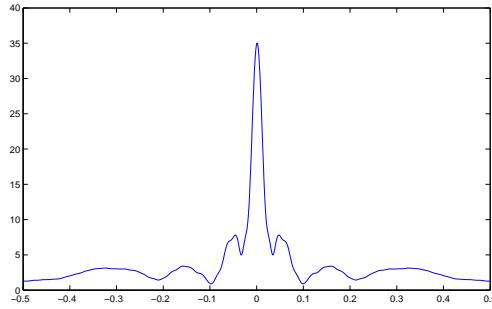


Figure 5.2: module de la TFtD du signal; comme le signal est réel, le module de la TFtD est une fonction paire

suite des échantillons non nuls de u pour obtenir une suite de taille M . Pour tout $k \in \{0, \dots, M-1\}$,

$$V_k = \sum_{n \in \{0, \dots, M-1\}} v_n e^{-2i\pi \frac{k}{M} n} = \hat{u}(k/M)$$

où \hat{u} est la TFtD de u . La dernière égalité provient du fait que u est nulle hors de $\{0, \dots, N-1\}$ et donc hors de $\{0, \dots, M-1\}$.

Estimation d'une fréquence

Soit $u_n = e^{2i\pi f_0 n}$. On voudrait déterminer la fréquence f_0 en ne se donnant le droit que d'observer les échantillons u_0, \dots, u_N . Pourquoi une telle contrainte? Dans un signal, musical par exemple, le contenu fréquentiel évolue au cours du temps. À chaque changement de note, le signal contient sinusoïdes de fréquences différentes. Si l'on veut, par exemple, transcrire un morceau de musique en notes, on ne peut pas se permettre une observation sur une trop longue période car cela aurait pour effet de mélanger

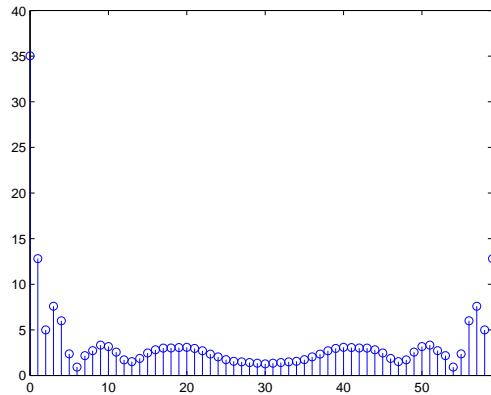


Figure 5.3: module de la TFD du signal

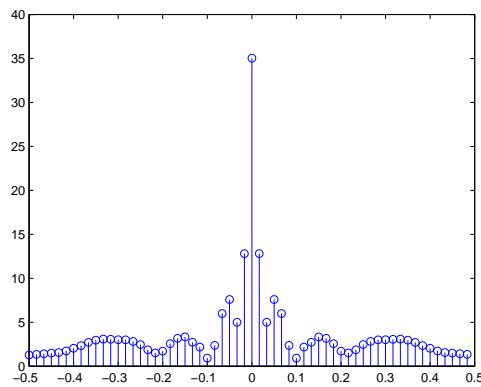


Figure 5.4: module de la TFD périodisé et remise à l'échelle

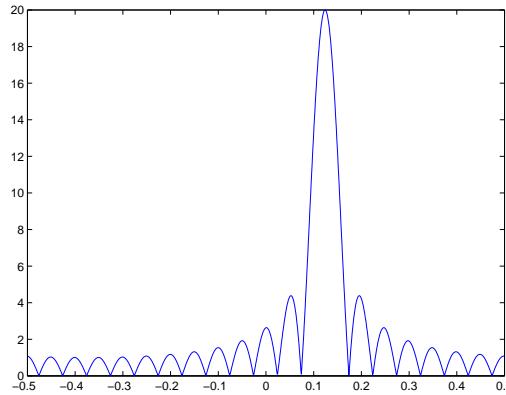


Figure 5.5: La TFD d'ordre 30 de l'harmonique complexe tronquée superposée à la TFtD. Le maximum de la TFD est atteint pour $k = 4$, soit une fréquence de $4/30 = 0,1333$ et une erreur d'estimation de 0,01

entre elles différentes notes. La même chose vaut pour l'analyse d'un signal de parole où l'on risque la confusion entre différents phonèmes.

On note u^T la suite définie sur \mathbb{Z} égale à u sur $\{0, \dots, N-1\}$ et nulle ailleurs. Ce sont les seules valeurs que nous nous donnons le droit d'utiliser pour déterminer v_0 . Sa TFtD est donnée pour tout $v \in [1/2, 1/2[$ par

$$\mathcal{F}(u^T)(v) = e^{-i\pi(N-1)(v-v_0)} \frac{\sin(N\pi(v-v_0))}{\sin(\pi(v-v_0))}$$

La Figure 5.5 représente le module de $\mathcal{F}(u^T)$.

Si on calcule une TFD d'ordre $M \geq N$, on sait que l'on va échantillonner la TFtD de u^T aux points k/M . Deux TFD d'ordres différents sont données Figure 5.6 et Figure 5.7.

Avec une TFD d'ordre M on peut estimer la fréquence de l'harmonique complexe v_0 avec une précision d'au moins $1/M$. En effet, quelque soit $v_0 \in [-1/2, 1/2[$ il existe au moins un k tel que $|k/M - v_0| \leq 1/M$.

5.3.2 Séparation de deux exponentielles complexes et fenêtrage

Cette fois-ci on possède un signal plus complexe qui est la somme de deux harmoniques complexes sur \mathbb{Z}

$$u_n = A_0 e^{2i\pi v_0 n} + A_1 e^{2i\pi v_1 n}$$

Les inconnues ici, sont les amplitudes A_0 et A_1 ainsi que les fréquences v_0 et v_1 . Encore une fois on ne se donne le droit que d'observer N échantillons, et on note u^T la suite ainsi tronquée.

Les graphiques de Figure 5.8 à Figure 5.11 illustrent le problème de la résolution fréquentielle en calculant la TFtD de u^T pour différentes valeurs de N .

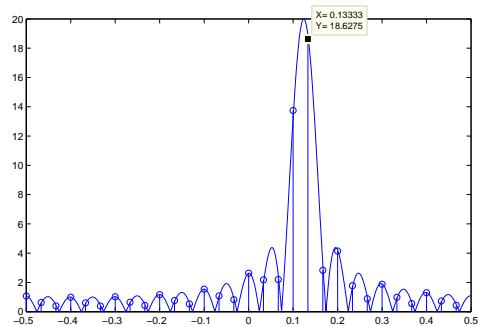


Figure 5.6: La TFD d'ordre 30 de l'harmonique complexe tronquée superposée à la TFtD. Le maximum de la TFD est atteint pour $k = 4$, soit une fréquence de $4/30 = 0,1333$ et une erreur d'estimation de 0,01

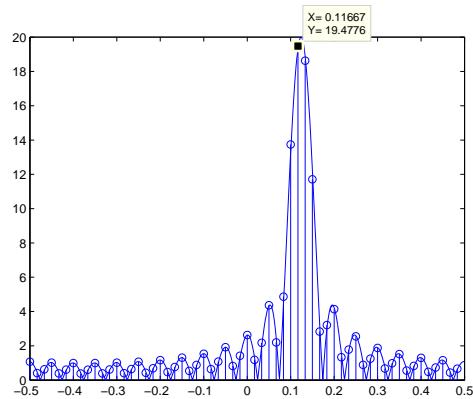


Figure 5.7: La TFD d'ordre 60 de l'harmonique complexe tronquée superposée à la TFtD. Le maximum de la TFD est atteint pour $k = 7$, soit une fréquence de $7/60 = 0,11667$ et une erreur d'estimation de 0,0063

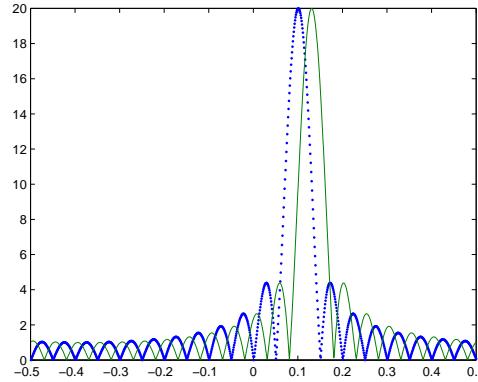


Figure 5.8: On a utilisé seulement $N = 20$ échantillons pour tracer la TFtD de deux ondes (l'une en pointillés, l'autre en trait plein) de même module et de fréquences 0,1 et 0,13.

On constate qu'il faut au moins avoir $|v_0 - v_1| > 1/N$ pour pouvoir distinguer deux pics sur la TFtD. Sinon, les deux pics se confondent en un seul et il sera impossible de distinguer v_0 et v_1 .

On dit que $1/N$ est la résolution fréquentielle. Il faut augmenter N pour pouvoir séparer des fréquences proches l'une de l'autre.

Les graphiques Figure 5.12 et Figure 5.12 montrent une situation où A_1 est beaucoup plus grand que A_0 . On constate que les lobes secondaires de la TFtD de l'harmonique complexe (tronquée) v_1 cachent jusqu'au lobe principal de l'harmonique complexe tronquée de fréquence v_0 . Cela est du au fait que la troncature choisie est trop brutale. Pour obtenir u^T nous avons multiplié u par une fenêtre rectangulaire que l'on va noter c :

$$c_n = \begin{cases} 1 & \text{si } 0 \leq n < N \\ 0 & \text{sinon} \end{cases}$$

et on a fait $u^T = u.c$. Comme vu plus haut la TFtD de u^T

est la convolution de la TFtD de u avec celle de c .

Figure 5.14 montre le module de la TFtD de c . Si l'on trouve une fenêtre dont la TFtD présente des lobes secondaires moins proéminents on peut espérer résoudre le problème que pose la séparation des deux ondes de notre mélange. Une fenêtre proposée est la fenêtre de Hamming définie par

$$h_n = \begin{cases} 0.54 - 0.46\cos(2\pi \frac{n}{N-1}) & \text{si } 0 \leq n < N \\ 0 & \text{sinon} \end{cases}$$

Figure 5.15 montre la TFtD de la fenêtre de Hamming. Par rapport à celle de c (créneau), on constate deux choses

1. Le lobe central est plus étalé : ceci implique une perte de résolution fréquentielle. On passe d'une résolution de l'ordre de $1/N$ à une résolution de l'ordre de $2/N$.
2. Les lobes secondaires sont très atténus par rapport au lobe central, ce qui permet de distinguer deux ondes dont les amplitudes diffèrent grandement.

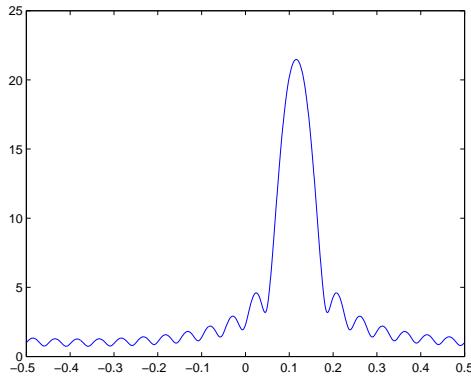


Figure 5.9: TFtD de la somme des deux ondes tronquées à 30 échantillons (Figure 5.8). On ne peut pas distinguer la superposition des deux harmoniques complexes dans le signal.

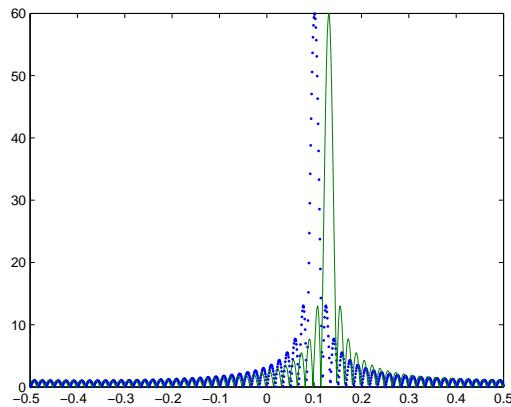


Figure 5.10: On a utilisé seulement $N = 20$ échantillons pour tracer la TFtD de deux ondes (l'une en pointillés, l'autre en trait plein) de même module et de fréquences 0,1 et 0,13.

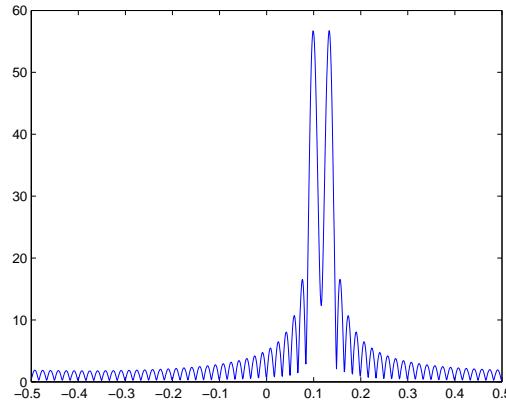


Figure 5.11: TFtD de la somme des deux ondes tronquées à 60 échantillons (Figure 5.10). Cette fois on distingue bien les deux ondes. Il a fallu prendre un nombre d'échantillons N supérieur à $1/(0.13 - 0.1) = 33.3$ pour arriver à distinguer les deux.

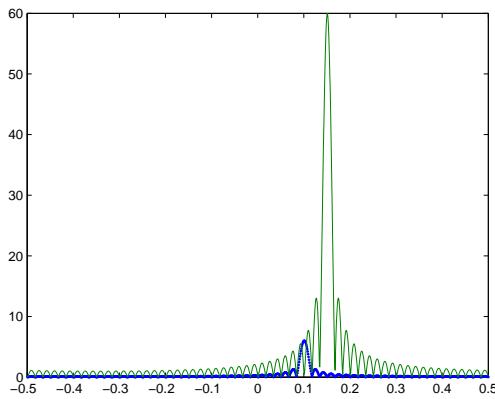


Figure 5.12: TFtD de deux ondes tronquées dont l'une a une amplitude 10 fois plus petite que l'autre. La plus petite des deux est cachée par les lobes secondaires de la TFtD de l'autre.

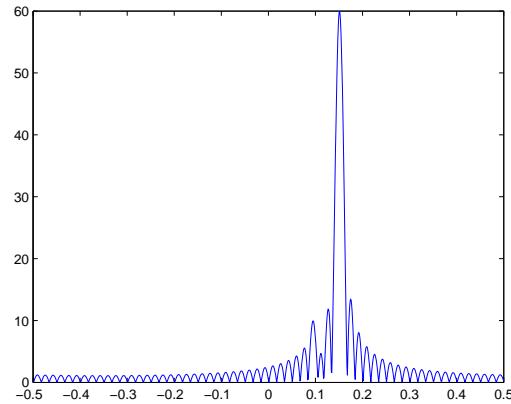


Figure 5.13: TFtD de la somme des deux ondes tronquées. Il est difficile de déceler la présence de l'harmonique complexe de faible amplitude.

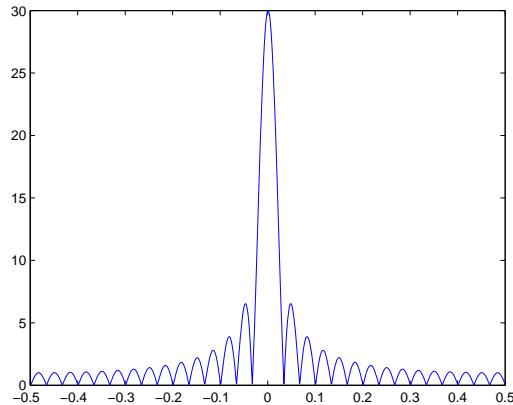


Figure 5.14: TFtD d'un créneau de taille 30.

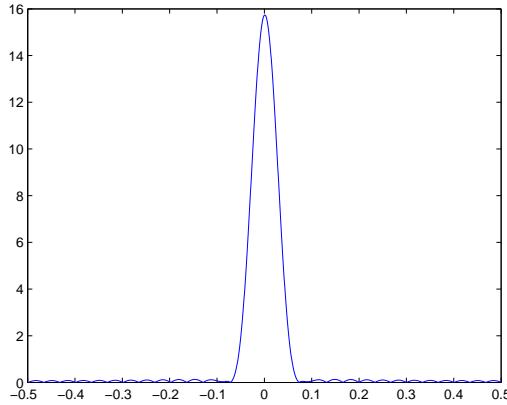


Figure 5.15: TFtD d'une fenêtre de Hamming de taille $N = 30$.

Figure 5.16 et Figure 5.17 montrent comment la multiplication par une fenêtre de Hamming plutôt qu'une troncature brutale permet de distinguer une onde de faible amplitude.

5.4 Transformée de Fourier à Court Terme (TFCT)

La Transformée de Fourier à Court Terme (TFCT) n'est pas à proprement parler une transformation de Fourier. Elle n'en a pas les propriétés algébriques remarquables, c'est pourtant un outil essentiel en traitement du signal. Cet outil est basé sur la constatation déjà faite plus haut que le contenu fréquentiel d'un signal peut évoluer au cours du temps. Elle se définit naïvement comme une analyse locale des composantes fréquentielles du signal. Plus précisément, pour chaque instant n , on extrait un certain nombre d'échantillons du signal étudié autour du point n que l'on étudie par les moyens vus ci-dessus (fenêtrage et TFD d'ordres arbitraires).

Si u est une suite définie sur \mathbb{Z} . On fixe une fenêtre w_0, \dots, w_n de taille N et on choisit un entier $M \geq N$. La Transformée de Fourier à Court Terme de u de fenêtre w et de précision $1/M$ est une fonction, que l'on note U définie sur $\mathbb{Z} \times \frac{\{0, \dots, M-1\}}{M}$ par

$$\forall (n, k) \in \mathbb{Z} \times \{0, \dots, M-1\}, U(n, \frac{k}{M}) = \sum_{m \in \mathbb{Z}} u_m w_{n-m} e^{-2i\pi \frac{k}{M} m}$$

On peut aussi considérer U comme une fonction définie sur $\mathbb{Z} \times [-1/2, 1/2[$ que l'on échantillonnera aussi finement que l'on veut suivant la seconde variable en augmentant la valeur de M (ordre de la TFD)

$$\forall (n, v) \in \mathbb{Z} \times [-1/2, 1/2[, U(n, v) = \sum_{m \in \mathbb{Z}} u_m w_{m-n} e^{-2i\pi 1/m}$$

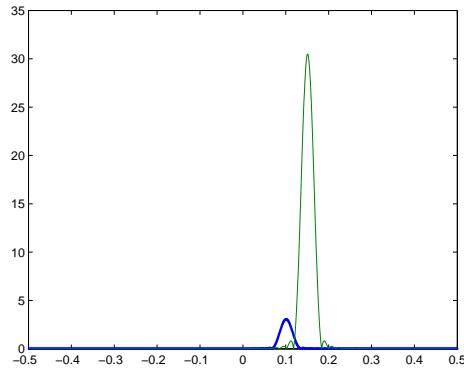


Figure 5.16: Même figure que Figure 5.12 mais en ayant multiplié le signal par une fenêtre de Hamming. Cette fois l'harmonique complexe de faible amplitude est bien au dessus des lobes secondaires de l'harmonique complexe de forte amplitude.

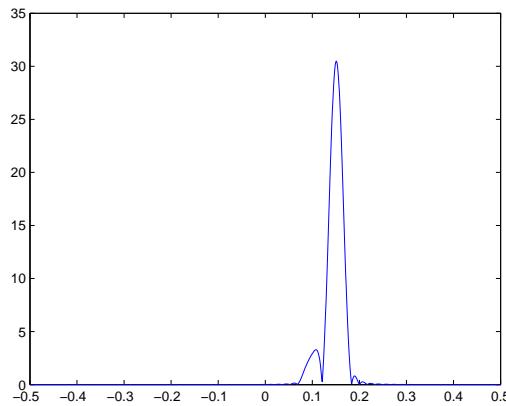


Figure 5.17: Même figure que Figure 5.13 mais en ayant multiplié le signal contre une fenêtre de hamming plutôt que de la tronquer brutalement. Cette fois-ci on constate bien un deuxième lobe qui ne peut être confondu avec le lobe secondaire engendré par l'harmonique complexe prédominante, car ces lobes secondaires sont bien plus faibles d'après Figure 5.15

On distingue cette notation de la notation U pour la *TFD* par le fait qu'elle dépend de deux variables.

Pour n fixé : On remarque que pour un entier n fixé, la fonction $v \mapsto U(n, v)$ est la TFtD de la suite $l \mapsto u_l w_{l-n}$, c'est à dire la suite u multipliée par la n -translatée de la fenêtre w . Il s'agit bien de ce que nous avions annoncé, autour de chaque instant n , on extrait une fenêtre de signal dont on calcule la TFtD (à l'aide d'une TFD dont le nombre de coefficients peut être choisi de façon arbitraire).

Pour v fixé : Pour une fréquence v fixée avec n variable, on a :

$$U(n, v) = \sum_{m \in \mathbb{Z}} u_m w_{m-n} e^{-2i\pi l/m} = e^{-2i\pi l/n} \sum_m u_m \gamma_{n-m}$$

où γ est la suite définie par

$$\gamma_l = w_{-l} e^{2i\pi v l}$$

Le module de U est alors

$$|U(n, v)| = |(u * \gamma)_n|$$

Ainsi le module de U est celui de la convolution de la symétrique de w multipliée par une harmonique complexe de fréquence v . Cela signifie qu'à v fixé, le module de U reflète à quel point la fréquence v est présente dans le signal autour du point n . En effet, la TFtD de γ est centrée autour de la fréquence v (la fenêtre w , si c'est une fenêtre de Hamming par exemple, a son spectre centré en zéro).

Le spectrogramme est le module au carré de la TFCT $((n, v) \mapsto |U(n, v)|^2)$. On le visualise comme une image, les deux axes sont ceux des variables n et v , on représente la valeur du spectrogramme soit en gris, suivant la valeur (sombre pour grand et clair pour faible).

i

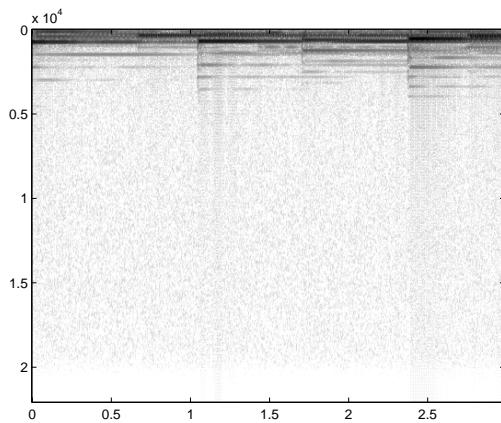


Figure 5.18: Spectrogramme d'un morceau de piano. On voit se succéder les notes. Chaque note est caractérisée par l'apparition de raies qui s'affaiblissent à mesure que le son s'atténue. (l'échelle des fréquences est du haut vers le bas et en 10000 Hz d'unité)

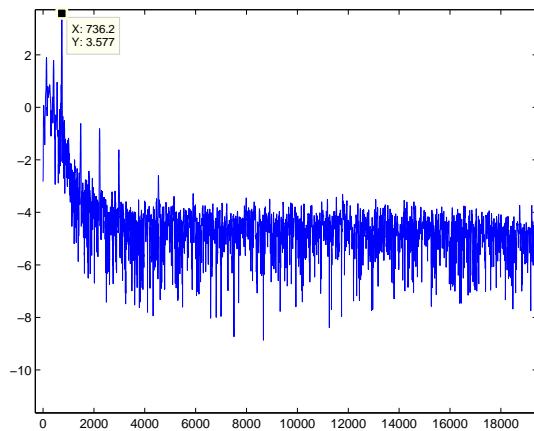


Figure 5.19: Une colonne du spectrogramme. C'est donc le contenu fréquentiel local autour d'un certain instant. Le pic le plus proéminent est pour la fréquence 736 Hz, qui correspond à peu près à un Fa dièse.

Chapitre 6

Transformée en z et filtrage

6.1 Transformée en z : définition

Définition 63 (Transformée en z). La transformée en z d'une suite $\{u_n, n \in \mathbb{Z}\}$ est définie comme la série $U(z)$ calculée comme suit:

$$U(z) = \sum_{n=-\infty}^{\infty} u_n z^{-n} \quad (6.1)$$

où z est une variable complexe.

On appelle encore (6.1) la transformée directe, car c'est la relation qui permet d'obtenir $U(z)$ à partir de $\{u_n, n \in \mathbb{Z}\}$. Cette transformée est bilatérale. L'opération inverse porte le nom de transformation inverse.

Comme cette transformation est une série infinie, elle n'existe que pour les valeurs de z pour lesquelles cette série converge. La région de convergence (RC) est l'ensemble des valeurs de z pour lesquelles la série prend une valeur finie. Dès lors, toute transformée en z doit être accompagnée de la région du plan complexe sur laquelle elle converge. Pour déterminer la région de convergence, on utilise le critère de Cauchy sur la convergence des séries entières. Rappelons que la série à termes positifs $\sum_{n=0}^{\infty} v_n$ converge si

$$\limsup_{n \rightarrow \infty} v_n^{1/n} < 1. \quad (6.2)$$

Pour appliquer le critère de Cauchy, on décompose la série en deux séries:

$$U(z) = \sum_{n=-\infty}^{-1} u_n z^{-n} + \sum_{n=0}^{\infty} u_n z^{-n} = U_1(z) + U_2(z).$$

L'application du critère de Cauchy à la série $U_2(z)$ mène à

$$\limsup_{n \rightarrow \infty} |u_n z^{-n}|^{1/n} < 1 \Rightarrow \limsup_{n \rightarrow \infty} |u_n|^{1/n} < |z|$$

En appelant R_- la limite

$$R_- = \limsup_{n \rightarrow \infty} |u_n|^{1/n} \quad (6.3)$$

la série $U_2(z)$ converge sur la couronne $\{z \in \mathbb{C} : |z| > R_-\}$. Pour ce qui est de la série $U_1(z)$, après un changement de variable, on a

$$U_1(z) = \sum_{n=-\infty}^{-1} u_n z^{-n} = \sum_{n=1}^{\infty} u_{-n} z^n$$

On a convergence si

$$\limsup_{n \rightarrow \infty} |u_{-n} z^n|^{1/n} < 1 \Rightarrow \limsup_{n \rightarrow \infty} |u_{-n}|^{1/n} |z| < 1$$

et donc la série $U_1(z)$ converge sur le disque

$$|z| < \left\{ \lim_{n \rightarrow \infty} |u_{-n}|^{1/n} \right\}^{-1} = R+ \quad (6.4)$$

En toute généralité, une série converge donc dans un anneau du plan complexe donné par

$$R_- < |z| < R_+$$

Lorsque $R_+ \leq R_-$, le domaine de convergence est vide et la transformée en z de la suite n'est alors pas définie.

La suite $U_2(z)$ représente la transformée en z d'une *suite causale*: les seules valeurs non-nulles de la suite correspondent aux indices positifs. La transformée en z d'une suite causale converge à l'extérieur d'un cercle de rayon R_- donné par (6.3).

La suite $U_1(z)$ représente la transformée en z d'une suite anti-causale, c'est-à-dire ne comportant des éléments que pour les valeurs négatives de l'indice. En général, une suite anti-causale converge à l'intérieur d'un cercle de rayon R_+ donné par (6.4). Quand une suite est à durée limitée, sa transformée est donnée par

$$U(z) = \sum_{n=n_1}^{n_2} u_n z^{-n} \quad (6.5)$$

Pour autant que dans l'intervalle $[n_1, n_2]$ le module de chaque élément de la suite soit fini, la série converge pour toutes les valeurs de z , sauf éventuellement en $z = 0$ ou $|z| \rightarrow \infty$. Les cas suivants peuvent être distingués:

1. si n_1 et n_2 sont positifs, on n'a pas convergence pour $z = 0$ car les termes z^{-n} divergent pour les n positifs.
2. si n_1 est négatif et n_2 est positif, on n'a pas convergence ni pour $z = 0$ ni $|z| \rightarrow \infty$.
3. si n_1 et n_2 sont négatifs, on n'a pas convergence pour $|z| \rightarrow \infty$.

Considérons les exemples suivants:

1. $u_n = \delta_n$: la définition fournit directement $U(z) = 1$. La transformée existe partout.

2. $u_n = 1$ si $n \geq 0$ et $u_n = 0$ sinon :

$$U(z) = \sum_{n=0}^{\infty} z^{-n} = \frac{1}{1 - z^{-1}}$$

$$\text{pour } R_- = \lim_{n \rightarrow \infty} 1^{1/n} = 1^1$$

3. $u_n = a^n$ pour $n \geq 0$, $u_n = 0$ sinon

$$\sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty} (az^{-1})^n = \frac{1}{1 - az^{-1}} \quad (6.6)$$

avec un domaine de convergence $|z| > |a|$.

4. $u_n = e^{2i\pi\nu_0 n}$

$$U(z) = \sum_{n=0}^{\infty} e^{2i\pi\nu_0 n} z^{-n} = \sum_{n=0}^{\infty} (e^{2i\pi\nu_0} z^{-1})^n = \frac{1}{1 - e^{2i\pi\nu_0} z^{-1}}$$

avec un domaine de convergence $|z| > 1$.

5. La suite $u_n = a^n$ pour $n \in \mathbb{Z}$ n'a pas de transformée en z .

6.2 Transformée inverse

Pour inverser une transformée en z , on peut s'aider utilement du *théorème de Cauchy* qui établit que

$$\frac{1}{2\pi i} \oint_{\Gamma} z^{l-1} dz = \begin{cases} 1 & l = 0 \\ 0 & \text{sinon} \end{cases} \quad (6.7)$$

où Γ est un contour qui entoure l'origine du plan et est parcouru dans le sens trigonométrique. En reprenant la définition de la transformée en z donnée par (6.1), en multipliant les deux membres par z^{l-1} et en intégrant le long d'un contour entourant l'origine et appartenant au domaine de convergence, on trouve

$$\begin{aligned} \oint_{\Gamma} U(z) z^{l-1} dz &= \oint_{\Gamma} \sum_{n=-\infty}^{\infty} u_n z^{-n+l-1} dz \\ &= \sum_{n=-\infty}^{\infty} u_n \oint_{\Gamma} z^{-n+l-1} dz \end{aligned}$$

où l'interversion de l'intégrale et de la somme est licite compte tenu du fait que l'on opère dans la zone de convergence de la transformée. En utilisant le théorème de Cauchy (6.7), on a finalement

$$u_n = \frac{1}{2\pi i} \oint_{\Gamma} U(z) z^{n-1} dz \quad (6.8)$$

avec les conditions déjà énoncées à propos du contour d'intégration.

L'évaluation de l'intégrale dans le plan complexe se fait à l'aide du théorème des résidus, qui établit que l'intégrale le long d'un contour est donné par la somme des résidus de la fonction à intégrer, soit ici $U(z)z^{n-1}$, dans le contour Γ . Le résidu r_q à un pôle d'ordre q en $z = a$ est donné par

$$r_q = \lim_{z \rightarrow a} \frac{1}{(q-1)!} \frac{d^{q-1}}{dz^{q-1}} [U(z)z^{n-1}(z-a)^q] \quad (6.9)$$

Pour un pôle simple ($q = 1$) en $z = a$, l'expression du résidu r_1 se réduit à

$$r_1 = \lim_{z \rightarrow a} [U(z)z^{n-1}(z-a)] \quad (6.10)$$

Exemple 64. Considérons la transformée en z donnée par

$$U(z) = \frac{1}{1 - z^{-1}}$$

et le domaine de convergence $|z| > 1$. En utilisant la formule d'inversion, on a donc que

$$u_n = \frac{1}{2\pi i} \oint_{\Gamma} \frac{z^{n-1}}{1 - z^{-1}} dz = \frac{1}{2\pi i} \oint_{\Gamma} \frac{z^n}{z-1} dz$$

où le contour Γ peut être un cercle de rayon plus grand que l'unité. Pour $n \geq 0$, on n'a qu'un pôle d'ordre 1 en $z = 1$ qui est entouré par le contour. Le résidu en ce pôle est donné par r_1 valant

$$r_1 = \lim_{z \rightarrow 1} [z^n] = 1$$

On a donc $u_n = 1$ pour tout $n \geq 0$. En ce qui concerne $n < 0$, on a cette fois un autre pôle d'ordre $(-n)$ en $z = 0$. L'application de la formule du résidu donne, pour $q = -n$,

$$r_q = \lim_{z \rightarrow 0} \frac{1}{(q-1)!} \frac{d^{q-1}}{dz^{q-1}} \left[\frac{1}{z} \right] = -1$$

autre résidu vaut toujours 1 et la somme donne donc 0. Pour éviter l'évaluation fastidieuse du résidu en un pôle d'ordre non égal à un, on peut recourir à un changement de variable $w = 1/z$. On a dès lors que le domaine de convergence devient $|w| < 1$, et l'intégrale à évaluer devient

$$\begin{aligned} u_n &= \frac{1}{2\pi j} \oint_{\Gamma} \frac{z^n}{z-1} dz \\ &= \frac{1}{2\pi j} \oint_{\Gamma^-} \frac{w^{-n}}{w^{-1}-1} \frac{(-1)}{w^2} dw \\ &= \frac{1}{2\pi j} \oint_{\Gamma^+} \frac{w^{-n-1}}{1-w} dw = 0 \end{aligned}$$

où le contour Γ^- peut être un cercle de rayon inférieur à l'unité et parcouru dans le sens anti-trogonométrique (à cause du changement de variable). Le contour Γ^+ est parcouru dans le sens trigonométrique et compense le signe -. Pour $n < 0$, on n'a pas de pôle en 0. Le seul pôle est en $w = 1$ mais n'est pas entouré par le domaine d'intégration qui doit appartenir au domaine de convergence, soit $|w| < 1$. Aucun pôle n'est donc entouré et la somme des résidus est nulle.

6.3 Décompositions en fractions simples

En utilisant d'ores et déjà la propriété de linéarité de la transformée en z , qui sera vue plus loin, on peut décomposer une fonction de forme compliquée en une somme de fonctions simples, et prendre la transformée inverse de chacun des éléments. Comme on le verra, de nombreux systèmes requièrent l'étude de transformées du type

$$U(z) = P(z)/Q(z) \quad (6.11)$$

où P et Q sont des polynômes en z ou z^{-1} . On peut alors décomposer $U(z)$ en fractions simples et obtenir la transformée inverse par la somme des transformées inverses. La transformée peut se mettre sous la forme

$$U(z) = S(z) + P_0(z)/Q(z)$$

où le degré de $P_0(z)$ est inférieur à celui de $Q(z)$. En fait, $S(z)$ et $P_0(z)$ sont respectivement le quotient et le reste de la division de P par Q . Si le degré de P est inférieur à celui de Q , le quotient S est nul.

Lorsque les racines p_i de $Q(z)$, appelées pôles sont simples, on peut mettre le quotient sous la forme

$$P_0(z)/Q(z) = \sum_{i=1}^N \frac{\alpha_i}{z - p_i} \quad (6.12)$$

Pour obtenir les poids α_i , il suffit d'effectuer le calcul suivant:

$$\alpha_i = [(z - p_i)P_0(z)/Q(z)]_{z=p_i}.$$

Si un pôle p_n est d'ordre $q > 1$, la décomposition prend la forme

$$P_0(z)/Q(z) = \sum_{i=1, \neq n}^N \frac{\alpha_i}{z - p_i} + \sum_{j=1}^q \frac{\beta_j}{(z - p_n)^j}$$

avec

$$\beta_j = \frac{1}{(q-j)!} \frac{d^{q-j}}{dz^{q-j}} [(z - p_n)^j P_0(z)/Q(z)]_{z=p_n} \quad (2.40)$$

En réalité, il est plus intéressant d'obtenir des fractions où z^{-1} apparaît au dénominateur, car les fractions dans ce cas correspondent directement à des transformées connues. Dès lors, tout ce qui vient d'être dit est appliqué pour des polynômes en z^{-1} et non pas en z .

Exemple 65. Considérons la transformée donnée par

$$U(z) = \frac{1}{1 - 3z^{-1} + 2z^{-2}}$$

pour $|z| > 2$. Comme nous considérons que la variable est z^{-1} , le degré du numérateur est bien inférieur à celui du dénominateur. Les pôles sont donnés par $z^{-1} = 1$ et $z^{-1} = 0.5$. On recherche donc une décomposition en fractions simples du type

$$\begin{aligned} U(z) &= \frac{0.5}{(z^{-1} - 1)(z^{-1} - 0.5)} \\ &= \frac{\alpha_1}{z^{-1} - 1} + \frac{\alpha_2}{z^{-1} - 0.5} \end{aligned}$$

On trouve en utilisant ce qui a été vu précédemment,

$$U(z) = \frac{1}{z^{-1} - 1} + \frac{-1}{z^{-1} - 0.5}$$

Dès lors,

$$U(z) = \frac{2}{1 - 2z^{-1}} - \frac{1}{1 - z^{-1}}$$

et il est possible d'identifier ces fractions au résultat obtenu en (6.6). On trouve de ce fait

$$u_n = 2 \times 2^n \mathbb{1}_{\mathbb{N}}(n) - \mathbb{1}_{\mathbb{N}}(n) = (2^{n+1} - 1) \mathbb{1}_{\mathbb{N}}(n).$$

Si on avait considéré les polynômes comme des fonctions de z, on aurait eu

$$\begin{aligned}
 U(z) &= \frac{1}{1 - 3z^{-1} + 2z^{-2}} \\
 &= \frac{z^2}{z^2 - 3z + 2} \\
 &= 1 + \frac{3z - 2}{z^2 - 3z + 2} \\
 &= 1 + \frac{3z - 2}{(z - 2)(z - 1)} \\
 &= 1 + \frac{4}{(z - 2)} + \frac{-1}{(z - 1)} \\
 &= 1 + \frac{4z^{-1}}{(1 - 2z^{-1})} - \frac{1}{(1 - z^{-1})}
 \end{aligned}$$

En s'aidant de la propriété de décalage qui sera vue plus loin, on a que

$$u_n = \delta_n + 4 \times 2^{n-1} \mathbb{1}_{\mathbb{N}}(n-1) - \mathbb{1}_{\mathbb{N}}(n-1).$$

Cette forme est moins concise que la précédente.

6.4 Propriétés de la transformée en z

6.4.1 Linéarité

La linéarité de la transformation signifie que la transformée d'une suite obtenue par combinaison linéaire d'autres suites n'est rien d'autre que la combinaison linéaire des transformées correspondantes. Sidonc

$$u_n = ax_n + by_n$$

alors

$$\begin{aligned}
 U(z) &= \sum_{n=-\infty}^{\infty} u_n z^{-n} = a \sum_{n=-\infty}^{\infty} x_n z^{-n} + b \sum_{n=-\infty}^{\infty} y_n z^{-n} \\
 &= a X(z) + b Y(z)
 \end{aligned}$$

La région de convergence est au moins l'intersection des régions associées à $X(z)$ et $Y(z)$ car la combinaison linéaire peut introduire des zéros qui compensent certains pôles.

Décalage et transformée bilatérale

Soit

$$v_n = u_{n-n_0}$$

La transformée en z de $\{v_n, n \in \mathbb{Z}\}$ est donnée par

$$\begin{aligned} V(z) &= \sum_{n=-\infty}^{\infty} v_n z^{-n} \\ &= \sum_{n=-\infty}^{\infty} u_{n-n_0} z^{-n} \\ &= \sum_{m=-\infty}^{\infty} u_m z^{-(m+n_0)} \\ &= z^{-n_0} U(z) \end{aligned}$$

Le domaine de convergence est le même que pour $U(z)$ sauf que si n_0 est positif (resp. négatif), on a un pôle en 0 (resp. $z \rightarrow \infty$). Il peut y avoir des compensations de zéros ou pôles.

Multiplier une transformée en z par z^{-1} revient donc à retarder la suite d'une unité (période d'échantillonnage). On se réfère souvent à z^{-1} comme opérateur de retard unité.

La propriété de décalage permet de traiter les équations aux différences vues précédemment. A partir de

$$v_n = u_n - \sum_{k=1}^p b_k u_{n-k}$$

on trouve par transformation en z ,

$$V(z) = U(z) - \sum_{k=1}^p b_k z^{-k} V(z)$$

ce qui implique que

$$V(z) = \frac{U(z)}{1 + \sum_{k=1}^p b_k z^{-k}}$$

Il faut faire attention toutefois avec ce calcul "formel" que les régions de convergence de $U(z)$ et de $1/(1 + \sum_{k=1}^p b_k z^{-k})$ soient compatibles. Nous reviendrons sur cet aspect plus loin dans l'exposé.

Décalage et transformée unilatérale

Dans le cas unilatéral, il faut être plus prudent dans ce que l'on écrit. La transformée unilatérale de la suite $\{u_n, n \in \mathbb{Z}\}$ est donnée par

$$\tilde{U}(z) = \sum_{k=0}^{\infty} u_k z^{-k}. \quad (6.13)$$

Notons que nous ne calculons la somme que pour les indices positifs ! Le rayon de convergence de la transformée monolatérale est une couronne donnée $z \in \mathbb{C}|z| \geq \limsup_{n \rightarrow \infty} |u_n|^{1/n}$.

Si on pose $v_n = u_{n-1}$, la transformée bilatérale est calculée par

$$\begin{aligned}
 \tilde{V}(z) &= \sum_{n=0}^{\infty} v_n z^{-n} \\
 &= \sum_{n=0}^{\infty} u_{n-1} z^{-n} \\
 &= z^{-1} \sum_{m=-1}^{\infty} u_m z^{-m} \\
 &= z^{-1}[u_{-1} z + \tilde{U}(z)] \\
 &= u_{-1} + z^{-1} \tilde{U}(z)
 \end{aligned} \tag{6.14}$$

On voit donc qu'un décalage vers la droite permet de faire apparaître des éléments qui ne sont pas pris en considération par la transformation unilatérale de départ. Cette propriété permet de prendre en compte les conditions initiales non nulles. Soit

$$v_n = u_n + av_{n-1}$$

avec $v_{-1} = K$ et une excitation $u_n = e^{2i\pi\nu_0 n} \mathbb{1}_N(n)$. En vertu de (6.14) la transformée en z donne

$$\begin{aligned}
 \tilde{V}(z) &= \tilde{U}(z) + av_{-1} + az^{-1} \tilde{U}(z) \\
 &= \frac{\tilde{U}(z) + av_{-1}}{1 - az^{-1}} \\
 &= \frac{aK}{1 - az^{-1}} + \frac{1}{(1 - az^{-1})(1 - e^{2rm i \pi \nu_0} z^{-1})} \\
 &= \frac{aK}{1 - az^{-1}} + \frac{1}{(1 - e^{2i\pi\nu_0}/a)(1 - az^{-1})} + \frac{1}{(1 - a/e^{2i\pi\nu_0})(1 - e^{2rm i \pi \nu_0} z^{-1})} \\
 &= \frac{aK}{1 - az^{-1}} + \frac{a}{(a - e^{2i\pi\nu_0})(1 - az^{-1})} + \frac{e^{2i\pi\nu_0}}{(e^{2i\pi\nu_0} - a)(1 - e^{2i\pi\nu_0} z^{-1})}
 \end{aligned}$$

De ce fait, en utilisant les transformées inverses déjà établies, on trouve

$$v_n = \mathbb{1}_N(n)[a^{n+1} K + \frac{a^{n+1}}{a - e^{2i\pi\nu_0}} - \frac{e^{2i\pi\nu_0(n+1)}}{a - e^{2i\pi\nu_0}}]$$

La première partie de la réponse correspond à la réponse libre, c'est-à-dire l'évolution du système sous l'effet des conditions initiales. La seconde partie correspond à la partie transitoire de la réponse forcée. La troisième partie correspond à la réponse de régime de la réponse forcée.

On peut bien évidemment généraliser ce qui précède au cas de plusieurs conditions initiales.

Dérivation

Comme

$$U(z) = \sum_{n=-\infty}^{\infty} u_n z^{-n}$$

on a aussi

$$\frac{dU(z)}{dz} = \sum_{n=-\infty}^{\infty} (-n)u_n z^{-n-1}$$

Et donc

$$-z \frac{dU(z)}{dz} = \sum_{n=-\infty}^{\infty} n u_n z^{-n}$$

De ce fait, il apparaît que $-z dU(z)/dz$ est la transformée de la suite $\{nu_n, n \in \mathbb{Z}\}$.

Convolution

Cette propriété est une des plus importantes et justifie à elle seule l'usage qui est fait de la transformée en z pour étudier les systèmes linéaires permanents en temps discret. Si $\{y_n, n \in \mathbb{Z}\}$ est obtenu par convolution de $\{x_n, n \in \mathbb{Z}\}$ et $\{g_n, n \in \mathbb{N}\}$, on a que

$$y_n = \sum_{m=-\infty}^{\infty} x_m g_{n-m}$$

La transformée en z , $Y(z)$, de $\{y_n, n \in \mathbb{N}\}$ est donc obtenue par

$$\begin{aligned} Y(z) &= \sum_{n=-\infty}^{\infty} y_n z^{-n} \\ &= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} x_m g_{n-m} z^{-n} \\ &= \left[\sum_{m=-\infty}^{\infty} x_m z^{-m} \right] \left[\sum_{n=-\infty}^{\infty} g_{n-m} z^{-(n-m)} \right] \\ &= X(z)G(z) \end{aligned}$$

Cette opération est valable pour les valeurs de z appartenant à l'intersection des domaines de convergence des deux transformées.

Considérons un système linéaire et permanent de réponse impulsionnelle

$$g_n = a^n \mathbb{1}_{\mathbb{N}}(n)$$

et $|a| < 1$ dont le signal à l'entrée est la suite

$$x_n = b^n \mathbb{1}_{\mathbb{N}}(n)$$

et $|b| < 1$ avec de plus $|b| > |a|$. La sortie $\{y_n, n \in \mathbb{N}\}$ est obtenue par $y = x * g$. Les transformées $X(z)$ et $G(z)$ sont données par

$$X(z) = \frac{1}{1 - bz^{-1}} \quad \text{pour } |z| > |b|$$

$$G(z) = \frac{1}{1 - az^{-1}} \quad \text{pour } |z| > |a|$$

En vertu de ce qui précède, on a donc pour $|z| > |b|$ qui est donc l'intersection des domaines de convergence,

$$\begin{aligned} Y(z) &= G(z)X(z) = \frac{1}{1-az^{-1}} \frac{1}{1-bz^{-1}} \\ &= \frac{1}{(1-b/a)(1-az^{-1})} + \frac{1}{(1-a/b)(1-bz^{-1})} \\ &= \frac{a}{(a-b)(1-az^{-1})} + \frac{b}{(b-a)(1-bz^{-1})} \end{aligned}$$

et donc, compte tenu des transformées calculées précédemment,

$$y_n = \mathbb{1}_{\mathbb{N}}(n) \left[\frac{a}{(a-b)} a^n + \frac{b}{(b-a)} b^n \right]$$

6.5 Pôles et zéros

Dans de nombreux cas d'intérêt, la transformée en z est une fraction rationnelle

$$H(z) = \frac{\prod_{m=1}^M (1-z_m z^{-1})}{\prod_{n=1}^N (1-p_n z^{-1})} \quad (6.15)$$

où les $\{z_m\}_{m=1}^M$ sont les *zéros*, et les $\{p_n\}_{n=1}^N$ sont les *pôles*. Pour des signaux et systèmes réels, ces polynômes sont à coefficients réels, et leurs racines (pôles et zéros) sont soit réelles, soit par paires complexes conjuguées.

Considérons à titre d'exemple la suite causale $h_n = a^n \mathbb{1}_{\mathbb{N}}(n)$ avec $|a| < 1$. La transformée en z de cette suite est donnée par

$$H(z) = \frac{1}{1-az^{-1}} = \frac{z}{z-a} \quad (6.16)$$

sur la couronne $\{z \in \mathbb{C} : |z| > |a|\}$. Remarquons que $z \mapsto U(z)$ est défini aussi sur le disque $\{z \in \mathbb{C} : |z| < a\}$ et donc que la donnée de (6.16) sans préciser le domaine de convergence ne permet pas de reconstruire la suite. Cette transformée possède un zéro en $z = z_1 = 0$ et un pôle en $z = p_1 = a$. La Figure 6.1 illustre le cas d'un pôle a réel.

Le calcul de la transformée de Fourier requiert d'évaluer la transformée en z sur le cercle de rayon unité. Pour une fréquence v , on pose $z = e^{+2i\pi v}$ et donc

$$U(e^{+2i\pi v}) = \frac{e^{+2i\pi v}}{e^{+2i\pi v}-a}$$

Quand la fréquence change, le module du numérateur ne change pas. Par contre le module du dénominateur décroît lorsque l'on se rapproche d'un pôle. Plus précisément, lorsque $2\pi v$ se rapproche de l'argument d'un pôle, le dénominateur devient petit, et le module de la transformée de Fourier devient grand. Le module de la transformée de Fourier prend donc des valeurs importantes pour des valeurs de pulsations proches de l'argument du pôle. Cette valeur sera d'autant plus élevée que le pôle a un module proche de l'unité. On aurait pu tenir le même raisonnement avec les zéros, qui

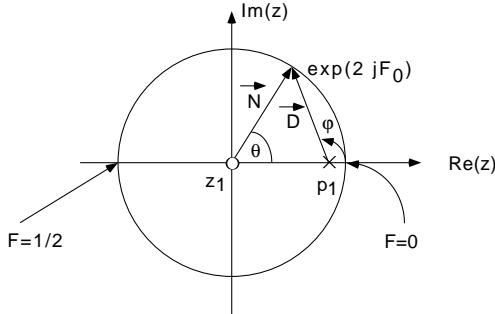


Figure 6.1: Réponse en fréquence de $z \mapsto U(z) = 1/(1 - az^{-1})$

occasionnent quant à eux des valeurs faibles (voire des zéros de transmission) de la transformée de Fourier. Ici pour $v = 0$, le module du dénominateur a une certaine valeur. Ce module croît au fur et à mesure que l'on s'éloigne de cette fréquence et atteint son maximum en $v = -1$. Par ailleurs, le signal est ici réel et on a en plus que le module de la transformée de Fourier est pair, et l'argument, impair. Ces considérations sont illustrées par la transformée de Fourier donnée à la Figure 6.2 pour $a = 0.9$.

Les considérations qui ont permis de prédire l'allure du spectre en module et argument peuvent aussi être utilisées dans des situations plus compliquées. Considérant la transformée en z donnée par (6.15). Dans ce cas le module et l'argument sont donnés par

$$|H(e^{2i\pi v})| = \frac{\prod_{m=1}^N |1 - z_m e^{-2i\pi v}|}{\prod_{n=1}^M |1 - p_n e^{-2i\pi v}|}$$

$$\begin{aligned} H(e^{2i\pi v}) &= \arg C + \sum_{m=1}^M \arg(e^{2i\pi v} - z_m) \\ &\quad - \sum_{n=1}^N \arg(e^{2i\pi v} - p_n) \end{aligned}$$

6.6 Fonction de transfert

On a vu que la sortie $\{v_n, n \in \mathbb{N}\}$ d'un système linéaire invariant dans le temps de réponse impulsionnelle $\{h_n, n \in \mathbb{N}\}$ est donnée par la convolution du signal d'entrée $\{u_n, n \in \mathbb{N}\}$ et de la réponse impulsionnelle $\{h_n, n \in \mathbb{N}\}$, i.e. $v_n = (h * x)_n$ pour tout $n \in \mathbb{Z}$. La transformée en z d'un produit de convolution étant égal au produit des transformées (la région de convergence étant l'intersection des régions de convergence

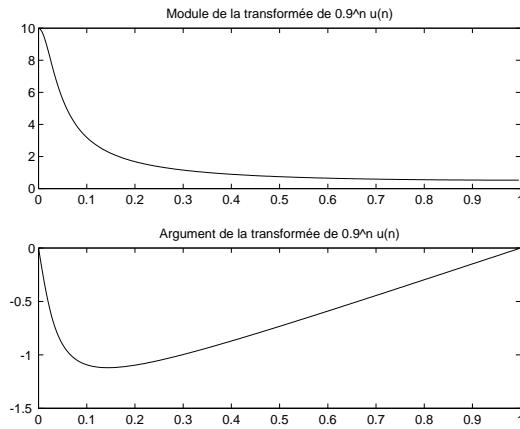


Figure 6.2: Transformée de Fourier, module en haut, argument en bas, de $h_n = a^n \mathbb{1}_{\mathbb{N}}(n)$ pour $a = 0.9$

de $U(z)$ et $H(z)$), nous avons donc

$$V(z) = H(z)U(z).$$

La transformée en z , $H(z)$ de la réponse impulsionale $\{h_n, n \in \mathbb{Z}\}$ porte le nom de *fonction de transfert* du système linéaire invariant. Cette transformée en z , évaluée sur le cercle de rayon unité donne la transformée de Fourier $H(e^{2i\pi v})$ de cette réponse impulsionale porte le nom de *transmittance du système* (on utilise aussi fonction de transfert pour cette quantité, bien que cette terminologie soit un peu incorrecte).

6.7 Causalité et Stabilité

Définition 66 (SLI causal). *On dit qu'un système linéaire invariant dans le temps est causal si $h_n = 0$ pour $n < 0$ où $\{h_n, n \in \mathbb{Z}\}$ est sa réponse impulsionale*

On a vu que la région de convergence de la transformée en z d'une suite causale $\{h_n, n \in \mathbb{N}\}$ est une couronne $\{z \in \mathbb{C} : |z| \geq \limsup_{n \rightarrow \infty} |h_n|^{1/n} = R_-\}$. Comme par définition une transformée $H(p) = \infty$ lorsque $z = p$ est un pôle, une suite est causale si et seulement si tous les pôles sont nécessairement contenus à l'intérieur d'un disque.

Proposition 67. *la fonction de transfert d'un système causal (c'est-à-dire dont la réponse impulsionale est causale) a ses pôles à l'intérieur d'un cercle.*

Définition 68 (SLI stable). *un système linéaire invariant dans le temps est stable si sa réponse impulsionale $\{h_n, n \in \mathbb{N}\}$ est absolument sommable,*

$$\sum_{n=-\infty}^{\infty} |h_n| < \infty.$$

Or,

$$|H(z)| = \left| \sum_{n=-\infty}^{\infty} h_n z^{-n} \right| \leq \sum_{n=-\infty}^{\infty} |h_n z^{-n}|$$

et sur le cercle de rayon unité,

$$|H(e^{2i\pi v})| = \left| \sum_{n=-\infty}^{\infty} |h_n| \right|$$

et si le dernier terme est borné, cela signifie bien que la transformée de Fourier à temps-discret existe. Donc, pour que le système soit stable, le cercle de rayon unité doit être inclus dans la région de convergence de la fonction de transfert qui est un anneau en toute généralité.

Pour qu'un système soit causal et stable, il faut que le cercle de rayon unité appartienne au domaine de convergence, qui pour réponse causale est la couronne $\{z \in \mathbb{C} : |z| \geq R\}$. Compte tenu de ce qui a été dit précédemment, les pôles du système doivent donc être contenus dans un disque $\{z \in \mathbb{C} : |z| < R\}$ avec $R < 1$.

6.8 Equations aux différences

6.8.1 fonction de transfert

Lorsque le système est régi par une équation aux différences du type

$$\sum_{l=0}^N a_l y_{n-l} = \sum_{m=0}^M b_m x_{n-m} \quad (6.17)$$

où $\{x_n, n \in \mathbb{Z}\}$ et $\{y_n, n \in \mathbb{Z}\}$ sont les suites d'entrée (excitation) et de sortie (réponse), on peut obtenir la réponse de régime en passant par la transformée en z bilatérale. En utilisant les propriétés de linéarité et décalage, on trouve finalement

$$Y(z) \sum_{l=0}^N a_l z^{-l} = X(z) \sum_{m=0}^M b_m z^{-m} \quad (6.18)$$

On en déduit que la fonction de transfert $G(z)$

$$G(z) = \frac{\sum_{m=0}^M b_m z^{-m}}{\sum_{l=0}^N a_l z^{-l}}$$

est une fraction rationnelle (en z^{-1}). Il est souvent pratique de factoriser les polynômes apparaissant au numérateur et au dénominateur, i.e. d'écrire

$$G(z) = \frac{b_0 \prod_{k=1}^M (1 - c_k z^{-1})}{a_0 \prod_{l=1}^N (1 - d_l z^{-1})}$$

Chaque facteur au numérateur ($1 - c_k z^{-1}$) sont associés à un zéro en $z = c_k$ et un pôle en $z = 0$. Chaque facteur au dénominateur est associé à un pôle en $z = d_k$ et un zéro en $z = 0$.

6.8.2 causalité et stabilité

Pour obtenir (6.18) à partir de (6.17), nous avons supposé que (6.17) représentait un système linéaire invariant dans le temps, mais nous n'avons pas fait d'hypothèses supplémentaires sur la stabilité ou la causalité du système. Pour spécifier réellement la relation d'entrée-sortie associée à (6.17), il est maintenant important de considérer précisément les domaines de convergence, car comme nous le verrons (6.17) ne spécifie pas une unique relation de filtrage mais un ensemble de relations de filtrages, associées à la fois aux différents domaines de convergence de la fonction de transfert $G(z)$ et du domaine de convergence de l'entrée $X(z)$. Pour une fraction rationnelle, chaque choix de la région de convergence sera associé à une réponse impulsionale différente, bien qu'étant associée à la *même* équation aux différences. Une certaine prudence s'impose donc ! Supposons que $M < N$ et que les pôles sont simples. Dans ce cas $G(z)$ peut se développer de la façon suivante

$$G(z) = \sum_{k=1}^N \frac{A_k}{1 - d_k z^{-1}},$$

où $A_k = (1 - d_k z^{-1})A(z)|_{z=d_k}$. Lorsque $M > N$, on doit tout d'abord procéder à la division euclidienne du numérateur par le dénominateur, puis on applique le résultat précédent au reste de cette division euclidienne

$$G(z) = \sum_{r=0}^{M-N} B_r z^{-r} + \sum_{k=1}^N \frac{A_k}{1 - d_k z^{-1}}.$$

Pour obtenir les solutions causales, nous développons

$$\frac{1}{1 - d_k z^{-1}} = \sum_{j=0}^{\infty} d_k^j z^{-j},$$

qui converge sur la couronne $\{z \in \mathbb{C} : |d_k| \leq |z|\}$. Le domaine de convergence de la solution causale est donc $R_- = \max(|d_k|, k \in \{1, \dots, N\})$ et la réponse associée est donc

$$G(z) = \sum_{r=0}^{M-N} B_r z^{-r} + \sum_{k=1}^N A_k \sum_{j=0}^{\infty} d_k^j z^{-j}.$$

La réponse impulsionale est stable si le cercle unité est élément de la région de convergence, ce qui implique que tous les pôles sont à l'intérieur du disque unité (dans ce cas, $R_- < 1$).

Si cette condition n'est pas satisfaite, il peut néanmoins exister une réponse stable. Supposons que $A(z) \neq 0$ pour $|z| = 1$ (le système n'a pas de pôles sur le cercle unité). Nous utilisons les développements suivants

$$\begin{cases} |d_k| < 1 & \frac{1}{1 - d_k z^{-1}} = \sum_{j=0}^{\infty} d_k^j z^{-j} \\ \text{DC } \{z \in \mathbb{C} : |d_k| \leq |z|\} & \\ |d_k| > 1 & \frac{1}{1 - d_k z^{-1}} = -\frac{d_k^{-1} z}{1 - d_k^{-1} z} = -\sum_{j=1}^{\infty} d_k^j z^{-j} \\ \text{DC } \{z \in \mathbb{C} : |z| < |d_k|\} & \end{cases} \quad (6.19)$$

Le domaine de convergence est alors une couronne $R_- = \max\{|d_k| < 1, k \in \{1, \dots, N\}\}$ et $R_+ = \min\{|d_k| > 1, k \in \{1, \dots, N\}\}$.

Exemple 69. Considérons la fonction de transfert

$$G(z) = \frac{1 - az^{-1}}{1 - bz^{-1}}.$$

Supposons que $|a| < 1$ et $|b| < 1$. La fonction de transfert a un zéro en $z = a$ et un pôle en $z = b$. Ce pôle est élément du disque unité ouvert et cette fonction de transfert admet donc un développement causal stable. Nous avons en effet

$$\frac{1 - az^{-1}}{1 - bz^{-1}} = c + \frac{1 - c}{1 - bz^{-1}}, \quad c = a/b.$$

Cette fraction rationnelle admet un développement causal stable sur la couronne $\{z \in \mathbb{C} : |b| \leq |z|\}$:

$$G(z) = c + (1 - c) \sum_{j=0}^{\infty} b^j z^{-j}.$$

On peut chercher à inverser ce système, c'est à dire trouver la fonction de transfert $G_1(z)$ qui soit telle que $G(z)G_1(z) = 1$. Nous avons alors

$$G_1(z) = \frac{1 - bz^{-1}}{1 - az^{-1}}$$

Cette fonction de transfert admet un pôle en a et comme $|a| < 1$, la fonction de transfert inverse admet aussi un développement causal stable.

Exemple 70 (filtre passe-tout). Considérons le filtre de fonction de transfert

$$G(z) = \frac{z^{-1} - \bar{a}}{1 - az^{-1}}, \quad |a| \neq 1$$

On remarque que le module de la réponse en fréquence de ce filtre est constant, i.e.

$$G(e^{+2i\pi\nu}) = \frac{e^{-2i\pi\nu} - \bar{a}}{1 - ae^{-2i\pi\nu}} = e^{-2i\pi\nu} \frac{1 - \bar{a}e^{+2i\pi\nu}}{1 - ae^{-2i\pi\nu}},$$

ce qui implique que $|G(e^{+2i\pi\nu})| = 1$ pour tout $\nu \in [-1/2, 1/2[$. Un tel système est appelé un "passe-tout". De façon générale, la fonction de transfert d'un filtre passe-tout est donnée par

$$G(z) = \prod_{k=1}^M \frac{z^{-1} - d_k}{1 - d_k z^{-1}} \prod_{k=1}^M \frac{(z^{-1} - \bar{e}_k)(z^{-1} - e_k)}{(1 - e_k z^{-1})(1 - \bar{e}_k z^{-1})},$$

où $d_k, k \in \{1, \dots, M\}$ sont les pôles réels et $e_k, k \in \{1, \dots, M\}$ sont les pôles complexes.

Exemple 71 (Filtre de Karplus-Strong).

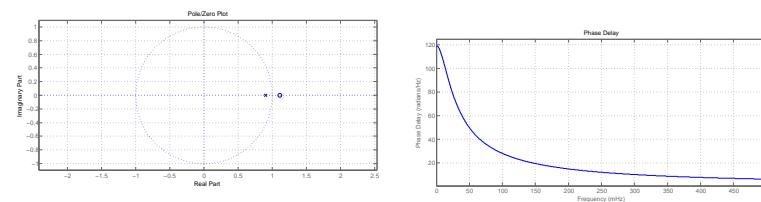


Figure 6.3: Pôles et zéros et phase d'un filtre passe tout du premier ordre avec $\alpha = 0.9$

Part III

Bases de traitement du signal aléatoire

Chapitre 7

Introduction au signal aléatoire à temps-discret

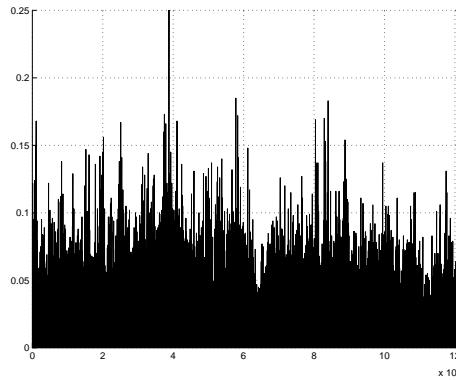


Figure 7.1: Trace de trafic Internet : temps d'inter-arrivées de paquets TCP.

Dans ce chapitre, nous introduisons des concepts de base concernant l’analyse des séries temporelles. En particulier, nous définissons les notions de stationnarité et de fonction d’autocovariance.

7.1 Introduction

Dans la suite, nous proposons de considérer les observations comme des réalisations d’un processus aléatoire $(X_t)_{t \in T}$ dont nous donnons la définition dans le paragraphe 7.2. Les quelques exemples qui suivent illustrent la diversité des situations dans lesquelles la modélisation stochastique (ou aléatoire) des séries temporelles joue un rôle important.

Exemple 72 (Trafic internet). *La figure 7.1 représente les temps d’inter-arrivées de paquets TCP, mesurés en secondes, sur la passerelle du laboratoire Lawrence Livermore. La trace représentée a été obtenue en enregistrant 2 heures de trafic. Pendant cette durée, environ 1.3 millions de paquets TCP, UDP, etc. ont été enregistrés. D’autres séries de ce type peuvent être obtenues sur The Internet Traffic Archive, <http://ita.ee.lbl.gov/>.*

Exemple 73 (Parole). *La figure 7.2 représente un segment de signal vocal échantillonné (la fréquence d’échantillonnage est de 8000 Hz). Ce segment de signal correspond à la réalisation du phonème ch (comme dans chat) qui est un son dit fricatif, c’est-à-dire produit par les turbulences du flux d’air au voisinage d’une constriction (ou resserrement) du conduit vocal.*

Exemple 74 (Indice financier). *La figure 7.3 représente les cours d’ouverture journaliers de l’indice Standard and Poor 500, du 2 Janvier 1990 au 25 Août 2000. L’indice S&P500 est calculé à partir de 500 actions choisies parmi les valeurs cotées au New York Stock Exchange (NYSE) et au NASDAQ en fonction de leur capitalisation, leur liquidité, leur représentativité dans différents secteurs d’activité. Cet indice est obtenu*

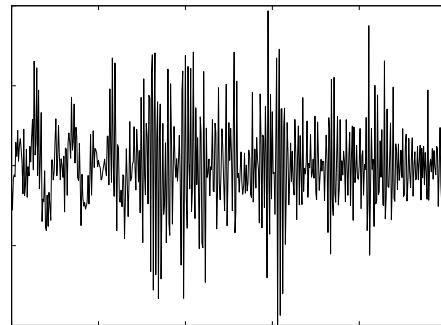


Figure 7.2: Signal de parole échantillonné à 8000 Hz : son non voisé *ch*.

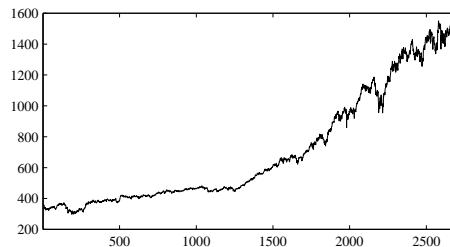


Figure 7.3: Cours quotidien d'ouverture de l'indice S&P500 : entre Janvier 1990 et Août 2000.

en pondérant le prix des actions par le nombre total d'actions, le poids de chaque valeur dans l'indice composite étant proportionnel à la capitalisation.

Exemple 75 (Battements cardiaques). *La figure 75 représente l'évolution, sur une durée totale de 900 secondes, du rythme cardiaque d'un sujet au repos. Ce rythme est mesuré en nombre de battements par minute avec un pas d'échantillonnage de 0.5 secondes.*

7.2 Définition et construction de la loi d'un processus aléatoire

7.2.1 Processus aléatoire

Définition 76 (Processus aléatoire). *Soient $(\Omega, \mathcal{F}, \mathbb{P})$ un espace de probabilité, T un ensemble d'indices et (E, \mathcal{E}) un espace mesurable. On appelle processus aléatoire une famille $(X_t)_{t \in T}$ de v.a. à valeurs dans (E, \mathcal{E}) indexées par $t \in T$.*

Le paramètre t représente par exemple le temps. Lorsque $T = \mathbb{Z}$ ou \mathbb{N} , nous dirons que le processus est à *temps discret* et, lorsque $T = \mathbb{R}$ ou \mathbb{R}_+ , que le processus est

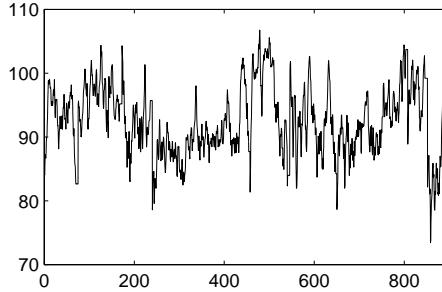


Figure 7.4: Nombre de pulsations par minutes en fonction du temps

à *temps continu*. Dans la suite, nous nous intéresserons sauf exception aux processus à temps discret avec $T = \mathbb{Z}$. Quant à (E, \mathcal{E}) , nous considérerons le plus souvent $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ (où $\mathcal{B}(\mathbb{R})$ est la tribu boréienne de \mathbb{R}) ou $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$. Dans le premier cas, on dira que le processus aléatoire est *scalaire*. Dans le second, nous dirons que le processus est *vectoriel*.

Notons qu'un processus peut être vu comme une application $X : \Omega \times T \rightarrow E$, $(\omega, t) \mapsto X_t(\omega)$ telle que, à chaque instant $t \in T$, l'application $\omega \mapsto X_t(\omega)$ est une variable aléatoire de (E, \mathcal{E}) .

Définition 7.7 (Trajectoire). *Pour chaque $\omega \in \Omega$, l'application $t \mapsto X_t(\omega)$ est une fonction de $T \rightarrow E$ qui s'appelle la trajectoire associée à l'épreuve ω .*

7.2.2 Répartitions finies

Etant donnés 2 espaces mesurables (E_1, \mathcal{E}_1) et (E_2, \mathcal{E}_2) , on définit l'espace mesurable produit $(E_1 \times E_2, \mathcal{E}_1 \otimes \mathcal{E}_2)$ où \times désigne le produit cartésien usuel des ensembles et \otimes l'opération correspondante sur les tribus: $\mathcal{E}_1 \otimes \mathcal{E}_2$ désigne la tribu engendrée par $\{A_1 \times A_2 : A_1 \in \mathcal{E}_1, A_2 \in \mathcal{E}_2\}$, ce que l'on écrira

$$\mathcal{E}_1 \otimes \mathcal{E}_2 = \sigma\{A_1 \times A_2 : A_1 \in \mathcal{E}_1, A_2 \in \mathcal{E}_2\}.$$

Comme la classe d'ensembles $\{A_1 \times A_2 : A_1 \in \mathcal{E}_1, A_2 \in \mathcal{E}_2\}$ est stable par intersection finie, une probabilité sur $\mathcal{E}_1 \otimes \mathcal{E}_2$ est caractérisée par sa restriction à cette classe (voir [?, Corollaire 6.1]).

On définit de même un espace mesurable produit $(E_1 \times \cdots \times E_n, \mathcal{E}_1 \otimes \cdots \otimes \mathcal{E}_n)$ à partir d'un nombre fini n d'espaces mesurables (E_t, \mathcal{E}_t) , $t \in T$. Si T n'est pas de cardinal fini, cette définition se généralise en considérant la tribu engendrée par les *cylindres* sur le produit cartésien $\prod_{t \in T} E_t$ qui contient l'ensemble des familles $(x_t)_{t \in T}$ telles que $x_t \in E_t$ pour tout $t \in T$. Examinons le cas qui nous servira par la suite où $(E_t, \mathcal{E}_t) = (E, \mathcal{E})$ pour tout $t \in T$. On note alors $E^T = \prod_{t \in T} E$ l'ensemble des trajectoires $(x_t)_{t \in T}$ telles que $x_t \in E$ pour tout t , que l'on munit de la tribu engendrée par les cylindres

$$\mathcal{E}^{\otimes T} = \sigma \left\{ \prod_{t \in I} A_t \times E^{T \setminus I} : I \in \mathcal{I}, \forall t \in I, A_t \in \mathcal{E} \right\},$$

où l'on note \mathcal{J} l'ensemble des parties finies de T .

Soit $X = (X_t)_{t \in T}$ un processus défini sur $(\Omega, \mathcal{F}, \mathbb{P})$ à valeurs dans (E, \mathcal{E}) et $I \in \mathcal{J}$. On note \mathbb{P}_I la loi du vecteur aléatoire $\{X_t, t \in I\}$, c'est-à-dire la mesure image de \mathbb{P} par ce vecteur : \mathbb{P}_I est la probabilité sur $(E^I, \mathcal{E}^{\otimes I})$ définie par

$$\mathbb{P}_I \left(\prod_{t \in I} A_t \right) = \mathbb{P}(X_t \in A_t, t \in I), \quad (7.1)$$

où $A_t, t \in T$ sont des éléments quelconques de la tribu \mathcal{E} . La probabilité \mathbb{P}_I est une *probabilité fini-dimensionnelle ou répartition finie* du processus X .

Définition 78. On appelle famille des répartitions finies *l'ensemble des répartitions finies* $(\mathbb{P}_I, I \in \mathcal{J})$.

La spécification de la mesure \mathbb{P}_I permet de calculer la probabilité d'événements de la forme $\mathbb{P}(\cap_{t \in I} \{X_t \in A_t\})$ où $\{A_t, t \in I\}$ est une famille d'éléments de la tribu \mathcal{E} , ou de manière équivalente, de calculer l'espérance $\mathbb{E}[\prod_{t \in I} f_t(X_t)]$ où pour tout $t \in I$, f_t est une fonction borélienne positive. Soit $J \subset I$ deux parties finies ordonnées. Soit $\Pi_{I,J}$ la projection canonique de E^I sur E^J définie par

$$\Pi_{I,J}[x] = (x_t)_{t \in J} \quad \text{pour tout } x = (x_t)_{t \in I} \in E^I. \quad (7.2)$$

La projection canonique préserve uniquement les coordonnées du vecteur appartenant au sous ensemble d'indices J . Par la définition (7.1), on observe que \mathbb{P}_J est la mesure image de $\Pi_{I,J}$ définie sur l'espace de probabilité $(E^J, \mathcal{E}^{\otimes J}, \mathbb{P}_I)$:

$$\mathbb{P}_I \circ \Pi_{I,J}^{-1} = \mathbb{P}_J. \quad (7.3)$$

Cette relation formalise le résultat intuitif que la distribution fini-dimensionnelle d'un sous-ensemble $J \subset I$ se déduit de la distribution fini-dimensionnelle P_I en "intégrant" par rapport aux variables X_t sur l'ensemble des t appartenant au complémentaire de J dans I . Cette propriété montre que la famille des répartitions finies d'un processus est fortement structurée. En particulier, les répartitions finies doivent, au moins, vérifier les conditions de compatibilité (7.3). Nous allons voir dans la suite que cette condition est en fait aussi *suffisante*.

Soit Π_I la projection canonique de E^T sur E^I ,

$$\Pi_I(x) = (x_t)_{t \in I} \quad \text{pour tout } x = (x_t)_{t \in T} \in E^T. \quad (7.4)$$

Si $I = \{s\}$ avec $s \in T$, on notera simplement

$$\Pi_s(x) = \Pi_{\{s\}}(x) = x_s \quad \text{pour tout } x = (x_t)_{t \in T} \in E^T. \quad (7.5)$$

Le théorème suivant montre comment on peut passer d'une famille de répartitions finies à une unique mesure de probabilité sur $(E^T, \mathcal{E}^{\otimes T})$, pourvu que la condition de compatibilité (7.3) soit satisfaite.

Théorème 79 (théorème de Kolmogorov). Soit $(v_I)_{I \in \mathcal{I}}$ une famille de probabilités indexées par l'ensemble des parties finies ordonnées de T telle, que pour tout $I \in \mathcal{I}$, v_I est une probabilité sur $(E^I, \mathcal{E}^{\otimes I})$. Supposons de plus que la famille $\{v_I, I \in \mathcal{I}\}$ vérifie les conditions de compatibilité (7.3): pour tout $I, J \in \mathcal{I}$, tel que $I \subset J$, $v_I \circ \Pi_{I,J}^{-1} = v_J$. Alors, il existe une unique probabilité \mathbb{P} sur l'espace mesurable $(E^T, \mathcal{E}^{\otimes T})$ telle que, pour tout $I \in \mathcal{I}$, $v_I = \mathbb{P} \circ \Pi_I^{-1}$.

Proof. Remarquons que la classe des cylindres est une semi-algèbre au sens de [?, p. 297]. On définit \mathbb{P} sur cette classe par

$$\mathbb{P} \left(\prod_{t \in I} A_t \times E^{T \setminus I} \right) = v_I \left(\prod_{t \in I} A_t \right),$$

où I décrit \mathcal{I} et $A_t \in \mathcal{E}$ pour tout $t \in I$. La condition de compatibilité implique que \mathbb{P} vérifie les hypothèses de [?, Proposition 9]. Il s'en suit une extension unique à l'algèbre engendrée par les cylindres, c'est-à-dire à la plus petite classe d'ensembles de E^T stable par intersection finie et par passage au complémentaire contenant les cylindres de E^T . Par le théorème de Carathéodory, voir [?, Théorème 8], on obtient une unique extension de \mathbb{P} à la tribu $\mathcal{E}^{\otimes T}$. \square

Ceci nous permet de décrire les répartitions finies d'un processus donné à partir d'une seule probabilité sur $(E^T, \mathcal{E}^{\otimes T})$, la *loi* (ou *mesure image*) du processus, définie comme suit.

Définition 80 (Loi d'un processus). Soit $X = (X_t)_{t \in T}$ un processus défini sur $(\Omega, \mathcal{F}, \mathbb{P})$ à valeurs dans (E, \mathcal{E}) . La mesure image \mathbb{P}_X est l'unique probabilité définie sur $(E^T, \mathcal{E}^{\otimes T})$ par $\mathbb{P}_X \circ \Pi_I^{-1} = \mathbb{P}_I$ pour tout $I \in \mathcal{I}$, i.e.

$$\mathbb{P}_X \left(\prod_{t \in I} A_t \times E^{T \setminus I} \right) = \mathbb{P}(X_t \in A_t, t \in I)$$

pour tout $(A_t)_{t \in I} \in \mathcal{E}^I$.

L'existence et l'unicité de \mathbb{P}_X est une conséquence du théorème 79. Cette loi est donc entièrement déterminée par la donnée des répartitions finies.

La définition suivante permet de voir \mathbb{P}_X comme la probabilité d'une variable aléatoire à valeurs dans $(E^T, \mathcal{E}^{\otimes T})$. Cette variable aléatoire est obtenue comme la trajectoire du *processus canonique* défini comme suit.

Définition 81 (Processus canonique). Soit (E, \mathcal{E}) un espace mesurable et (E^T, \mathcal{E}^T) l'espace mesurable des trajectoires correspondants. La famille canonique sur (E^T, \mathcal{E}^T) est la famille des fonctions mesurables $(\xi_t)_{t \in T}$ définies sur (E^T, \mathcal{E}^T) à valeurs dans (E, \mathcal{E}) par $\xi_t(\omega) = \omega_t$ pour tout $\omega = (\omega_t)_{t \in T} \in E^T$.

Quand on munit (E^T, \mathcal{E}^T) de la mesure image \mathbb{P}_X , on appelle la famille canonique $(\xi_t)_{t \in T}$ définies sur $(E^T, \mathcal{E}^T, \mathbb{P}_X)$ le processus canonique associé à X .

On a supposé jusqu'à présent le processus $X = (X_t)_{t \in T}$ donné. Le théorème 79 peut aussi être utilisé pour le construire, sous la forme d'un processus canonique, comme le montre l'exemple suivant, puis le paragraphe 7.2.3 qui introduit une classe particulière de processus : la classe des processus gaussiens.

Exemple 82 (Suite de v.a. indépendantes). Soit $(v_t)_{t \in T}$ une suite de probabilités sur (E, \mathcal{E}) . Pour $I \in \mathcal{J}$, on pose

$$v_I = \bigotimes_{t \in I} v_t , \quad (7.6)$$

où \otimes désigne le produit tensoriel sur les probabilités (loi du vecteur à composantes indépendantes et de lois marginales données par les v_t , $t \in I$). Il est clair que l'on définit ainsi une famille $(v_I)_{I \in \mathcal{J}}$ compatible, c'est-à-dire, vérifiant la condition donnée par l'équation (7.3). Donc, si $\Omega = E^T$, $X_t(\omega) = \omega_t$ et $\mathcal{F} = \sigma(X_t, t \in T)$, il existe une unique probabilité \mathbb{P} sur (Ω, \mathcal{F}) telle que $(X_t)_{t \in T}$ soit une suite de v.a. indépendantes telles que $X_t \sim v_t$ pour tout $t \in T$.

7.2.3 Processus gaussiens réels

Nous introduisons à présent une classe importante de processus aléatoires en modélisation stochastique : la classe des processus gaussiens. Rappelons tout d'abord la définition des variables aléatoires gaussiennes, univariées puis multivariées.

Définition 83 (Variable aléatoire gaussienne réelle). On dit que X est une variable aléatoire réelle gaussienne si sa loi de probabilité a pour fonction caractéristique :

$$\phi_X(u) = \mathbb{E} [e^{iuX}] = \exp(i\mu u - \sigma^2 u^2 / 2)$$

où $\mu \in \mathbb{R}$ et $\sigma \in \mathbb{R}^+$.

On en déduit que $\mathbb{E}[X] = \mu$ et que $\text{Var}(X) = \sigma^2$. Si $\sigma \neq 0$, la loi possède une densité de probabilité qui a pour expression :

$$p_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) . \quad (7.7)$$

Si $\sigma = 0$, on a alors $X = \mu$ p.s. La définition suivante étend cette définition aux vecteurs aléatoires de dimension n .

Définition 84 (Vecteur gaussien réel). Un vecteur aléatoire réel de dimension n $[X_1, \dots, X_n]^T$ est un vecteur gaussien si toute combinaison linéaire de X_1, \dots, X_n est une variable aléatoire gaussienne réelle.

Notons μ le vecteur moyenne de $[X_1, \dots, X_n]^T$ et Γ sa matrice de covariance. Par définition d'un vecteur aléatoire gaussien, pour tout $u \in \mathbb{R}^n$, la variable aléatoire $Y = \sum_{k=1}^n u_k X_k = u^T X$ est une variable aléatoire réelle gaussienne. Par conséquent, sa loi est complètement déterminée par sa moyenne et sa variance qui ont pour expressions respectives :

$$\mathbb{E}[Y] = \sum_{k=1}^n u_k \mathbb{E}[X_k] = u^T \mu \quad \text{et} \quad \text{Var}(Y) = \sum_{j,k=1}^n u_j u_k \text{cov}(X_j, X_k) = u^T \Gamma u$$

On en déduit l'expression, en fonction de μ et de Γ , de la fonction caractéristique de la loi de probabilité d'un vecteur gaussien $[X(1), \dots, X(n)]^T$:

$$\phi_X(u) = \mathbb{E} [\exp(iu^T X)] = \mathbb{E} [\exp(iY)] = \exp\left(iu^T \mu - \frac{1}{2} u^T \Gamma u\right) \quad (7.8)$$

Réiproquement, si un vecteur aléatoire X de taille n a une fonction caractéristique de cette forme, on obtient immédiatement que X est un vecteur gaussien en calculant la fonction caractéristique de ses produits scalaires. Cette propriété permet d'obtenir la proposition suivante.

Proposition 85. *La loi d'un vecteur gaussien X de taille n est entièrement caractérisée par son vecteur moyenne μ et sa matrice d'autocovariance Γ . On notera*

$$X \sim \mathcal{N}_n(\mu, \Gamma) .$$

Réiproquement pour tout vecteur $\mu \in \mathbb{R}^n$ et toute matrice symétrique positive Γ , il existe un vecteur aléatoire X tel que $X \sim \mathcal{N}_n(\mu, \Gamma)$.

Proof. La première partie de l'énoncé découle directement de (7.8). Démontrons maintenant la réciproque. Tout d'abord le résultat est vrai pour $n = 1$ comme nous l'avons rappelé plus haut. On passe aisément au cas où Γ est diagonale. En effet, notons σ_i^2 , $i = 1, \dots, n$ ses éléments diagonaux et $\mu = [\mu_1, \dots, \mu_n]^T$. Alors il suffit de prendre X_1, \dots, X_n indépendants tels que $X_i \sim \mathcal{N}_1(\mu_i, \sigma_i^2)$ pour $i = 1, \dots, n$. On vérifie aisément que $X \sim \mathcal{N}_n(\mu, \Gamma)$ en calculant sa fonction caractéristique. Pour passer du cas des matrices diagonales à une matrice Γ symétrique positive quelconque, on utilise le lemme suivant dont la preuve est laissée à titre d'exercice.

Lemme 86. *Soit $X \sim \mathcal{N}_n(\mu, \Gamma)$ avec $\mu \in \mathbb{R}^n$ et Γ matrice symétrique positive $n \times n$. Alors pour toute matrice A de taille $p \times n$, on a $AX \sim \mathcal{N}_p(A\mu, A\Gamma A^T)$.*

Pour conclure la preuve de la proposition 85, il suffit de remarquer que toute matrice symétrique positive Γ est diagonalisable en base orthonormée et s'écrit donc $\Gamma = U\Sigma U^T$ avec Σ matrice diagonale positive et U matrice orthogonale. Il suffit alors de prendre $Y \sim \mathcal{N}_n(U^T\mu, \Sigma)$ et de poser $X = UY$ et le lemme donne $X \sim \mathcal{N}_n(\mu, \Gamma)$ comme recherché. \square

On montre facilement la proposition suivante:

Proposition 87. *Soit $X \sim \mathcal{N}_n(\mu, \Gamma)$ avec $\mu \in \mathbb{R}^n$ et Γ matrice symétrique positive $n \times n$. Alors X a des composantes indépendantes si et seulement si Γ est une matrice diagonale.*

En utilisant le même procédé de preuve que pour la proposition 85, i.e. en considérant le cas Γ diagonale puis la diagonalisation de Γ pour passer au cas général, on obtient aussi le résultat suivant:

Proposition 88. *Soit $X \sim \mathcal{N}_n(\mu, \Gamma)$ avec $\mu \in \mathbb{R}^n$ et Γ matrice symétrique positive $n \times n$. Si Γ est de rang plein, alors la loi de probabilité de X possède une densité dans \mathbb{R}^n dont l'expression est :*

$$p_X(x) = \frac{1}{(2\pi)^{n/2} \sqrt{\det(\Gamma)}} \exp\left(-\frac{1}{2}(x - \mu)^T \Gamma^{-1}(x - \mu)\right), \quad x \in \mathbb{R}^n .$$

Dans le cas où Γ est de rang $r < n$, c'est à dire où Γ possède $n - r$ valeurs propres nulles, X se trouve, avec probabilité 1, dans un sous espace affine de dimension r de \mathbb{R}^n . En effet, il existe alors $r - n$ vecteurs a_i formant une famille libre tels que $\text{cov}(a_i^T X) = 0$ et donc $a_i^T X = a_i^T \mu$ p.s. X n'admet donc évidemment pas de densité dans ce cas.

Nous étendons maintenant la notion de vecteur gaussien à celle de *processus gaussien*.

Définition 89 (Processus gaussien réel). *On dit qu'un processus réel $X = (X_t)_{t \in T}$ est gaussien si, pour tout ensemble fini d'indices $I = \{t_1, t_2, \dots, t_n\}$, $[X_{t_1}, X_{t_2}, \dots, X_{t_n}]^T$ est un vecteur gaussien.*

Ainsi un vecteur gaussien $[X_1, \dots, X_n]^T$ peut être lui-même vu comme un processus gaussien $\{X_t, t \in \{1, \dots, n\}\}$. Cette définition n'a donc un intérêt que dans le cas où T est de cardinal infini. D'après (7.8), la famille des répartitions finies est caractérisée par la donnée de la fonction moyenne $\mu : t \in T \mapsto \mu(t) \in \mathbb{R}$ et de la fonction de covariance $\gamma : (t, s) \in (T \times T) \mapsto \gamma(t, s) \in \mathbb{R}$. De plus, pour tout ensemble fini d'indices $I = \{t_1, t_2, \dots, t_n\}$, la matrice Γ_I d'éléments $\Gamma_I(m, k) = \gamma(t_m, t_k)$, où $1 \leq m, k \leq n$, est une matrice de covariance d'un vecteur aléatoire de dimension n . Elle est donc symétrique positive. Réciproquement, donnons nous une fonction $\mu : t \in T \mapsto \mu(t) \in \mathbb{R}$ et une fonction $\gamma : (t, s) \in (T \times T) \mapsto \gamma(t, s) \in \mathbb{R}$ telle que, pour tout ensemble fini d'indices I , la matrice Γ_I est symétrique positive. On peut alors définir, pour tout ensemble fini d'indices $I = \{t_1, t_2, \dots, t_n\}$, une probabilité gaussienne v_I sur \mathbb{R}^n par :

$$v_I \stackrel{\text{def}}{=} \mathcal{N}_n(\mu_I, \Gamma_I) \quad (7.9)$$

où $\mu_I = [\mu(t_1), \dots, \mu(t_n)]^T$. La famille $(v_I, I \in \mathcal{I})$, ainsi définie, vérifie les conditions de compatibilité et l'on a ainsi établi, d'après le théorème 79, le résultat suivant :

Théorème 90. *Soit T un ensemble d'indices quelconque, μ une fonction réelle définie sur T et γ une fonction réelle définie sur $T \times T$ dont toutes les restrictions Γ_I aux ensembles $I \times I$ avec $I \subseteq T$ fini forment des matrices symétriques positives. Il existe un espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$ et un processus aléatoire $\{X_t, t \in T\}$ gaussien défini sur cet espace vérifiant*

$$\mu(t) = \mathbb{E}[X_t] \quad \text{et} \quad \gamma(s, t) = \mathbb{E}[(X_s - \mu(s))(X_t - \mu(t))].$$

7.3 Stationnarité stricte d'un processus à temps discret

7.3.1 Définition

La notion de stationnarité joue un rôle central dans la théorie des processus aléatoires. On distingue ci-dessous deux versions de cette propriété, la *stationnarité stricte* qui fait référence à l'invariance des répartitions finies par translation de l'origine des temps, et une notion plus faible, la *stationnarité au second ordre*, qui impose l'invariance par translation des moments d'ordre un et deux uniquement, lorsque ceux-ci existent.

Définition 91 (Opérateurs de retard). *On note S et l'on appelle opérateur de décalage (Shift) l'application $E^{\mathbb{Z}} \rightarrow E^{\mathbb{Z}}$ définie par*

$$S(x) = (x_{t-1})_{t \in \mathbb{Z}} \quad \text{pour tout } x = (x_t)_{t \in \mathbb{Z}} \in E^{\mathbb{Z}}.$$

Pour tout $\tau \in \mathbb{Z}$, on définit S^τ par

$$S^\tau(x) = (x_{t-\tau})_{t \in \mathbb{Z}} \quad \text{pour tout } x = (x_t)_{t \in \mathbb{Z}} \in E^{\mathbb{Z}}.$$

Définition 92 (Stationnarité stricte). *Un processus aléatoire $X = \{X_t, t \in \mathbb{Z}\}$ est stationnaire au sens strict si X et $S \circ X$ ont même loi, i.e. $\mathbb{P}_{S \circ X} = \mathbb{P}_X$.*

Par caractérisation de la loi image par les répartitions finies, on a $\mathbb{P}_{S \circ X} = \mathbb{P}_X$ si et seulement si

$$\mathbb{P}_{S \circ X} \circ \Pi_I^{-1} = \mathbb{P}_X \circ \Pi_I^{-1}$$

pour toute partie finie $I \in \mathcal{I}$. Or $\mathbb{P}_{S \circ X} \circ \Pi_I^{-1} = \mathbb{P}_X \circ (\Pi_I \circ S)^{-1}$ et $\Pi_I \circ S = \Pi_{I-1}$, où $I-1 = \{t-1, t \in I\}$. On en conclut que $\{X_t, t \in \mathbb{Z}\}$ est stationnaire au sens strict si et seulement si, pour toute partie finie $I \in \mathcal{I}$,

$$\mathbb{P}_I = \mathbb{P}_{I-1}.$$

On remarque aussi que la stationnarité au sens strict implique que X et $S^\tau \circ X$ ont même loi pour tout $\tau \in \mathbb{Z}$ et donc aussi $\mathbb{P}_I = \mathbb{P}_{I+\tau}$, où $I+\tau = \{t+\tau, t \in I\}$.

Exemple 93 (Processus i.i.d.). Soit $(Z_t)_{t \in \mathbb{Z}}$ une suite de variables aléatoires indépendantes et identiquement distribuées (i.i.d.) à valeurs dans \mathbb{R}^d . Alors $(Z_t)_{t \in \mathbb{Z}}$ est un processus stationnaire au sens strict, car, pour toute partie finie ordonnée $I = \{t_1, < t_2 < \dots < t_n\}$ et tous boréliens A_1, \dots, A_n de \mathbb{R}^d , nous avons :

$$\mathbb{P}(Z_{t_1} \in A_1, \dots, Z_{t_n} \in A_n) = \prod_{j=1}^n \mathbb{P}(Z_0 \in A_j),$$

qui ne dépend pas de t_1, \dots, t_n . Notons que d'après Exemple 82, pour toute probabilité v sur \mathbb{R}^d , on sait construire un processus (Z_t) i.i.d. de loi marginale v , c'est-à-dire tel que $Z_t \sim v$ pour tout $t \in \mathbb{Z}$.

Exemple 94 (Transformation d'un processus i.i.d.). Soit Z un processus i.i.d. (voir Exemple 93). Soient k un entier et g une fonction borélienne de \mathbb{R}^k dans \mathbb{R} . On peut vérifier que le processus aléatoire $(X_t)_{t \in \mathbb{Z}}$ défini par

$$X_t = g(Z(t), Z(t-1), \dots, Z(t-k+1))$$

est encore un processus aléatoire stationnaire au sens strict. Par contre, ce processus obtenu par transformation n'est plus i.i.d dans la mesure où, dès que $k \geq 1$, $X_t, X_{t+1}, \dots, X_{t+k-1}$ ont bien la même distribution marginale mais sont, en général, dépendants car fonctions de variables aléatoires communes. Un tel processus est dit k -dépendant car pour $\tau \geq k$, $(X_s)_{s \leq t}$ et $(X_s)_{s \geq t+\tau}$ sont indépendants pour tout t . Les processus m -dépendants peuvent être utilisés pour approcher une grande classe de processus dépendants afin d'étudier le comportement asymptotique de statistiques usuelles telles que la moyenne empirique.

7.3.2 Transformations préservant la stationnarité

On pose $E = \mathbb{C}^d$ et $\mathcal{E} = \mathcal{B}(\mathbb{C}^d)$ pour un entier $d \geq 1$.

Définition 95 (Filtrage). Soit ϕ une application mesurable de $(E^\mathbb{Z}, \mathcal{E}^{\otimes\mathbb{Z}})$ dans $(F^\mathbb{Z}, \mathcal{F}^{\otimes\mathbb{Z}})$ et $X = (X_t)_{t \in \mathbb{Z}}$ un processus à valeurs dans (E, \mathcal{E}) . On appelle filtré du processus X par la transformation ϕ le processus $Y = (Y_t)_{t \in \mathbb{Z}}$ à valeurs dans (F, \mathcal{F}) défini par $Y = \phi \circ X$, c'est-à-dire $Y_t = \Pi_t(\phi(X))$ pour tout $t \in \mathbb{Z}$, où Π_t est défini par (7.5). Si ϕ est une application linéaire, on parlera de filtrage linéaire.

L'exemple 94 est un exemple de filtrage (en général non-linéaire, à moins que g soit une forme linéaire). La transformation associée à cet exemple est l'application $\phi : \mathbb{R}^\mathbb{Z} \rightarrow \mathbb{R}^\mathbb{Z}$ définie par

$$\phi((x_t)_{t \in \mathbb{Z}}) = (g(x_t, x_{t-1}, \dots, x_{t-k+1}))_{t \in \mathbb{Z}}.$$

Exemple 96 (Décalage). Un exemple fondamental de filtrage linéaire de processus est obtenu en prenant $\phi = S$ où S est l'opérateur de décalage de la définition 91. Dans ce cas $Y_t = X_{t+1}$ pour tout $t \in \mathbb{Z}$.

Exemple 97 (Filtre à réponse impulsionnelle finie (RIF)). Soient $n \geq 1$ et $t_1 < \dots < t_n$ des éléments de \mathbb{Z} et $\alpha_1, \dots, \alpha_n \in E$. Alors $\sum_i \alpha_i S^{-t_i}$ définit un filtrage linéaire pour n'importe quel processus $X = (X_t)_{t \in \mathbb{Z}}$ pour lequel la sortie est donnée par

$$Y_t = \sum_{i=1}^n \alpha_i X_{t-t_i}, \quad t \in \mathbb{Z}.$$

Exemple 98 (Différentiation). Un cas particulier de l'exemple précédent est donné par l'opérateur de différentiation $I - S^{-1}$ où I dénote l'opérateur identité. Le processus obtenu en sortie s'écrit

$$Y_t = X_t - X_{t-1}, \quad t \in \mathbb{Z}.$$

On pourra itérer l'opérateur de différentiation, ainsi $Y = (I - S^{-1})^k X$ est donnée par

$$Y_t = \sum_{j=0}^k \binom{k}{j} (-1)^j X_{t-j}, \quad t \in \mathbb{Z}.$$

Exemple 99 (Retournement du temps). Etant donné un processus $X = \{X_t, t \in \mathbb{Z}\}$, on appellera processus retourné le processus obtenu par retournement du temps défini par

$$Y_t = X_{-t}, \quad t \in \mathbb{Z}.$$

Exemple 100 (Intégration). Etant donné un processus $X = (X_t)_{t \in \mathbb{Z}}$ qui vérifie $\sum_{t=-\infty}^0 |X_t| < \infty$ p.s., on appellera processus intégré le processus défini par

$$Y_t = \sum_{s=0}^{\infty} X_{t-s}, \quad t \in \mathbb{Z}.$$

Contrairement aux exemples précédents, l'application ϕ qui définit ce filtrage doit être définie avec quelques précautions. Il faut en effet tout d'abord définir ϕ sur

$$A = \left\{ x = (x_t)_{t \in \mathbb{Z}} \in E^{\mathbb{Z}} : \sum_{t=-\infty}^0 |x_t| < \infty \right\},$$

par $\phi(x) = \sum_{s=0}^{\infty} x_{t-s}$. Comme A est un espace vectoriel, on peut prolonger ϕ linéairement sur $(E^{\mathbb{Z}}, \mathcal{E}^{\otimes \mathbb{Z}})$. Le point important est que ce filtrage ne sera appliqué à X que sous l'hypothèse $\sum_{t=-\infty}^0 |X_t| < \infty$ p.s. et que ce prolongement est donc défini de façon quelconque.

On remarque que dans tous les exemples précédents les opérateurs introduits préservent la stationnarité stricte, c'est-à-dire, si X est strictement stationnaire alors Y l'est aussi. Il est facile de construire des filtrages linéaires qui ne préserve pas la stationnarité stricte, par exemple, $y = \phi(x)$ avec $y_t = x_t$ pour t pair et $y_t = x_t + 1$ pour t impaire. Une propriété plus forte que la conservation de la stationnarité est donnée par la définition suivante.

Définition 101. Un filtrage linéaire est invariant par translation s'il commute avec S : $\phi \circ S = S \circ \phi$.

Cette propriété implique la préservation de la stationnarité mais ne lui est pas équivalente. Le retournement du temps est en effet un exemple de filtrage qui ne commute pas avec S puisque dans ce cas on a $\phi \circ S = S^{-1} \circ \phi$. En revanche tous les autres exemples ci-dessus satisfont la propriété d'invariance par translation.

Remarque 102. Un filtrage ϕ invariant par translation est entièrement déterminé par sa composition avec sa composition avec la projection canonique Π_0 , voir (7.5). En effet, notons $\phi_0 = \Pi_0 \circ \phi$. Alors pour tout $s \in \mathbb{Z}$, $\Pi_s \circ \phi = \Pi_0 \circ S^s \circ \phi = \Pi_0 \circ \phi \circ S^s$. Il suffit enfin d'observer que pour tout $x \in E^{\mathbb{Z}}$, $\phi(x)$ est la suite $(\pi_s \circ \phi)_{s \in \mathbb{Z}}$.

7.4 Processus du second ordre

Définition 103 (Processus du second ordre). *Le processus $X = (X_t)_{t \in T}$ à valeurs dans \mathbb{C}^d est dit du second ordre, si $\mathbb{E}[|X_t|^2] < \infty$ pour tout $t \in T$, où $|x|$ est la norme hermitienne de $x \in \mathbb{C}^d$.*

Notons que la *fonction moyenne* définie sur T par $\mu(t) = \mathbb{E}[X_t]$ est à valeurs dans \mathbb{C}^d et que la *fonction d'autocovariance* définie sur $T \times T$ par

$$\Gamma(s, t) = \text{cov}(X_s, X_t) = \mathbb{E}[(X_s - \mu(s))(X_t - \mu(t))^H].$$

Elle prend ses valeurs dans l'espace des matrices de dimension $d \times d$. Pour tout $s \in T$, $\Gamma(s, s)$ est une matrice d'autocovariance. C'est donc une matrice hermitienne positive. Plus généralement, toute fonction d'autocovariance vérifie les propriétés suivantes.

Proposition 104. *Soit Γ la fonction d'autocovariance d'un processus du second ordre indexé par T à valeurs dans \mathbb{C}^d . Elle vérifie alors les propriétés suivantes.*

1. *Symétrie hermitienne: pour tout $s, t \in T$,*

$$\Gamma(s, t) = \Gamma(t, s)^H \quad (7.10)$$

2. *Type positif: pour tout $n \geq 1$, pour tout $t_1, \dots, t_n \in T$ et pour tout $a_1, \dots, a_n \in \mathbb{C}^d$,*

$$\sum_{1 \leq k, m \leq n} a_k^H \Gamma(t_k, t_m) a_m \geq 0 \quad (7.11)$$

Proof. La propriété (7.10) est immédiate par définition de la covariance. Pour montrer (7.11), formons la combinaison linéaire $Y = \sum_{k=1}^n a_k^H X_{t_k}$. Y est une variable aléatoire complexe. En utilisant les propriétés de forme hermitienne de la covariance, on obtient

$$\text{Var}(Y) = \sum_{1 \leq k, m \leq n} a_k^H \Gamma(t_k, t_m) a_m$$

ce qui établit (7.11).

□

Dans le cas scalaire ($d = 1$), on note en général $\gamma(s, t)$ la covariance, en réservant la notation $\Gamma(s, t)$ au cas des processus vectoriels ($d > 1$).

7.5 Covariance d'un processus stationnaire au second ordre

Dorénavant, dans ce chapitre, on prend $T = \mathbb{Z}$. On définit la stationnarité au second ordre en ne retenant que les propriétés du second ordre (moyenne et covariance) d'un processus stationnaire au sens strict indexé par \mathbb{Z} . En effet, soit $X = (X_t)_{t \in \mathbb{Z}}$ un processus stationnaire au sens strict à valeurs dans \mathbb{C}^d . Supposons de plus qu'il est du second ordre. Alors sa fonction moyenne est constante puisque la loi marginale l'est,

et sa fonction d'autocovariance Γ vérifie $\Gamma(s, t) = \Gamma(s - t, 0)$ pour tout $s, t \in \mathbb{Z}$ puisque les lois bi-dimensionnelles sont invariantes par translation. Cela donne la définition suivante.

Définition 105 (Stationnarité au second ordre). *Soit $\mu \in \mathbb{C}^d$ et $\Gamma : \mathbb{Z} \rightarrow \mathbb{C}^{d \times d}$. Un processus $(X_t)_{t \in \mathbb{Z}}$ à valeurs dans \mathbb{C}^d est dit stationnaire au second ordre (ou faiblement stationnaire) de moyenne μ et de fonction d'auto-covariance Γ si :*

- (a) *X est un processus du second ordre, i.e. $\mathbb{E}[|X_t|^2] < +\infty$,*
- (b) *pour tout $t \in \mathbb{Z}$, $\mathbb{E}[X_t] = \mu$,*
- (c) *pour tout couple $(s, t) \in \mathbb{Z} \times \mathbb{Z}$, $\text{cov}(X_s, X_t) = \Gamma(s - t)$.*

Par convention la fonction d'autocovariance d'un processus stationnaire au second ordre indexé par T est définie sur T au lieu de $T \times T$ pour le cas général.

Comme expliqué en préambule de la définition, un processus du second ordre stationnaire au sens strict est stationnaire au second ordre. L'implication inverse est vraie pour la classe des processus gaussiens définies au paragraphe 7.2.3 d'après la proposition 85.

On remarque qu'un processus $(X_t)_{t \in \mathbb{Z}}$ à valeurs dans \mathbb{C}^d est stationnaire au second ordre de moyenne μ et de fonction d'auto-covariance Γ si et seulement si pour tout $\lambda \in \mathbb{C}^d$, le processus $(\lambda^H X_t)_{t \in \mathbb{Z}}$ à valeurs dans \mathbb{C} est stationnaire au second ordre de moyenne $\lambda^H \mu$ et de fonction d'auto-covariance $\lambda^H \Gamma \lambda$. L'étude des processus stationnaires au second ordre peut donc se restreindre au cas $d = 1$ sans grande perte de généralité.

7.5.1 Propriétés

Les propriétés de la proposition 104 se déclinent pour un processus stationnaire au second ordre de la façon suivante.

Proposition 106. *La fonction d'autocovariance $\gamma : \mathbb{Z} \rightarrow \mathbb{C}$ d'un processus stationnaire au second ordre à valeurs complexes vérifie les propriétés suivantes qui sont une conséquence directe de la proposition 104.*

1. *Symétrie hermitienne : Pour tout $s \in \mathbb{Z}$,*

$$\gamma(-s) = \overline{\gamma(s)}$$

2. *Type positif : Pour tout entier $n \geq 1$ et tout vecteur (a_1, \dots, a_n) de valeurs complexes,*

$$\sum_{s=1}^n \sum_{t=1}^n \overline{a_s} \gamma(s-t) a_t \geq 0$$

La matrice de covariance de n valeurs consécutives X_1, \dots, X_n du processus possède de plus une structure particulière, dite de *Toeplitz*, caractérisée par le fait que $(\Gamma_n)_{ij} =$

$\gamma(i-j)$. On obtient une matrice de la forme

$$\begin{aligned}\Gamma_n &= \text{cov}([X_1 \dots X_n]^T) \\ &= \begin{bmatrix} \gamma(0) & \gamma(-1) & \cdots & \gamma(1-n) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(2-n) \\ \vdots & & & \\ \gamma(n-1) & \gamma(n-2) & \cdots & \gamma(0) \end{bmatrix} \quad (7.12)\end{aligned}$$

Lorsque $\gamma(0)$ est non-nul il peut être pratique de normaliser la fonction d'autocovariance. On obtient la définition suivante.

Définition 107 (Fonction d'autocorrélation). *Pour un processus stationnaire au second ordre de variance non nulle, on appelle fonction d'autocorrélation ρ la fonction définie sur $s \in \mathbb{Z}$ par $\rho(s) = \gamma(s)/\gamma(0)$. Il s'agit d'une quantité normalisée dans le sens où $\rho(0) = 1$ et $|\rho(s)| \leq 1$ pour tout $s \in \mathbb{Z}$.*

En effet, l'inégalité de Cauchy-Schwarz appliquée à γ implique

$$|\gamma(s)| = |\text{cov}(X_s, X_0)| \leq \sqrt{\text{Var}(X_s) \text{Var}(X_0)} = \gamma(0)$$

la dernière inégalité découlant de l'hypothèse de stationnarité.

Exemple 108 (Retournement du temps (suite)). *Soit $(X_t)_{t \in \mathbb{Z}}$ un processus aléatoire stationnaire au second ordre à valeurs réelles de moyenne μ_X et de fonction d'autocovariance γ_X . On note, pour tout $t \in \mathbb{Z}$, $Y_t = X_{-t}$ le processus retourné, comme dans l'exemple 99. Alors Y_t est un processus stationnaire au second ordre de même moyenne et de même fonction d'autocovariance que le processus X_t . En effet on a :*

$$\begin{aligned}\mathbb{E}[Y_t] &= \mathbb{E}[X_{-t}] = \mu_X \\ \text{cov}(Y_{t+h}, Y_t) &= \text{cov}(X_{-t-h}, X_{-t}) = \gamma_X(-h) = \gamma_X(h)\end{aligned}$$

Définition 109 (Bruit blanc faible). *On appelle bruit blanc faible un processus aléatoire stationnaire au second ordre à valeurs complexes ou réelles, centré, de fonction d'autocovariance γ définie par $\gamma(0) = \sigma^2 > 0$ et $\gamma(s) = 0$ pour tout $s \neq 0$. On le notera $(X_t) \sim \text{B.B.}(0, \sigma^2)$.*

Définition 110 (Bruit blanc fort). *On appelle bruit blanc fort une suite de variables aléatoires (X_t) , centrées, indépendantes et identiquement distribuées (i.i.d.) de variance $\mathbb{E}[X_t^2] = \sigma^2 < \infty$. On le notera $(X_t) \sim \text{B.B.F.}(0, \sigma^2)$.*

Par définition un bruit blanc fort est un bruit blanc faible. La structure de bruit blanc fort est clairement plus contraignante que celle du bruit blanc faible. Notons que, de même que la stationnarité stricte d'un processus gaussien découle de la stationnarité faible, un bruit blanc gaussien est un bruit blanc fort.

Exemple 111 (Processus MA(1)). *Soit (X_t) le processus stationnaire au second ordre défini par :*

$$X_t = Z_t + \theta Z_{t-1}, \quad (7.13)$$

où $(Z_t) \sim \text{B.B.}(0, \sigma^2)$ réel et $\theta \in \mathbb{R}$. On vérifie aisément que $\mathbb{E}[X_t] = 0$ et que sa fonction d'autocovariance est définie par

$$\gamma(s) = \begin{cases} \sigma^2(1 + \theta^2) & \text{si } s = 0, \\ \sigma^2\theta & \text{si } s = \pm 1, \\ 0 & \text{sinon.} \end{cases} \quad (7.14)$$

Le processus (X_t) est donc bien stationnaire au second ordre. Un tel processus est appelé processus à moyenne ajustée d'ordre 1. Cette propriété se généralise, sans difficulté, à un processus MA(q). Nous reviendrons plus en détail, paragraphe 8.2, sur la définition et les propriétés de ces processus.

Exemple 112 (Processus harmonique réel). Soient $(A_k)_{1 \leq k \leq N}$ N v.a. réelles de variance finie. On note $\sigma_k^2 = \text{Var}(A_k)$. Soient $(\Phi_k)_{1 \leq k \leq N}$, N variables aléatoires indépendantes et identiquement distribuées (i.i.d), de loi uniforme sur $[-\pi, \pi]$, et indépendantes de $(A_k)_{1 \leq k \leq N}$. On définit :

$$X_t = \sum_{k=1}^N A_k \cos(\lambda_k t + \Phi_k), \quad (7.15)$$

où $(\lambda_k)_{1 \leq k \leq N} \in [-\pi, \pi]$ sont N pulsations. Le processus (X_t) est appelé processus harmonique. On vérifie aisément que $\mathbb{E}[X_t] = 0$ et que, pour tout $s, t \in \mathbb{Z}$,

$$\mathbb{E}[X_s X_t] = \frac{1}{2} \sum_{k=1}^N \sigma_k^2 \cos(\lambda_k(s-t)).$$

Le processus harmonique est donc stationnaire au second ordre.

Exemple 113 (Marche aléatoire). Soit (S_t) le processus défini sur $t \in \mathbb{N}$ par $S_t = X_0 + X_1 + \dots + X_t$, où (X_t) est un bruit blanc fort réel. Un tel processus est appelé une marche aléatoire. On en déduit que $\mathbb{E}[S_t] = 0$, $\mathbb{E}[S_t^2] = t\sigma^2$ et, pour $s \leq t \in \mathbb{N}$, on a :

$$\mathbb{E}[S_s S_t] = \mathbb{E}[(S_s + X_{s+1} + \dots + X_t)S_s] = s\sigma^2$$

Le processus (S_t) n'est donc pas stationnaire au second ordre.

Exemple 114. Nous allons montrer que la fonction χ définie sur \mathbb{Z} , par

$$\chi(s) = \begin{cases} 1 & \text{si } s = 0, \\ \rho & \text{si } s = \pm 1, \\ 0 & \text{sinon.} \end{cases} \quad (7.16)$$

est la fonction d'autocovariance d'un processus stationnaire au second ordre réel si et seulement si $\rho \in [-1/2, 1/2]$. Nous avons déjà montré exemple 111 que la fonction d'autocovariance γ d'un processus MA(1) est donnée par (7.14). La fonction χ est donc la fonction d'autocovariance d'un processus MA(1) si et seulement si $\sigma^2(1 + \theta^2) = 1$ et $\sigma^2\theta = \rho$. Lorsque $|\rho| \leq 1/2$, ce système d'équations admet comme solution :

$$\theta = (2\rho)^{-1}(1 \pm \sqrt{1 - 4\rho^2}) \quad \text{et} \quad \sigma^2 = (1 + \theta^2)^{-1}.$$

Lorsque $|\rho| > 1/2$, ce système d'équations n'admet pas de solution réelles et la fonction χ n'est donc pas la fonction d'autocovariance d'un processus MA(1). Plus généralement, si $|\rho| > 1/2$, alors χ n'est en fait pas de type positif et n'est donc pas une fonction de covariance, voir Proposition 106.

7.5.2 Interprétation de la fonction d'autocovariance

Dans les exemples précédents, nous avons été amenés à évaluer la fonction d'autocovariance de processus pour quelques exemples simples de séries temporelles. Dans la plupart des problèmes d'intérêt pratique, nous ne partons pas de modèles de série temporelle définis *a priori*, mais d'*observations*, $\{x_1, \dots, x_n\}$ associées à une *réalisation* du processus. Afin de comprendre la structure de dépendance entre les différentes observations, nous serons amenés à *estimer* la loi du processus, ou du moins des caractéristiques de ces lois. Pour un processus stationnaire au second ordre, nous pourrons, à titre d'exemple, estimer sa moyenne par la *moyenne empirique* :

$$\hat{\mu}_n = n^{-1} \sum_{k=1}^n x_k$$

et les fonctions d'autocovariance et d'autocorrélation par les fonctions d'autocorrélation et d'autocovariance *empiriques*

$$\hat{\gamma}(h) = n^{-1} \sum_{k=1}^{n-|h|} (x_k - \hat{\mu}_n)(x_{k+|h|} - \hat{\mu}_n) \quad \text{et} \quad \hat{\rho}(h) = \hat{\gamma}(h)/\hat{\gamma}(0).$$

Lorsqu'il est *a priori* raisonnable de penser que la série considérée est stationnaire au second ordre, la moyenne empirique, la fonction d'autocovariance empirique et la fonction d'autocorrélation empirique sont des estimateurs consistants de la fonction d'autocovariance et de la fonction d'autocorrélation.

L'analyse de la fonction d'autocovariance empirique est un élément permettant de guider le choix d'un modèle approprié pour les observations. Par exemple, le fait que la fonction d'autocovariance empirique soit *proche* de zéro pour tout $h \neq 0$ (proximité qu'il faudra définir dans un sens statistique précis) indique par exemple qu'un bruit blanc est un modèle adéquat pour les données. La figure 7.5 représente les 100 premières valeurs de la fonction d'autocorrélation empirique de la série des battements cardiaques représentée figure 75. On observe que cette série est *positivement corrélée* c'est-à-dire que les fonctions coefficients d'autocorrélation sont positifs et significativement non nuls. Nous avons, à titre de comparaison, représenté aussi la fonction d'autocorrélation empirique d'une trajectoire de même longueur d'un bruit blanc gaussien. La figure 7.6 montre que le fait que $\hat{\rho}(1) = 0.966$ pour la série des battements cardiaques se traduit par une forte prédictibilité de X_{t+1} en fonction de X_t (les couples de points successifs s'alignent quasiment sur une droite). Nous montrerons au chapitre 9, que dans un tel contexte, $\mathbb{E}[(X_{t+1} - \mu) - \rho(1)(X_t - \mu)] = (1 - \rho^2)\text{cov}(X_t)$, c'est-à-dire, compte tenu de la valeur estimée pour $\rho(1)$, que la variance de "l'erreur de prédiction" $X_{t+1} - [\mu + \rho(1)(X_t - \mu)]$ est 15 fois plus faible que celle du signal original.

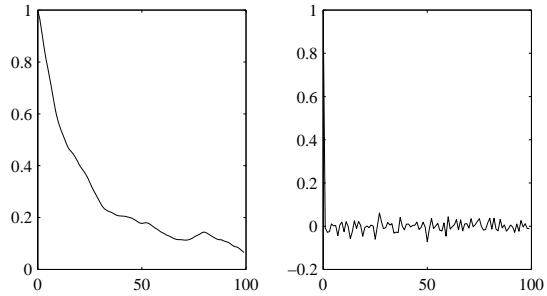


Figure 7.5: Courbe de gauche : fonction d'autocorrélation empirique de la série des battements cardiaques (figure 75). Courbe de droite : fonction d'autocorrélation empirique d'une trajectoire de même longueur d'un bruit blanc gaussien.

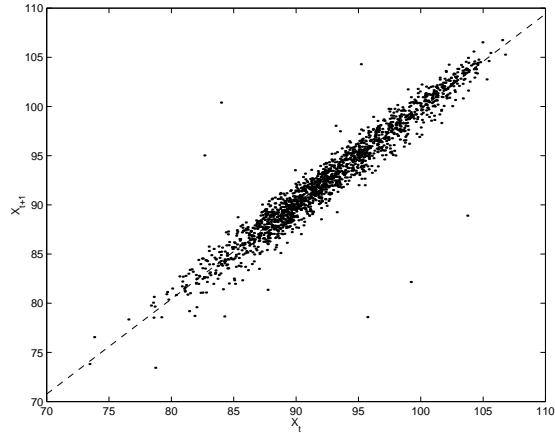


Figure 7.6: X_{t+1} en fonction de X_t pour la série des battements cardiaques de la figure 75. Les tirets représentent la meilleure droite de régression linéaire de X_{t+1} sur X_t .

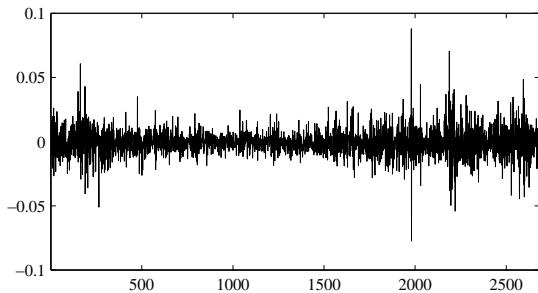


Figure 7.7: Log-Retours de la série S&P 500

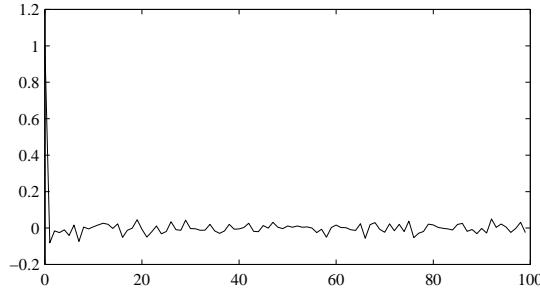


Figure 7.8: Fonction d'autocorrélation empirique de la série des log-retours de l'indice S&P 500.

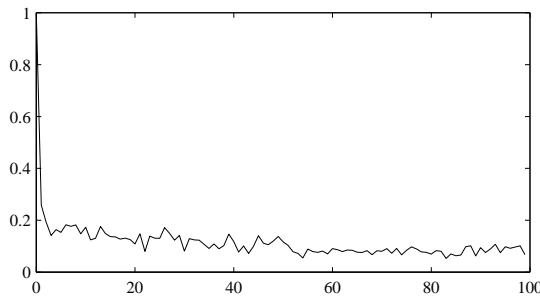


Figure 7.9: Fonction d'autocorrélation empirique de la série des valeurs absolues des log-retours de l'indice S&P 500.

L'indice S&P500 tracé (fig. 7.3) présente un cas de figure plus difficile, d'une part parce que la série n'est clairement pas stationnaire ; d'autre part, parce que selon le choix de la transformation des données considérées, la série transformée présente ou non des effets de corrélation. On définit tout d'abord les *log-retours* de l'indice S&P500 comme les différences des logarithmes de l'indice à deux dates successives :

$$X_t = \log(S_t) - \log(S_{t-1}) = \log\left(1 + \frac{S_t - S_{t-1}}{S_{t-1}}\right)$$

La série des log-retours de la série S&P 500 est représentée dans la figure 7.7.

Les coefficients d'autocorrélation empiriques de la série des log-retours sont représentés dans la figure 7.8. On remarque qu'ils sont approximativement nuls pour $h \neq 0$ ce qui suggère de modéliser la série des log-retours par un bruit blanc faible. Il est intéressant d'étudier aussi la série des log-retours absolu, $A(t) = |X_t|$. On peut, de la même façon, déterminer la suite des coefficients d'autocorrélation empirique de cette série, qui est représentée dans la figure 7.9. On voit, qu'à l'inverse de la série des log-retours, la série des valeurs absolues des log-retours est positivement corrélée, les valeurs d'autocorrélation étant significativement non nulles pour $|h| \leq 100$. On en déduit, en particulier, que la suite des log-retours peut être modélisée comme un

bruit blanc, mais pas un bruit blanc fort : en effet, pour un bruit blanc fort X_t , nous avons, pour toute fonction f telle que $\mathbb{E}[f(X_t)^2] = \sigma_f^2 < \infty$, $\text{cov}(f(X_{t+h}), f(X_t)) = 0$ pour $h \neq 0$ (les variables $f(X_{t+h})$ et $f(X_t)$ étant indépendantes, elles sont a fortiori non corrélées).

7.6 Mesure spectrale d'un processus stationnaire

Dans toute la suite, \mathbb{T} désigne le tore $]-\pi, \pi]$ et $\mathcal{B}(\mathbb{T})$ la tribu borélienne associée. Le théorème d'Herglotz ci dessous établit l'équivalence entre la fonction d'autocovariance et une mesure finie définie sur $(\mathbb{T}, \mathcal{B}(\mathbb{T}))$. Cette mesure, appelée *mesure spectrale du processus*, joue un rôle analogue à celui de la transformation de Fourier pour les fonctions de carré intégrable.

Théorème 115 (Herglotz). *Une suite $(\gamma(h))_{h \in \mathbb{Z}}$ est de type positif si et seulement si il existe une unique mesure positive ν sur $(\mathbb{T}, \mathcal{B}(\mathbb{T}))$ telle que :*

$$\gamma(h) = \int_{\mathbb{T}} e^{ih\lambda} \nu(d\lambda), \quad \forall h \in \mathbb{Z}. \quad (7.17)$$

Lorsque γ est la fonction d'autocovariance d'un processus stationnaire au second ordre, on sait d'après la proposition 106 que $\{\gamma(h)\}_{h \in \mathbb{Z}}$ est de type positif. Les hypothèses du théorème de Herglotz sont donc vérifiées et dans ce cas la mesure ν est appelée la *mesure spectrale* du processus. Si la mesure ν possède une densité f par rapport à la mesure de Lebesgue sur $(\mathbb{T}, \mathcal{B}(\mathbb{T}))$ alors f est appelée la *densité spectrale de puissance* du processus.

Proof. Si $\gamma(n)$ a la représentation (7.17), montrons que $\gamma(n)$ est de type positif. En effet, pour tout n et toute suite $\{a_k \in \mathbb{C}\}_{1 \leq k \leq n}$,

$$\sum_{k,m} a_k \overline{a_m} \gamma(k-m) = \int_{\mathbb{T}} \sum_{k,m} a_k \overline{a_m} e^{ik\lambda} e^{-im\lambda} \nu(d\lambda) = \int_{\mathbb{T}} \left| \sum_k a_k e^{ik\lambda} \right|^2 \nu(d\lambda) \geq 0.$$

Réciproquement, supposons que $\gamma(n)$ soit une suite de type positif et considérons la suite de fonctions indexée par n :

$$f_n(\lambda) = \frac{1}{2\pi n} \sum_{k=1}^n \sum_{m=1}^n \gamma(k-m) e^{-ik\lambda} e^{im\lambda} = \frac{1}{2\pi} \sum_{k=-(n-1)}^{n-1} \left(1 - \frac{|k|}{n} \right) \gamma(k) e^{-ik\lambda}.$$

γ étant de type positif, $f_n(\lambda) \geq 0$, pour tout $\lambda \in \mathbb{T}$. Notons ν_n la mesure (positive) de densité f_n par rapport à la mesure de Lebesgue sur \mathbb{T} . On a alors

$$\begin{aligned} \int_{\mathbb{T}} e^{ih\lambda} \nu_n(d\lambda) &= \int_{\mathbb{T}} e^{ih\lambda} f_n(\lambda) d\lambda = \frac{1}{2\pi} \sum_{k=-(n-1)}^{n-1} \left(1 - \frac{|k|}{n} \right) \gamma(k) \int_{\mathbb{T}} e^{i(h-k)\lambda} d\lambda \\ &= \begin{cases} \left(1 - \frac{|h|}{n} \right) \gamma(h), & \text{si } |h| < n, \\ 0, & \text{sinon.} \end{cases} \end{aligned} \quad (7.18)$$

Quitte à renormaliser v_n pour en faire une mesure de probabilité, le théorème de Prohorov implique qu'il existe une mesure positive v et une sous-suite v_{n_k} de v_n telle que

$$\int_{\mathbb{T}} e^{ih\lambda} v_{n_k}(d\lambda) \longrightarrow \int_{\mathbb{T}} e^{ih\lambda} v(d\lambda), \text{ lorsque } k \rightarrow \infty.$$

En remplaçant n par n_k dans (7.18) et en faisant tendre k vers l'infini, on a

$$\gamma(h) = \int_{\mathbb{T}} e^{ih\lambda} v(d\lambda), \forall h \in \mathbb{Z}.$$

Montrons à présent que v est unique. En effet, s'il existait une autre mesure μ telle que pour tout $h \in \mathbb{Z}$: $\int_{\mathbb{T}} e^{ih\lambda} v(d\lambda) = \int_{\mathbb{T}} e^{ih\lambda} \mu(d\lambda)$ alors d'après le lemme 193, $\int_{\mathbb{T}} g(\lambda) v(d\lambda) = \int_{\mathbb{T}} g(\lambda) \mu(d\lambda)$ pour toute fonction continue g telle que $g(\pi) = g(-\pi)$. On en déduit donc que $v = \mu$.

□

Corollaire 116 (Corollaire du théorème d'Herglotz). *Une suite $(\gamma(h))_{h \in \mathbb{Z}}$ à valeurs complexes telle que $\sum_{h \in \mathbb{Z}} |\gamma(h)|^2 < \infty$ est de type positif si et seulement si la fonction définie par*

$$f(\lambda) = \frac{1}{2\pi} \sum_{h \in \mathbb{Z}} \gamma(h) e^{-ih\lambda}$$

est positive pour tout $\lambda \in \mathbb{T}$.

Proof. D'après le théorème de Herglotz (Théorème 115), $(\gamma(h))_{h \in \mathbb{Z}}$ est de type positif si et seulement si il existe une mesure positive v sur $(\mathbb{T}, \mathcal{B}(\mathbb{T}))$ telle que :

$$\gamma(h) = \int_{\mathbb{T}} e^{ih\lambda} v(d\lambda).$$

D'après le théorème 180 et le corollaire 195, comme $\sum_{h \in \mathbb{Z}} |\gamma(h)|^2 < \infty$, on peut considérer la série de Fourier associée convergente dans $L_2(\mathbb{T}, \lambda^{\text{Leb}})$: $(2\pi)^{-1} \sum_{k \in \mathbb{Z}} \gamma(k) e^{-ik\lambda} \stackrel{\text{def}}{=} f(\lambda)$. Ainsi, $\gamma(h) = \int_{\mathbb{T}} e^{ih\lambda} f(\lambda) d\lambda$ et donc la positivité de v revient à la positivité de f , ce qui conclut la preuve.

□

Exemple 117. En reprenant l'exemple 114, on vérifie immédiatement que $(\chi(h))$ est de module sommable et que :

$$f(\lambda) = \frac{1}{2\pi} \sum_h \chi(h) e^{-ih\lambda} = \frac{1}{2\pi} (1 + 2\rho \cos(\lambda))$$

et donc que la séquence est une fonction d'autocovariance uniquement lorsque $|\rho| \leq 1/2$.

Exemple 118 (Densité spectrale de puissance du bruit blanc). *La fonction d'autocovariance d'un bruit blanc est donnée par $\gamma(h) = \sigma^2 \delta(h)$, d'où l'expression de la densité spectrale correspondante*

$$f(\lambda) = \frac{\sigma^2}{2\pi}$$

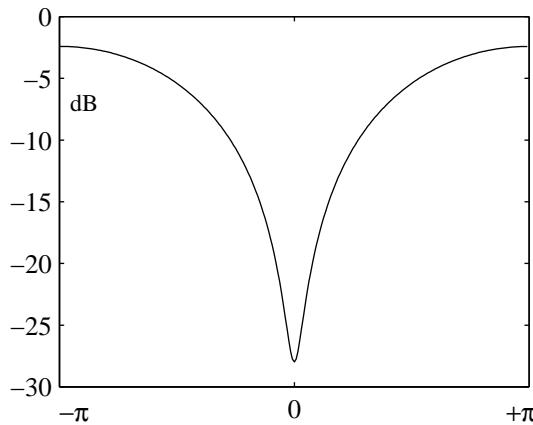


Figure 7.10: Densité spectrale (en dB) d'un processus MA-1, défini par l'équation (7.13) pour $\sigma = 1$ et $\theta = -0.9$.

La densité spectrale d'un bruit blanc est donc constante. Cette propriété est à l'origine de la terminologie "bruit blanc" qui provient de l'analogie avec le spectre de la lumière blanche constant dans toute la bande de fréquences visibles.

Exemple 119 (Densité spectrale de puissance du processus MA(1)). Le processus MA(1) introduit dans l'exemple 111 possède une séquence d'autocovariance donnée par $\gamma(0) = \sigma^2(1 + \theta^2)$, $\gamma(1) = \gamma(-1) = \sigma^2\theta$ et $\gamma(h) = 0$ sinon (cf. exemple 111). D'où l'expression de sa densité spectrale :

$$f(\lambda) = \frac{\sigma^2}{2\pi} (2\theta \cos(\lambda) + (1 + \theta^2)) = \frac{\sigma^2}{2\pi} |1 + \theta e^{-i\lambda}|^2$$

La densité spectrale d'un tel processus est représentée figure 7.10 pour $\theta = -0.9$ et $\sigma^2 = 1$ avec une échelle logarithmique (dB).

Exemple 120 (Mesure spectrale du processus harmonique). La fonction d'autocovariance du processus harmonique $X_t = \sum_{k=1}^N A_k \cos(\lambda_k t + \Phi_k)$ (voir exemple 112) est donnée par :

$$\gamma(h) = \frac{1}{2} \sum_{k=1}^N \sigma_k^2 \cos(\lambda_k h) \quad (7.19)$$

où $\sigma_k^2 = \mathbb{E}[A_k^2]$. Cette suite de coefficients d'autocovariance n'est pas sommable et la mesure spectrale n'admet pas de densité. En notant cependant que :

$$\cos(\lambda_k h) = \frac{1}{2} \int_{-\pi}^{\pi} e^{ih\lambda} (\delta_{\lambda_k}(d\lambda) + \delta_{-\lambda_k}(d\lambda))$$

où $\delta_{x_0}(d\lambda)$ désigne la mesure de Dirac au point x_0 (cette mesure associe la valeur 1 à tout borélien de $[-\pi, \pi]$ contenant x_0 et la valeur 0 sinon), la mesure spectrale du

processus harmonique peut s'écrire :

$$v(d\lambda) = \frac{1}{4} \sum_{k=1}^N \sigma_k^2 \delta_{\lambda_k}(d\lambda) + \frac{1}{4} \sum_{k=1}^N \sigma_k^2 \delta_{-\lambda_k}(d\lambda)$$

Elle apparaît donc comme une somme de mesures de Dirac, dont les masses σ_k^2 sont localisées aux pulsations des différentes composantes harmoniques.

Contrairement aux autres exemples étudiés, le processus harmonique possède une fonction d'autocovariance, donnée par 7.19, non absolument sommable ($\gamma(h)$ ne tend pas même vers 0 pour les grandes valeurs de h). Par suite, il admet une mesure spectrale mais pas une densité spectrale. La propriété suivante, à démontrer à titre d'exercice, implique que le processus harmonique est en fait entièrement prédictible à partir de quelques-unes de ses valeurs passées.

Proposition 121. *S'il existe un rang n pour lequel la matrice de covariance Γ_n définie en (7.12) est non inversible, le processus correspondant X_t est prédictible dans le sens où il existe une combinaison linéaire a_1, \dots, a_l avec $l \leq n - 1$ telle que $X_t = \sum_{k=1}^l a_k X_{t-k}$, l'égalité ayant lieu presque sûrement.*

L'expression de la fonction d'autocovariance, obtenue en (7.19) pour le processus harmonique, montre que les matrices de covariances associées s'écrivent comme la somme de $2N$ matrices complexes de rang 1. Par conséquent, les matrices Γ_n ne sont pas inversibles dès que $n > 2N$, ce qui implique que le processus harmonique est prédictible dès lors que l'on en a observé $2N$ valeurs. Ce résultat est sans surprise compte tenu du fait que les trajectoires de ce processus sont des sommes de sinusoïdes de fréquences $\lambda_1, \dots, \lambda_N$ dont seules les amplitudes et les phases sont aléatoires. La propriété suivante donne une condition suffisante simple pour éviter ce type de comportements "extrêmes". Cette propriété implique en particulier que, pour une fonction d'autocovariance absolument sommable (tous les exemples vus ci-dessus en dehors du processus harmoniques), les valeurs futures du processus correspondant ne sont pas prédictibles sans erreur à partir d'un ensemble fini de valeurs passées du processus. Nous reviendrons en détail sur ces problèmes de prédiction au chapitre 9.

Proposition 122. *Soit $\gamma(h)$ la fonction d'autocovariance d'un processus stationnaire au second ordre. On suppose que $\gamma(0) > 0$ et que $\gamma(h) \rightarrow 0$ quand $h \rightarrow \infty$. Alors, quel que soit n , la matrice de covariance définie en (7.12) est de rang plein et donc inversible.*

Proof. Supposons qu'il existe une suite de valeurs complexes (a_1, \dots, a_n) non toutes nulles, telle que $\sum_{k=1}^n \sum_{m=1}^n a_k \overline{a_m} \gamma(k-m) = 0$. En notant v_X la mesure spectrale de X_t , on peut écrire :

$$0 = \sum_{k=1}^n \sum_{m=1}^n a_k \overline{a_m} \int_{\mathbb{T}} e^{i(k-m)\lambda} v_X(d\lambda) = \int_{\mathbb{T}} \left| \sum_{k=1}^n a_k e^{ik\lambda} \right|^2 v_X(d\lambda)$$

Ce qui implique que $\left| \sum_{k=1}^n a_k e^{ik\lambda} \right|^2 = 0$ v_X presque partout, c'est à dire que

$$v_X(\{\lambda : \left| \sum_{k=1}^n a_k e^{ik\lambda} \right|^2 \neq 0\}) = v_X(\mathbb{T} - Z) = 0$$

où $Z = \{\lambda_1, \dots, \lambda_M : \sum_{k=1}^n a_k e^{ik\lambda_m} = 0\}$ désigne l'ensemble *fini* ($M < n$) des racines $x \in \mathbb{T}$ du polynôme trigonométrique $\sum_{k=1}^n a_k e^{ik\lambda}$. Par conséquent, les seuls éléments de $\mathcal{B}(\mathbb{T})$, qui peuvent être de mesure non nulle pour v_X , sont les singletons $\{\lambda_m\}$. Ce qui implique que $v_X = \sum_{m=1}^M a_m \delta_{\lambda_m}$ (où $a_m \geq 0$ ne peuvent être tous nuls si $\gamma(0) \neq 0$). Mais, dans ce cas, $\gamma(h) = \sum_{m=1}^M a_m e^{ih\lambda_m}$, ce qui contredit l'hypothèse que $\gamma(h)$ tend vers 0 quand n tend vers l'infini. \square

Chapitre 8

Filtrage des signaux aléatoires à temps-discret

Dans ce chapitre nous nous intéressons à une classe très importante de processus du second ordre, les processus autorégressifs à moyenne ajustée ou processus ARMA. Afin de pouvoir étudier leurs propriétés, nous allons tout d'abord établir les propriétés des processus obtenus par un filtrage linéaire de processus stationnaires au second ordre.

8.1 Filtrages linéaires de processus au second ordre

On s'intéresse dans ce paragraphe aux propriétés du processus (Y_t) obtenu comme image du processus (X_t) par le filtre linéaire suivant :

$$Y_t = \sum_{k \in \mathbb{Z}} \psi_k X_{t-k}, \quad (8.1)$$

où (ψ_k) est une suite de nombres complexes. Lorsqu'il n'y a qu'un nombre fini de ψ_k non nuls, la somme (8.1) est bien définie. On dit dans ce cas-là que le filtre est à réponse impulsionnelle finie. La question devient plus délicate lorsque l'on considère des filtres à réponse impulsionnelle infinie c'est à dire lorsque le nombre de ψ_k non nuls est infini. En effet, Y_t défini par (8.1) est la limite dans un sens à préciser, d'une suite de variables aléatoires. Le théorème 123 donne un sens précis à cette limite.

Théorème 123. Soit $(\psi_k)_{k \in \mathbb{Z}}$ une suite absolument sommable, i.e. $\sum_{k=-\infty}^{\infty} |\psi_k| < \infty$ et soit (X_t) un processus aléatoire tel que $\sup_{t \in \mathbb{Z}} \mathbb{E}[|X_t|] < \infty$. Alors, pour tout $t \in \mathbb{Z}$, la suite :

$$Y_{n,t} = \sum_{k=-n}^n \psi_k X_{t-k}$$

converge presque sûrement, quand n tend vers l'infini, vers une limite Y_t que nous notons

$$Y_t = \sum_{k=-\infty}^{\infty} \psi_k X_{t-k}.$$

De plus, la variable aléatoire Y_t est intégrable, i.e. $\mathbb{E}[|Y_t|] < \infty$ et la suite $(Y_{n,t})_{n \geq 0}$ converge vers Y_t dans $L^1(\Omega, \mathcal{A}, \mathbb{P})$, i.e.

$$\lim_{n \rightarrow \infty} \mathbb{E}[|Y_{n,t} - Y_t|] = 0.$$

Supposons que $\sup_{t \in \mathbb{Z}} \mathbb{E}[|X_t|^2] < \infty$ alors $\mathbb{E}[|Y_t|^2] < \infty$ et la suite $(Y_{n,t})_{n \geq 0}$ converge en moyenne quadratique vers la variable aléatoire Y_t , i.e.

$$\lim_{n \rightarrow \infty} \mathbb{E}[|Y_{n,t} - Y_t|^2] = 0.$$

Proof. Notons pour tout $t \in \mathbb{Z}$ et $n \in \mathbb{N}$, $U_{n,t} = \sum_{k=-n}^n |\psi_k| |X_{t-k}|$. La suite $(U_{n,t})_{n \geq 0}$ est une suite de variables aléatoires intégrables. Puisque $\lim_{n \rightarrow \infty} \mathbb{E}[U_{n,t}] = \sum_{k \in \mathbb{Z}} |\psi_k| |X_{t-k}|$, on en déduit que (théorème de Beppo-Levi)

$$\lim_{n \rightarrow \infty} \mathbb{E}[U_{n,t}] = \mathbb{E}\left[\sum_{k \in \mathbb{Z}} |\psi_k| |X_{t-k}|\right],$$

où $\lim_{n \rightarrow \infty} \uparrow$ signifie qu'il s'agit d'une limite croissante. Comme

$$\mathbb{E}[U_{n,t}] \leq \sum_{k=-n}^n |\psi_k| \mathbb{E}[|X_{t-k}|] \leq \sup_{t \in \mathbb{Z}} \mathbb{E}[|X_t|] \sum_{k \in \mathbb{Z}} |\psi_k| < \infty,$$

on en déduit que

$$\mathbb{E}\left[\sum_{k \in \mathbb{Z}} |\psi_k| |X_{t-k}|\right] < \infty.$$

Par conséquent, il existe un ensemble $\Omega_0 \in \mathcal{F}$ tel que $\mathbb{P}(\Omega_0) = 1$ et tel que, pour tout $\omega \in \Omega_0$,

$$\sum_{k \in \mathbb{Z}} |\psi_k| |X_{t-k}(\omega)| < \infty.$$

Donc pour tout $\omega \in \Omega_0$,

$$|Y_{n,t}(\omega) - Y_t(\omega)| \leq \sum_{|k|>n} |\psi_k| |X_{t-k}(\omega)| \rightarrow 0, \text{ lorsque } n \rightarrow \infty.$$

Ainsi, pour tout $\omega \in \Omega_0$, $Y_{n,t}(\omega)$ est convergente et converge vers $Y_t(\omega)$, ce qui montre que $\lim_{n \rightarrow \infty} Y_{n,t} = Y_t$ p.s.. Le lemme de Fatou montre que

$$\mathbb{E}[|Y_t|] = \mathbb{E}\left[\liminf_n |Y_{n,t}|\right] \leq \liminf_n \mathbb{E}[|Y_{n,t}|] \leq \sup_t \mathbb{E}[|X_t|] \sum_{j=-\infty}^{\infty} |\psi_j| < \infty,$$

et donc que $Y_t \in L_1(\Omega, \mathcal{A}, \mathbb{P})$. Comme $|Y_{n,t} - Y_t| \leq \sum_{k \in \mathbb{Z}} |\psi_k| |X_{t-k}|$, le théorème de convergence dominée montre que $\lim_n \mathbb{E}[|Y_{n,t} - Y_t|] = 0$ et donc que la suite $\{Y_{n,t}\}$ converge vers Y_t dans $L_1(\Omega, \mathcal{A}, \mathbb{P})$.

Considérons maintenant le cas où $\sup_{t \in \mathbb{Z}} \mathbb{E}[|X_t|^2] < \infty$. Remarquons tout d'abord que $\mathbb{E}[|X_t|] \leq (\mathbb{E}[|X_t|^2])^{1/2}$ et donc que cette condition implique que $\sup_{t \in \mathbb{Z}} \mathbb{E}[|X_t|] < \infty$. La suite $(Y_{m,t})_{m \geq 0}$ est une suite de Cauchy dans $L^2(\Omega, \mathcal{A}, \mathbb{P})$. En effet, pour $p \geq q$, nous avons en notant $\|X\|_2 = (\mathbb{E}[|X|^2])^{1/2}$

$$\|Y_{p,t} - Y_{q,t}\|_2 = \left\| \sum_{|k|=q+1}^p \psi_k X_{t-k} \right\|_2 \leq \sup_t \|X_t\| \sum_{|k|=q+1}^p |\psi_k| \xrightarrow[q,p \rightarrow \infty]{} 0.$$

Comme $L^2(\Omega, \mathcal{A}, \mathbb{P})$ est complet, la suite $(Y_{n,t})$ converge vers une variable Y_t^* . En utilisant le lemme de Fatou, nous avons

$$\mathbb{E}[|Y_t - Y_t^*|^2] = \mathbb{E}\left[\liminf_n |Y_{n,t} - Y_t^*|^2\right] \leq \liminf_n \mathbb{E}[|Y_{n,t} - Y_t^*|^2] = 0,$$

ce qui montre que les limites p.s. Y_t et $L_2(\Omega, \mathcal{A}, \mathbb{P})$, Y_t^* coïncident p.s.. \square

Le résultat suivant établit que le processus (Y_t) obtenu par filtrage linéaire d'un processus stationnaire au second ordre (X_t) via l'équation (8.1) est lui-même stationnaire au second ordre, à condition que la suite des (ψ_k) soit absolument sommable i.e. $\sum_{k \in \mathbb{Z}} |\psi_k| < \infty$.

Théorème 124 (Filtrage des processus stationnaires au second ordre). Soit (ψ_k) une suite telle que $\sum_{k=-\infty}^{\infty} |\psi_k| < \infty$ et soit (X_t) un processus stationnaire au second ordre de moyenne $\mu_X = \mathbb{E}[X_t]$ et de fonction d'autocovariance $\gamma_X(h) = \text{cov}(X_{t+h}, X_t)$ alors le processus $Y_t = \sum_{k=-\infty}^{\infty} \psi_k X_{t-k}$ est stationnaire au second ordre de moyenne :

$$\mu_Y = \mu_X \sum_{k=-\infty}^{\infty} \psi_k, \quad (8.2)$$

de fonction d'autocovariance :

$$\gamma_Y(h) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \psi_j \bar{\psi}_k \gamma_X(h+k-j), \quad (8.3)$$

et de mesure spectrale :

$$v_Y(d\lambda) = |\psi(e^{-i\lambda})|^2 v_X(d\lambda), \quad (8.4)$$

où $\psi(e^{-i\lambda}) = \sum_{k \in \mathbb{Z}} \psi_k e^{-ik\lambda}$.

Proof. D'après la continuité du produit scalaire dans $L^2(\Omega, \mathcal{A}, \mathbb{P})$, voir théorème 167, on a

$$\begin{aligned} \mathbb{E} \left[\sum_{k \in \mathbb{Z}} \psi_k X_{t-k} \right] &= \mathbb{E} \left[\lim_{n \rightarrow \infty} \sum_{k=-n}^n \psi_k X_{t-k} \right] = \lim_{n \rightarrow \infty} \mathbb{E} \left[\sum_{k=-n}^n \psi_k X_{t-k} \right] \\ &= \mu_X \left(\lim_{n \rightarrow \infty} \sum_{k=-n}^n \psi_k \right) = \mu_X \sum_{k \in \mathbb{Z}} \psi_k. \end{aligned}$$

Montrons à présent le résultat sur la fonction d'auto-covariance. D'après la continuité du produit scalaire dans $L^2(\Omega, \mathcal{A}, \mathbb{P})$, voir théorème 167, on a

$$\text{Cov}(Y_t, Y_{t+h}) = \lim_{n \rightarrow \infty} \sum_{k,j=-n}^n \psi_k \bar{\psi}_j \text{Cov}(X_{t-k}, X_{t+h-j}) = \sum_{k \in \mathbb{Z}} \psi_k \bar{\psi}_j \gamma_X(h+k-j),$$

ce qui montre Eq. (8.3)

D'après le théorème 115, $\gamma_X(h) = \int_{\mathbb{T}} e^{ih\lambda} v_X(d\lambda)$ où v_X désigne la mesure spectrale du processus (X_t) . En reportant cette expression de $\gamma_X(h)$ dans (8.3), nous obtenons

$$\gamma_Y(h) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \psi_j \bar{\psi}_k \int_{\mathbb{T}} e^{i(h+k-j)\lambda} v_X(d\lambda). \quad (8.5)$$

Puisque

$$\sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \int_{\mathbb{T}} |\psi_j| |\psi_k| v_X(d\lambda) \leq \gamma_X(0) \left(\sum_{j=-\infty}^{\infty} |\psi_j| \right)^2 < \infty,$$

on peut appliquer le théorème de Fubini et permute les signes somme et intégrale dans (8.5). Ce qui donne :

$$\gamma_Y(h) = \int_{\mathbb{T}} e^{ih\lambda} \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \psi_j \bar{\psi}_k e^{ik\lambda} e^{-ij\lambda} = \int_{\mathbb{T}} e^{ih\lambda} |\psi(e^{-i\lambda})|^2 v_X(d\lambda),$$

où $\psi(e^{-i\lambda}) = \sum_{k \in \mathbb{Z}} \psi_k e^{-ik\lambda}$. On en déduit, d'après le théorème 115 que $v_Y(d\lambda) = |\psi(e^{-i\lambda})|^2 v_X(d\lambda)$.

□

Nous définissons à présent une classe très importante de processus obtenus par filtrage : les *processus linéaires* qui sont obtenus en filtrant un bruit blanc.

Définition 125 (Processus linéaire). *Nous dirons que (X_t) est un processus linéaire s'il existe un bruit blanc $Z_t \sim \text{B.B.}(0, \sigma^2)$ et une suite de coefficients $(\psi_k)_{k \in \mathbb{Z}}$ absolument sommable telle que :*

$$X_t = \mu + \sum_{k=-\infty}^{\infty} \psi_k Z_{t-k}, \quad t \in \mathbb{Z}, \quad (8.6)$$

où μ est un nombre complexe. On dira que $(X_t)_{t \in \mathbb{Z}}$ est un processus linéaire causal par rapport à $(Z_t)_{t \in \mathbb{Z}}$ si (8.6) est vérifiée avec $\psi_k = 0$ pour tout $k < 0$. On dira que $(X_t)_{t \in \mathbb{Z}}$ est un processus linéaire inversible par rapport à $(Z_t)_{t \in \mathbb{Z}}$ si (8.6) est vérifiée et qu'il existe de plus une suite $(\pi_k)_{k \geq 0}$ absolument sommable telle que

$$Z_t = \sum_{k=0}^{\infty} \pi_k (X_{t-k} - \mu), \quad t \in \mathbb{Z}. \quad (8.7)$$

D'après le théorème 124, un processus linéaire est stationnaire au second ordre de moyenne μ , de fonction d'autocovariance :

$$\gamma_X(h) = \sigma^2 \sum_{j=-\infty}^{\infty} \psi_j \bar{\psi}_{j+h} = \sigma^2 \sum_{\ell=-\infty}^{\infty} \psi_{\ell-h} \bar{\psi}_{\ell}, \quad (8.8)$$

et dont la mesure spectrale admet une densité donnée par :

$$f_X(\lambda) = \frac{\sigma^2}{2\pi} |\psi(e^{-i\lambda})|^2, \quad (8.9)$$

où $\psi(e^{-i\lambda}) = \sum_{k \in \mathbb{Z}} \psi_k e^{-ik\lambda}$.

8.2 Processus ARMA

Avant de passer au cas général des processus ARMA, nous nous intéressons à deux classes de processus ARMA particuliers : les processus à moyenne ajustée (MA) et les processus autorégressifs (AR).

8.2.1 Processus MA(q)

Définition 126 (Processus MA(q)). *On dit que le processus (X_t) est à moyenne ajustée d'ordre q (ou MA(q)) si X_t est donné par :*

$$X_t = Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q} \quad (8.10)$$

où $Z_t \sim \text{B.B.}(0, \sigma^2)$ et les θ_i sont des nombres complexes.

Le terme “moyenne ajustée” est la traduction assez malheureuse du nom anglo-saxon “moving average” (moyenne mobile). Observons que $X_t = \sum_{k=0}^q \theta_k Z_{t-k}$, avec la convention $\theta_0 = 1$. En utilisant les résultats du théorème 124, on obtient $\mathbb{E}[X_t] = 0$, et

$$\gamma_X(h) = \begin{cases} \sigma^2 \sum_{t=0}^{q-h} \theta_k \bar{\theta}_{k+h}, & \text{si } 0 \leq h \leq q, \\ \sigma^2 \sum_{t=0}^{q+h} \bar{\theta}_k \theta_{k-h}, & \text{si } -q \leq h \leq 0, \\ 0, & \text{sinon.} \end{cases} \quad (8.11)$$

Enfin, d’après la formule (8.9), le processus admet une densité spectrale dont l’expression est :

$$f_X(\lambda) = \frac{\sigma^2}{2\pi} \left| 1 + \sum_{k=1}^q \theta_k e^{-ik\lambda} \right|^2.$$

Un exemple de densité spectrale pour le processus MA(1) est représenté sur la figure 7.10.

8.2.2 Processus AR(p)

Définition 127 (Processus AR(p)). *On dit que le processus $\{X_t\}$ est un processus autorégressif d’ordre p (ou AR(p)) si $\{X_t\}$ est un processus stationnaire au second ordre et s’il est solution de l’équation de récurrence :*

$$X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + Z_t, \quad (8.12)$$

où $Z_t \sim \text{B.B.}(0, \sigma^2)$ est un bruit blanc et les ϕ_k sont des nombres complexes.

Le terme “autorégressif” provient de la forme de l’équation (8.12) dans laquelle la valeur courante du processus s’exprime sous la forme d’une régression des p valeurs précédentes du processus plus un bruit additif.

L’existence et l’unicité d’une solution stationnaire au second ordre de l’équation (8.12) sont des questions délicates (qui ne se posaient pas lorsque nous avions défini les modèles MA). Nous détaillons ci-dessous la réponse à cette question dans le cas $p = 1$.

Cas : $|\phi_1| < 1$

L’équation de récurrence (8.12) s’écrit dans le cas $p = 1$:

$$X_t = \phi_1 X_{t-1} + Z_t, \quad (8.13)$$

où $(Z_t) \sim \text{B.B.}(0, \sigma^2)$. En itérant (8.13), on obtient :

$$\begin{aligned} X_t &= \phi_1 (\phi_1 X_{t-2} + Z_{t-1}) + Z_t = \phi_1^2 X_{t-2} + \phi_1 Z_{t-1} + Z_t \\ &= \phi_1^{k+1} X_{t-k-1} + \phi_1^k Z_{t-k} + \cdots + \phi_1^2 Z_{t-2} + \phi_1 Z_{t-1} + Z_t. \end{aligned}$$

En prenant la limite quand $k \rightarrow \infty$, on en déduit que

$$X_t = \sum_{j=0}^{\infty} \phi_1^j Z_{t-j}, \quad (8.14)$$

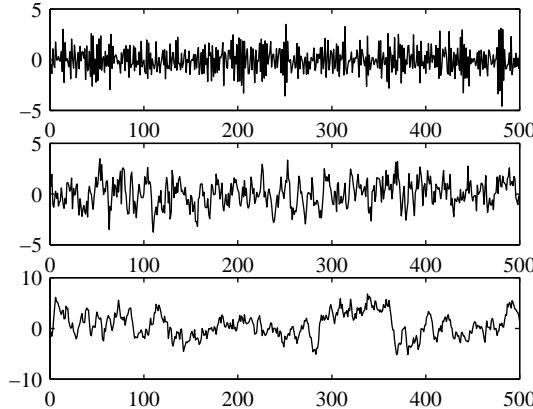


Figure 8.1: Trajectoires de longueur 500 d'un processus AR(1) gaussien. Courbe du haut : $\phi_1 = -0.7$. Courbe du milieu : $\phi_1 = 0.5$. Courbe du bas : $\phi_1 = 0.9$

la série convergeant dans $L_2(\Omega, \mathcal{A}, \mathbb{P})$ et p.s.. En effet, si on suppose que X_t une solution stationnaire,

$$\mathbb{E} \left[\left| X_t - \sum_{j=0}^k \phi_1^j Z_{t-j} \right|^2 \right] = |\phi_1|^{2k+2} \mathbb{E} [|X_{t-k-1}|^2] = |\phi_1|^{2k+2} \mathbb{E} [|X_0|^2] \rightarrow 0, k \rightarrow \infty,$$

puisque $|\phi_1| < 1$. De plus, d'après la définition 125, (X_t) défini par (8.14) est un processus linéaire et est donc stationnaire au second ordre. On peut vérifier que (X_t) défini par (8.14) est bien solution de (8.13) en notant que :

$$X_t = Z_t + \phi_1 \sum_{k=0}^{+\infty} \phi_1^k Z_{t-1-k} = Z_t + \phi_1 X_{t-1}.$$

Remarquons que la solution donnée par (8.14) peut être obtenu en utilisant le développement la fraction rationnelle $\psi(z) = (1 - \phi_1 z^{-1})^{-1}$ en série entière

$$\psi(z) = \frac{1}{1 - \phi_1 z^{-1}} = \sum_{k=0}^{+\infty} \phi_1^k z^{-k}$$

convergeant sur le disque $\{z \in \mathbb{C} : |\phi_1| < |z|\}$. Ce lien n'a rien de fortuit, comme nous le verrons dans le Section 8.2.3.

La fonction d'autocovariance de (X_t) solution stationnaire de (8.13) est donnée par la formule (8.8) qui s'écrit ;

$$\gamma_X(h) = \sigma^2 \sum_{k=0}^{\infty} \phi_1^k \bar{\phi}_1^{k+h} = \sigma^2 \frac{\bar{\phi}_1^h}{1 - |\phi_1|^2}, \text{ si } h \geq 0, \quad (8.15)$$

$$= \overline{\gamma(-h)}, \text{ sinon.} \quad (8.16)$$

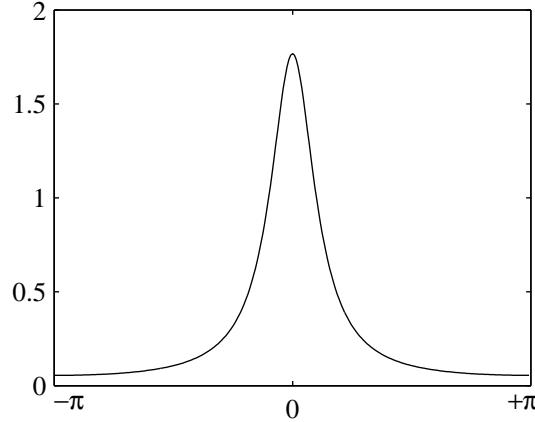


Figure 8.2: Densité spectrale d'un processus AR(1), défini par (8.13) pour $\sigma = 1$ et $\phi_1 = 0.7$.

Lorsque ϕ_1 est un réel strictement positif, le processus (X_t) est positivement corrélé, dans le sens où tous ses coefficients d'auto-covariance sont positifs. Les exemples de trajectoires représentées sur la figure 8.1 montrent que des valeurs de ϕ_1 proches de 1 correspondent à des trajectoires “persistantes”. Inversement, des valeurs de ϕ_1 réelles et négatives conduisent à des trajectoires où une valeur positive a tendance à être suivie par une valeur négative. La densité spectrale de (X_t) est donnée par

$$f_X(\lambda) = \frac{\sigma^2}{2\pi} \left| \sum_{k=0}^{\infty} \phi_1^k e^{-ik\lambda} \right|^2 = \frac{\sigma^2}{2\pi} \frac{1}{|1 - \phi_1 e^{-i\lambda}|^2}. \quad (8.17)$$

Cas $|\phi_1| > 1$

Dans ce cas-là, $\mathbb{E} \left[|X_t - \sum_{j=0}^k \phi_1^j Z_{t-j}|^2 \right] = |\phi_1|^{2k+2} \mathbb{E} [X_{t-k-1}^2]$ diverge lorsque k tend vers l'infini. Par contre, on peut réécrire l'équation définissant X_t en fonction de Z_t comme suit

$$X_t = -\phi_1^{-1} Z_{t+1} + \phi_1^{-1} X_{t+1}.$$

En itérant l'équation précédente, on obtient

$$\begin{aligned} X_t &= -\phi_1^{-1} Z_{t+1} - \phi_1^{-2} Z_{t+2} + \phi_1^{-2} X_{t+2} = \dots \\ &= -\phi_1^{-1} Z_{t+1} - \phi_1^{-2} Z_{t+2} - \dots - \phi_1^{-k-1} Z_{t+k+1} + \phi_1^{-k-1} X_{t+k+1}. \end{aligned}$$

En utilisant exactement les mêmes arguments que ceux employés précédemment, on déduit que la solution stationnaire dans ce cas vaut

$$X_t = - \sum_{j \geq 1} \phi_1^{-j} Z_{t+j}. \quad (8.18)$$

Cette solution est **non causale** : elle dépend uniquement du “futur” du processus (Z_t) .

Remarquons que, comme précédemment, la solution donnée par (8.18) est obtenu en choisissant le développement fraction rationnelle $\psi(z) = (1 - \phi_1 z^{-1})^{-1}$

$$\psi(z) = \frac{1}{1 - \phi_1 z^{-1}} = \frac{-(\phi_1 z^{-1})^{-1}}{1 - (\phi_1 z^{-1})^{-1}} = -(\phi_1 z^{-1})^{-1} \sum_{k=0}^{+\infty} (\phi_1 z^{-1})^{-k} = - \sum_{k \geq 1} \phi_1^{-k} z^k,$$

qui converge sur le disque $\{z \in \mathbb{C} : |z| < |\phi_1|\}$. Nous remarquons que nous avons choisi dans les deux cas $|\phi_1| < 1$ et $|\phi_1| > 1$ les développements convergeant dans des domaines incluant le cercle unité, $\{z \in \mathbb{C} : |z| = 1\}$. Ce choix est justifié précisément dans le paragraphe 8.2.3.

Cas $|\phi_1| = 1$

Supposons qu'il existe une solution stationnaire dans ce cas alors, par stationnarité de X_t ,

$$\mathbb{E} \left[\left| X_t - \sum_{j=0}^{k-1} \phi_1^j Z_{t-j} \right|^2 \right] = |\phi_1|^{2k} \mathbb{E} [|X_{t-k}|^2] = |\phi_1|^{2k} \mathbb{E} [|X_t|^2] = \mathbb{E} [|X_t|^2].$$

Or, le terme de gauche est aussi égal à

$$\mathbb{E} [|X_t|^2] + \mathbb{E} \left[\left| \sum_{j=0}^{k-1} \phi_1^j Z_{t-j} \right|^2 \right] - 2\mathbb{E} \left[\bar{X}_t \sum_{j=0}^{k-1} \phi_1^j Z_{t-j} \right].$$

Ainsi, $\mathbb{E} \left[\left| \sum_{j=0}^{k-1} \phi_1^j Z_{t-j} \right|^2 \right] = 2\mathbb{E} \left[\bar{X}_t \sum_{j=0}^{k-1} \phi_1^j Z_{t-j} \right]$. De plus, $\mathbb{E} \left[\left| \sum_{j=0}^{k-1} \phi_1^j Z_{t-j} \right|^2 \right] = \sum_{j=0}^{k-1} |\phi_1|^{2j} \sigma^2 = k\sigma^2$. D'où, en utilisant l'inégalité de Cauchy-Schwarz,

$$k\sigma^2 \leq 2\mathbb{E} [|X_t|^2]^{1/2} \mathbb{E} \left[\left| \sum_{j=0}^{k-1} \phi_1^j Z_{t-j} \right|^2 \right]^{1/2} \leq 2(\gamma_X(0) + |\mu_X|^2)^{1/2} k^{1/2} \sigma,$$

ce qui est impossible pour k grand. Donc, dans ce cas, **il n'existe pas de solution stationnaire**.

Conclusion

Nous avons donc montré, dans le cas $p = 1$, que l'équation de récurrence (8.12) n'admettait pas de solution stationnaire lorsque $|\phi_1| = 1$ et qu'elle admettait une solution stationnaire lorsque $|\phi_1| \neq 1$, donnée par :

$$X_t = \sum_{j \geq 0} \phi_1^j Z_{t-j}, \text{ si } |\phi_1| < 1,$$

et

$$X_t = - \sum_{j \geq 1} \phi_1^{-j} Z_{t+j}, \text{ si } |\phi_1| > 1.$$

8.2.3 Cas général

Avant d'énoncer le théorème 129 qui donne une condition nécessaire et suffisante d'existence d'une solution stationnaire à l'équation récurrente (8.25) définissant un processus ARMA(p, q), nous introduisons un nouvel opérateur qui sera utile dans la preuve du théorème 129.

Soit $\mathcal{S}(\Omega, \mathcal{A}, \mathbb{P})$ l'ensemble des processus indexés par \mathbb{Z} stationnaires au second ordre et à valeurs complexes. A toute suite de coefficients complexes (α_k) vérifiant : $\sum_{k \in \mathbb{Z}} |\alpha_k| < \infty$, on associe un opérateur F_α qui à $X \in \mathcal{S}(\Omega, \mathcal{A}, \mathbb{P})$ associe le processus Y défini par :

$$F_\alpha : X \mapsto Y = (Y_t)_{t \in \mathbb{Z}} = \left(\sum_{k \in \mathbb{Z}} \alpha_k X_{t-k} \right)_{t \in \mathbb{Z}} .$$

D'après le théorème 124, Y est aussi dans $\mathcal{S}(\Omega, \mathcal{A}, \mathbb{P})$.

Le lemme 128 montre comment composer deux opérateurs de type F_α .

Lemme 128. Soient (α_k) et (β_k) des suites de coefficients complexes telles que : $\sum_{k \in \mathbb{Z}} |\alpha_k| < \infty$ et $\sum_{k \in \mathbb{Z}} |\beta_k| < \infty$. Si $X \in \mathcal{S}(\Omega, \mathcal{A}, \mathbb{P})$ alors

$$F_\alpha \circ F_\beta X = F_\beta \circ F_\alpha X = F_{\alpha * \beta} X , \text{ dans } L^2(\Omega, \mathcal{A}, \mathbb{P}) ,$$

où $(\alpha * \beta)_k = \sum_{j \in \mathbb{Z}} \alpha_j \beta_{k-j}$ est la convolution discrète des suites α et β .

Proof. Soit $Y = F_\beta X$. D'après le théorème 124, puisque $\sum_k |\beta_k| < \infty$, Y est dans $\mathcal{S}(\Omega, \mathcal{A}, \mathbb{P})$. Pour les mêmes raisons, $F_\alpha Y$ est lui aussi dans $\mathcal{S}(\Omega, \mathcal{A}, \mathbb{P})$. Posons $Z = F_\alpha[F_\beta X]$ et $W = [F_{\alpha * \beta}]X$. On a alors, pour tout $t \in \mathbb{Z}$,

$$\begin{aligned} Z_t &= \sum_{j \in \mathbb{Z}} \alpha_j Y_{t-j} & Y_t &= \sum_{k \in \mathbb{Z}} \beta_k X_{t-k} \\ W_t &= \sum_{k \in \mathbb{Z}} \left(\sum_{j \in \mathbb{Z}} \alpha_j \beta_{k-j} \right) X_{t-k} . \end{aligned}$$

Ainsi, $Z_t = \sum_{j \in \mathbb{Z}} \alpha_j (\sum_{k \in \mathbb{Z}} \beta_k X_{t-j-k})$.

Définissons $Z_{t,m,n}$ et $W_{t,m,n}$ par :

$$\begin{aligned} Z_{t,m,n} &= \sum_{j=-m}^m \alpha_j \left(\sum_{k=-n}^n \beta_k X_{t-j-k} \right) \\ W_{t,m,n} &= \sum_{k=-m}^m \left(\sum_{j=-n}^n \alpha_j \beta_{k-j} \right) X_{t-k} . \end{aligned}$$

En posant $\ell = j + k$, on en déduit que

$$Z_{t,m,n} = \sum_{\ell=-m+n}^{m+n} \left(\sum_{j=-m}^m \alpha_j \beta_{\ell-j} \right) X_{t-\ell} = W_{t,m+n,m} . \quad (8.19)$$

En notant $\|X\|_2 = (\mathbb{E}[|X|^2])^{1/2}$, nous pouvons écrire en utilisant l'inégalité triangulaire que :

$$\|Z_t - W_t\|_2 \leq \|Z_t - Z_{t,m,n}\|_2 + \|Z_{t,m,n} - W_{t,m+n,m}\|_2 + \|W_{t,m+n,m} - W_t\|_2 , \quad (8.20)$$

le deuxième terme du membre de droite de (8.20) étant nul d'après (8.19). D'autre part, avec : $Z_{t,m} = \sum_{j=-m}^m \alpha_j (\sum_{k \in \mathbb{Z}} \beta_k X_{t-j-k}) = \sum_{j=-m}^m \alpha_j Y_{t-j}$, on a :

$$\|Z_t - Z_{t,m,n}\|_2 \leq \|Z_t - Z_{t,m}\|_2 + \|Z_{t,m} - Z_{t,m,n}\|_2 . \quad (8.21)$$

En utilisant l'inégalité de Minkovsky, le fait que Y est dans $\mathcal{S}(\Omega, \mathcal{A}, \mathbb{P})$ et $\sum_{k \in \mathbb{Z}} |\alpha_k| < \infty$, on a

$$\|Z_t - Z_{t,m}\|_2 = \left\| \sum_{|j|>m} \alpha_j Y_{t-j} \right\|_2 \leq \|Y_0\| \sum_{|j|>m} |\alpha_j| \rightarrow 0 , m \rightarrow \infty . \quad (8.22)$$

D'autre part, en utilisant l'inégalité de Cauchy-Schwarz, le fait que X est dans $\mathcal{S}(\Omega, \mathcal{A}, \mathbb{P})$ et l'absolue sommabilité de (α_k) et (β_k) :

$$\sup_{m \in \mathbb{N}} \|Z_{t,m} - Z_{t,m,n}\|_2 = \sup_{m \in \mathbb{N}} \left\| \sum_{|j| \leq m} \alpha_j \left(\sum_{|k|>n} \beta_k X_{t-j-k} \right) \right\|_2 \quad (8.23)$$

$$\leq \|X_0\|_2 \sum_{j \in \mathbb{Z}} |\alpha_j| \sum_{|k|>n} |\beta_k| \rightarrow 0 , n \rightarrow \infty . \quad (8.24)$$

En utilisant (8.21), (8.22) et (8.23), on obtient que le premier terme du membre de droite de (8.20) tend vers 0 lorsque m et n tendent vers l'infini. On peut montrer en utilisant le même type d'arguments que $\|W_{t,m+n,m} - W_t\|_2$ tend vers 0 lorsque m et n tendent vers l'infini ce qui conclut la preuve avec (8.20). \square

Théorème 129 (Existence et unicité des processus ARMA(p, q)). *Soit l'équation récurrente :*

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q} , \quad (8.25)$$

où $Z_t \sim \text{B.B.}(0, \sigma^2)$ et les ϕ_j et les θ_j sont des nombres complexes. On note $\phi(z)$ et $\theta(z)$ les transformées en z

$$\phi(z) = 1 - \phi_1 z^{-1} - \cdots - \phi_p z^{-p} \quad (8.26)$$

$$\theta(z) = 1 + \theta_1 z^{-1} + \cdots + \theta_q z^{-q} . \quad (8.27)$$

On suppose que $\phi(z)$ et $\theta(z)$ n'ont pas de zéros communs. Alors l'équation (8.25) admet une solution stationnaire au second ordre si et seulement si le polynôme $\phi(z) \neq 0$ pour $|z| = 1$. Cette solution est unique et a pour expression :

$$X_t = \sum_{k=-\infty}^{\infty} \psi_k Z_{t-k} , \quad (8.28)$$

où les (ψ_k) sont donnés par les coefficients du développement

$$\frac{\theta(z)}{\phi(z)} = \sum_{k \in \mathbb{Z}} \psi_k z^{-k}, \quad (8.29)$$

convergeant dans la couronne

$$\{z \in \mathbb{C} : \delta_1 < |z| < \delta_2\}, \quad (8.30)$$

où $\delta_1 < 1$ et $\delta_2 > 1$ sont définis par

$$\delta_1 = \max\{z \in \mathbb{C}, |z| < 1, \phi(z) = 0\} \quad (8.31)$$

et

$$\delta_2 = \min\{z \in \mathbb{C}, |z| > 1, \phi(z) = 0\}, \quad (8.32)$$

avec la convention $\max(\emptyset) = 0$ et $\min(\emptyset) = \infty$.

Proof. Nous commençons par énoncer et prouver un lemme utile pour la preuve du théorème 129.

Lemme 130. Soient θ et ϕ deux polynômes à coefficients complexes tels que $\phi(z) \neq 0$ pour $|z| = 1$ et $\phi(0) = 1$ alors la fraction rationnelle $\theta(z)/\phi(z)$ est développable en série de Laurent, c'est-à-dire

$$\frac{\theta(z)}{\phi(z)} = \sum_{k \in \mathbb{Z}} c_k z^{-k},$$

où la série $\sum_{k \in \mathbb{Z}} c_k z^{-k}$ est uniformément convergente dans la couronne définie par $\{z \in \mathbb{C}, r_1 < |z| < r_2\}$, où

$$r_1 = \max\{|z| : z \in \mathbb{C}, |z| < 1, \phi(z) = 0\}$$

$$r_2 = \min\{|z| : z \in \mathbb{C}, |z| > 1, \phi(z) = 0\}.$$

avec la convention $\max(\emptyset) = 0$ et $\min(\emptyset) = \infty$.

Le cas $r_1 = 0$ correspond à $\phi(z) \neq 0$ dans le disque unité, $\{z \in \mathbb{C} : |z| \leq 1\}$.

Le cas $r_2 = \infty$ correspond à $\phi(z) \neq 0$ dans la couronne $\{z \in \mathbb{C} : |z| \geq 1\}$. Dans ce cas on a $c_k = 0$ pour tout $k > \max(-1, \deg(\theta) - \deg(\phi))$.

Proof. La décomposition en éléments simples de la fraction rationnelle $\theta(z)/\phi(z)$ s'écrit comme la somme d'un polynôme de degré $\deg(\theta) - \deg(\phi)$ (avec la convention que tout polynôme de degré strictement négatif est le polynôme nul) et de termes de la forme : $a/(z - z_0)^r$, où z_0 est une racine de ϕ de multiplicité supérieure ou égale à r et a est une constante. On écrit :

$$\text{si } |z_0| < 1, \frac{1}{(1 - z_0 z^{-1})^r} = \frac{z^{-r}}{(1 - z_0/z)^r}, \text{ lorsque } |z_0| < |z|,$$

$$\text{si } |z_0| > 1, \frac{1}{(z - z_0)^r} = \frac{(-z_0)^{-r}}{(1 - z/z_0)^r}, \text{ lorsque } |z| < |z_0|.$$

On utilise que :

$$\begin{aligned}(1-u)^{-r} &= \frac{(-1)^{r-1}}{(r-1)!} \sum_{k \geq r-1} \frac{k!}{(k-r+1)!} u^{k-r+1} \\ &= \frac{(-1)^{r-1}}{(r-1)!} \sum_{k \geq 0} \frac{(k+r-1)!}{k!} u^k, \text{ lorsque } |u| < 1,\end{aligned}$$

Ainsi,

$$\text{si } |z_0| < 1, \frac{1}{(z-z_0)^r} = \frac{z^{-r}}{(1-z_0/z)^r} = z^{-r} \frac{(-1)^{r-1}}{(r-1)!} \sum_{k \geq 0} \frac{(k+r-1)!}{k!} (z_0/z)^k,$$

qui converge si $|z_0| < |z|$,

$$\text{si } |z_0| > 1, \frac{1}{(z-z_0)^r} = \frac{(-z_0)^{-r}}{(1-z/z_0)^r} = -\frac{z_0^{-r}}{(r-1)!} \sum_{k \geq 0} \frac{(k+r-1)!}{k!} (z/z_0)^k,$$

qui converge si $|z| < |z_0|$.

En majorant $(k+r-1)!/k!$ par k^{r-1} , on en déduit que

$$\begin{aligned}\text{si } |z_0| < 1, \frac{1}{(z-z_0)^r} &= \sum_{k \leq -r} v_k z^k, \text{ qui converge si } |z| > |z_0|, \\ \text{si } |z_0| > 1, \frac{1}{(z-z_0)^r} &= \sum_{k \geq 0} w_k z^k, \text{ qui converge si } |z| < |z_0|,\end{aligned}$$

où $|v_k|$ et $|w_k|$ sont majorés par $C\eta^{|k|}$, C étant une constante strictement positive pour tout η choisi dans $(0, r_1)$ ou $(0, 1/r_2)$, respectivement.

□

Retour à la preuve du théorème 129

Supposons que $\phi(z) \neq 0$ pour $|z| = 1$, alors d'après le Lemme 130 il existe $r_1 < 1$ et $r_2 > 1$ tels que

$$\psi(z) = \frac{\theta(z)}{\phi(z)} = \sum_{k=-\infty}^{\infty} \psi_k z^{-k}, \quad r_1 < |z| < r_2, \quad (8.33)$$

où la suite $(\psi_k)_{k \in \mathbb{Z}}$ vérifie $\sum_k |\psi_k| < \infty$. Vérifions que le processus (X_t) défini par : $X_t = \sum_{k \in \mathbb{Z}} \psi_k Z_{t-k} = (\mathbf{F}_\psi Z)_t$, pour tout $t \in \mathbb{Z}$ est une solution stationnaire de (8.25). D'après la définition 125, (X_t) est stationnaire. De plus, d'après le Lemme 128,

$$\mathbf{F}_\phi \circ \mathbf{F}_\psi Z = \mathbf{F}_{\phi * \psi} Z = \mathbf{F}_\theta Z,$$

ce qui montre l'existence d'une solution stationnaire à (8.25).

D'autre part, si X est un processus stationnaire au second ordre solution de (8.25) alors X vérifie :

$$\mathbf{F}_\phi X = \mathbf{F}_\theta Z. \quad (8.34)$$

Comme $\phi(z) \neq 0$ pour $|z| = 1$, alors d'après le Lemme 130 il existe $r_1 < 1$ et $r_2 > 1$ tels que :

$$\xi(z) = \frac{1}{\phi(z)} = \sum_{k \in \mathbb{Z}} \xi_k z^k, \quad r_1 < |z| < r_2,$$

où la suite $(\xi_k)_{k \in \mathbb{Z}}$ vérifie $\sum_k |\xi_k| < \infty$. On peut donc appliquer l'opérateur F_ξ aux deux membres de l'équation (8.34) d'où l'on déduit en utilisant le lemme 128 que

$$X = F_{\xi * \theta} Z = F_\psi Z$$

où (ψ_k) est définie dans (8.33). Donc

$$X_t = \sum_{k \in \mathbb{Z}} \psi_k Z_{t-k} = (F_\psi Z)_t,$$

pour tout $t \in \mathbb{Z}$, ce qui assure l'unicité de la solution.

Réciproquement, si (X_t) est un processus stationnaire solution de (8.25) de la forme

$$X_t = \sum_{k \in \mathbb{Z}} \eta_k Z_{t-k} \quad \text{où} \quad \sum_k |\eta_k| < \infty,$$

montrons que $\phi(z) \neq 0$ pour $|z| = 1$. En effet, puisque X est solution de (8.34) alors :

$$F_\phi X = F_\phi [F_\eta Z] = F_\theta Z.$$

D'après le lemme 128,

$$F_\phi [F_\eta Z] = F_{\phi * \eta} Z = F_\theta Z.$$

Posons $\zeta_k = \sum_{j \in \mathbb{Z}} \phi_j \eta_{k-j}$. On a alors, pour tout $t \in \mathbb{Z}$,

$$\sum_{k \in \mathbb{Z}} \zeta_k Z_{t-k} = \sum_{j=1}^q \theta_j Z_{t-j}.$$

En multipliant les deux membres de cette équation par $Z_{t-\ell}$ et en prenant l'espérance, on déduit que $\zeta_\ell = \theta_\ell$, $\ell = 0, \dots, q$ et $\zeta_\ell = 0$, sinon. Ainsi, $\theta(z) = \phi(z)\eta(z)$, $|z| = 1$. Puisque θ et ϕ n'ont pas de racines communes et que $|\eta(z)| \leq \sum_{k \in \mathbb{Z}} |\eta_k| < \infty$, si $|z| = 1$, $\phi(z)$ ne s'annule pas sur le cercle unité : $\{z, |z| = 1\}$, ce qui conclut la preuve du théorème 129. \square

Dans le cas où $\phi(z)$ et $\theta(z)$ ont des zéros communs, deux configurations sont possibles :

- (a) Les zéros communs ne sont pas sur le cercle unité. Dans ce cas on se ramène au cas sans zéro commun en annulant les facteurs communs.
- (b) Certains des zéros communs se trouvent sur le cercle unité. L'équation (8.25) admet une infinité de solutions stationnaires au second ordre.

Du point de vue de la modélisation, la présence de zéros communs ne présente aucun intérêt puisqu'elle est sans influence sur la densité spectrale de puissance. Elle conduit de plus à une ambiguïté sur l'ordre réel des parties AR et MA.

ARMA(p, q) causal

Le théorème 132 donne une condition nécessaire et suffisante d'existence d'une solution causale à l'équation (8.25).

Définition 131 (Représentation ARMA causale). *Sous les hypothèses du théorème 129, on dira que l'équation (8.25) fournit une représentation causale de la solution stationnaire au second ordre (X_t) si (X_t) $_{t \in \mathbb{Z}}$ est un processus linéaire causal par rapport à (Z_t) $_{t \in \mathbb{Z}}$.*

Théorème 132 (ARMA(p, q) causal). *Soit l'équation récurrente :*

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q} \quad (8.35)$$

où $Z_t \sim \text{B.B.}(0, \sigma^2)$ et $\{\phi_j\}_{j=1}^p$ et $\{\theta_j\}_{j=1}^q$ sont des nombres complexes. On pose $\phi(z) = 1 - \phi_1 z^{-1} - \cdots - \phi_p z^{-p}$ et $\theta(z) = 1 + \theta_1 z^{-1} + \cdots + \theta_q z^{-q}$. On suppose que $\phi(z)$ et $\theta(z)$ n'ont pas de zéros communs. Alors l'équation (8.35) fournit une représentation causale de la solution stationnaire au second ordre si et seulement si le polynôme $\phi(z) \neq 0$ pour $|z| \geq 1$. Cette solution a pour expression :

$$X_t = \sum_{k \geq 0} \psi_k Z_{t-k} \quad (8.36)$$

où la suite (ψ_k) est donnée par les coefficients du développement

$$\frac{\theta(z)}{\phi(z)} = \sum_{k=0}^{\infty} \psi_k z^{-k}$$

qui converge dans la couronne $\{z \in \mathbb{C}, |z| \geq 1\}$.

Proof. Le théorème 129 montre l'existence et l'unicité de la solution de l'équation (8.35) et vérifie

$$X_t = \sum_{k \in \mathbb{Z}} \psi_k Z_{t-k}$$

où la suite (ψ_k) est caractérisée par l'équation

$$\frac{\theta(z)}{\phi(z)} = \sum_{k=0}^{\infty} \psi_k z^{-k}, \quad z \in \mathbb{C}, |z| = 1.$$

Si maintenant $\phi(z) \neq 0$ pour $|z| \leq 1$, alors, d'après le lemme 130, comme $r_1 = 0$, on a $\psi_k = 0$ pour $k < 0$ et (8.36) suit.

Réciproquement, si X est une solution stationnaire de (8.35) causale alors

$$X_t = \sum_{k \geq 0} \eta_k Z_{t-k} \quad \text{où} \quad \sum_k |\eta_k| < \infty.$$

Montrons que $\phi(z) \neq 0$ pour $|z| \leq 1$. En effet, puisque X est solution de (8.35), alors :

$$F_\phi X = F_\phi [F_\eta Z] = F_\theta Z.$$

D'après le lemme 128, nous avons

$$F_{\phi * \eta} Z = F_\theta Z.$$

Posons, pour tout $k \in \mathbb{N}$, $\zeta_k = \sum_{j=0}^k \phi_j \eta_{k-j}$. On a alors, pour tout $t \in \mathbb{Z}$,

$$\sum_{k=0}^{\infty} \zeta_k Z_{t-k} = \sum_{j=0}^q \theta_j Z_{t-j},$$

avec la convention $\theta_0 = 1$. En multipliant les deux membres de cette équation par $Z_{t-\ell}$ et en prenant l'espérance, on déduit que $\zeta_\ell = \theta_\ell$, $\ell = 0, \dots, q$ et $\zeta_\ell = 0$, sinon. Ainsi, $\theta(z) = \phi(z)\eta(z)$, $|z| \leq 1$. Puisque θ et ϕ n'ont pas de racines communes et que $|\eta(z)| \leq \sum_{k \geq 0} |\eta_k| < \infty$, si $|z| \leq 1$, $\phi(z)$ ne s'annule pas sur le disque unité : $\{z, |z| \leq 1\}$, ce qui conclut la preuve du théorème 132. \square

Définition 133 (Représentation ARMA inversible). *Sous les hypothèses du théorème 129, on dira que l'équation (8.25) fournit une représentation inversible de la solution stationnaire au second ordre (X_t) si $(X_t)_{t \in \mathbb{Z}}$ est un processus linéaire inversible par rapport à $(Z_t)_{t \in \mathbb{Z}}$.*

Théorème 134 (ARMA(p, q) inversible). *Soit l'équation récurrente :*

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q} \quad (8.37)$$

où $Z_t \sim \text{B.B.}(0, \sigma^2)$ et les ϕ_j et les θ_j sont des nombres complexes. On pose $\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$ et $\theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q$. On suppose que $\phi(z)$ et $\theta(z)$ n'ont pas de zéros communs. Alors l'équation (8.37) fournit une représentation inversible de la solution stationnaire au second ordre si et seulement si le polynôme $\theta(z) \neq 0$ pour $|z| \geq 1$. Cette solution est unique et a pour expression :

$$Z_t = \sum_{k \geq 0} \pi_k X_{t-k} \quad (8.38)$$

où la suite (π_k) est donnée par les coefficients du développement

$$\frac{\phi(z)}{\theta(z)} = \sum_{k=0}^{\infty} \pi_k z^{-k}$$

qui converge dans $\{z \in \mathbb{C}, |z| \geq 1\}$.

La preuve de ce théorème est tout à fait analogue à celle du théorème 132 et n'est donc pas détaillée ici.

Un modèle ARMA(p, q) est causal et inversible lorsque les racines des polynômes $\phi(z)$ et $\theta(z)$ sont toutes situées à l'intérieur du disque unité. Dans ce cas, X_t et Z_t se déduisent mutuellement l'un de l'autre par des opérations de filtrage causal.

Calcul des covariances d'un processus ARMA(p, q) causal

Une première méthode consiste à utiliser l'expression (8.8) où la suite (ψ_k) se détermine de façon récurrente à partir de l'égalité $\psi(z)\theta(z) = \phi(z)$ par identification du terme en z^k . Pour les premiers termes on trouve :

$$\begin{aligned}\psi_0 &= 1 \\ \psi_1 &= \theta_1 + \psi_0\phi_1 \\ \psi_2 &= \theta_2 + \psi_0\phi_2 + \psi_1\phi_1 \\ &\dots\end{aligned}$$

La seconde méthode utilise une formule de récurrence, vérifiée par la fonction d'autocovariance d'un processus ARMA(p, q), qui s'obtient en multipliant les deux membres de (8.25) par \bar{X}_{t-k} et en prenant l'espérance. On obtient :

$$\gamma(k) - \phi_1\gamma(k-1) - \cdots - \phi_p\gamma(k-p) = \sigma^2 \sum_{k \leq j \leq q} \theta_j \bar{\psi}_{j-k}, \quad 0 \leq k < \max(p, q+1) \quad (8.39)$$

$$\gamma(k) - \phi_1\gamma(k-1) - \cdots - \phi_p\gamma(k-p) = 0, \quad k \geq \max(p, q+1) \quad (8.40)$$

où nous avons utilisé la causalité du processus pour écrire que $\mathbb{E}[Z_t \bar{X}_{t-k}] = 0$ pour tout $k \geq 1$. Le calcul de la suite $\{\psi_k\}$ pour $k = 1, \dots, p$ se fait comme précédemment. En reportant ces valeurs dans (8.39) pour $0 \leq k \leq p$, on obtient $(p+1)$ équations linéaires aux $(p+1)$ inconnues $(\gamma(0), \dots, \gamma(p))$ que l'on peut résoudre. Pour déterminer les valeurs suivantes on utilise l'expression (8.40).

Densité spectrale d'un processus ARMA(p, q)

Théorème 135 (Densité spectrale d'un processus ARMA(p, q)). *Soit (X_t) un processus ARMA(p, q) (pas nécessairement causal ou inversible) i.e. la solution stationnaire de l'équation (8.25) où les polynômes $\theta(z)$ et $\phi(z)$ sont des polynômes de degré q et p n'ayant pas de zéros communs. Alors (X_t) possède une densité spectrale qui a pour expression :*

$$f(\lambda) = \frac{\sigma^2}{2\pi} \frac{|1 + \sum_{k=1}^q \theta_k e^{-ik\lambda}|^2}{|1 - \sum_{k=1}^p \phi_k e^{-ik\lambda}|^2}, \quad -\pi \leq \lambda \leq \pi. \quad (8.41)$$

Remarque 136. *D'après le théorème 129, l'expression de f est bien définie puisque ϕ ne s'annule pas sur le cercle unité.*

Chapitre 9

Prédiction des signaux aléatoires

9.1 Prédiction linéaire de processus stationnaires

Soit $(X_t)_{t \in \mathbb{Z}}$ un processus stationnaire au second ordre à valeurs réelles, **d'espérance nulle** et de fonction d'autocovariance $\gamma(h) = \text{cov}(X_h, X_0)$. On cherche à *prédirer* la valeur du processus à la date t à partir d'une combinaison linéaire des p derniers échantillons du passé X_{t-1}, \dots, X_{t-p} . La meilleure combinaison linéaire (*i.e.* le prédicteur linéaire optimal) est la projection orthogonale de X_t sur $\mathcal{H}_{t-1,p}$ notée $\text{proj}(X_t | \mathcal{H}_{t-1,p})$, où $\mathcal{H}_{t-1,p}$ est défini par :

$$\mathcal{H}_{t-1,p} = \text{Vect}(X_{t-1}, X_{t-2}, \dots, X_{t-p}). \quad (9.1)$$

Les indices dans la notation $\mathcal{H}_{t-1,p}$ doivent être compris ainsi : $\mathcal{H}_{t-1,p}$ est le sous-espace vectoriel engendré par les p observations précédant X_{t-1} à savoir X_{t-1}, \dots, X_{t-p} . D'après le théorème 196,

$$\text{proj}(X_t | \mathcal{H}_{t-1,p}) = \sum_{k=1}^p \phi_{k,p} X_{t-k}, \quad (9.2)$$

où les coefficients $(\phi_{k,p})_{1 \leq k \leq p}$ satisfont

$$\left\langle X_t - \sum_{k=1}^p \phi_{k,p} X_{t-k}, X_{t-j} \right\rangle = 0, \quad j = 1, \dots, p, \quad (9.3)$$

la notation $\langle \cdot, \cdot \rangle$ correspondant au produit scalaire dans $L_2(\Omega, \mathcal{A}, \mathbb{P})$ défini pour X et Y dans $L_2(\Omega, \mathcal{A}, \mathbb{P})$ par $\langle X, Y \rangle = \mathbb{E}[XY]$. L'équation (9.3) se réécrit encore sous la forme

$$\langle X_t, X_{t-j} \rangle = \sum_{k=1}^p \phi_{k,p} \langle X_{t-k}, X_{t-j} \rangle, \quad j = 1, \dots, p, \quad (9.4)$$

soit encore

$$\sum_{k=1}^p \phi_{k,p} \gamma(k-j) = \gamma(j), \quad j = 1, \dots, p. \quad (9.5)$$

En posant Γ_p la matrice de covariance du vecteur $(X_{t-1}, \dots, X_{t-p})$ définie par

$$\Gamma_p = \begin{bmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(p-1) \\ \gamma(1) & \gamma(0) & \gamma(1) & \vdots \\ \vdots & \ddots & \ddots & \ddots \\ \vdots & & & \gamma(1) \\ \gamma(p-1) & \gamma(p-2) & \cdots & \gamma(1) & \gamma(0) \end{bmatrix},$$

on peut réécrire (9.5) comme suit :

$$\Gamma_p \phi_p = \gamma_p, \quad (9.6)$$

où $\phi_p = (\phi_{1,p}, \dots, \phi_{p,p})^T$ et $\gamma_p = (\gamma(1), \gamma(2), \dots, \gamma(p))^T$.

Définition 137. Nous appellerons dans la suite erreur de prédition directe d'ordre p ou innovation partielle d'ordre p le processus :

$$\varepsilon_{t,p}^+ = X_t - \text{proj}(X_t | \mathcal{H}_{t-1,p}) = X_t - \sum_{k=1}^p \phi_{k,p} X_{t-k}. \quad (9.7)$$

La variance de l'erreur de prédition directe d'ordre p est notée σ_p^2 et définie par

$$\sigma_p^2 = \|X_t - \text{proj}(X_t | \mathcal{H}_{t-1,p})\|^2 = \mathbb{E}[\|X_t - \text{proj}(X_t | \mathcal{H}_{t-1,p})\|^2]. \quad (9.8)$$

D'après (9.2) et la proposition 198, la variance de l'erreur de prédition directe d'ordre p a pour expression :

$$\sigma_p^2 = \langle X_t, X_t - \text{proj}(X_t | \mathcal{H}_{t-1,p}) \rangle = \gamma(0) - \sum_{k=1}^p \phi_{k,p} \gamma(k) = \gamma(0) - \phi_p^T \gamma_p. \quad (9.9)$$

Les équations (9.6) et (9.9) sont appelées les *équations de Yule-Walker*.

Notons que (9.6) a une unique solution si et seulement si la matrice Γ_p est inversible auquel cas la solution vaut :

$$\phi_p = \Gamma_p^{-1} \gamma_p. \quad (9.10)$$

La proposition 122 fournit les conditions suffisantes assurant que Γ_p est inversible pour tout p . On a ainsi des conditions sous lesquelles on peut calculer le prédicteur de X_t à partir de X_{t-1}, \dots, X_{t-p} .

Exemple 138 (Cas d'un processus AR(m) causal). Soit (X_t) le processus AR(m) causal solution de l'équation récurrente :

$$X_t = \phi_1 X_{t-1} + \dots + \phi_m X_{t-m} + Z_t, \quad (9.11)$$

où $Z_t \sim \text{B.B.}(0, \sigma^2)$ et où $\phi(z) = 1 - \sum_{k=1}^m \phi_k z^k \neq 0$ lorsque $|z| \leq 1$. Dans ce cas, pour tout $p \geq m$:

$$\phi_{k,p} = \begin{cases} \phi_k, & \text{lorsque } 1 \leq k \leq m, \\ 0, & \text{lorsque } m < k \leq p. \end{cases}$$

En effet, (X_t) étant causal on a, pour tout $h \geq 1$, $\mathbb{E}[Z_t X_{t-h}] = 0$ et donc, d'après (9.11), $\mathbb{E}[(X_t - \sum_{k=1}^m \phi_k X_{t-k}) X_{t-h}] = 0$. Ainsi, pour tout $p \geq m$, $\sum_{k=1}^m \phi_k X_{t-k} \in \mathcal{H}_{t-1,p}$ et $(X_t - \sum_{k=1}^m \phi_k X_{t-k}) \perp \mathcal{H}_{t-1,p}$ et donc, d'après le théorème 196, pour tout $p \geq m$,

$$\sum_{k=1}^m \phi_k X_{t-k} = \text{proj}(X_t | \mathcal{H}_{t-1,p}).$$

Les coefficients de prédition d'un processus stationnaire au second ordre fournissent une décomposition particulière de la matrice de covariance Γ_{p+1} sous la forme d'un produit de matrices triangulaires explicitée dans le théorème 139.

Théorème 139. Soit (X_t) un processus stationnaire au second ordre, centré, de fonction d'autocovariance $\gamma(h)$. On note :

$$A_{p+1} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -\phi_{1,1} & 1 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & 0 \\ -\phi_{p,p} & -\phi_{p-1,p} & \cdots & -\phi_{1,p} & 1 \end{bmatrix} \text{ et } D_{p+1} = \begin{bmatrix} \sigma_0^2 & 0 & \cdots & 0 \\ 0 & \sigma_1^2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & \sigma_p^2 \end{bmatrix},$$

où les coefficients $(\phi_{k,p})_{1 \leq k \leq p}$ et $(\sigma_k^2)_{1 \leq k \leq p}$ sont respectivement définis dans (9.2) et (9.8). On a alors :

$$\Gamma_{p+1} = A_{p+1}^{-1} D_{p+1} (A_{p+1}^T)^{-1}. \quad (9.12)$$

Proof. Pour simplifier les notations, posons $\mathcal{H}_k = \mathcal{H}_{k,k} = \text{Vect}(X_k, \dots, X_1)$ et montrons tout d'abord que, pour $k \neq \ell$, nous avons :

$$\langle X_k - \text{proj}(X_k | \mathcal{H}_{k-1}), X_\ell - \text{proj}(X_\ell | \mathcal{H}_{\ell-1}) \rangle = 0. \quad (9.13)$$

En effet, pour $k < \ell$, on a $X_k - \text{proj}(X_k | \mathcal{H}_{k-1}) \in \mathcal{H}_k \subseteq \mathcal{H}_{\ell-1}$ et $X_\ell - \text{proj}(X_\ell | \mathcal{H}_{\ell-1}) \perp \mathcal{H}_{\ell-1}$. D'autre part, si on note \mathbf{X}_{p+1} le vecteur : $(X_1, \dots, X_{p+1})^T$, alors, par définition des coefficients de prédiction (9.2), on peut écrire :

$$A_{p+1} \mathbf{X}_{p+1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -\phi_{1,1} & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ -\phi_{p,p} & -\phi_{p-1,p} & \cdots & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{p+1} \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 - \text{proj}(X_2 | \mathcal{H}_1) \\ \vdots \\ X_{p+1} - \text{proj}(X_{p+1} | \mathcal{H}_p) \end{bmatrix},$$

qui donne :

$$\mathbb{E}[A_{p+1} \mathbf{X}_{p+1} \mathbf{X}_{p+1}^T A_{p+1}^T] = D_{p+1},$$

d'après (9.13) et (9.8). Par ailleurs,

$$\mathbb{E}[A_{p+1} \mathbf{X}_{p+1} \mathbf{X}_{p+1}^T A_{p+1}^T] = A_{p+1} \Gamma_{p+1} A_{p+1}^T,$$

ce qui démontre (9.12) puisque la matrice A_{p+1} est inversible, son déterminant étant égal à 1. \square

D'après l'équation (9.12) lorsque la matrice Γ_{p+1} est inversible, la variance $\sigma_p^2 = \|\varepsilon_{t,p}^+\|^2$ est strictement positive. D'autre part, la suite σ_p^2 est décroissante. En effet, par définition de $\mathcal{H}_{t-1,p}$, $\mathcal{H}_{t-1,p}$ est inclus dans $\mathcal{H}_{t-1,p+1}$ donc $\text{proj}(X_{p+1} | \mathcal{H}_{t-1,p})$ est dans $\mathcal{H}_{t-1,p+1}$. On déduit donc du théorème 196 que $\sigma_{p+1}^2 \leq \sigma_p^2$. La suite (σ_p^2) étant décroissante et minorée, elle possède donc une limite quand p tend vers l'infini.

Définition 140 (Processus régulier/déterministe). Soit $(X_t)_{t \in \mathbb{Z}}$ un processus aléatoire stationnaire au second ordre. On note $\sigma^2 = \lim_{p \rightarrow \infty} \sigma_p^2$ où σ_p^2 est la variance de l'innovation partielle d'ordre p . On dit que le processus (X_t) est régulier si $\sigma^2 > 0$ et déterministe si $\sigma^2 = 0$.

Par ailleurs, nous pouvons remarquer que le problème de la recherche des coefficients de prédiction pour un processus stationnaire au second ordre se ramène à celui de la minimisation de l'intégrale :

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |\psi(e^{-i\lambda})|^2 v_X(d\lambda)$$

sur l'ensemble \mathcal{P}_p des polynômes à coefficients réels de degré p de la forme $\psi(z) = 1 + \psi_1 z + \dots + \psi_p z^p$. En effet, en utilisant la relation (8.4) de filtrage des mesures spectrales, on peut écrire que la variance de $\|\epsilon_{t,p}^+\|^2$, qui minimise l'erreur de prédiction, a pour expression :

$$\sigma_p^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |\phi_p(e^{-i\lambda})|^2 v_X(d\lambda) \quad (9.14)$$

où :

$$\phi_p(z) = 1 - \sum_{k=1}^p \phi_{k,p} z^{-k}$$

désigne le *polynôme prédicteur d'ordre p*.

Théorème 141. Si $\{X_t\}$ est un processus régulier, alors, pour tout p , $\phi_p(z) \neq 0$ pour $|z| \leq 1$. Tous les zéros des polynômes prédicteurs sont à l'extérieur du cercle unité.

Preuve du théorème 141. Nous allons tout d'abord montrer que le prédicteur optimal n'a pas de racines sur le cercle unité. Raisonnons par contradiction. Supposons que le polynôme $\phi_p(z)$ ait deux racines complexes conjuguées, de la forme $\exp(\pm i\theta)$, sur le cercle unité (on traite de façon similaire le cas de racines réelles, $\theta = 0$ ou π). Nous pouvons écrire :

$$\phi_p(z) = \phi_p^*(z)(1 - 2\cos(\theta)z + z^2)$$

On note $\bar{v}_X(d\lambda) = v_X(d\lambda)|\phi_p^*(e^{-i\lambda})|^2$. \bar{v}_X est une mesure positive sur $[-\pi, \pi]$ de masse finie. On note :

$$\bar{\gamma}(\tau) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i\tau\lambda} \bar{v}_X(d\lambda).$$

Nous avons donc :

$$\begin{aligned} \sigma_p^2 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} (1 - 2\cos(\theta)e^{-i\lambda} + e^{-2i\lambda}) \bar{v}_X(d\lambda) \\ &= \inf_{\psi \in \mathcal{P}_2} \frac{1}{2\pi} \int_{-\pi}^{\pi} |1 + \psi_1 e^{-i\lambda} + \psi_2 e^{-2i\lambda}|^2 \bar{v}_X(d\lambda). \end{aligned}$$

La minimisation de σ_p^2 est équivalente à la résolution des équations de Yule-Walker à l'ordre $p = 2$ pour la suite des covariances $\bar{\gamma}(h)$. Par conséquent la suite des coefficients $\{1, -2\cos(\theta), 1\}$ doit vérifier l'équation :

$$\begin{bmatrix} \bar{\gamma}(0) & \bar{\gamma}(1) & \bar{\gamma}(2) \\ \bar{\gamma}(1) & \bar{\gamma}(0) & \bar{\gamma}(1) \\ \bar{\gamma}(2) & \bar{\gamma}(1) & \bar{\gamma}(0) \end{bmatrix} \begin{bmatrix} 1 \\ -2\cos(\theta) \\ 1 \end{bmatrix} = \begin{bmatrix} \sigma_p^2 \\ 0 \\ 0 \end{bmatrix}$$

De cette équation il s'en suit (les première et troisième lignes sont égales) que $\sigma_p^2 = 0$, ce qui est contraire à l'hypothèse que le processus est régulier.

Démontrons maintenant que les racines des polynômes prédicteurs sont toutes *strictement à l'intérieur du disque unité*.

Raisonnons encore par l'absurde. Supposons que le polynôme prédicteur à l'ordre p ait m racines $\{a_k, |a_k| < 1, 1 \leq k \leq m\}$ à l'intérieur du cercle unité et $(p-m)$ racines $\{b_\ell, |b_\ell| > 1, 1 \leq \ell \leq p-m\}$ à l'extérieur du cercle unité. Le polynôme prédicteur à l'ordre p s'écrit donc :

$$\phi_p(z) = \prod_{k=1}^m (1 - a_k z^{-1}) \prod_{\ell=1}^{p-m} (1 - b_\ell z^{-1}).$$

Considérons alors le polynôme :

$$\bar{\phi}_p(z) = \prod_{k=1}^m (1 - a_k z^{-1}) \prod_{\ell=1}^{p-m} (1 - (1/b_\ell^*) z^{-1}).$$

Il a d'une part toutes ses racines strictement à l'intérieur du cercle unité et d'autre part il vérifie $|\bar{\phi}_p(e^{-i\lambda})|^2 < |\phi_p(e^{-i\lambda})|^2$. On a en effet $|1 - (1/b_\ell^*)e^{-i\lambda}| = |b_\ell| |1 - b_\ell e^{-i\lambda}|$ et donc $|\bar{\phi}_p(e^{-i\lambda})|^2 = \left(\prod_{\ell=1}^{p-m} |b_\ell|^{-2} \right) |\phi_p(e^{-i\lambda})|^2$, ce qui démontre le résultat annoncé puisque $|b_\ell| > 1$, $\ell \in \{1, \dots, p-m\}$. On en déduit alors que :

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} |\bar{\phi}_p(e^{-i\lambda})|^2 v_X(d\lambda) < \sigma_p^2,$$

ce qui contredit que ϕ_p est le prédicteur linéaire optimal, i.e. minimise la variance de l'erreur de prédiction

$$\phi_p = \inf_{\psi \in \mathcal{P}_p} (2\pi)^{-1} \int_{-\pi}^{\pi} |\psi(e^{-i\lambda})|^2 v_X(d\lambda).$$

□

Une conséquence directe du théorème 141 est qu'à toute matrice de covariance de type défini positif, de dimension $(p+1) \times (p+1)$, on peut associer un processus AR(p) causal dont les $(p+1)$ premiers coefficients de covariance sont précisément la première ligne de cette matrice. Ce résultat n'est pas général. Ainsi il existe bien un processus AR(2) causal ayant $\gamma(0) = 1$ et $\gamma(1) = \rho$, comme premiers coefficients de covariance, à condition toutefois que la matrice de covariance soit positive c'est-à-dire que $|\rho| < 1$, tandis qu'il n'existe pas, pour cette même matrice de processus MA(1). Il faut en effet, en plus du caractère positif, que $|\rho| \leq 1/2$ (voir exemple 114).

9.2 Algorithme de Levinson-Durbin

La solution directe du système des équations de Yule-Walker requiert de l'ordre de p^3 opérations : la résolution classique de ce système implique en effet la décomposition

de la matrice Γ_p sous la forme du produit d'une matrice triangulaire inférieure et de sa transposée, $\Gamma_p = L_p L_p^T$ (décomposition de Choleski) et la résolution par substitution de deux systèmes triangulaires. Cette procédure peut s'avérer coûteuse lorsque l'ordre de prédiction est grand (on utilise généralement des ordres de prédiction de l'ordre de quelques dizaines à quelques centaines), ou lorsque, à des fins de modélisation, on est amené à évaluer la qualité de prédiction pour différents horizons de prédiction. L'algorithme de Levinson-Durbin exploite la structure géométrique particulière des processus stationnaires au second ordre pour établir une formule de récurrence donnant les coefficients de prédiction à l'ordre $(p+1)$ à partir des coefficients de prédiction obtenus à l'ordre p . Il fournit également une relation de récurrence entre l'erreur de prédiction directe à l'ordre $p+1$ et l'erreur de prédiction directe à l'ordre p .

On supposera dans toute cette partie que Γ_p est inversible pour tout $p \geq 1$.

Supposons que les coefficients de prédiction linéaire et la variance de l'erreur de prédiction directe à l'ordre p , pour $p \geq 0$, sont connus :

$$\text{proj}(X_t | \mathcal{H}_{t-1,p}) = \sum_{k=1}^p \phi_{k,p} X_{t-k} \quad \text{et} \quad \sigma_p^2 = \|X_t - \text{proj}(X_t | \mathcal{H}_{t-1,p})\|^2,$$

et déterminons, à partir de la projection à l'ordre p de X_t , la projection de X_t à l'ordre $p+1$ sur le sous-espace $\mathcal{H}_{t-1,p+1} = \text{Vect}(X_{t-1}, \dots, X_{t-p-1})$.

Pour cela, on décompose cet espace en somme orthogonale de la façon suivante :

$$\begin{aligned} \mathcal{H}_{t-1,p+1} &= \mathcal{H}_{t-1,p} \overset{\perp}{\oplus} \text{Vect}(X_{t-p-1} - \text{proj}(X_{t-p-1} | \mathcal{H}_{t-1,p})) \\ &= \mathcal{H}_{t-1,p} \overset{\perp}{\oplus} \text{Vect}(\varepsilon_{t-p-1,p}^-), \end{aligned}$$

où, de façon générale, $\varepsilon_{t,p}^-$ correspond à l'*erreur de prédiction rétrograde à l'ordre p* définie par :

$$\varepsilon_{t,p}^- = X_t - \text{proj}(X_t | \mathcal{H}_{t+p,p}) = X_t - \text{proj}(X_t | \text{Vect}(X_{t+p}, \dots, X_{t+1})).$$

Elle représente la différence entre la valeur à l'instant courant X_t et la projection orthogonale de X_t sur les p échantillons qui suivent l'instant courant $\{X_{t+1}, \dots, X_{t+p}\}$. Le qualificatif *rétrograde* est clair : il traduit le fait que l'on cherche à prédire la valeur courante en fonction des valeurs futures.

D'après la proposition 198,

$$\text{proj}(X_t | \mathcal{H}_{t-1,p+1}) = \text{proj}(X_t | \mathcal{H}_{t-1,p}) + \text{proj}(X_t | \text{Vect}(\varepsilon_{t-p-1,p}^-)),$$

où d'après l'exemple 197 :

$$\text{proj}(X_t | \text{Vect}(\varepsilon_{t-p-1,p}^-)) = \alpha \varepsilon_{t-p-1,p}^- \quad \text{avec} \quad \alpha = \langle X_t, \varepsilon_{t-p-1,p}^- \rangle / \|\varepsilon_{t-p-1,p}^-\|^2.$$

On en déduit donc que :

$$\begin{aligned} \text{proj}(X_t | \mathcal{H}_{t-1,p+1}) &= \text{proj}(X_t | \mathcal{H}_{t-1,p}) \\ &\quad + k_{p+1} [X_{t-p-1} - \text{proj}(X_{t-p-1} | \mathcal{H}_{t-1,p})], \end{aligned} \quad (9.15)$$

où

$$k_{p+1} = \frac{\langle X_t, \varepsilon_{t-p-1,p}^- \rangle}{\|\varepsilon_{t-p-1,p}^-\|^2}. \quad (9.16)$$

Montrons à présent que les coefficients de prédiction rétrograde coïncident avec les coefficients de prédiction directe. Plus précisément, si

$$\text{proj}(X_t | \mathcal{H}_{t-1,p}) = \sum_{k=1}^p \phi_{k,p} X_{t-k}, \quad (9.17)$$

alors

$$\text{proj}(X_{t-p-1} | \mathcal{H}_{t-1,p}) = \sum_{k=1}^p \phi_{k,p} X_{t-p-1+k} = \sum_{k=1}^p \phi_{p+1-k,p} X_{t-k}. \quad (9.18)$$

En effet, les coefficients des deux développements (9.17) et (9.18) sont tous les deux donnés par (9.10). En utilisant (9.17) et (9.18) dans (9.15), on a :

$$\begin{aligned} \text{proj}(X_t | \mathcal{H}_{t-1,p+1}) &= \sum_{k=1}^{p+1} \phi_{k,p+1} X_{t-k} \\ &= \sum_{k=1}^p (\phi_{k,p} - k_{p+1} \phi_{p+1-k,p}) X_{t-k} + k_{p+1} X_{t-p-1}. \end{aligned}$$

On en déduit, par unicité, les formules de récurrence donnant les coefficients de prédiction à l'ordre $p+1$ à partir de ceux à l'ordre p :

$$\begin{cases} \phi_{k,p+1} = \phi_{k,p} - k_{p+1} \phi_{p+1-k,p}, & \text{pour } k \in \{1, \dots, p\}, \\ \phi_{p+1,p+1} = k_{p+1}. \end{cases} \quad (9.19)$$

Explicitons à présent la relation (9.16) définissant k_{p+1} . En utilisant (9.18), on a :

$$\begin{aligned} \langle X_t, \varepsilon_{t-p-1,p}^- \rangle &= \langle X_t, X_{t-p-1} - \text{proj}(X_{t-p-1} | \mathcal{H}_{t-1,p}) \rangle \\ &= \gamma(p+1) - \left\langle X_t, \sum_{k=1}^p \phi_{k,p} X_{t-p-1+k} \right\rangle = \gamma(p+1) - \sum_{k=1}^p \phi_{k,p} \gamma(p+1-k). \end{aligned}$$

D'autre part,

$$\begin{aligned} \|\varepsilon_{t-p-1,p}^-\|^2 &= \left\langle X_{t-p-1}, X_{t-p-1} - \sum_{k=1}^p \phi_{k,p} X_{t-p-1+k} \right\rangle \\ &= \gamma(0) - \sum_{k=1}^p \phi_{k,p} \gamma(k) = \sigma_p^2 = \|\varepsilon_{t,p}^+\|^2, \quad (9.20) \end{aligned}$$

ce qui donne

$$k_{p+1} = \frac{\gamma(p+1) - \sum_{k=1}^p \phi_{k,p} \gamma(p+1-k)}{\sigma_p^2}. \quad (9.21)$$

Il nous reste maintenant à déterminer l'erreur de prédition σ_{p+1}^2 à l'ordre $(p+1)$ en fonction de σ_p^2 . En utilisant l'équation (9.15), on a

$$\begin{aligned}\boldsymbol{\varepsilon}_{t,p+1}^+ &= X_t - \text{proj}(X_t | \mathcal{H}_{t-1,p+1}) \\ &= X_t - \text{proj}(X_t | \mathcal{H}_{t-1,p}) - k_{p+1}[X_{t-p-1} - \text{proj}(X_{t-p-1} | \mathcal{H}_{t-1,p})] \\ &= X_t - \text{proj}(X_t | \mathcal{H}_{t-1,p}) - k_{p+1}\boldsymbol{\varepsilon}_{t-p-1,p}^-,\end{aligned}$$

dont on déduit d'après (9.20) :

$$\sigma_{p+1}^2 = \|\boldsymbol{\varepsilon}_{t,p+1}^+\|^2 = \sigma_p^2 + k_{p+1}^2\sigma_p^2 - 2k_{p+1}\left\langle X_t - \text{proj}(X_t | \mathcal{H}_{t-1,p}), \boldsymbol{\varepsilon}_{t-p-1,p}^- \right\rangle.$$

En utilisant que $\text{proj}(X_t | \mathcal{H}_{t-1,p})$ et $\boldsymbol{\varepsilon}_{t-p-1,p}^-$ sont orthogonaux, (9.16) et (9.20), on obtient

$$\sigma_{p+1}^2 = \sigma_p^2(1 - k_{p+1}^2). \quad (9.22)$$

A partir de ces récursions, nous allons à présent décrire l'algorithme de Levinson-Durbin.

Pour initialiser l'algorithme, nous nous intéressons au cas $p = 0$. Dans ce cas, la meilleure prédition de X_t est $\mathbb{E}[X_t] = 0$ et la variance de l'erreur de prédition est donnée par $\sigma_0^2 = \mathbb{E}[(X_t - 0)^2] = \gamma(0)$. Au pas suivant on a $k_1 = \gamma(1)/\gamma(0)$, en posant $p = 0$ dans (9.21), $\phi_{1,1} = \gamma(1)/\gamma(0)$, en posant $p = 0$ dans (9.19) et $\sigma_1^2 = \gamma(0)(1 - k_1^2)$, en posant $p = 0$ dans (9.22).

L'algorithme de *Levinson-Durbin* qui permet de déterminer les coefficients de prédition $\{\phi_{m,p}\}_{1 \leq m \leq p, 1 \leq p \leq K}$ à partir de $\gamma(0), \dots, \gamma(K)$ s'écrit alors de la façon suivante :

Algorithm 142 (Levinson-Durbin).

Initialisation $k_1 = \gamma(1)/\gamma(0)$, $\phi_{1,1} = \gamma(1)/\gamma(0)$ et $\sigma_1^2 = \gamma(0)(1 - k_1^2)$

Récursion Pour $p = \{2, \dots, K\}$ répéter :

(a) Calculer

$$\begin{aligned}k_p &= \sigma_{p-1}^{-2} \left(\gamma(p) - \sum_{k=1}^{p-1} \phi_{k,p-1} \gamma(p-k) \right) \\ \phi_{p,p} &= k_p \\ \sigma_p^2 &= \sigma_{p-1}^2(1 - k_p^2)\end{aligned}$$

(b) Pour $m \in \{1, \dots, p-1\}$ calculer :

$$\phi_{m,p} = \phi_{m,p-1} - k_p \phi_{p-m,p-1}$$

Proposition 143. Soit (X_t) un processus stationnaire au second ordre de fonction d'autocovariance $\gamma(h)$. Le coefficient k_{p+1} défini par (9.16) vérifie, pour tout $p \geq 0$:

$$k_{p+1} = \frac{\left\langle \boldsymbol{\varepsilon}_{t,p}^+, \boldsymbol{\varepsilon}_{t-p-1,p}^- \right\rangle}{\|\boldsymbol{\varepsilon}_{t,p}^+\| \|\boldsymbol{\varepsilon}_{t-p-1,p}^-\|}, \quad (9.23)$$

et

$$|k_{p+1}| \leq 1. \quad (9.24)$$

Proof. En utilisant que $\text{proj}(X_t | \mathcal{H}_{t-1,p})$ est orthogonal à $\varepsilon_{t-p-1,p}^-$, (9.16) et (9.7), on a

$$k_{p+1} = \frac{\langle \varepsilon_{t,p}^+, \varepsilon_{t-p-1,p}^- \rangle}{\|\varepsilon_{t-p-1,p}^-\|^2}.$$

Or, d'après (9.20), $\|\varepsilon_{t-p-1,p}^-\|^2 = \sigma_p^2 = \|\varepsilon_{t,p}^+\|^2$, la dernière égalité venant de (9.8), d'où l'on déduit (9.23). L'inégalité (9.24) se déduit alors de (9.23) en utilisant l'inégalité de Cauchy-Schwarz. \square

Définition 144 (Fonction d'autocorrélation partielle). *Soit (X_t) un processus stationnaire au second ordre de fonction d'autocovariance $\gamma(h)$. On appelle fonction d'autocorrélation partielle la suite des coefficients d'autocorrélation partielle $(k_p)_{p \geq 1}$ définie par :*

$$k_p = \text{corr}(X_t, X_{t-1}) = \frac{\langle X_t, X_{t-1} \rangle}{\|X_t\| \|X_{t-1}\|}, \text{ si } p = 1, \quad (9.25)$$

et

$$\begin{aligned} k_p &= \text{corr}(\varepsilon_{t,p-1}^+, \varepsilon_{t-p,p-1}^-) \\ &= \frac{\langle X_t - \text{proj}(X_t | \mathcal{H}_{t-1,p-1}), X_{t-p} - \text{proj}(X_{t-p} | \mathcal{H}_{t-1,p-1}) \rangle}{\|X_t - \text{proj}(X_t | \mathcal{H}_{t-1,p-1})\| \|X_{t-p} - \text{proj}(X_{t-p} | \mathcal{H}_{t-1,p-1})\|}, \text{ si } p \geq 2. \end{aligned} \quad (9.26)$$

Remarque 145. Dans (9.25), l'expression pour $p = 1$ est en accord avec celle pour $p \geq 2$ dans la mesure où on peut noter que $\varepsilon_{t,0}^+ = X_t$ et que $\varepsilon_{t-1,0}^- = X_{t-1}$. Notons aussi que, dans l'expression de k_p , X_t et X_{t-p} sont projetés sur le même sous-espace $\text{Vect}(X_{t-1}, \dots, X_{t-p+1})$. Le résultat remarquable est que la suite des coefficients de corrélation partielle est donnée par :

$$k_p = \phi_{p,p} \quad (9.27)$$

où $\phi_{p,p}$ est défini au moyen des équations de Yule-Walker (9.6).

Dans le cas particulier d'un processus AR(m) causal, on a alors :

$$k_p = \begin{cases} \phi_{p,p} & \text{pour } 1 \leq p < m, \\ \phi_m & \text{pour } p = m, \\ 0 & \text{pour } p > m. \end{cases}$$

9.3 Algorithme des innovations

L'algorithme des innovations est une application directe de la méthode de Gram-Schmidt et est, à cet égard, plus élémentaire que l'algorithme de Levinson-Durbin. De plus, il

ne suppose pas que le processus $(X_t)_{t \in \mathbb{Z}}$ soit stationnaire. L'espérance de X_t étant supposée nulle dans ce chapitre, nous notons

$$\kappa(i, j) = \langle X_i, X_j \rangle = \mathbb{E}[X_i X_j],$$

la fonction d'autocovariance de ce processus. Notons, pour $n \geq 1$,

$$\mathcal{H}_n = \text{Vect}(X_1, \dots, X_n) \text{ et } \sigma_n^2 = \|X_{n+1} - \text{proj}(X_{n+1} | \mathcal{H}_n)\|^2.$$

La procédure d'orthogonalisation de Gram-Schmidt permet alors d'écrire pour tout $n \geq 1$:

$$\mathcal{H}_n = \text{Vect}(X_1, X_2 - \text{proj}(X_2 | X_1), \dots, X_n - \text{proj}(X_n | \mathcal{H}_{n-1})),$$

où on utilise la convention suivante : $\text{proj}(X_1 | \mathcal{H}_0) = 0$. On a alors :

$$\text{proj}(X_{n+1} | \mathcal{H}_n) = \sum_{j=1}^n \theta_{n,j} (X_{n+1-j} - \text{proj}(X_{n+1-j} | \mathcal{H}_{n-j})). \quad (9.28)$$

L'algorithme des innovations décrit dans la proposition suivante fournit une méthode récursive permettant de calculer $(\theta_{n,j})_{1 \leq j \leq n}$ et σ_n^2 pour $n \geq 1$.

Proposition 146. Soit (X_t) un processus à moyenne nulle tel que la matrice $[\kappa(i, j)]_{1 \leq i, j \leq n}$ soit inversible pour tout $n \geq 1$ alors

$$\text{proj}(X_{n+1} | \mathcal{H}_n) = \begin{cases} 0, & \text{si } n = 0, \\ \sum_{j=1}^n \theta_{n,j} (X_{n+1-j} - \text{proj}(X_{n+1-j} | \mathcal{H}_{n-j})), & \text{si } n \geq 1, \end{cases}$$

où

$$\begin{cases} \sigma_0^2 = \kappa(1, 1), \\ \theta_{n,n-k} = \sigma_k^{-2} [\kappa(n+1, k+1) - \sum_{j=0}^{k-1} \theta_{k,k-j} \theta_{n,n-j} \sigma_j^2], \quad 0 \leq k \leq n-1, \\ \sigma_n^2 = \kappa(n+1, n+1) - \sum_{j=0}^{n-1} \theta_{n,n-j}^2 \sigma_j^2, \quad n \geq 1. \end{cases}$$

Proof. Remarquons tout d'abord que les vecteurs $(X_i - \text{proj}(X_i | \mathcal{H}_{i-1}))_{i \geq 1}$ sont orthogonaux. En effet, pour $i < j$, $X_i - \text{proj}(X_i | \mathcal{H}_{i-1}) \in \mathcal{H}_{j-1}$ et $X_j - \text{proj}(X_j | \mathcal{H}_{j-1}) \perp \mathcal{H}_{j-1}$. On en déduit, en faisant le produit scalaire de (9.28) par $X_{k+1} - \text{proj}(X_{k+1} | \mathcal{H}_k)$ que, pour $0 \leq k < n$:

$$\langle \text{proj}(X_{n+1} | \mathcal{H}_n), X_{k+1} - \text{proj}(X_{k+1} | \mathcal{H}_k) \rangle = \theta_{n,n-k} \sigma_k^2.$$

Puisque $\langle X_{n+1} - \text{proj}(X_{n+1} | \mathcal{H}_n), X_{k+1} - \text{proj}(X_{k+1} | \mathcal{H}_k) \rangle = 0$, les coefficients $\theta_{n,n-k}$, $k = 0, \dots, n-1$ sont donnés par

$$\theta_{n,n-k} = \sigma_k^{-2} \langle X_{n+1}, X_{k+1} - \text{proj}(X_{k+1} | \mathcal{H}_k) \rangle. \quad (9.29)$$

En utilisant la représentation (9.28),

$$\begin{aligned} \text{proj}(X_{k+1} | \mathcal{H}_k) &= \sum_{j=1}^k \theta_{k,j} (X_{k+1-j} - \text{proj}(X_{k+1-j} | \mathcal{H}_{k-j})) \\ &= \sum_{j=0}^{k-1} \theta_{k,k-j} (X_{j+1} - \text{proj}(X_{j+1} | \mathcal{H}_j)), \end{aligned}$$

d'où l'on déduit que

$$\theta_{n,n-k} = \sigma_k^{-2} \left(\kappa(n+1, k+1) - \sum_{j=0}^{k-1} \theta_{k,k-j} \langle X_{n+1}, X_{j+1} - \text{proj}(X_{j+1} | \mathcal{H}_j) \rangle \right).$$

D'après (9.29), $\langle X_{n+1}, X_{j+1} - \text{proj}(X_{j+1} | \mathcal{H}_j) \rangle = \sigma_j^2 \theta_{n,n-j}$ pour $0 \leq j < n$, nous avons donc pour $k \in \{1, \dots, n-1\}$,

$$\theta_{n,n-k} = \sigma_k^{-2} \left(\kappa(n+1, k+1) - \sum_{j=0}^{k-1} \theta_{k,k-j} \theta_{n,n-j} \sigma_j^2 \right). \quad (9.30)$$

L'équation (9.30) est encore valable lorsque $k = 0$ en utilisant la convention que la somme sur j dans le membre de droite est nulle dans ce cas. Par ailleurs, la proposition 198 (Pythagore) implique que

$$\begin{aligned} \sigma_n^2 &= \|X_{n+1} - \text{proj}(X_{n+1} | \mathcal{H}_n)\|^2 = \|X_{n+1}\|^2 - \|\text{proj}(X_{n+1} | \mathcal{H}_n)\|^2 \\ &= \kappa(n+1, n+1) - \sum_{k=0}^{n-1} \theta_{n,n-k}^2 \sigma_k^2. \end{aligned} \quad (9.31)$$

□

Alors que l'algorithme de Levinson-Durbin permet de déterminer les coefficients du développement de $\text{proj}(X_{n+1} | \mathcal{H}_n)$ sur X_1, \dots, X_n donnés par $\text{proj}(X_{n+1} | \mathcal{H}_n) = \sum_{j=1}^n \phi_{n,j} X_{n+1-j}$, l'algorithme des innovations calcule les coefficients du développement de $\text{proj}(X_{n+1} | \mathcal{H}_n)$ sur $X_1, X_2 - \text{proj}(X_2 | X_1), \dots, X_n - \text{proj}(X_n | \mathcal{H}_{n-1})$.

Exemple 147 (Prédiction d'un processus MA(1)). *Considérons le processus $X_t = Z_t + \theta Z_{t-1}$ où $(Z_t) \sim \text{B.B.}(0, \sigma^2)$. Nous avons donc $\kappa(i, j) = 0$ pour $|i - j| > 1$, $\kappa(i, i) = \sigma^2(1 + \theta^2)$ et $\kappa(i, i+1) = \theta \sigma^2$. Dans ce cas, nous avons*

$$\begin{cases} \theta_{n,j} = 0, & 2 \leq j \leq n, \\ \theta_{n,1} = \sigma_{n-1}^{-2} \theta \sigma^2, \end{cases}$$

et les variances des innovations qui sont données par

$$\begin{cases} \sigma_0^2 = (1 + \theta^2) \sigma^2, \\ \sigma_n^2 = [1 + \theta^2 - \sigma_{n-1}^{-2} \theta^2 \sigma^2] \sigma^2. \end{cases}$$

Si nous posons $r_n = \sigma_n^2 / \sigma^2$, nous avons

$$\text{proj}(X_{n+1} | \mathcal{H}_n) = \theta (X_n - \text{proj}(X_n | \mathcal{H}_{n-1})) / r_{n-1},$$

avec $r_0 = 1 + \theta^2$, et pour $n \geq 1$, $r_{n+1} = 1 + \theta^2 - \theta^2 / r_n$.

Part IV

Information et codage de signaux et d'images

Chapitre 10

Éléments de théorie de l'information

La complexité d'une suite de symboles peut se mesurer par la taille minimum d'un code permettant de reconstruire cette suite. La théorie de l'information de Shannon montre que le nombre de bit moyen pour coder chaque symbole dépend de l'entropie du processus aléatoire sous-jacent. Pour coder des suites de nombre réels, il est nécessaire de les approximer avec une quantification avant d'effectuer un codage entropique. L'optimisation de cette quantification est étudiée. Ces résultats donnent les bases mathématiques et algorithmiques permettant de comprimer des signaux audios ou des images.

10.1 Complexité et Entropie

La théorie de l'information définit la complexité d'une série numérique en évaluant la taille des codes permettant de reproduire cette série. Les fondations de cette théorie sont mises en place par Shannon en 1948, qui modélise des séries numériques comme des réalisations d'un processus aléatoire. Il démontre alors l'existence d'une complexité intrinsèque associée à tout processus aléatoire, qu'il appelle *entropie*. En 1965, Kolmogorov introduit une définition plus générale de la complexité d'une série numérique, comme étant la longueur minimum du programme binaire permettant de reproduire cette série avec un ordinateur. Le modèle d'ordinateur est une machine de Turing ayant un nombre fini d'états. Cette définition n'a pas recours à un modèle probabiliste mais est plus délicate à manipuler mathématiquement. Nous suivront donc ici l'approche de Shannon qui donne des résultats suffisamment précis pour la plupart des problèmes de traitement du signal.

10.1.1 Suites typiques

Considérons des suites de symboles de taille n prenant leurs valeurs dans un alphabet $A = \{a_k\}_{1 \leq k \leq K}$ de taille K . L'approche probabiliste de Shannon modélise ces séquences de symboles comme étant les valeurs prises par des variables aléatoires $X_1 X_2 \dots X_n$. Pour simplifier l'analyse, nous nous placerons dans le cas le plus simple où les X_i sont des variables aléatoires indépendantes et de même loi. On note

$$p(a_k) = \mathbb{P}\{X_i = a_k\}.$$

Comme les variables X_i sont indépendantes, la probabilité d'une suite de valeurs est:

$$p(x_1, \dots, x_n) = \mathbb{P}\{X_1 = x_1, \dots, X_n = x_n\} = \prod_{k=1}^n \mathbb{P}\{X_k = x_k\} = \prod_{k=1}^n p(x_k).$$

On peut définir $p(X_1, \dots, X_n) = \prod_{k=1}^n p(X_k)$ qui est une variable aléatoire donnant la probabilité d'une suite de valeurs tirée au hasard. Le théorème suivant montre que pour n fixé et suffisamment grand, alors pour la plupart des tirages, cette probabilité est presque constante et égale à l'entropie

$$H = - \sum_{k=1}^K p(a_k) \log_2 p(a_k) = -E\{\log_2 p(X_i)\}.$$

L'entropie H peut s'interpréter comme l'incertitude moyenne sur les valeurs que prennent les variables aléatoires X_i . On peut vérifier que

$$0 \leq H \leq \log_2 K.$$

L'entropie est maximum, $H = \log_2 K$, si $p(a_k) = \frac{1}{K}$ pour $1 \leq k \leq K$. Il y a en effet une incertitude maximum sur les valeurs prises par X_i . L'entropie est minimum, $H = 0$, si l'un des symboles a_k a une probabilité 1. On connaît alors à l'avance la valeur de X_i .

Théorème 148. *Si les X_i sont des variables aléatoires indépendantes et de même probabilité $p(x)$ alors*

$$-\frac{1}{n} \log_2 p(X_1, \dots, X_n) \text{ tend vers } H \text{ avec une probabilité 1}$$

lorsque n tend vers $+\infty$.

Proof. On calcule

$$-\frac{1}{n} \log_2 p(X_1, \dots, X_n) = -\frac{1}{n} \sum_{i=1}^n \log_2 p(X_i).$$

Comme les X_i sont indépendants, les $\log_2 p(X_i)$ sont aussi des variables aléatoires indépendantes. En appliquant la loi forte des grands nombres on démontre que $-\frac{1}{n} \sum_{i=1}^n \log_2 p(X_i)$ tend vers $-E\{\log p(X_i)\} = H$ lorsque n tend vers $+\infty$, avec probabilité 1. \square

Bien qu'a priori X_1, \dots, X_n puisse prendre des valeurs quelconques dans l'ensemble A^n des vecteurs de symboles de taille n , ce théorème permet de montrer qu'il y a une probabilité presque 1 pour que ce vecteur soit une suite typique appartenant à un ensemble beaucoup plus petit. On appelle *ensemble typique* T_ε^n relativement à $p(x)$ l'ensemble des suites $(x_1, \dots, x_n) \in A^n$ telles que

$$2^{-n(H+\varepsilon)} \leq p(x_1, \dots, x_n) \leq 2^{-n(H-\varepsilon)}. \quad (10.1)$$

On note $|T_\varepsilon^n|$ le cardinal de T_ε^n . Le théorème suivant montre que $|T_\varepsilon^n|$ est de l'ordre de 2^{nH} , et que toutes les suites typiques ont une probabilité presque égale à 2^{-nH} .

Proposition 149 (Ensembles typiques). (i) Si $(x_1, \dots, x_n) \in T_\varepsilon^n$ alors

$$H - \varepsilon \leq -\frac{1}{n} \log_2 p(x_1, \dots, x_n) \leq H + \varepsilon. \quad (10.2)$$

(ii) Lorsque n est suffisamment grand

$$\mathbb{P}\{(X_1, \dots, X_n) \in T_\varepsilon^n\} > 1 - \varepsilon. \quad (10.3)$$

(iii) Lorsque n est suffisamment grand

$$2^{n(H-\varepsilon)} \leq |T_\varepsilon^n| \leq 2^{n(H+\varepsilon)}. \quad (10.4)$$

Proof. La propriété (10.2) est une conséquence directe de la définition (10.1) de T_ε^n .

L'inégalité (10.3) se déduit du théorème 148 qui montre que pour tout $\varepsilon > 0$ et $\delta > 0$ il existe n_0 tel que pour tout $n \geq n_0$

$$\mathbb{P}\left\{\left|-\frac{1}{n} \log_2 p(X_1, \dots, X_n) - H\right| < \varepsilon\right\} > 1 - \delta.$$

En prenant $\delta = \varepsilon$ on obtient (10.3).

On note $\vec{x} = (x_1, \dots, x_n)$,

$$\begin{aligned} 1 &= \sum_{\vec{x} \in A^n} p(\vec{x}) \geq \sum_{\vec{x} \in T_\varepsilon^n} p(\vec{x}) \\ &\geq \sum_{\vec{x} \in T_\varepsilon^n} 2^{-n(H+\varepsilon)} = |T_\varepsilon^n| 2^{-n(H+\varepsilon)}, \end{aligned}$$

ce qui démontre l'inégalité (10.4) à droite.

Lorsque n est suffisamment grand, on a montré en (10.3) que

$$\begin{aligned} 1 - \varepsilon &< \mathbb{P}\{(X_1, \dots, X_n) \in T_\varepsilon^n\} \\ &\leq \sum_{x \in T_\varepsilon^n} 2^{-n(H-\varepsilon)} = |T_\varepsilon^n| 2^{-n(H-\varepsilon)}, \end{aligned}$$

ce qui démontre l'inégalité (10.4) □

10.1.2 Codage

On peut effectuer un codage “ ε -typique” des valeurs de X_1, \dots, X_n qui utilise des mots binaires plus courts pour coder les séquences typiques qui sont les plus probables. Comme il y a moins de $2^{n(H+\varepsilon)}$ éléments dans T_ε^n , ces éléments peuvent être indexés par des mots binaires de $\lfloor n(H+\varepsilon) \rfloor + 1$ bits, où $\lfloor x \rfloor$ est le plus grand entier inférieur à x . Comme il y a K^n éléments dans A , les éléments qui n'appartiennent pas à T_ε^n peuvent être indexés par des mots binaires de $\lfloor \log_2 K \rfloor + 1$ bits. Afin de savoir si $x \in T_\varepsilon^n$ on rajoute un 0 au début de son code binaire, qui est donc de longueur $\lfloor n(H+\varepsilon) \rfloor + 2$. Si $x \in T_\varepsilon^n$ on rajoute un 1 au début de son code binaire, dont la taille est donc $\lfloor \log_2 K \rfloor + 2$. On note R le nombre moyen de bits pour coder chaque symbole d'une séquence X_1, \dots, X_n .

Théorème 150. *Il existe $C > 0$ tel que pour tout $\varepsilon > 0$, et n suffisamment grand, le nombre moyen R de bits par symbole d'un codage ε -typique satisfait*

$$R \leq H + C\varepsilon.$$

Proof. On note $\vec{X} = (X_1, \dots, X_n)$, $\vec{x} = (x_1, \dots, x_n)$. Soit $l(x_i)$ la longueur du mot binaire utilisé par un code typique pour coder x_i . Le nombre total de bits pour coder \vec{x} est

$$l(\vec{x}) = \sum_{i=1}^n l(x_i).$$

Le nombre moyen de bits par symbole est donc

$$\begin{aligned} R &= E\{l(\vec{X})\} = \sum_{\vec{x} \in A^n} l(\vec{x}) p(\vec{x}) = \sum_{\vec{x} \in T_\epsilon^n} l(\vec{x}) p(\vec{x}) + \sum_{\vec{x} \notin T_\epsilon^n} l(\vec{x}) p(\vec{x}) \\ &\leq \mathbb{P}\{\vec{X} \in T_\epsilon^n\} (n(H + \epsilon) + 2) + \mathbb{P}\{\vec{X} \notin T_\epsilon^n\} (n \log_2 K + 2) \\ &\leq n(H + \epsilon) + 2 + \epsilon(n \log_2 K + 2) \leq H + C\epsilon \end{aligned}$$

avec $C = 5 + \log_2 K$ pour $n \geq 1/\epsilon$. \square

Ce théorème démontre que l'on peut construire un code dont le nombre de bit moyen par symbole est arbitrairement près de l'entropie H . Par ailleurs, on peut montrer que tout code nécessite un nombre moyen de bit par symbole $R \geq H$. Le paragraphe suivant démontre ce résultat pour les codes par blocs.

10.1.3 Codage entropique

Nous considerons dans un premier temps les codes instantanés, qui définissent un code binaire w_k pour chaque symbole a_k de l'alphabet A . Cela permet de décoder symbole par symbole toute séquence x_1, \dots, x_n . Si $\log_2 K$ est un entier, chaque symbole a_k peut être codé par un mot binaire de $\lfloor \log_2 K \rfloor + 1$ bits. Ce code peut cependant être amélioré en utilisant des mots binaires plus courts pour des symboles qui apparaissent plus souvent.

Soit l_k la longueur du code binaire w_k associée à a_k . Le nombre moyen de bits nécessaires pour coder les symboles d'une suite de variables aléatoires $X_1 \dots X_n$ de même probabilité $p(x)$ est

$$R = \sum_{k=1}^K l_k p(a_k). \quad (10.5)$$

Le but est de trouver un code instantané qui soit décodable et qui minimise R .

10.1.4 Condition de préfixe

Un code instantané n'est pas toujours uniquement décodable. Par exemple, le code qui associe à $\{a_k\}_{1 \leq k \leq 4}$ les mots binaires

$$\{w_1 = 0, w_2 = 10, w_3 = 110, w_4 = 101\}$$

n'est pas décodable de façon unique. La suite 1010 peut soit correspondre à $w_2 w_2$ ou à $w_4 w_1$. La condition de préfixe impose qu'aucun mot binaire n'est le début d'un autre mot binaire. Cette condition est clairement nécessaire et suffisante pour garantir que toute suite de mots binaires se décode de façon unique. Dans l'exemple précédent, w_2 est le préfixe de w_4 . Le code suivant

$$\{w_1 = 0, w_2 = 10, w_3 = 110, w_4 = 111\}$$

satisfait la condition de préfixe.

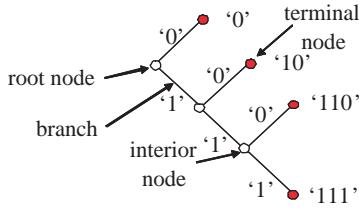


Figure 10.1: Arbre binaire d'un code de 4 symboles, qui satisfait la condition de préfixe.

Un code qui satisfait la condition de préfixe peut être associé à un arbre binaire, dont les K feuilles correspondent aux symboles $\{a_k\}_{1 \leq k \leq K}$. Cette représentation est utile pour construire le code qui minimise le nombre de bits moyen R . Les branches de gauche et de droite de l'arbre binaire sont respectivement codées par 0 et 1. La figure 10.1 montre un exemple pour un code de 4 symboles. Le mot binaire w_k associé au symbole a_k est la succession de 0 et de 1 correspondant aux branches de gauche et de droite, le long du chemin de la racine de l'arbre à la feuille correspondant à a_k . Le code binaire généré par un tel arbre satisfait toujours la condition de préfixe. En effet, w_m est un préfixe de w_k si et seulement si a_m est un ancêtre de a_k dans l'arbre binaire. Ceci n'est pas possible puisque les deux symboles correspondent à des feuilles de l'arbre. Inversement, tout code préfixe peut être représenté par un tel arbre binaire. La longueur l_k du mot binaire w_k est la profondeur de la feuille a_k dans l'arbre binaire. L'optimisation d'un code de préfixe est donc équivalente à la construction d'un arbre binaire optimal qui distribue les profondeurs des feuilles de façon à minimiser (10.5).

10.1.5 Entropie de Shannon

Le théorème de Shannon prouve que le nombre moyen de bit R par symbole est plus grand que l'entropie.

Théorème 151 (Shannon). *On suppose que les symboles $\{a_k\}_{1 \leq k \leq K}$ apparaissent avec la distribution de probabilité $\{p(a_k)\}_{1 \leq k \leq K}$. Le nombre moyen R de bit d'un code ayant la propriété du préfixe satisfait*

$$R \geq H = - \sum_{k=1}^K p(a_k) \log_2 p(a_k). \quad (10.6)$$

Il existe un code ayant la propriété du préfixe tel que

$$R \leq H + 1. \quad (10.7)$$

Proof. Le théorème de Shannon se démontre à partir de l'inégalité de Kraft.

Lemme 152 (Inégalité de Kraft). *Tout code ayant la propriété du préfixe satisfait*

$$\sum_{k=1}^K 2^{-l_k} \leq 1. \quad (10.8)$$

Inversement, si $\{l_k\}_{1 \leq k \leq K}$ sont des entiers positifs tels que l'inégalité (10.8) est satisfaite alors il existe un code de mots binaires $\{w_k\}_{1 \leq k \leq K}$ de longueurs $\{l_k\}_{1 \leq k \leq K}$ et qui satisfait la condition de préfixe.

Pour démontrer (10.8) on associe un arbre binaire au code considéré. Chaque l_k correspond à un noeud de l'arbre à une profondeur l_k qui dépend du mot binaire w_k . Soit

$$m = \max\{l_1, l_2, \dots, l_K\}. \quad (10.9)$$

On considère l'arbre binaire complet dont toutes les feuilles sont à la profondeur m . On note T_k le sous-arbre issu du noeud correspondant au mot binaire w_k . Ce sous arbre a une profondeur $m - l_k$ et contient donc 2^{m-l_k} noeud au niveau m . Comme il y a 2^m noeud à la profondeur m de l'arbre binaire complet et que la propriété du préfixe implique que tous les sous arbres T_1, \dots, T_K sont distincts, on déduit que

$$\sum_{k=1}^K 2^{m-l_k} \leq 2^m,$$

d'où (10.8).

Inversement, on considère $\{l_k\}_{1 \leq k \leq K}$ satisfaisant (10.8) avec $l_1 \leq l_2 \leq \dots \leq l_K$ et $m = \max\{l_1, l_2, \dots, l_K\}$. On définit les ensembles N_1 des 2^{m-l_1} premiers noeud au niveau m sur la gauche de l'arbre, puis N_2 l'ensemble des 2^{m-l_2} noeuds suivants et ainsi de suite. Les noeuds des ensembles N_k sont les noeuds terminaux de sous-arbres T_k qui sont disjoints. On associe à la racine de l'arbre T_k qui est à la profondeur l_k le mot binaire w_k . Cela définit un code qui satisfait la condition du préfixe où chaque mot a la longueur l_k voulue. Cela termine la démonstration du lemme.

Pour démontrer les deux inégalités (10.6) et (10.7) du théorème, on considère la minimisation de

$$R = \sum_{k=1}^K p(a_k) l_k$$

sous la contrainte de Kraft

$$\sum_{k=1}^K 2^{-l_k} \leq 1.$$

Dans un premier temps, nous supposons que l_k peut être un réel quelconque. Le minimum se calcule en utilisant un multiplicateur de Lagrange λ et en minimisant

$$J = \sum_{k=1}^K p(a_k) l_k + \lambda \sum_{k=1}^K 2^{-l_k}.$$

L'annulation de la dérivée par rapport à l_k donne

$$\frac{\partial J}{\partial l_k} = p(a_k) - \lambda 2^{-l_k} \log_{\exp} 2 = 0.$$

Le minimum est obtenu pour $\sum_{k=1}^K 2^{-l_k} = 1$ et comme $\sum_{k=1}^K p(a_k) = 1$ on obtient $\lambda = 1/\log_{\exp} 2$. La longueur optimale minimisant R est donc

$$l_k = -\log_2 p(a_k),$$

et

$$R = \sum_{k=1}^K p(a_k) l_k = - \sum_{k=1}^K p(a_k) \log_2 p(a_k) = H.$$

Pour garantir que l_k est entier, on choisit

$$l_k = \lceil -\log_2 p(a_k) \rceil$$

où $\lceil x \rceil$ est la plus petite valeur entière supérieure à x . Cela correspond au code de Shannon. Comme $l_k \geq -\log_2 p(a_k)$, l'inégalité de Kraft est satisfaite puisque

$$\sum_{k=1}^K 2^{-l_k} \leq \sum_{k=1}^K 2^{\log_2 p(a_k)} = 1.$$

Il existe donc un code préfixe dont les mots de code ont une longueur l_k . Pour ce code

$$\sum_{k=1}^K p(a_k) l_k \leq \sum_{k=1}^K p(a_k) (-\log_2 p(a_k) + 1) = H + 1.$$

□

10.1.6 Codage par blocs

L'ajout de 1 bit dans l'inégalité (10.7) vient du fait que $-\log_2 p_i$ n'est pas nécessairement un entier alors que la longueur d'un mot binaire doit être un entier. On peut construire des codes tels que R est plus proche de H en répartissant ce bit supplémentaire sur un bloc de n éléments. Au lieu de faire un codage instantané, symbole par symbole, on code d'un coup le bloc de symboles $\vec{X} = X_1, \dots, X_n$, qui peut être considéré comme une variable aléatoire à valeurs dans l'alphabet A^n de taille K^n . A tout bloc de symboles $\vec{a} \in A^n$ on associe un mot binaire de longueur $l(\vec{a})$. Le nombre de bits R par symbole pour un tel code par bloc est

$$R = \frac{1}{n} \sum_{\vec{a} \in A^n} p(\vec{a}) l(\vec{a}).$$

Proposition 153. *Le nombre moyen R de bit d'un code par bloc de taille n ayant la propriété du préfixe satisfait*

$$R \geq H = - \sum_{k=1}^K p(a_k) \log_2 p(a_k). \quad (10.10)$$

Il existe un code par blocs de taille n ayant la propriété du préfixe tel que

$$R \leq H + \frac{1}{n}. \quad (10.11)$$

Proof. L'entropie associée à \vec{X} est

$$\vec{H} = \sum_{\vec{x} \in A^n} p(\vec{x}) \log_2 p(\vec{x}).$$

Comme les variables aléatoires X_i sont indépendantes

$$p(\vec{x}) = p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i).$$

On démontre par récurrence sur n que $\vec{H} = nH$. Soit \vec{R} le nombre de bits moyen pour coder les n symboles \vec{X} . Le théorème de Shannon 151 montre que $\vec{R} \geq \vec{H}$ et qu'il existe un code par bloc tel que $\vec{R} \leq \vec{H} + 1$. On déduit donc (10.10,10.11) pour $R = \frac{\vec{R}}{n}$, qui est le nombre de bits moyen par symbole. \square

Ce théorème démontre que des codes par blocs utilisent un nombre moyen de bits par symbole qui tendent vers l'entropie lorsque la taille du bloc augmente.

10.1.7 Code de Huffman

L'algorithme de Huffman est un algorithme de programmation dynamique qui construit de bas en haut un arbre correspondant à un code préfixe et qui minimise

$$R = \sum_{k=1}^K p(a_k) l_k. \quad (10.12)$$

Nous ordonnons $\{a_k\}_{1 \leq k \leq K}$ pour que $p(a_k) \leq p(a_{k+1})$. Pour minimiser (10.12) les symboles de plus petites probabilités doivent être associés aux mots binaires w_k de longueur maximale, ce qui correspond à un noeud au bas de l'arbre. Nous commençons donc par représenter les deux symboles de plus petite probabilité a_1 et a_2 comme les enfants d'un noeud commun. Ce noeud peut être interprété comme un symbole $a_{1,2}$ correspondant à “ a_1 ou a_2 ” et dont la probabilité est $p(a_1) + p(a_2)$. La proposition suivante prouve que l'on peut itérer ce regroupement élémentaire et construire un code optimal.

Proposition 154. *On considère K symboles avec leurs probabilités ordonnées en ordre croissant: $p(a_k) \leq p(a_{k+1})$. On regroupe les deux symboles a_1 et a_2 de probabilité minimum en un seul symbole $a_{1,2}$ de probabilité*

$$p(a_{1,2}) = p(a_1) + p(a_2).$$

Un arbre correspondant à un code préfixe optimal pour les K symboles se construit à partir d'un arbre de code préfixe optimal pour les $K - 1$ symboles $\{a_{1,2}\} \cup \{a_k\}_{3 \leq k \leq K}$, en divisant la feuille de $a_{1,2}$ en deux noeuds correspondant à a_1 et a_2 .

La démonstration de cette proposition se trouve dans [2].

Cette proposition réduit la construction d'un code optimal de K symboles à la construction d'un code optimal pour les $K - 1$ symboles. Le code de Huffman itère $K - 1$ fois ce regroupement et fait progressivement pousser l'arbre d'un code de préfixe optimal depuis le bas jusqu'en haut. Le Théorème 151 de Shannon prouve que

$$H \leq R \leq H + 1. \quad (10.13)$$

10.1.8 Exemple

Les probabilités des $\{a_k\}_{1 \leq k \leq 6}$ sont

$$\{p(a_k)\}_{1 \leq k \leq 6} = \{0.05, 0.1, 0.1, 0.15, 0.2, 0.4\}. \quad (10.14)$$

La figure 10.2 donne l'arbre binaire construit avec l'algorithme de Huffman. Les symboles a_1 et a_2 sont regroupés en un symbole $a_{1,2}$ de probabilité $p(a_{1,2}) = p(a_1) + p(a_2) = 0.15$. A l'itération suivante, les symboles de plus basse probabilité sont $p(a_3) = 0.1$ et $p(a_{1,2}) = 0.15$. On regroupe donc $a_{1,2}$ et a_3 en un symbole $a_{1,2,3}$ dont la probabilité est 0.25. Les deux symboles de probabilités les plus faibles sont alors a_4 et a_5 qui sont regroupés en $a_{4,5}$ de probabilité 0.35. On regroupe ensuite $a_{4,5}$ et $a_{1,2,3}$ pour obtenir un symbole $a_{1,2,3,4,5}$ de probabilité 0.6 qui est finalement regroupé avec a_6 , ce qui finit le code, comme l'illustre l'arbre de la figure 10.2. Le nombre moyen de bits obtenu par ce code est $R = 2.35$ alors que l'entropie est $H = 2.28$.

=6cm NewFig/fig3.eps

Figure 10.2: Arbre correspondant au code de Huffman pour une source dont les probabilités sont données par (10.14) [14].

10.1.9 Sensibilité au bruit

Un code de Huffman est plus compact qu'un code de taille fixe $\log_2 K$ mais est aussi plus sensible au bruit. Pour un code de taille constante, une erreur de transmission d'un bit modifie seulement la valeur d'un symbole. Au contraire, une erreur d'un bit dans un code de taille variable peut modifier toute la suite des symboles. Lors de transmissions bruitées, de telles erreurs peuvent se produire. Il est alors nécessaire d'utiliser un code correcteur qui introduit une légère redondance de façon à identifier les erreurs.

10.2 Quantification scalaire

Si une variable aléatoire X prend des valeurs réelles quelconques, on ne peut pas obtenir un code exact de taille finie. Il est alors nécessaire d'approximer X par \tilde{X} qui prend ses valeurs dans un alphabet fini, et l'erreur résultante est

$$D = E\{|X - \tilde{X}|^2\}.$$

Un quantificateur scalaire décompose l'axe réel en K intervalles $\{[y_{k-1}, y_k]\}_{1 \leq k \leq K}$ de tailles variables, avec $y_0 = -\infty$ et $y_K = +\infty$. Le quantificateur associe à tout $x \in [y_{k-1}, y_k]$ une valeur $Q(x) = a_k$. Si les K niveaux de quantification $\{a_k\}_{1 \leq k \leq K}$ sont fixés a priori, pour minimiser $|x - Q(x)| = |x - a_k|$, il faut que la quantification associe à x son niveau de quantification a_k le plus proche. On doit alors choisir des intervalles de quantification qui satisfont

$$y_k = \frac{a_k + a_{k+1}}{2} \quad (10.15)$$

10.2.1 Quantification haute résolution

Soit $p(x)$ la densité de probabilité de X . On note $\tilde{X} = Q(X)$ la variable quantifiée. L'erreur quadratique moyenne est

$$D = E\{(X - \tilde{X})^2\} = \int_{-\infty}^{+\infty} |x - Q(x)|^2 p(x) dx. \quad (10.16)$$

On dit que le quantificateur a une haute résolution si $p(x)$ peut être approximé par une constante sur tout intervalle de quantification $[y_{k-1}, y_k]$. La taille de ces intervalles est $\Delta_k = y_k - y_{k-1}$. L'hypothèse de haute résolution implique que

$$p(x) = \frac{p_k}{\Delta_k} \text{ pour } x \in [y_{k-1}, y_k], \quad (10.17)$$

avec

$$p_k = \mathbb{P}\{X \in [y_{k-1}, y_k]\} = \mathbb{P}\{\tilde{X} = a_k\}.$$

La proposition suivant calcule l'erreur D sous cette hypothèse.

Proposition 155. Pour un quantificateur de haute résolution sur des intervalles $[y_{k-1}, y_k]$, l'erreur D minimum obtenue en optimisant la position des niveaux $\{a_k\}_{0 \leq k \leq K}$ est

$$D = \frac{1}{12} \sum_{k=1}^K p_k \Delta_k^2. \quad (10.18)$$

Proof. Comme $Q(x) = a_k$ si $x \in [y_{k-1}, y_k]$, on peut reécrire (10.16)

$$D = \sum_{k=1}^K \int_{y_{k-1}}^{y_k} (x - a_k)^2 p(x) dx.$$

En remplaçant $p(x)$ par son expression (10.17) on a

$$D = \sum_{k=1}^K \frac{p_k}{\Delta_k} \int_{y_{k-1}}^{y_k} (x - a_k)^2 dx. \quad (10.19)$$

Cette erreur est minimum pour $a_k = \frac{1}{2}(y_k + y_{k-1})$, et l'intégration donne (10.18). \square

10.2.2 Quantification uniforme

Le quantificateur uniforme est un cas particulier important où tous les intervalles de quantification sont de même taille

$$y_k - y_{k-1} = \Delta \text{ pour } 1 \leq k \leq K.$$

L'erreur quadratique moyenne (10.18) devient

$$D = \frac{\Delta^2}{12} \sum_{k=1}^K p_k = \frac{\Delta^2}{12}. \quad (10.20)$$

Elle est indépendante de la distribution de probabilité $p(x)$ de la source.

10.2.3 Quantification optimale

On veut optimiser le quantificateur pour minimiser le nombre de bits nécessaires pour coder les valeurs quantifiées \tilde{X} , étant donnée une distortion D admissible. Le théorème de Shannon 151 prouve que la valeur moyenne minimum de bits nécessaire pour coder \tilde{X} est supérieure à l'entropie H de la variable aléatoire X . Comme le code de Huffman donne un résultat proche de cette entropie, il nous faut minimiser l'entropie H pour D fixe.

La source quantifiée \tilde{X} prend K valeurs différentes $\{a_k\}_{1 \leq k \leq K}$ avec probabilités $\{p_k\}_{1 \leq k \leq K}$. L'entropie du signal quantifié est donc

$$H = - \sum_{k=1}^K p_k \log_2 p_k.$$

On définit l'entropie différentielle de la variable aléatoire X à valeurs réelles

$$H_d = - \int_{-\infty}^{+\infty} p(x) \log_2 p(x) dx. \quad (10.21)$$

Le théorème suivant montre que pour un quantificateur de haute résolution produisant une erreur D , l'entropie est minimum lorsque le quantificateur est uniforme.

Théorème 156. *L'entropie de tout quantificateur de haute résolution satisfait*

$$H \geq H_d - \frac{1}{2} \log_2 (12D). \quad (10.22)$$

Le minimum est atteint si et seulement si Q est un quantificateur uniforme.

Proof. Pour un quantificateur de haute résolution $p(x)$ est approximativement constant sur $[y_{k-1}, y_k]$ et donc

$$p_k = \int_{y_{k-1}}^{y_k} p(x) dx = p_k \Delta_k$$

avec $\Delta_k = y_k - y_{k-1}$. Donc

$$\begin{aligned} H &= - \sum_{k=1}^K p_k \log_2 (p(a_k) \Delta_k) \\ &= - \sum_{k=1}^K \int_{y_{k-1}}^{y_k} p(x) \log_2 p(a_k) dx - \sum_{k=1}^K p_k \log_2 \Delta_k \\ &= H_d - \frac{1}{2} \sum_{k=1}^K p_k \log_2 \Delta_k^2, \end{aligned}$$

car $p(x) = p(a_k)$ pour $x \in [y_{k-1}, y_k]$. Pour toute fonction concave $\phi(x)$, l'inégalité de Jensen montre que pour tout $\sum_{k=1}^K p_k = 1$ et $\{a_k\}_{1 \leq k \leq K}$ alors

$$\sum_{k=1}^N p_k \phi(a_k) \leq \phi\left(\sum_{k=1}^N p_k a_k\right). \quad (10.23)$$

Si $\phi(x)$ est strictement concave, l'inégalité devient une égalité si et seulement si tous les a_k sont égaux lorsque $p_k \neq 0$. Comme $\log_2(x)$ est strictement concave, (10.18) montre que

$$\frac{1}{2} \sum_{k=1}^N p_k \log_2 \Delta_k^2 \leq \frac{1}{2} \log_2 \sum_{k=1}^N p_k \Delta_k^2 = \frac{1}{2} \log_2(12D).$$

On en déduit donc que

$$H \geq H_d - \frac{1}{2} \log_2(12D).$$

Cette inégalité devient une égalité si et seulement si tous les Δ_k sont égaux, ce qui correspond à un quantificateur uniforme. \square

Ce théorème montre que pour un quantificateur haute résolution le nombre minimum de bits $R = H$ est obtenu pour un quantificateur uniforme et

$$R = H_d - \frac{1}{2} \log_2(12D). \quad (10.24)$$

La distortion en fonction du nombre de bits est donc

$$D(R) = \frac{1}{12} 2^{2H_d} 2^{-2R}.$$

Chapitre 11

Application au codage de parole

Principles of Speech Coding

W. B. Kleijn

Speech coding is the art of reducing the bit rate required to describe a speech signal. In this chapter, we discuss the attributes of speech coders as well as the underlying principles that determine their behavior and their architecture. The ubiquitous class of linear-prediction-based coders is used as an illustration. Speech is generally modeled as a sequence of stationary signal segments, each having unique statistics. Segments are encoded using a two-step procedure: (1) find a model describing the speech segment, (2) encode the segment assuming it is generated by the model. We show that the bit allocation for the model (the predictor parameters) is independent of overall rate and of perception, which is consistent with existing experimental results. The modeling of perception is an important aspect of efficient coding and we discuss how various perceptual distortion measures can be integrated into speech coders.

14.1 The Objective of Speech Coding	283
14.2 Speech Coder Attributes	284
14.2.1 Rate	284
14.2.2 Quality.....	285

14.2.3 Robustness to Channel Imperfections	285
14.2.4 Delay	286
14.2.5 Computational and Memory Requirements	286
14.3 A Universal Coder for Speech.....	286
14.3.1 Speech Segment as Random Vector	286
14.3.2 Encoding Random Speech Vectors ..	287
14.3.3 A Model of Quantization.....	288
14.3.4 Coding Speech with a Model Family	289
14.4 Coding with Autoregressive Models	293
14.4.1 Spectral-Domain Index of Resolvability	293
14.4.2 A Criterion for Model Selection	294
14.4.3 Bit Allocation for the Model.....	295
14.4.4 Remarks on Practical Coding.....	296
14.5 Distortion Measures and Coding Architecture	296
14.5.1 Squared Error	297
14.5.2 Masking Models and Squared Error.	298
14.5.3 Auditory Models and Squared Error	299
14.5.4 Distortion Measure and Coding Architecture.....	301
14.6 Summary	302
References	303

14.1 The Objective of Speech Coding

In modern communication systems, speech is represented by a sequence of bits. The main advantage of this *binary* representation is that it can be recovered exactly (without distortion) from a noisy channel (assuming proper system design), and does not suffer from decreasing quality when transmitted over many transmission legs. In contrast, analog transmission generally results in an increase of distortion with the number of legs.

An acoustic speech signal is inherently analog. Generally, the resulting analog microphone output is converted to a binary representation in a manner con-

sistent with Shannon's sampling theorem. That is, the analog signal is first band-limited using an anti-aliasing filter, and then simultaneously sampled and quantized. The output of the *analog-to-digital (A/D) converter* is a digital speech signal that consists of a sequence of numbers of finite precision, each representing a sample of the band-limited speech signal. Common sampling rates are 8 and 16 kHz, rendering *narrowband speech* and *wideband speech*, respectively, usually with a precision of 16 bits per sample. For the 8 kHz sampling rate a logarithmic 8-bit-per-sample representation is also common.

Particularly at the time of the introduction of the binary speech representation, the bit rate produced by the A/D converter was too high for practical applications such as cost-effective mobile communications and secure telephony. A search ensued for more-efficient digital representations. Such representations are possible since the digital speech contains *irrelevancy* (the signal is described with a higher precision than is needed) and *redundancy* (the rate can be decreased without affecting precision). The aim was to trade off computational effort at the transmitter and receiver for the bit rate required for the speech representation. Efficient representations generally involve a *model* and a set of *model parameters*, and sometimes a set of coefficients that form the input to the model. The algorithms used to reduce the required rate are called speech-coding algorithms, or *speech codecs*.

The performance of speech codecs can be measured by a set of properties. The fundamental codec attributes are bit rate, speech quality, quality degradation due to channel errors and packet loss, delay, and computational effort. Good performance for one of the attributes generally leads to lower performance for the others. The

interplay between the attributes is governed by the fundamental laws of information theory, the properties of the speech signal, limitations in our knowledge, and limitations of the equipment used.

To design a codec, we must know the desired values for its attributes. A common approach to develop a speech codec is to constrain all attributes but one quantitatively. The design objective is then to optimize the remaining attribute (usually quality or rate) subject to these constraints. A common objective is to maximize the average quality over a given set of channel conditions, given the rate, the delay, and the computational effort.

In this chapter, we attempt to discuss speech coding at a generic level and yet provide information useful for practical coder design and analysis. Section 14.2 describes the basic attributes of a speech codec. Section 14.3 discusses the underlying principles of coding and Sect. 14.4 applies these principles to a commonly used family of linear predictive (autoregressive model-based) coders. Section 14.5 discusses distortion criteria and how they affect the architecture of codecs. Section 14.6 provides a summary of the chapter.

14.2 Speech Coder Attributes

The usefulness of a speech coder is determined by its attributes. In this section we describe the most important attributes and the context in which they are relevant in some more detail. The attributes were earlier discussed in [14.1, 2].

14.2.1 Rate

The rate of a speech codec is generally measured as the average number of bits per second. For *fixed-rate* coders the bit rate is the same for each coding block, while for *variable-rate* coders it varies over time.

In traditional circuit-switched communication systems, a fixed rate is available for each communication direction. It is then natural to exploit this rate at all times, which has resulted in a large number of standardized fixed-rate speech codecs. In such coders each particular parameter or variable is encoded with the same number of bits for each block. This a priori knowledge of the bit allocation has a significant effect on the structure of the codec. For example, the mapping of the quantization indices to the transmitted codewords is trivial. In more-flexible circuit-switched networks (e.g., modern

mobile-phone networks), codecs may have a variable number of modes, each mode having a different fixed rate [14.3, 4]. Such codecs with a set of fixed coding rates should not be confused with true variable-rate coders.

In variable-rate coders, the bit allocation within a particular block for the parameters or variable depends on the signal. The bit allocation for a parameter varies with the quantization index and the mapping from the quantization index to the transmitted codeword is performed by means of a table lookup or computation, which can be very complex. The major benefit of variable-rate coding is that it leads to higher coding efficiency than fixed-rate coders because the rate constraint is less strict.

In general, network design evolves towards the facilitation of variable-rate coders. In packet-switched communication systems, both packet rate and size can vary, which naturally leads to variable-rate codecs. While variable-rate codecs are common for audio and video signals, they are not yet commonplace for speech. The requirements of low rates and delays lead to a small packet payload for speech signals. The relatively large packet header size limits the benefits of the low rate and,

consequently, the benefit of variable-rate speech coding. However, with the removal of the fixed-rate constraint, it is likely that variable-rate speech codecs will become increasingly common.

14.2.2 Quality

To achieve a significant rate reduction, the parameters used to represent the speech signal are generally transmitted at a reduced precision and the reconstructed speech signal is not a perfect copy of the original digital signal. It is therefore important to ensure that its quality meets a certain standard.

In speech coding, we distinguish two applications for quality measures. First, we need to evaluate the *overall quality* of a particular codec. Second, we need a *distortion measure* to decide how to encode each signal block (typically of duration 5–25 ms). The distortion measure is also used during the design of the coder (in the training of its codebooks). Naturally, these quality measures are not unrelated, but in practice their formulation has taken separate paths. Whereas overall quality can be obtained directly from scoring of speech utterances by humans, distortion measures used in coding algorithms have been defined (usually in an ad hoc manner) based on knowledge about the human auditory system.

The only true measure of the overall quality of a speech signal is its rating by humans. Standardized conversational and listening tests have been developed to obtain reliable and repeatable (at least to a certain accuracy) results. For speech coding, listening tests, where a panel of listeners evaluates performance for a given set of utterances, are most common. Commonly used standardized listening tests use either an absolute category rating, where listeners are asked to score an utterance on an absolute scale, or a degradation category rating, where listeners are asked to provide a relative score. The most common overall measure associated with the absolute category rating of speech quality is the mean opinion score (**MOS**) [14.5]. The **MOS** is the mean value of a numerical score given to an utterance by a panel of listeners, using a standardized procedure. To reduce the associated cost, subjective measures can be approximated by objective, repeatable algorithms for many practical purposes. Such measures can be helpful in the development of new speech coders. We refer to [14.6–8] and to Chap. 5 for more detail on the subject of overall speech quality.

As a distortion measure for speech segments variants of the squared-error criterion are most commonly used. The squared-error criterion facilitates fast evaluation for

coding purposes. Section 14.5 discusses distortion measures in more detail. It is shown that adaptively weighted squared error criteria can be used for a large range of perceptual models.

14.2.3 Robustness to Channel Imperfections

Early terrestrial digital communication networks were generally designed to have very low error rates, obviating the need for measures to correct errors for the transmission of speech. In contrast, bit errors and packet loss are inherent in modern communication infrastructures.

Bit errors are common in wireless networks and are generally addressed by introducing channel codes. While the integration of source and channel codes can result in higher performance, this is not commonly used because it results in reduced modularity. Separate source and channel coding is particularly advantageous when a codec is faced with different network environments; different channel codes can then be used for different network conditions.

In packet networks, the open systems interconnection reference (**OSI**) model [14.9] provides a separation of various communication functionalities into seven layers. A speech coder resides in the application layer, which is the seventh and highest layer. Imperfections in the transmission are removed in both the physical layer (the first layer) and the transport layer (the fourth layer). The physical layer removes *soft* information, which consists of a probability for the allowed symbols, and renders a sequence of bits to the higher layers. Error control normally resides in the transport layer. However, the error control of the transport layer, as specified by the transmission control protocol (**TCP**) [14.10], and particularly the automatic repeat requests that **TCP** uses is generally not appropriate for real-time communication of audiovisual data because of delay. **TCP** is also rarely used for broadcast and multicast applications to reduce the load on the transmitter. Instead, the user datagram protocol (**UDP**) [14.11] is used, which means that the coded signal is handed up to the higher network layers without error correction. It is possible that in future systems cross-layer interactions will allow the application layer to receive information about the soft information available at the physical layer.

Handing the received coded signal with its defects directly to the application layer allows the usage of both the inherent redundancy in the signal and our knowledge of the perception of distortion by the user. This leads to coding systems that exhibit a graceful degrada-

tion with increasing error rate. We refer to the chapter on voice over internet protocol (IP) for more detail on techniques that lead to robustness against bit errors and packet loss.

14.2.4 Delay

From coding theory [14.12], we know that optimal coding performance generally requires a delay in the transfer of the message. Long delays are impractical because they are generally associated with methods with high computational and storage requirements, and because in real-time environments (common for speech) the user does not tolerate a long delay.

Significant delay directly affects the quality of a conversation. Impairment to conversations is measurable at one-way delays as low as 100 ms [14.13], although 200 ms is often considered a useful bound.

Echo is perceivable at delays down to 20 ms [14.14]. Imperfections in the network often lead to so-called network echo. Low-delay codecs have been designed to keep the effect of such echo to a minimum, e.g., [14.15]. However, echo cancellation has become commonplace in communication networks. Moreover, packet networks have an inherent delay that requires echo cancellation even for low-delay speech codecs. Thus, for most applications codecs can be designed without consideration of echo.

In certain applications the user may hear both an acoustic signal and a signal transmitted by a network. Examples are flight control rooms and wireless systems for hearing-impaired persons. In this class of applications, coding delays of less than 10 ms are needed to attain an acceptable overall delay.

14.2.5 Computational and Memory Requirements

Economic cost is generally a function of the computational and memory requirements of the coding system. A common measure of computational complexity used in applications is the number of instructions required on a particular silicon device. This is often translated into the number of channels that can be implemented on a single device.

A complicating factor is that speech codecs are commonly implemented on fixed-point signal processing devices. Implementation on a fixed-point device generally takes significant development effort beyond that of the development of the floating-point algorithm.

It is well known that vector quantization facilitates an optimal rate versus quality trade-off. Basic vector quantization techniques require very high computational effort and the introduction of vector quantization in speech coding resulted in promising but impractical codecs [14.16]. Accordingly, significant effort was spent to develop vector quantization structures that facilitate low computational complexity [14.17–19]. The continuous improvement in vector quantization methods and an improved understanding of the advantages of vector quantization over scalar quantization [14.20, 21] has meant that the computational effort of speech codecs has not changed significantly over the past two decades, despite significant improvement in codec performance. More effective usage of scalar quantization and the development of effective lattice vector quantization techniques make it unlikely that the computational complexity of speech codecs will increase significantly in the future.

14.3 A Universal Coder for Speech

In this section, we consider the encoding of a speech signal from a fundamental viewpoint. In information-theoretic terminology, speech is our *source* signal. We start with a discussion of the direct encoding of speech segments, without imposing any structure on the coder. This discussion is not meant to lead directly to a practical coding method (the computational effort would not be reasonable), but to provide an insight into the structure of existing coders. We then show how a signal model can be introduced. The signal model facilitates coding at a reasonable computational cost and the resulting coding paradigm is used by most speech codecs.

14.3.1 Speech Segment as Random Vector

Speech coders generally operate on a sequence of subsequent signal segments, which we refer to as *blocks* (also commonly known as *frames*). Blocks consist generally, but not always, of a fixed number of samples. In the present description of a basic coding system, we divide the speech signal into subsequent blocks of equal length and denote the block length in samples by k . We neglect dependencies across block boundaries, which is not always justified in a practical implementation, but simplifies the discussion; it is generally straightforward to cor-

rect this omission on implementation. We assume that the blocks can be described by k -dimensional random vectors \mathbf{X}^k with a probability density function $p_{\mathbf{X}^k}(\mathbf{x}^k)$ for any $\mathbf{x}^k \in \mathbb{R}^k$, the k -dimensional Euclidian space (following convention, we denote random variables by capital letters and realizations by lower case letters).

For the first part of our discussion (Sect. 14.3.2), it is sufficient to assume the existence of the probability density function. It is natural, however, to consider some structure of the probability density $p_{\mathbf{X}^k}(\cdot)$ based on the properties of speech. We commonly describe speech in terms of a particular set of sounds (a distinct set of phones). A speech vector then corresponds to one sound from a countable set of speech sounds. We impose the notion that speech consists of a set of sounds on our probabilistic speech description. We can think of each sound as having a particular probability density. A particular speech vector then has one of a set of possible probability densities. Each member probability density of the set has an *a prior* probability, denoted as $p_I(i)$, where i indexes the set. The prior probability $p_I(i)$ is the probability that a random vector \mathbf{X}^k is drawn from the particular member probability density i . The overall probability function of the random speech vector $p_{\mathbf{X}^k}(\cdot)$ is then a *mixture* of probability density functions

$$p_{\mathbf{X}^k}(\mathbf{x}^k) = \sum_{i \in \mathcal{A}} p_I(i) p_{\mathbf{X}^k|I}(\mathbf{x}^k|i), \quad (14.1)$$

where \mathcal{A} is the set of indexes for component densities and $p_{\mathbf{X}^k|I}(\cdot|\cdot)$ is the density of component i . These densities are commonly referred to as *mixture components*. If the set of mixture components is characterized by continuous parameters, then the summation must be replaced by an integral.

A common motivation for the mixture formulation of (14.1) is that a good approximation to the true probability density function can be achieved with a mixture of a finite set of probability densities from a particular family. This eliminates the need for the physical motivation. The family is usually derived from a single *kernel* function, such as a Gaussian. The kernel is selected for mathematical tractability.

If a mixture component does correspond to a physically reasonable speech sound, then it can be considered a statistical *model* of the signal. As described in Sect. 14.3.4, it is possible to interpret existing speech coding paradigms from this viewpoint. For example, linear prediction identifies a particular autoregressive model appropriate for a block. Each of the autoregressive models of speech has a certain prior probability and this

in turn leads to an overall probability for the speech vector. According to this interpretation, mixture models have long been standard tools in speech coding, even if this was not explicitly stated.

The present formalism does not impose stationarity conditions on the signal within the block. In the mixture density, it is reasonable to include densities that correspond to signal transitions. In practice, this is not common, and the probability density functions are usually defined based on the definition that the signal is stationary within a block. On the other hand, the assumption that all speech blocks are drawn from the same distribution is implicit in the commonly used coding methods. It is consistent with our neglect of interblock dependencies. Thus, if we consider the speech signal to be a vector signal, then we assume stationarity for this vector signal (which is a rather inaccurate approximation). Strictly speaking, we do not assume ergodicity, as averaging over a database is best interpreted an averaging over an ensemble of signals, rather than time averaging over a single signal.

14.3.2 Encoding Random Speech Vectors

To encode observed speech vectors \mathbf{x}^k that form *realizations* of the random vector \mathbf{X}^k , we use a speech codebook $C_{\mathbf{X}^k}$ that consists of a countable set of k -dimensional vectors (the code vectors). We can write $C_{\mathbf{X}^k} = \{\mathbf{c}_q^k\}_{q \in \mathcal{Q}}$, where $\mathbf{c}_q^k \in \mathbb{R}^k$ and \mathcal{Q} is a countable (but not necessarily finite) set of indices. A decoded vector is simply the entry of the codebook that is pointed to by a transmitted index.

The encoding with codebook vectors results in the removal of both redundancy and irrelevancy. It removes irrelevancy by introducing a reduced precision version of the vector \mathbf{x}^k , i. e., by quantizing \mathbf{x}^k . The quantized vector requires fewer bits to encode than the unquantized vector. The mechanism of the redundancy removal depends on the coding method and will be discussed in Sect. 14.3.3.

We consider the speech vector, \mathbf{X}^k , to have a continuous probability density function in \mathbb{R}^k . Thus, coding based on the finite-size speech codebook $C_{\mathbf{X}^k}$ introduces *distortion*. To minimize the distortion associated with the coding, the encoder selects the code vector (codebook entry) \mathbf{c}_q^k that is nearest to the observed vector \mathbf{x}^k according to a particular distortion measure,

$$q = \operatorname{argmin}_{q' \in \mathcal{Q}} d(\mathbf{x}^k, \mathbf{c}_{q'}^k). \quad (14.2)$$

Quantization is the operation of finding the nearest neighbor in the codebook. The set of speech vectors

that is mapped to a particular code vector \mathbf{c}_q^k is called a *quantization cell* or *Voronoi region*. We denote the Voronoi region as \mathcal{V}_q ,

$$\mathcal{V}_q = \{\mathbf{x}^k : d(\mathbf{x}^k, \mathbf{c}_q^k) < d(\mathbf{x}^k, \mathbf{c}_m^k) \forall m \neq q\}, \quad (14.3)$$

where we have ignored that generally points exist for which the inequality is not strict. These are boundary points that can be assigned to any of the cells that share the boundary.

Naturally, the average [averaged over $p_{X^k}(\cdot)$] distortion of the decoded speech vectors differs for different codebooks. A method for designing a coder is to find the codebook, i.e., the set $C_{X^k} = \{\mathbf{c}_q^k\}_{q \in Q}$, that minimizes the average distortion over the speech probability density, given a constraint on the transmission rate. It is not known how to solve this problem in a general manner. Iterative methods (the Lloyd algorithm and its variants, e.g., [14.22–24]) have been developed for the case where $|Q|$ (the *cardinality* or number of vectors in C_{X^k}) is finite. The iterative approach is not appropriate for our present discussion for two reasons. First, we ultimately are interested in structured quantizers that allow us to approximate the optimal codebook and structure is difficult to determine from the iterative method. Second, as we will see below for the constrained-entropy case, practical codebooks do not necessarily have finite cardinality. Instead of the iterative approach, we use an approach where we make simplifying assumptions, which are asymptotically accurate for high coding rates.

14.3.3 A Model of Quantization

To analyze the behavior of the speech codebook, we construct a model of the quantization (encoding–decoding) operation. (This *quantization* model is not to be confused with the probabilistic *signal* model described in the next subsection.) Thus, we make the quantization problem mathematically tractable. For simplicity, we use the squared error criterion (Sect. 14.5 shows that this criterion can be used over a wide range of coding scenarios). We also make the standard assumption that the quantization cells are convex (for any two points in a cell, all points on the line segment connecting the two points are in the cell). To construct our encoding–decoding model, we make three additional assumptions that cannot always be justified:

1. The density $p_{X^k}(\mathbf{x}^k)$ is constant within each quantization cell. This implies that the probability that a speech vector is inside a cell with index q is

$$p_Q(q) = V_q p_{X^k}(\mathbf{x}^k), \mathbf{x}^k \in \mathcal{V}_q, \quad (14.4)$$

2. where V_q is the volume of the k -dimensional cell.
2. The average distortion for speech data falling within cell q is

$$D_q = CV_q^{\frac{2}{k}}, \quad (14.5)$$

where C is a constant. The assumption made in (14.5) essentially means that the cell shape is fixed. *Gersho* [14.25], conjectured that this assumption is correct for optimal codebooks.

3. We assume that the countable set of code vectors C_{X^k} can be represented by a code-vector density, denoted as $g(\mathbf{x}^k)$. This means that the cell volume now becomes a function of \mathbf{x}^k rather than the cell index q ; we replace V_q by $V(\mathbf{x}^k)$. To be consistent we must equate the density with the inverse of the cell volume:

$$g(\mathbf{x}^k) = \frac{1}{V(\mathbf{x}^k)}. \quad (14.6)$$

The third assumption also implies that we can replace D_q by $D(\mathbf{x}^k)$.

The three assumptions listed above lead to solutions that can generally be shown to hold asymptotically in the limit of infinite rate. The theory has been observed to make reasonable predictions of performance for practical quantizers at rates down to two bits per dimension [14.26, 27], but we do not claim accuracy here. The theory serves as a vehicle to understand quantizer behavior and *not* as an accurate predictor of performance.

The code-vector density $g(\mathbf{x}^k)$ of our quantization model replaces the set of code vectors as the description of the codebook. Our objective of finding the codebook that minimizes the average distortion subject to a rate constraint has become the objective of finding the optimal density $g(\mathbf{x}^k)$ that minimizes the distortion

$$\begin{aligned} D &= \int D(\mathbf{x}^k) p_{X^k}(\mathbf{x}^k) d\mathbf{x}^k \\ &= C \int V(\mathbf{x}^k)^{\frac{2}{k}} p_{X^k}(\mathbf{x}^k) d\mathbf{x}^k \\ &= C \int g(\mathbf{x}^k)^{-\frac{2}{k}} p_{X^k}(\mathbf{x}^k) d\mathbf{x}^k, \end{aligned} \quad (14.7)$$

subject to a rate constraint.

Armed with our quantization model, we now attempt to find the optimal density $g(\mathbf{x}^k)$ (the optimal codebook) for encoding speech. We consider separately two commonly used constraints on the rate: a given fixed rate and a given average rate. As mentioned in Sect. 14.2.1, the former rate constraint applies to circuit-switched networks and the latter rate constraint represents situations

where the rate can be varied continuously, such as, for example, in storage applications and packet networks.

We start with the fixed-rate requirement, where each codebook vector c_q^k is encoded with a codeword of a fixed number of bits. This is called *constrained-resolution* coding. If we use a rate of R bits per speech vector then we have a codebook cardinality of $N = 2^R$ and the density $g(\mathbf{x}^k)$ must be consistent with this cardinality:

$$N = \int_{\mathbb{R}^k} g(\mathbf{x}^k) d\mathbf{x}^k. \quad (14.8)$$

We have to minimize the average distortion of (14.7) subject to the constraint (14.8) (i. e., subject to given N). This constrained optimization problem is readily solved with the calculus of variations. The solution is

$$\begin{aligned} g(\mathbf{x}^k) &= N \frac{p_{X^k}(\mathbf{x}^k)^{\frac{k}{k+2}}}{\int p_{X^k}(\mathbf{x}^k)^{\frac{k}{k+2}} d\mathbf{x}^k} \\ &= 2^R \frac{p_{X^k}(\mathbf{x}^k)^{\frac{k}{k+2}}}{\int p_{X^k}(\mathbf{x}^k)^{\frac{k}{k+2}} d\mathbf{x}^k} \\ &= 2^R \underline{p_{X^k}(\mathbf{x}^k)^{\frac{k}{k+2}}}, \end{aligned} \quad (14.9)$$

where the underlining denotes normalization to unit integral over \mathbb{R}^k and where R is the bit rate per speech vector. Thus, our encoding-decoding model suggests that, for constrained-resolution coding, the density of the code vectors varies with the data density. At dimensionalities $k \gg 1$ the density of the code vectors approximates a simple scaling of the probability density of the speech vectors since $k/(k+2) \rightarrow 1$ with increasing k .

In the constrained-resolution case, *redundancy* is removed by placing the codebook vectors such that they reflect the density of the data vectors. For example, as shown by (14.9), regions of \mathbb{R}^k without data have no vectors placed in them. This means no codewords are used for regions that have no data. If we had placed codebook vectors there, these would have been redundant. Note that scalar quantization of the k -dimensional random vector X^k would do precisely that. Similarly, regions of low data density get relatively few code vectors, reducing the number of codewords spent in such regions.

Next, we apply our quantization model to the case where the average rate is constrained. That is, the codeword length used to encode the cell indices q varies. Let us denote the random index associated with the random vector X^k as \mathcal{Q} . The source coding theorem [14.12] tells us the lowest possible average rate for uniquely (so it can be decoded) encoding the indices with separate codewords is within one bit of the index entropy (in bits)

$$H(Q) = - \sum_{q \in \mathcal{Q}} p_Q(q) \log_2[p_Q(q)]. \quad (14.10)$$

The entropy can be interpreted as the average of a bit allocation, $-\log_2[p_Q(q)]$, for each index q . Neglecting the aforementioned *within one bit*, the average rate constraint is $H(Q) = R$, where R is the selected rate. For this reason, this coding method is known as *constrained-entropy* coding. This neglect is reasonable as the difference can be made arbitrarily small by encoding sequences of indices, as in arithmetic coding [14.28], rather than single indices. We minimize the distortion of (14.7) subject to the constraint (14.10), i. e., subject to given $H(Q) = R$. Again, the constrained optimization problem is readily solved with the calculus of variations. In this case the solution is

$$g(\mathbf{x}^k) = 2^{H(Q)-h(X^k)} = 2^{R-h(X^k)}, \quad (14.11)$$

where $h(X^k) = - \int p_{X^k}(\mathbf{x}^k) \log_2[p_{X^k}(\mathbf{x}^k)] d\mathbf{x}^k$ is the *differential entropy* of X^k in bits, and where $H(Q)$ is specified in bits. It is important to realize that special care must be taken if $p_{X^k}(\cdot)$ is singular, i. e., if the data lie on a manifold.

Equation (14.11) implies that the the code vector density is uniform across \mathbb{R}^k . The number of code vectors is countably infinite despite the fact that the rate itself is finite. The codeword length $-\log_2[p_Q(q)]$ increases very slowly with decreasing probability $p_Q(q)$ and, roughly speaking, long codewords make no contribution to the mean rate.

In the constrained-entropy case, redundancy is removed through the lossless encoding of the indices. Given the probabilities of the code vectors, (ideal) lossless coding provides the most efficient bit assignment that allows unique decoding, and this rate is precisely the entropy of the indices. Code vectors in regions of high probability density receive short codewords and code vectors in regions of low probability density receive long codewords.

An important result that we have found for both the constrained-resolution and constrained-entropy cases is that the structure of the codebook is independent of the overall rate. The code-vector density simply increases as 2^R (cf. (14.9) and (14.11), respectively) anywhere in \mathbb{R}^k . Furthermore, for the constrained-entropy case, the code vector density depends only through the global variable $h(X^k)$ on the probability density.

14.3.4 Coding Speech with a Model Family

Although the quantization model of Sect. 14.3.3 provides interesting results, a general implementation of

a codebook for the random speech vector \mathbf{X}^k leads to practical problems, except for small k . For the constrained-resolution case, larger values of k lead to codebook sizes that do not allow for practical training procedures for storage on conventional media. For the constrained-entropy case, the codebook itself need not be stored, but we need access to the probability density of the codebook entries to determine the corresponding codewords (either offline or through computation during encoding). We can resolve these practical coding problems by using a *model* of the density. Importantly, to simplify the computational effort, we do *not* assume that the model is an accurate representation of the density of the speech signal vector, we simply make a best effort given the tools we have.

The model-based approach towards reducing computational complexity is suggested by the mixture model that we discussed in Sect. 14.3.1. If we classify each speech vector first as corresponding to a particular sound, then we can specify a probability density for that sound. A signal model specifies the probability density, typically by means of a formula for the probability density. The probability densities of the models are typically selected to be relatively simple. The signal models reduce computational complexity, either because they reduce codebook size or because the structural simplicity of the model simplifies the lossless coder. We consider models of a similar structure to be member of a *model family*. The selection of a particular model from the family is made by specifying *model parameters*.

Statistical signal models are commonly used in speech coding, with autoregressive modeling (generally referred to as linear prediction coding methods) perhaps being the most common. In this section, we discuss the selection of a particular model from the model family (i.e., the selection of the model parameters) and the balance in bit allocation between the model and the specification of the speech vector.

Our starting point is that a family of signal models is available for the coding operation. The model family can be any model family that provides a probability assignment for the speech vector \mathbf{x}^k . We discuss relevant properties for coding with signal models. We do not make the assumption that the resulting coding method is close to a theoretical performance bound on the rate versus distortion trade-off. As said, we also do not make an assumption about the appropriateness of the signal model family for the speech signal. The model probabilities may not be accurate. However it is likely that models that are based on knowledge of speech production result in better performance.

The reasoning below is based on the early descriptions of the minimum description length (MDL) principle for finding signal models [14.29–31]. These methods separate a code for the model and a code for the signal realization, making them relevant to practical speech coding methods whereas later MDL methods use a single code. Differences from the MDL work include a stronger focus on distortion, and the consideration of the constrained-resolution case, which is of no interest to modeling theory.

Constrained-Entropy Case

First we consider the constrained-entropy case, i. e., we consider the case of a uniform codebook. Each speech vector is encoded with a codebook where each cell is of identical volume, which we denote as V . Let the model distribution be specified by a set of model parameters, θ . We consider the models to have a probability density, which means that a particular parameter set θ corresponds to a particular realization of a random parameter vector Θ . We write the probability density of \mathbf{X}^k assuming the particular parameter set θ as $p_{\mathbf{X}^k|\Theta}(\mathbf{x}^k|\theta)$. The corresponding overall model density is

$$\tilde{p}_{\mathbf{X}^k}(\mathbf{x}^k) = - \sum_{\theta} p_{\mathbf{X}^k|\Theta}(\mathbf{x}^k|\theta) \cdot p_{\Theta}(\theta), \quad (14.12)$$

where the summation is over all parameter sets. The advantage of selecting and then using models $p_{\mathbf{X}^k|\Theta}(\mathbf{x}^k|\theta)$ from the family over using the composite model density $\tilde{p}_{\mathbf{X}^k}(\mathbf{x}^k)$ is a decrease of the computational effort.

The quantization model of Sect. 14.3.3 and in particular (14.4) and (14.10), show that the constrained-entropy encoding of a vector \mathbf{x}^k assuming the model with parameters θ requires $-\log_2[V p_{\mathbf{X}^k|\Theta}(\mathbf{x}^k|\theta)]$ bits. In addition, the decoder must receive side information specifying the model.

Let $\hat{\theta}(\mathbf{x}^k)$ be the parameter vector that maximizes $p_{\mathbf{X}^k|\Theta}(\mathbf{x}^k|\theta)$ and, thus, minimizes the bit allocation $-\log_2[V p_{\mathbf{X}^k|\Theta}(\mathbf{x}^k|\theta)]$. That is, $p_{\mathbf{X}^k|\Theta}[\mathbf{x}^k|\hat{\theta}(\mathbf{x}^k)]$ is the maximum-likelihood model (from the family) for encoding the speech vector \mathbf{x}^k . The random speech vectors \mathbf{X}^k do not form a countable set and as a result the random parameter vector $\hat{\Theta}(\mathbf{X}^k)$ generally does not form a countable set for conventional model families such as autoregressive models. To encode the model, we must discretize it.

To facilitate transmission of the random model index, J , the model parameters must be quantized and we write the random parameter set corresponding to random index J as $\theta(J)$. If $p_J(j)$ is a prior probability of the model index, the overall bit allocation for the vector

\mathbf{x}^k when encoded with model j is

$$\begin{aligned} l &= -\log_2[p_J(j)] - \log_2\{V(\mathbf{x}^k)p_{\mathbf{x}^k|\Theta}[\mathbf{x}^k|\theta(j)]\} \\ &= -\log_2[p_J(j)] + \log_2\left(\frac{p_{X^k|\Theta}[\mathbf{x}^k|\hat{\theta}(\mathbf{x}^k)]}{p_{X^k|\Theta}[\mathbf{x}^k|\theta(j)]}\right) \\ &\quad - \log_2[V(\mathbf{x}^k)p_{X^k|\Theta}(\mathbf{x}^k|\hat{\theta})], \end{aligned} \quad (14.13)$$

where the term $\log_2\left(\frac{p_{X^k|\Theta}[\mathbf{x}^k|\hat{\theta}(\mathbf{x}^k)]}{p_{X^k|\Theta}[\mathbf{x}^k|\theta(j)]}\right)$ represents the additional (excess) bit allocation required to encode \mathbf{x}^k with model j over the bit allocation required to encode \mathbf{x}^k with the true maximum-likelihood model from the model family.

With some abuse of notation, we denote by $j(\mathbf{x}^k)$ the function that provides the index for a given speech vector \mathbf{x}^k . In the following, we assume that the functions $\theta(j)$ and $j(\mathbf{x}^k)$ minimize l . That is, we quantize θ so as to minimize the total number of bits required to encode \mathbf{x}^k .

We are interested in the bit allocation that results from averaging over the probability density $p_{X^k}(\cdot)$ of the speech vectors,

$$\begin{aligned} E\{L\} &= -E\{\log_2[p_J(j(X^k))]\} \\ &\quad - E\left\{\log_2\left(\frac{p_{X^k|\Theta}(X^k|\theta(j(X^k)))}{p_{X^k|\Theta}(X^k|\hat{\theta}(X^k))}\right)\right\} \\ &\quad - E\{\log_2[V(X^k)p_{X^k|\Theta}(X^k|\hat{\theta}(X^k))]\}, \end{aligned} \quad (14.14)$$

where $E\{\cdot\}$ indicates averaging over the speech vector probability density and where L is the random bit allocation that has I as realization. In (14.14), the first term describes the mean bit allocation to specify the model, the second term specifies the mean excess in bits required to encode X^k assuming $\theta(j(\mathbf{x}^k))$ instead of assuming the optimal $\hat{\theta}(\mathbf{x}^k)$, and the third term specifies the mean number of bits required to encode X^k if the optimal model is available. Importantly, only the third term contains the cell volume that determines the mean distortion of the speech vectors through (14.7).

Assuming validity of the encoding model of Sect. 14.3.3, the optimal trade-off between the bit allocation for the model index and the bit allocation for the speech vectors X^k depends only on the mean of

$$\begin{aligned} \eta &= -\log_2\{p_J[j(\mathbf{x}^k)]\} \\ &\quad - \log_2\left(\frac{p_{X^k|\Theta}[\mathbf{x}^k|\theta(j(\mathbf{x}^k))]}{p_{X^k|\Theta}[\mathbf{x}^k|\hat{\theta}(\mathbf{x}^k)]}\right), \end{aligned} \quad (14.15)$$

which is referred to as the [14.32]. The goal is to find the functions $\theta(\cdot)$ and $j(\cdot)$ that minimize the index of

resolvability over the ensemble of speech vectors. An important consequence of our logic is that these functions, and therefore the rate allocation for the model index, are dependent only on the excess rate and the probability of the quantized model. As the third term of (14.14) is missing, no relation to the speech distortion exists. That is *the rate allocation for the model index J is independent of distortion and overall bit rate*. While the theory is based on assumptions that are accurate only for high bit rates, this suggests that the bit allocation for the parameters becomes proportionally more important at low rates.

The fixed entropy for the model index indicates, for example, that for the commonly used linear-prediction-based speech coders, the rate allocation for the linear prediction parameters is independent of the overall rate of the coder. As constrained-entropy coding is not commonly used for predictive coding, this result is not immediately applicable to conventional speech coders. However, the new result we derive below is applicable to such coders.

Constrained-Resolution Case

Most current speech coders were designed with a constrained-resolution (fixed-rate) constraint, making it useful to study modeling in this context. We need some preliminary results. For a given model, with parameter set θ , and optimal code vector density, the average distortion over a quantization cell centered at location \mathbf{x}^k can be written

$$\begin{aligned} D(\mathbf{x}^k) &= CV(\mathbf{x}^k)^{\frac{2}{k}} \\ &= Cg(\mathbf{x}^k)^{-\frac{2}{k}} \\ &= CN^{-\frac{2}{k}} \left[p_{X^k|\Theta}(\mathbf{x}^k|\theta)^{\frac{k}{k+2}} \right]^{-\frac{2}{k}}, \end{aligned} \quad (14.16)$$

where we have used (14.7) and (14.9). We take the expectation of (14.16) with respect to the true probability density function $p_{X^k}(\mathbf{x}^k)$ and obtain the mean distortion for the constrained-resolution case:

$$D_{\text{CR}} = CN^{-\frac{2}{k}} E\left\{\left[p_{X^k|\Theta}(X^k|\theta)^{\frac{k}{k+2}} \right]^{-\frac{2}{k}}\right\}. \quad (14.17)$$

Equation (14.17) can be rewritten as

$$\begin{aligned} \frac{2}{k} \log_2(N) &= \log_2\left(E\left\{\left[p_{X^k|\Theta}(X^k|\theta)^{\frac{k}{k+2}} \right]^{-\frac{2}{k}}\right\}\right) \\ &\quad - \log_2\left(\frac{D_{\text{CR}}}{C}\right). \end{aligned} \quad (14.18)$$

We assume that k is sufficiently large that, in the region where $p_{X^k}(\mathbf{x}^k)$ is significant, we can use the expansion

$u \approx 1 + \log(u)$ for the term $\underline{[p_{X^k|\Theta}(x^k|\theta)^{k/(k+2)}]}^{-2/k}$ and write

$$\begin{aligned} & \log \left(\mathbb{E} \left\{ \left[p_{X^k|\Theta}(x^k|\theta)^{\frac{k}{k+2}} \right]^{-\frac{2}{k}} \right\} \right) \\ & \approx \log \left(1 - \frac{2}{k} \mathbb{E} \left\{ \log \left[p_{X^k|\Theta}(x^k|\theta)^{\frac{k}{k+2}} \right] \right\} \right) \\ & \approx -\frac{2}{k} \mathbb{E} \left\{ \log \left[p_{X^k|\Theta}(x^k|\theta)^{\frac{k}{k+2}} \right] \right\}. \end{aligned} \quad (14.19)$$

Having completed the preliminaries, we now consider the encoding of a speech vector x^k . Let $L_{(m)}$ be the fixed bit allocation for the model index. The total rate is then

$$\begin{aligned} L &= L_{(m)} + L(x^k) \\ &= L_{(m)} + \log_2(N) \\ &= L_{(m)} - \mathbb{E} \left\{ \log_2 \left[p_{X^k|\Theta}(x^k|\theta)^{\frac{k}{k+2}} \right] \right\} \\ &\quad - \frac{k}{2} \log_2 \left(\frac{D_{\text{CR}}}{C} \right). \end{aligned} \quad (14.20)$$

The form of (14.20) shows that, given the assumptions made, we can define an *equivalent codeword length* $\log_2[p_{X^k|\Theta}(x^k|\theta)^{k/(k+2)}]$ for each speech codebook entry. The equivalent codeword length represents the spatial variation of the distortion. Note that this equivalent codeword length does *not* correspond to the true codeword length of the speech vector codebook, which is fixed for the constrained-resolution case. For a particular codebook vector x^k , the equivalent codeword length is

$$\begin{aligned} L &= L_{(m)} - \log_2 \left[p_{X^k|\Theta}(x^k|\theta)^{\frac{k}{k+2}} \right] \\ &\quad - \frac{k}{2} \log_2 \left(\frac{D_{\text{CR}}}{C} \right). \end{aligned} \quad (14.21)$$

Similarly to the constrained-entropy case, we can decompose (14.21) into a rate component that relates to the encoding of the model parameters, a component that describes the excess equivalent rate resulting from limiting the precision of the model parameters, and a rate component that relates to optimal encoding with optimal

(uncoded) model parameters:

$$\begin{aligned} L &= L_{(m)} - \log_2 \left(p_{X^k|\Theta}(x^k|\theta(j))^{\frac{k}{k+2}} \right) - \frac{k}{2} \log_2 \left(\frac{D_{\text{CR}}}{C} \right) \\ &= L_{(m)} - \log_2 \left(\frac{p_{X^k|\Theta}(x^k|\theta(j))^{\frac{k}{k+2}}}{p_{X^k|\Theta}(x^k|\hat{\theta}(x^k))^{\frac{k}{k+2}}} \right) \\ &\quad - \log_2 \left[p_{X^k|\Theta}(x^k|\hat{\theta}(x^k))^{\frac{k}{k+2}} \right] - \frac{k}{2} \log_2 \left(\frac{D_{\text{CR}}}{C} \right), \end{aligned} \quad (14.22)$$

We can identify the last two terms as the bit allocation for x^k for the optimal constrained-resolution model for the speech vector x^k . The second term is the excess equivalent bit allocation required to encode the speech vector with model j over the bit allocation required for the optimal model from the model family. The first two terms determine the trade-off between the bits spent on the model, and the bits spent on the speech vectors. These two terms form the index of resolvability for the constrained-resolution case:

$$\eta = L_{(m)} - \log_2 \left(\frac{p_{X^k|\Theta}(x^k|\theta(j))^{\frac{k}{k+2}}}{p_{X^k|\Theta}(x^k|\hat{\theta}(x^k))^{\frac{k}{k+2}}} \right). \quad (14.23)$$

As for the constrained-entropy case, the optimal set of functions $\theta(j)$ and $j(x^k)$ (and, therefore, the bit allocation for the model) are dependent only on the speech vector density for the constrained-resolution case. The rate for the model is independent of the distortion selected for the speech vector and of the overall rate. With increasing k , the second term in (14.23) and (14.15) becomes identical. That is the expression for the excess rate for using the quantized model parameters corresponding to model j instead of the optimal parameters is identical.

The independence of the model-parameter bit allocation of the overall codec rate for the constrained-entropy case is of great significance for practical coding systems. We emphasize again that this result is valid only under the assumptions made in Sect. 14.3.3. We expect the independence to break down at lower rates, where the codebook C_{X^k} describing the speech cannot be approximated by a density.

Table 14.1 Bit rates of the AMR-WB coder [14.4]

Rate (bits)	6.6	8.85	12.65	14.25	15.85	18.25	19.85	23.05
AR model	36	46	46	46	46	46	46	46
Pitch parameter	23	26	30	30	30	30	30	30
Excitation	48	80	144	176	208	256	288	352

The results described in this section are indeed supported, at least qualitatively, by the configuration of practical coders. Table 14.1 shows the most important bit allocations used in the adaptive-multirate wideband (**AMR-WB**) speech coder [14.4]. The **AMR-WB** coder is a constrained-resolution coder. It is seen that the design of the codec satisfies the predicted behavior: the bit allocation for the model parameters is essentially independent of the rate of the codec, except at low rates.

Model-Based Coding

In signal-model-based coding we assume the family is known to the encoder and decoder. An index to the specific model is transmitted. Each model corresponds to a unique speech-domain codebook. The advantage of the model-based approach is that the structure of the density is simplified (which is advan-

tageous for constrained-entropy coding) and that the required number of codebook entries for the constrained-resolution case is smaller. This facilitates searching through the codebook and/or the definition of the lossless coder.

The main result of this section is that we can determine the set of codebooks for the models independently of the overall rate (and speech-vector distortion). The result is consistent with existing results. The result of this section leads to fast codec design as there is no need to check the best trade-off in bit allocation between model and signal quantization.

When encoding with a model-based coding it is advantageous first to identify the *best* model, encode the model index j , and then encode the signal using codebook $C_{X^k, j}$ that is associated with that particular model j . The model selection can be made based on the index of resolvability.

14.4 Coding with Autoregressive Models

We now apply the methods of Sect. 14.3 to a practical model family. Autoregressive model families are commonly used in speech coding. In speech coding this class of coders is generally referred to as being based on *linear prediction*. We discuss coding based on a family that consists of a set of autoregressive models of a particular order (denoted as p). To match current practice, we consider the constrained-resolution case.

We first formulate the index of resolvability in terms of a spectral formulation of the autoregressive model. We show that this corresponds to the definition of a distortion measure for the model parameters. The distortion measure is approximated by the commonly used Itakura–Saito and log spectral distortion measures. Thus, starting from a squared error criterion for the speech signal, we obtain the commonly used (e.g., [14.33–37]) distortion measures for the linear-prediction parameters. Finally, we show that our reasoning leads to an estimate for the bit allocation for the model. We discuss how this result relates to results on autoregressive model estimation.

14.4.1 Spectral-Domain Index of Resolvability

Our objective is to encode a particular speech vector \mathbf{x}^k using the autoregressive model. To facilitate insight, it is beneficial to make a spectral formulation of the problem.

To this purpose, we assume that k is sufficiently large to neglect edge effects. Thus, we neglect the difference between circular and linear convolution.

The autoregressive model assumption implies that \mathbf{x}^k has a multivariate Gaussian probability density

$$p_{\mathbf{X}^k|\Theta}(\mathbf{x}^k|\theta) = \frac{1}{\sqrt{2\pi \det(\mathbf{R}_\theta)}} \exp\left(-\frac{1}{2}\mathbf{x}^{kT}\mathbf{R}_\theta^{-1}\mathbf{x}^k\right). \quad (14.24)$$

\mathbf{R}_θ is the *model* autocorrelation matrix

$$\mathbf{R}_\theta = \mathbf{A}^{-1}\mathbf{A}^{-H}, \quad (14.25)$$

where \mathbf{A} a lower-triangular Toeplitz matrix with first column $\sigma[1, a_1, a_2, \dots, a_p, 0, \dots, 0]^T$, where the a_i are the autoregressive model parameters (linear-prediction parameters), and p is the autoregressive model order and the superscript H is the Hermitian transpose. Thus, the set of model parameters is $\theta = \{\sigma, a_1, \dots, a_p\}_{i=1,\dots,p}$. We note that typically $p = 10$ for 8 kHz sampling rate and $p = 16$ for 12 kHz and 16 kHz sampling rate.

When k is sufficiently large, we can perform our analysis in terms of power spectral densities. The transfer function of the autoregressive model is

$$A(z)^{-1} = \frac{\sigma}{1 + a_1 z^{-1} + \dots + a_p z^{-p}}, \quad (14.26)$$

where σ is a gain. This corresponds to the model power spectral density

$$\mathbf{R}_\theta(z) = |A(z)|^{-2}. \quad (14.27)$$

In the following, we make the standard assumption that $A(z)$ is minimum-phase.

Next we approximate (14.24) in terms of power spectral densities and the transfer function of the autoregressive model. Using Szegő's theorem [14.38], it is easy to show that, asymptotically in k ,

$$\det(\mathbf{R}_\theta) = \exp \left\{ \frac{k}{2\pi} \int_0^{2\pi} \log [\mathbf{R}_\theta(e^{i\omega})] d\omega \right\}. \quad (14.28)$$

We also use the asymptotic equality

$$\begin{aligned} \frac{1}{2} \mathbf{x}^{kT} \mathbf{R}_\theta^{-1} \mathbf{x}^k &= \frac{1}{4\pi} \int_0^{2\pi} \frac{|x(e^{i\omega})|^2}{\mathbf{R}_\theta(e^{i\omega})} d\omega \\ &= \frac{k}{4\pi} \int_0^{2\pi} \frac{R_x(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} d\omega, \end{aligned} \quad (14.29)$$

where $x(z) = \sum_{i=0}^{k-1} x_i z^{-i}$ for $\mathbf{x}^k = (x_1, \dots, x_k)$ and $R_x(e^{i\omega}) = \frac{1}{k} |x(e^{i\omega})|^2$.

Equations (14.28) and (14.29) can be used to rewrite the multivariate density of (14.24) in terms of power spectral densities. It is convenient to write the log density:

$$\begin{aligned} \log[p_{X^k|\Theta}(\mathbf{x}^k|\theta)] &= -\frac{1}{2} \log(2\pi) - \frac{k}{4\pi} \int_0^{2\pi} \log(\mathbf{R}_\theta(e^{i\omega})) d\omega \\ &\quad - \frac{k}{4\pi} \int_0^{2\pi} \frac{R_x(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} d\omega. \end{aligned} \quad (14.30)$$

We use (14.30) to find the index of resolvability for the constrained-resolution case. We make the approximation that k is sufficiently large that it is reasonable to approximate the exponent $k/(k+2)$ by unity in (14.23). This implies that we do not have to consider the normalization in this equation. Inserting (14.30) into (14.23) results in

$$\begin{aligned} \eta &= L_{(m)} + \frac{k}{4\pi} \int_0^{2\pi} \left[-\log \left(\frac{R_{\hat{\theta}}(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} \right) \right. \\ &\quad \left. + \frac{R_x(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} - \frac{R_x(e^{i\omega})}{R_{\hat{\theta}}(e^{i\omega})} \right] d\omega. \end{aligned} \quad (14.31)$$

The maximum-likelihood estimate of the autoregressive model $\hat{\theta}$ given a data vector \mathbf{x}^k is a well-understood problem, e.g., [14.39, 40]. The predictor parameter estimate of the standard Yule–Walker solution method has the same asymptotic density as the maximum-likelihood estimate [14.41].

To find the optimal bit allocation for the model we have to minimize the expectation of (14.31) over the ensemble of all speech vectors. We study the behavior of this minimization. For notational convenience we define a cost function

$$\begin{aligned} \psi(\theta, \hat{\theta}) &= \frac{k}{4\pi} \int_0^{2\pi} \left[-\log \left(\frac{R_{\hat{\theta}}(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} \right) \right. \\ &\quad \left. + \frac{R_x(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} - \frac{R_x(e^{i\omega})}{R_{\hat{\theta}}(e^{i\omega})} \right] d\omega. \end{aligned} \quad (14.32)$$

Let θ be a particular model from a countable model set $\mathcal{C}_\Theta(L_{(m)})$ with a bit allocation $L_{(m)}$ for the model. Finding the optimal model set $\mathcal{C}_\Theta(L_{(m)})$ is then equivalent to

$$\begin{aligned} \min_{L_{(m)} \in \mathbb{N}} E[\eta] &= \min_{L_{(m)} \in \mathbb{N}} \left\{ L_{(m)} + \min_{\mathcal{C}_\Theta(L_{(m)})} E \left[\min_{\theta \in \mathcal{C}_\Theta(L_{(m)})} \psi(\theta, \hat{\theta}) \right] \right\}. \end{aligned} \quad (14.33)$$

If we write

$$D(L_{(m)}) = \min_{\mathcal{C}_\Theta(L_{(m)})} E \left[\min_{\theta \in \mathcal{C}_\Theta(L_{(m)})} \psi(\theta, \hat{\theta}) \right] \quad (14.34)$$

then (14.33) becomes

$$\min_{L_{(m)} \in \mathbb{N}} E[\eta] = \min_{L_{(m)} \in \mathbb{N}} [L_{(m)} + D(L_{(m)})]. \quad (14.35)$$

If we interpret $D(L_{(m)})$ as a minimum mean distortion, minimizing (14.35) is equivalent to finding a particular point on a rate-distortion curve. We can minimize the cost function of (14.34) for all $L_{(m)}$ and then select the $L_{(m)}$ that minimizes the overall expression of (14.35). Thus only one particular distortion level, corresponding to one particular rate, is relevant to our speech coding system. This distortion–rate pair for the model is dependent on the distribution of the speech models. Assuming that $D(L_{(m)})$ is once differentiable towards $L_{(m)}$, then (14.35) shows that its derivative should be -1 at the optimal rate for the model.

14.4.2 A Criterion for Model Selection

We started with the notion of using an autoregressive model family to quantize the speech signal. We found

that we could do so by first finding the maximum-likelihood estimate $\hat{\theta}$ of the autoregressive model parameters, then selecting from a set of models $\mathcal{C}_\Theta(L_{(m)})$ the model nearest to the maximum-likelihood model based on the cost function $\psi(\theta, \hat{\theta})$ and then quantizing the speech given the selected model. As quantization of the predictor parameters corresponds to our model selection, it is then relevant to compare the distortion measure of (14.32) with the distortion measures that are commonly used for the linear-prediction parameters in existing speech coders.

To provide insight, it is useful to write $R_x(e^{i\omega}) = R_{\hat{\theta}}(e^{i\omega})R_w(e^{i\omega})$, where $R_w(e^{i\omega})$ represents a *remainder* power-spectral density that captures the spectral error of the maximum likelihood model. If the model family is of low order, then $R_w(e^{i\omega})$ includes the spectral *fine structure*. We can rewrite (14.32) as

$$\begin{aligned} \psi(\theta, \hat{\theta}) &= \frac{k}{4\pi} \int_0^{2\pi} \left[-\log \left(\frac{R_{\hat{\theta}}(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} \right) \right. \\ &\quad \left. + \left(\frac{R_{\hat{\theta}}(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} - 1 \right) R_w(e^{i\omega}) \right] d\omega. \end{aligned} \quad (14.36)$$

Interestingly, (14.36) reduces to the well-known *Itakura–Saito criterion* [14.42] if $R_w(e^{i\omega})$ is set to unity.

It is common (e.g., [14.43]) to relate different criteria through the series expansion $u = 1 + \log(u) + \frac{1}{2}[\log(u)]^2 + \dots$. Assuming small differences between the optimal model $\hat{\theta}$ and the model from the set θ , (14.36) can be written

$$\begin{aligned} \psi(\theta, \hat{\theta}) &\cong \frac{k}{4\pi} \int_0^{2\pi} \left\{ [R_w(e^{i\omega}) - 1] \log \left(\frac{R_{\hat{\theta}}(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} \right) \right. \\ &\quad \left. + \frac{1}{2} R_w(e^{i\omega}) \left[\log \left(\frac{R_{\hat{\theta}}(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} \right) \right]^2 \right\} d\omega. \end{aligned} \quad (14.37)$$

Equation (14.37) needs to be accurate only for nearest neighbors of $\hat{\theta}$.

We can simplify (14.37) further. With our assumptions for the autoregressive models, $\mathbf{R}_\theta(z)$ is related to monic minimum-phase polynomials through (14.27) and the further assumption that their gains σ are identical (i.e., is not considered here), this implies that

$$\frac{1}{2\pi} \int_0^{2\pi} \log(\mathbf{R}_\theta) d\omega = \frac{1}{2\pi} \int_0^{2\pi} \log(R_{\hat{\theta}}) d\omega = \log(\sigma^2). \quad (14.38)$$

This means that we can rewrite (14.37) as

$$\begin{aligned} \psi(\theta, \hat{\theta}) &\cong \frac{k}{4\pi} \int_0^{2\pi} R_w(e^{i\omega}) \left\{ \log \left(\frac{R_{\hat{\theta}}(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} \right) \right. \\ &\quad \left. + \frac{1}{2} \left[\log \left(\frac{R_{\hat{\theta}}(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} \right) \right]^2 \right\} d\omega. \end{aligned} \quad (14.39)$$

Equation (14.39) forms the basic measure that must be optimized for the selection of the model from a set of models, i.e., for the optimal quantization of the model parameters.

If we can neglect the impact of $R_w(z)$, then (using the result of (14.38)) minimizing (14.39) is equivalent to minimizing

$$\psi(\theta, \hat{\theta}) \cong \frac{k}{8\pi} \int_0^{2\pi} \left[\log \left(\frac{R_{\hat{\theta}}(e^{i\omega})}{\mathbf{R}_\theta(e^{i\omega})} \right) \right]^2 d\omega, \quad (14.40)$$

which is the well-known *mean squared log spectral distortion*, scaled by the factor $k/4$. Except for this scaling factor, (14.39) is precisely the criterion that is commonly used (e.g., [14.35, 44, 45]) to evaluate performance of quantizers for autoregressive (AR) model parameters. This is not unreasonable as the neglected modeling error $R_w(e^{i\omega})$ is likely uncorrelated with the model quantization error.

14.4.3 Bit Allocation for the Model

The AR model is usually described with a small number of parameters (as mentioned, $p = 10$ is common for 8 kHz sampling rate). Thus, the spectral data must lie on a manifold of dimension p or less in the log spectrum space. At high bit allocations, where measurement noise dominates (see also the end of Sect. 14.3.3), the manifold dimension is p and the spectral distortion is expected to scale as

$$D(L_{(m)}) = \frac{k}{4} \beta^2 N_{\text{AR}}^{-\frac{2}{p}} = \frac{k}{4} \beta^2 e^{-\frac{2}{p} L_{(m)}}, \quad (14.41)$$

where β is a constant and N_{AR} is the number of spectral models in the family and $L_{(m)} = \log(N_{\text{AR}})$. At higher spectral distortion levels, it has been observed that the physics of the vocal tract constrains the dimensionality of the manifold. This means that (14.41) is replaced by

$$D(L_{(m)}) = \frac{k}{4} \beta^2 e^{-\frac{2}{\kappa} L_{(m)}} \quad (14.42)$$

with $\kappa < p$. This behavior was observed for trained codebooks over a large range in [14.44] (similar behavior was

observed for cepstral parameters in [14.46]) and for specific vowels in [14.47]. The results of [14.44] correspond to $\kappa = 7.1$ and $\beta = 0.80$.

The mean of (14.31) becomes

$$E[\eta] = L_{(m)} + \frac{k}{4} \beta^2 e^{-\frac{2}{k} L_{(m)}}. \quad (14.43)$$

Differentiating towards $L_{(m)}$ we find that the optimal bit allocation for the AR model selection to be

$$L_{(m)} = \frac{\kappa}{2} \log \left(\beta^2 \frac{k}{2\kappa} \right), \quad (14.44)$$

which is logarithmically dependent on k . Using the observed data of [14.44], we obtain an optimal rate of about 17 bits for 8 kHz sampled speech at a 20 ms block size. The corresponding mean spectral distortion is about 1.3 dB. The distortion is similar to the mean estimation errors found in experiments on linear predictive methods on speech sounds [14.48].

The 17 bit requirement for the prediction parameter quantizer is similar to that obtained by the best available prediction parameter quantizers that operate on single blocks and bounds obtained for these methods [14.35, 49–52]. In these systems the lowest bit allocation for 20 ms blocks is about 20 bits. However, the performance of these coders is entirely based on the often quoted 1 dB threshold for transparency [14.35]. The definition of this empirical threshold is consistent with the conventional two-step approach: the model parameters are first quantized using a separately defined criterion, and the speech signal is quantized thereafter based on a weighted squared error criterion. In contrast, we have shown that a single distortion measure operating on the speech vector suffices for this purpose.

We conclude that the definition of a squared-error criterion for the speech signal leads to a bit allocation for the autoregressive model. No need exists to introduce perception based thresholds on log spectral distortion.

14.4.4 Remarks on Practical Coding

The two-stage approach is standard practice in linear-prediction-based (autoregressive-model-based) speech codecs. In the selection stage, weighted squared error criteria in the so-called line-spectral frequency (LSF) representation of the prediction parameters are commonly used, e.g., [14.34–37]. If the proper weighting is used, then the criterion can be made to match the log spectral distortion measure [14.53] that we derived above.

The second stage is the selection of a speech codebook entry from a codebook corresponding to the selected model. The separation into a set of models simplifies this selection. In general, this means that a speech-domain codebook must be available for each model. It was recently shown that the computational or storage requirements for optimal speech-domain codebooks can be made reasonable by using a single codebook for each set of speech sounds that are similar except for a unitary transform [14.54]. The method takes advantage of the fact that different speech sounds may have similar statistics after a suitable unitary transform and can, therefore, share a codebook. As the unitary transform does not affect the Euclidian distance, it also does not affect the optimality of the codebook.

In the majority of codecs the speech codebooks are generated in real time, with the help of the model obtained in the first stage. This approach is the so-called *analysis-by-synthesis* approach. It can be interpreted as a method that requires the *synthesis* of candidate speech vectors (our speech codebook), hence the name. Particularly common is the usage of the analysis-by-synthesis approach for the autoregressive model [14.16, 55]. While the analysis-by-synthesis approach has proven its merit and is used in hundreds of millions of communication devices, it is not optimal. It was pointed out in [14.54] that analysis-by-synthesis coding inherently results in a speech-domain codebook with quantization cells that have a suboptimal shape, limiting performance.

14.5 Distortion Measures and Coding Architecture

An objective of coding is the removal of irrelevancy. This means that precision is lost and that we introduce a difference between the original and the decoded signal, the error signal. So far we have considered basic quantization theory and how modeling can be introduced in this quantization structure. We based our discussion on

a mean squared error distortion measure for the speech vector. As discussed in Sect. 14.2.2, the proper measure is the decrease in signal quality as perceived by human listeners. That is, the goal in speech coding is to minimize the perceived degradation resulting from an encoding at a particular rate. This section discusses

methods for integrating perceptually motivated criteria into a coding structure.

To base coding on perceived quality degradation, we must define an appropriate quantitative measure of the perceived distortion. Reasonable objectives for a good distortion measure for a speech codec are a good prediction of experimental data on human perception, mathematical tractability, low delay, and low computational requirements.

A major aspect in the definition of the criterion is the representation of the speech signal the distortion measure operates on. Most straightforward is to quantize the speech signal itself and use the distortion measure as a selection criterion for code vectors and as a means to design the quantizers. This coding structure is commonly used in speech coders based on linear-predictive coding. An alternative coding structure is to apply a transform towards a domain that facilitates a simple distortion criterion. Thus, in this approach, we first perform a mapping to a *perceptual domain* (preprocessing) and then quantize the mapped signal in that domain. At the decoder we apply the inverse mapping (postprocessing). This second architecture is common in transform coders aimed at encoding audio signals at high fidelity.

We start this section with a subsection discussing the squared error criterion, which is commonly used because of its mathematical simplicity. In Subsects. 14.5.2 and 14.5.3 we then discuss models of perception and how the squared error criterion can be used to represent these models. We end the section with a subsection discussing in some more detail the various coding architectures.

14.5.1 Squared Error

The squared-error criterion is commonly used in coding, often without proper physical motivation. Such usage results directly from its mathematical tractability. Given a data sequence, optimization of the model parameters for a model family often leads to a set of linear equations that is easily solved.

For the k -dimensional speech vector \mathbf{x}^k , the basic squared-error criterion is

$$\eta = (\mathbf{x}^k - \hat{\mathbf{x}}^k)^H (\mathbf{x}^k - \hat{\mathbf{x}}^k), \quad (14.45)$$

where the superscript ‘H’ denotes the Hermitian conjugate and $\hat{\mathbf{x}}^k$ is the reconstruction vector upon encoding and decoding. Equation (14.45) quantifies the variance of the signal error. Unfortunately, variance cannot be

equated to loudness, which is the psychological correlate of variance. At most we can expect that, for a given original signal, a scaling of the error signal leads to a positive correlation between perceived distortion and squared error.

While the squared error in its basic form is not representative of human perception, adaptive weighting of the squared-error criterion can lead to improved correspondence. By means of weighting we can generalize the squared-error criterion to a form that allows inclusion of knowledge of perception (the formulation of the weighted squared error criterion for a specific perceptual model is described in Subsects. 14.5.2 and 14.5.3). To allow the introduction of perceptual effects, we linearly weight the error vector $\mathbf{x}^k - \hat{\mathbf{x}}^k$ and obtain

$$\eta = (\mathbf{x}^k - \hat{\mathbf{x}}^k)^H \mathbf{H}^H \mathbf{H} (\mathbf{x}^k - \hat{\mathbf{x}}^k), \quad (14.46)$$

where \mathbf{H} is an $m \times k$ matrix, where m depends on the weighting invoked. As we will see below, many different models of perception can be approximated with the simple weighted squared-error criterion of (14.46). In general, the weighting matrix \mathbf{H} adapts to \mathbf{x}^k , that is $\mathbf{H}(\mathbf{x}^k)$ and

$$\mathbf{y}^m = \mathbf{H}(\mathbf{x}^k) \mathbf{x}^k \quad (14.47)$$

can be interpreted as a perceptual-domain representation of the signal vector for a region of \mathbf{x}^k where $\mathbf{H}(\mathbf{x}^k)$ is approximately constant.

The inclusion of the matrix \mathbf{H} in the formulation of the squared-error criterion generally results in a significantly higher computational complexity for the evaluation of the criterion. Perhaps more importantly, when the weighted criterion of (14.46) is adaptive, then the optimal distribution of the code vectors (Sect. 14.3.3) for constrained-entropy coding is no longer uniform in the speech domain. This has significant implications for the computational effort of a coding system.

The formulation of (14.46) is commonly used in coders that are based on an autoregressive model family, i.e., linear-prediction-based analysis-by-synthesis coding [14.16]. (The matrix \mathbf{H} then usually includes the autoregressive model, as the speech codebook is defined as a filtering of an excitation codebook.) Also in the context of this class of coders, the vector $\mathbf{H}\mathbf{x}^k$ can be interpreted as a perceptual-domain vector. However, because \mathbf{H} is a function of \mathbf{x}^k it is not straightforward to define a codebook in this domain.

The perceptual weighting matrix \mathbf{H} often represents a filter operation. For a filter with impulse response $[h_0, h_1, h_2, \dots]$, the matrix \mathbf{H} has a Toeplitz structure:

$$\mathbf{H} = \begin{pmatrix} h_0 & 0 & \dots \\ h_1 & h_0 & \dots \\ h_2 & h_1 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}. \quad (14.48)$$

For computational reasons, it may be convenient to make the matrix $\mathbf{H}^H \mathbf{H}$ Toeplitz. If the impulse response has time support p then $\mathbf{H}^H \mathbf{H}$ is Toeplitz if \mathbf{H} is selected to have dimension $(m + p) \times m$ [14.19].

Let us consider how the impulse response $[h_0, h_1, h_2, \dots]$ of (14.48) is typically constructed for the case of linear-predictive coding. The impulse response is constructed from the signal model. Let the transfer function of the corresponding autoregressive model be, as in (14.26)

$$A(z)^{-1} = \frac{\sigma}{1 + a_1 z^{-1} + \dots + a_p z^{-p}}, \quad (14.49)$$

where the a_i are the prediction parameters and σ is the gain. A weighting that is relatively flexible and has low computational complexity is then [14.56]

$$H(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)}, \quad (14.50)$$

where γ_1 and γ_2 are parameters that are selected to accurately describe the impact of the distortion on perception. The sequence $[h_0, h_1, h_2, \dots]$ of (14.48) is now simply the impulse response of $H(z)$. The parameters γ_1 and γ_2 are selected to approximate perception where $1 \geq \gamma_1 > \gamma_2 > 0$. The filter $A(z/\gamma_1)$ deemphasizes the envelope of the power spectral density, which corresponds to decreasing the importance of spectral peaks. The filter $1/A(z/\gamma_2)$ undoes some of this emphasis for a smoothed version of the spectral envelope. The effect is roughly that $1/A(z/\gamma_2)$ limits the spectral reach of the deemphasis $A(z/\gamma_1)$. In other words, the deemphasis of the spectrum is made into a local effect.

To understand coders of the transform model family, it is useful to interpret (14.46) in the frequency domain. We write the discrete Fourier transform (DFT) as the unitary matrix \mathbf{F} and define a frequency-domain weighting matrix \mathbf{W} such that

$$\mathbf{H}\mathbf{x}^k = \mathbf{F}^H \mathbf{W} \mathbf{F}\mathbf{x}^k, \quad (14.51)$$

The matrix \mathbf{W} provides a weighting of the frequency-domain vector $\mathbf{F}\mathbf{x}^k$. If, for the purpose of our discussion, we neglect the difference between circular and linear convolution and if \mathbf{H} represents a filtering operation (convolution) as in (14.48), then \mathbf{W} is diagonal. To account for perception, we must adapt \mathbf{W} to the input vector \mathbf{x}^k (or equivalently, to the frequency-domain vector $\mathbf{F}\mathbf{x}^k$) and it becomes a function $\mathbf{W}(\mathbf{x}^k)$: Equation (14.50) could be used as a particular mechanism for such weighting. However, in the transform coding context, so-called *masking* methods, which are described in Sect. 14.5.2, are typically used to find $\mathbf{W}(\mathbf{x}^k)$.

As mentioned before, the random vector $\mathbf{Y}^m = \mathbf{H}\mathbf{X}^m$ (or, equivalently, the vector $\mathbf{W}\mathbf{F}\mathbf{X}^m$) can be considered as a perceptual-domain description. Assuming smooth behavior of $\mathbf{H}(\mathbf{x}^k)$ as a function of \mathbf{x}^k , this domain can then be used as the domain for coding. A codebook must be defined for the perceptual-domain vector \mathbf{Y}^m and we select entries from this codebook with the unweighted squared error criterion. This approach is common in transform coding. When this coding in the perceptual domain is used, the distortion measure does not vary with the vector \mathbf{y}^m , and a uniform quantizer is optimal for \mathbf{Y}^m for the constrained-entropy case. If the mapping to the perceptual domain is unique and invertible (which is not guaranteed by the formulation), then $\mathbf{y}^m = \mathbf{H}(\mathbf{x}^k)\mathbf{x}^k$ ensures that \mathbf{x}^k is specified when \mathbf{y}^m is known and only indices to the codebook for \mathbf{Y}^m need to be encoded. In practice, the inverse mapping may not be unique, resulting in problems at block boundaries and the inverse may be difficult to compute. As a result it is common practice to quantize and transmit the weighting \mathbf{W} , e.g., [14.57, 58].

14.5.2 Masking Models and Squared Error

Extensive quantitative knowledge of auditory perception exists and much of the literature on quantitative descriptions of auditory perception relates to the concept of *masking*, e.g., [14.59–63]. The masking-based description of the operation of the auditory periphery can be used to include the effect of auditory perception in speech and audio coding. Let us define an arbitrary signal that we call the *masker*. The masker implies a set of second signals, called *maskees*, which are defined as signals that are not audible when presented in the presence of the masker. That is, the maskee is below the *masking threshold*. Masking explains, for example, why a radio must be made louder in a noisy environment such as a car. We can think of masking as being

a manifestation of the internal precision of the auditory periphery.

In general, laws for the masking threshold are based on psychoacoustic measurements for the masker and maskee signals that are constructed independently and then added. However, it is clear that the coding error is correlated to the original signal. In the context of masking it is a commonly overlooked fact that, for ideal coding, the coding error signal is, under certain common conditions, independent of the reconstructed signal [14.12]. Thus, a reasonable objective of audio and speech coding is to ensure that the coding error signal is below the masking threshold of the *reconstructed* signal.

Masking is quantified in terms of a so-called masking curve. We provide a generalized definition of such a curve. Let us consider a signal vector \mathbf{x}^k with k samples that is defined in \mathbb{R}^k . We define a perceived-error measurement domain by any invertible mapping $\mathbb{R}^k \rightarrow \mathbb{R}^m$. Let $\{\mathbf{e}_i^m\}_{i \in \{0, \dots, m-1\}}$ be the unit-length basis vectors that span \mathbb{R}^m . We then define the m -dimensional *masking curve* as [14.64] JND_i , $i \in 0, \dots, m-1$, where the scalar JND_i is the *just-noticeable difference* (JND) for the basis vector \mathbf{e}_i^m . That is, the vector $JND_i \mathbf{e}_i^m$ is precisely at the threshold of being audible for the given signal vector.

Examples of the masking curve can be observed in the time and the frequency domain. The frequency-domain representation of \mathbf{x}^k is $\mathbf{F}\mathbf{x}^k$. *Simultaneous* masking is defined as the masking curve for $\mathbf{F}\mathbf{x}^k$, i.e., the just-noticeable amplitudes for the frequency unit vectors $\mathbf{e}_1, \mathbf{e}_2$, etc. In the time domain we refer to *nonsimultaneous* (or *forward* and *backward*) masking depending on whether the time index i of the unit vectors \mathbf{e}_i is prior to or after the main event in the masker (e.g., an onset). Both the time-domain (temporal) and frequency-domain masking curves are asymmetric and dependent on the loudness of the masker. A loud sound leads to a rapid decrease in auditory acuity, followed by a slow recovery to the default level. The recovery may take several hundreds of ms and causes forward masking. The decrease in auditory acuity before a loud sound, backward masking, extends only over very short durations (at most a few ms). Similar asymmetry occurs in the frequency domain, i.e., in simultaneous masking. Let us consider a tone. The auditory acuity is decreased mostly at frequencies higher than the tone. The acuity increases more rapidly from the masker when moving towards lower frequencies than when moving towards higher frequencies, which is related to the decrease in frequency resolution with

increasing frequency. A significant difference exists in the masking between tonal and noise-like signals. We refer to [14.63, 65, 66] for further information on masking.

The usage of masking is particularly useful for coding in the perceptual domain with a constraint that the quality is to be transparent (at least according to the perceptual knowledge provided). For example, consider a transform coder (based on either the discrete cosine transform or the DFT). In this case, the quantization step size can be set to be the JND as provided by the simultaneous masking curve [14.57].

Coders are commonly subject to a bit-rate constraint, which means knowledge of the masking curve is not sufficient. A distortion criterion must be defined based on the perceptual knowledge given. A common strategy in audio coding to account for simultaneous masking is to use a weighted squared error criterion, with a diagonal weighting matrix \mathbf{H} that is reciprocal of the masking threshold [14.27, 58, 67–70]. In fact, this is a general approach that is useful to convert a masking curve in any measurement domain:

$$\mathbf{H} = \begin{pmatrix} \frac{1}{JND_1} & 0 & \cdots \\ 0 & \frac{1}{JND_2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}, \quad (14.52)$$

where it is understood that the weighting matrix \mathbf{H} is defined in the measurement domain. To see this consider the effect of the error vector $JND_i \mathbf{e}_i^m$ on the squared error:

$$\begin{aligned} \eta &= JND_i^2 \mathbf{e}_i^{mH} \mathbf{H}^H \mathbf{H} \mathbf{e}_i^m \\ &= JND_i^2 JND_i^{-2} = 1. \end{aligned} \quad (14.53)$$

Thus, the points on the masking curve are defined as the amplitudes of basis vectors that lead to a unit distortion. This is a reasonable motivation for the commonly used reciprocal-weighting approach for the squared-error criterion defined by the weighting described in (14.52). However, it should be noted that for this formulation the distortion measure does not vanish below the masking threshold. A more-complex approach where the distortion measure does vanish below the JND is given in [14.71].

14.5.3 Auditory Models and Squared Error

The weighting procedure of (14.52) (possibly in combination with the transform to the measurement domain)

is an operation that transforms the signal to a perceptually relevant domain. Thus, the operation can be interpreted as a simple auditory model. Sophisticated models of the auditory periphery that directly predict the input to the auditory nerve have also been developed, e.g., [14.72–77]. Despite the existence of such quantitative models of perception, their application in speech coding has been limited. Only a few examples [14.78, 79] of the explicit usage of existing quantitative knowledge of auditory perception in speech coding exist. In contrast, in the field of audio coding the usage of quantitative auditory knowledge is common. Transform coders can be interpreted as methods that perform coding in the perceptual domain, using a simple perceptual model, usually based on (simultaneous) masking results.

We can identify a number of likely causes for the lack of usage of auditory knowledge in speech coding. First, the structure of speech coders and the constraint on computational complexity naturally leads to speech-coding-specific models of auditory perception, such as (14.50). The parameters of these simple speech-coding-based models are optimized directly based on coding performance. Second, the perception of the periodicity nature of voiced speech, often referred to as the perception of *pitch*, is not well understood in a quantitative manner. It is precisely the distortion associated with the near-periodic nature of voiced speech that is often critical for the perceived quality of the reconstructed signal. An argument against using a quantitative model based on just-noticeable differences (JNDs) is that JNDs are often exceeded significantly in speech coding. While the weighting of (14.52) is reasonable near the JND threshold value, it may not be accurate in the actual operating region of the speech coder. Major drawbacks of using sophisticated models based on knowledge of the auditory periphery are that they tend to be computationally expensive, have significant latency, and often lead to a representation that has many more dimensions than the input signal. Moreover the complexity of the model structure makes inversion difficult, although not impossible [14.80].

The complex structure of auditory models that describe the functionality of the auditory periphery is time invariant. We can replace it by a much simpler structure at the cost of making it time variant. That is, the mapping from the speech domain to the perceptual domain can be simplified by approximating this mapping as locally linear [14.79]. Such an approximation leads to the *sensitivity matrix approach*, which was first introduced in a different context by [14.53] and

described in a rigorous general manner in [14.81]. If a mapping from the speech domain vector \mathbf{x}^k to an auditory domain vector \mathbf{y}^m (as associated with a particular model of the auditory periphery) can be approximated as locally linear, then for a small coding error $\mathbf{x}^k - \hat{\mathbf{x}}^k$, we can write $\mathbf{y}^m - \hat{\mathbf{y}}^m$ as a matrix multiplication of $\mathbf{x}^k - \hat{\mathbf{x}}^k$:

$$\mathbf{y}^m - \hat{\mathbf{y}}^m \approx \mathbf{H}(\mathbf{x}^k - \hat{\mathbf{x}}^k), \quad (14.54)$$

where \mathbf{H} is an $m \times k$ matrix. This means that, for the set of codebook vectors from C_{X^k} that is sufficiently close to \mathbf{x}^k , the squared-error criterion of (14.46) forms an approximation to the psychoacoustic measure. Moreover, selecting the nearest codebook entry from C_{X^k} using (14.46) results in the globally optimal codebook vector for the input vector \mathbf{x}^k . In the sensitivity matrix approach, the first step for each speech vector \mathbf{x}^k is to find the $k \times k$ *sensitivity matrix* $\mathbf{H}^T \mathbf{H}$. This operation is based on an analysis of the distortion criterion [14.79]. Once this has been done, the selection of the codebook entries is similar to that for a signal-invariant weighted squared-error distortion measure.

In the sensitivity matrix approach, the matrix \mathbf{H} is a function of the past and future signal:

$$\mathbf{H} = \mathbf{H}(\dots, \mathbf{x}_{i-1}^k, \mathbf{x}_i^k, \mathbf{x}_{i+1}^k, \dots), \quad (14.55)$$

where \mathbf{x}_i^k is the current speech vector, \mathbf{x}_{i-1}^k is the previous speech vector, etc. To avoid the introduction of latency, the future speech vectors can be replaced by a prediction of these vectors from the present and past speech vectors.

The sensitivity matrix approach requires that the mapping from speech domain to perceptual domain is continuous and differentiable, which is not the case for psychoacoustic models. The approximation of such discontinuities by continuous functions generally leads to satisfactory results.

The sensitivity matrix approach is well motivated in the context of a speech-domain codebook C_{X^k} and a criterion that consists of a perceptual transform followed by the squared error criterion. The search through the speech-domain codebook with a perceptual criterion then reduces to searching with a weighted squared-error criterion.

The benefit of the sensitivity matrix approach is not so obvious if the signal vector codebook, C_{X^k} , is defined in the perceptual domain. However, it can be useful if the perceptual transform is known to the decoder (by, for example, transmission of an index). The perceptual domain often has higher dimensionality than the corresponding

speech block. The singular-value decomposition of \mathbf{H} can then be used to reduce the dimensionality of the perceptual domain error vector to k . Let $\mathbf{H} = \mathbf{V}\mathbf{D}\mathbf{U}$ be a singular value decomposition, where \mathbf{V} is an $m \times m$ unitary matrix, \mathbf{D} is a $m \times k$ diagonal matrix and \mathbf{U} is a unitary $k \times k$ matrix

14.5.4 Distortion Measure and Coding Architecture

As we have seen, the distortion measure has a significant impact on the architecture of a speech coder. In this subsection, we summarize the above discussion from a codec-architecture viewpoint.

Speech-Domain Codebook

The most straightforward architecture for a speech coder is to define the codebook in the speech domain and use an appropriate distortion measure during encoding and during training of the codebook. In general, the measure is adaptive. This approach, which is shown in Fig. 14.1, is most common in speech coding. An advantage is that the decoder does not require knowledge of the time-varying distortion measure.

We saw in Sect. 14.3.4 that it can be advantageous to use speech codebooks that are associated with models. This simplifies the codebook structure and, in the case of constrained-resolution coding, its size. The codebooks can be generated in real time as, for example, in the case of linear-prediction (autoregressive model)-based analysis-by-synthesis coding [14.16, 55].

The underlying aim of the speech-domain codebook architecture is generally to approximate auditory perception by an adaptively weighted squared-error criterion. Usually, this criterion is heuristic and based on tuning within the context of the coder (as is typically done for (14.50)). However, as shown in Sect. 14.5.3, the sensitivity analysis method facilitates the usage of complex auditory models. This approach requires, at least in principle, no further tuning.

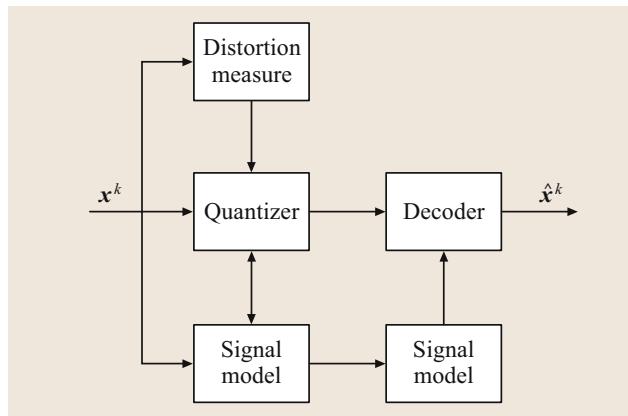


Fig. 14.1 Common architecture for coding with a distortion measure. A signal quantization index and a signal model index are transmitted

Disadvantages of the weighted squared-error criterion are its computational complexity and that, in contrast to the unweighted squared-error measure, it does not lead to uniform codebooks for the constrained-entropy case. In the context of autoregressive-model-based analysis-by-synthesis coding, many procedures have been developed to reduce the computational complexity of the weighted squared-error criterion [14.18, 19, 82].

Perceptual-Domain Codebook

As an alternative to defining the codebook in the speech domain, we can define the codebook in a perceptual domain, as is shown in Fig. 14.2. We define a perceptual domain as a domain where the unweighted squared error criterion can be applied. The most elegant paradigm requires no information about the speech vector other than its index in the perceptual domain codebook. This elegance applies if the mapping to the perceptual domain is injective (one to one). Then if y^m is known, x^k is also known. An example is an auditory model that is a weighted DFT or DCT with a one-to-one function $\mathbb{R}^k \rightarrow \mathbb{R}^k$ that maps x^k into y^k . An inverse func-

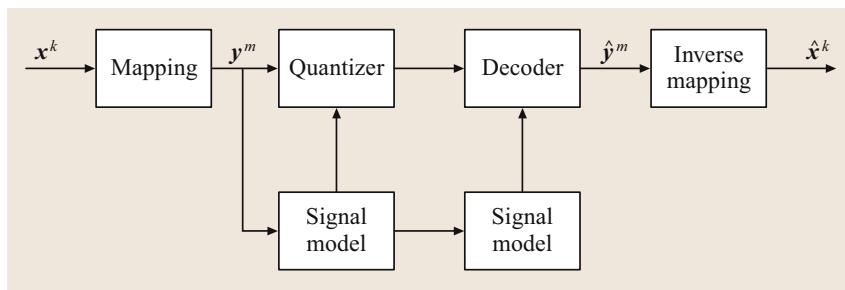


Fig. 14.2 Ideal architecture for coding in the perceptual domain, with invertible mapping. A perceptual-domain quantization index and a signal model index are transmitted. The signal model can be omitted

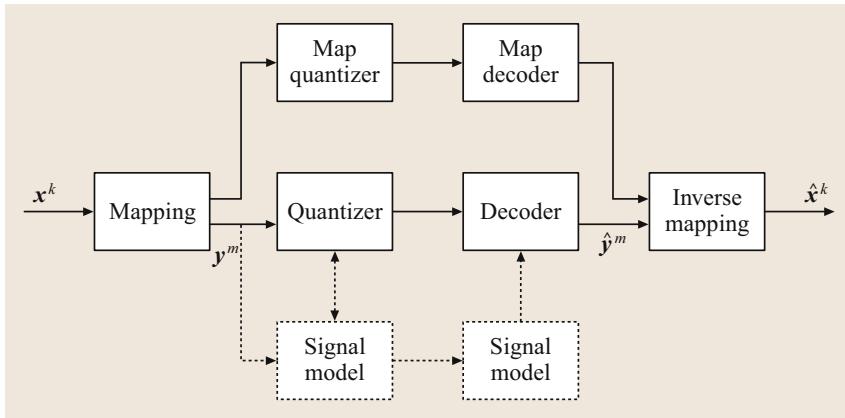


Fig. 14.3 Architecture for coding in the perceptual domain with encoded mapping. This architecture is common in transform coding. A mapping index, a perceptual-domain quantization index, and a signal model index are transmitted. The signal model is commonly omitted

tion can be derived by the decoder from the quantized vector \hat{y}^k .

The non-uniqueness of the auditory mapping from the speech domain to the auditory domain results in practical problems. Particularly if the models are not accurate, the nonuniqueness can result in mismatches between coding blocks and severe audible distortion.

The uniqueness issue for the mapping can be solved, at the cost of increased rate, by transmitting information

about the mapping as is shown in Fig. 14.3. For example, for transform coders used for audio signals, the masking curve is commonly transmitted, e.g., [14.52, 58, 67–69]. To reduce the rate required, this is commonly done on a per-frequency-band basis. The bands generally are uniformly spaced on an equivalent rectangular bandwidth (ERB) or mel scale. In practice the masking curve is not always transmitted directly, but a maximum amplitude, and the number of quantization levels for each band are transmitted, e.g., [14.27, 68, 70].

14.6 Summary

This chapter discussed the principles underlying the transmission of speech (and audio) signals. The main attributes of coding, rate, quality, robustness to channel errors, delay, and computational complexity were discussed first. We then provided a generic perspective of speech coding.

Each block of speech samples was described as a random vector. Information about the vector was transmitted in the form of a codebook index. The relation between the reconstruction vector density and the data density was given. We then modeled the probability density of the speech vector as a weighted sum of component probability density functions, each describing the speech vector probability density for a particular speech sound. Each such component density function corresponds to a *model*. In the case of linear-prediction-based (equivalent to autoregressive-model-based) coding, each component function (and thus model) is characterized by a set of predictor parameters. The approach results in the standard two-step speech coding approach, in which we first extract the model and then code the speech vector given the model.

We showed that this approach leads to standard distortion measures used for the quantization of the predictor parameters.

We showed that the number of models (component probability density functions) is independent of the overall coding rate. Thus, the rate spent on linear-prediction parameters in linear-predictive coding should not vary with rate (at least when the rate is high). It was shown that practical coders indeed have this behavior. We emphasized also that the rate allocated to the model (the predictor parameters) is not a direct function of a perceptual threshold, but the result of an optimal trade-off between the rate allocated for the speech given the model and the rate allocated for the model. We showed that it is possible to calculate the rate allocation for the model (the bit allocation for the predictor parameters) and that the result provided is close to practical codec configurations. We discussed analysis-by-synthesis coding as a particular application of the two-stage coding method. We noted that analysis-by-synthesis coding is not optimal because the speech-domain codebook has suboptimal quantization cell shapes.

Finally we discussed how perception can be integrated into the coding structure. We distinguished coding in a perceptually relevant domain, which is commonly used in audio coding, from coding in the speech-signal domain, which is commonly used in speech coding. The advantage of coding in the per-

ceptual domain is that a simple squared-error criterion can be used. However, in practice the method generally requires some form of encoding of the mapping, so that the decoder can perform an inverse mapping. Improved mapping procedures may change this requirement.

References

- 14.1 W.B. Kleijn, K.K. Paliwal: An introduction to speech coding. In: *Speech Coding and Synthesis*, ed. by W.B. Kleijn, K.K. Paliwal (Elsevier, Amsterdam 1995) pp. 1–47
- 14.2 R.V. Cox: Speech coding standards. In: *Speech Coding and Synthesis*, ed. by W.B. Kleijn, K.K. Paliwal (Elsevier, Amsterdam 1995) pp. 49–78
- 14.3 R. Salami, C. Laflamme, J. Adoul, A. Kataoka, S. Hayashi, T. Moriya, C. Lamblin, D. Massaloux, S. Proust, P. Kroon, Y. Shoham: Design and description of CS-ACELP: a toll quality 8 kb/s speech coder, *IEEE Trans. Speech Audio Process.* **6**(2), 116–130 (1998)
- 14.4 B. Bessette, R. Salami, R. Lefebvre, M. Jelinek, J. Rotola-Pukkila, J. Vainio, H. Mikkola: The adaptive multirate wideband speech codec (amr-wb), *IEEE Trans. Speech Audio Process.* **6**(8), 620–636 (2002)
- 14.5 ITU-T Rec. P.800: *Methods for Subjective Determination of Transmission Quality* (1996)
- 14.6 A.W. Rix: Perceptual speech quality assessment – a review, *Proc. IEEE ICASSP*, Vol. 3 (2004) pp. 1056–1059
- 14.7 S. Möller: *Assessment and Prediction of Speech Quality in Telecommunications* (Kluwer Academic, Boston 2000)
- 14.8 P. Kroon: Evaluation of speech coders. In: *Speech Coding and Synthesis*, ed. by W.B. Kleijn, K.K. Paliwal (Elsevier, Amsterdam 1995) pp. 467–493
- 14.9 W. Stallings: *High-speed networks: TCP/IP and ATM design principles* (Prentice Hall, Englewood Cliffs 1998)
- 14.10 Information Sciences Institute: Transmission control protocol, IETF RFC793 (1981)
- 14.11 J. Postel: User datagram protocol, IETF RFC768 (1980)
- 14.12 T.M. Cover, J.A. Thomas: *Elements of Information Theory* (Wiley, New York 1991)
- 14.13 N. Kitawaki, K. Itoh: Pure delay effects on speech quality in telecommunications, *IEEE J. Sel. Area. Comm.* **9**(4), 586–593 (1991)
- 14.14 J. Cox: The minimum detectable delay of speech and music, *Proc. IEEE ICASSP*, Vol. 1 (1984) pp. 136–139
- 14.15 J. Chen: A robust low-delay CELP speech coder at 16 kb/s. In: *Advances in Speech Coding*, ed. by B.S. Atal, V. Cuperman, A. Gersho (Kluwer Academic, Dordrecht 1991) pp. 25–35
- 14.16 B.S. Atal, M.R. Schroeder: Stochastic coding of speech at very low bit rates, *Proc. Int. Conf. Comm.* (1984) pp. 1610–1613
- 14.17 J.-P. Adoul, P. Mabilieu, M. Delprat, S. Morisette: Fast CELP coding based on algebraic codes, *Proc. IEEE ICASSP* (1987) pp. 1957–1960
- 14.18 I.M. Trancoso, B.S. Atal: Efficient procedures for selecting the optimum innovation in stochastic coders, *IEEE Trans. Acoust. Speech* **38**(3), 385–396 (1990)
- 14.19 W.B. Kleijn, D.J. Krasinski, R.H. Ketchum: Fast methods for the CELP speech coding algorithm, *IEEE Trans. Acoust. Speech* **38**(8), 1330–1342 (1990)
- 14.20 T. Lookabough, R. Gray: High-resolution theory and the vector quantizer advantage, *IEEE Trans. Inform. Theory* **IT-35**(5), 1020–1033 (1989)
- 14.21 S. Na, D. Neuhoff: Bennett's integral for vector quantizers, *IEEE Trans. Inform. Theory* **41**(4), 886–900 (1995)
- 14.22 S.P. Lloyd: Least squares quantization in PCM, *IEEE Trans. Inform. Theory* **IT-28**, 129–137 (1982)
- 14.23 Y. Linde, A. Buzo, R.M. Gray: An algorithm for vector quantizer design, *IEEE Trans. Commun.* **COM-28**, 84–95 (1980)
- 14.24 P. Chou, T. Lookabough, R. Gray: Entropy-constrained vector quantization, *IEEE Trans. Acoust. Speech* **38**(1), 31–42 (1989)
- 14.25 A. Gersho: Asymptotically optimal block quantization, *IEEE Trans. Inform. Theory* **25**, 373–380 (1979)
- 14.26 P. Swaszek, T. Ku: Asymptotic performance of unrestricted polar quantizers, *IEEE Trans. Inform. Theory* **32**(2), 330–333 (1986)
- 14.27 R. Vafin, W.B. Kleijn: Entropy-constrained polar quantization and its application to audio coding, *IEEE Trans. Speech Audio Process.* **13**(2), 220–232 (2005)
- 14.28 J.J. Rissanen, G. Langdon: Arithmetic coding, *IBM J. Res. Devel.* **23**(2), 149–162 (1979)
- 14.29 J. Rissanen: Modeling by the shortest data description, *Automatica* **14**, 465–471 (1978)
- 14.30 J. Rissanen: A universal prior for integers and estimation by minimum description length, *Ann. Stat.* **11**(2), 416–431 (1983)

- 14.31** P. Grunwald: A tutorial introduction to the minimum description length principle. In: *Advances in Minimum Description Length: Theory and Applications*, ed. by P. Grunwald, I.J. Myung, M. Pitt (MIT, Boston 2005)
- 14.32** A. Barron, T.M. Cover: Minimum complexity density estimation, *IEEE Trans. Inform. Theory* **37**(4), 1034–1054 (1991)
- 14.33** A.H. Gray, J.D. Markel: Distance measures for speech process, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-24**(5), 380–391 (1976)
- 14.34** R. Hagen, P. Hedelin: Low bit-rate spectral coding in CELP a new LSP method, *Proc. IEEE ICASSP* (1990) pp.189–192
- 14.35** K.K. Paliwal, B.S. Atal: Efficient vector quantization of LPC parameters at 24 bits/frame, *IEEE Trans. Speech Audio Process.* **1**(1), 3–14 (1993)
- 14.36** C. Xydeas, C. Papanastasiou: Split matrix quantization of lpc parameters, *IEEE Trans. Speech Audio Process.* **7**(2), 113–125 (1999)
- 14.37** A. Subramaniam, B. Rao: Speech LSF quantization with rate independent complexity, bit scalability, and learning, *Proc. IEEE ICASSP* (2001) pp.705–708
- 14.38** U. Grenander, G. Szego: *Toeplitz Forms and their Applications* (Chelsea, New York 1984)
- 14.39** F. Itakura, S. Saito: Speech information compression based on the maximum likelihood estimation, *J. Acoust. Soc. Jpn.* **27**(9), 463 (1971)
- 14.40** S. Saito, K. Nakata: *Fundamentals of Speech Signal Process* (Academic, New York 1985)
- 14.41** P.J. Brockwell, R.A. Davis: *Time Series: Theory and Methods* (Springer, New York 1996)
- 14.42** F. Itakura, S. Saito: Analysis Synthesis Telephony Based Upon the Maximum Likelihood Method, Reports of 6th Int. Cong. Acoust.,C-5-5, C17-20, ed. by Y. Kohasi (1968)
- 14.43** R.M. Gray, A. Buzo, A.H. Gray, Y. Matsuyama: Distortion measures for speech process, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-28**(4), 367–376 (1980)
- 14.44** K.K. Paliwal, W.B. Kleijn: Quantization of LPC parameters. In: *Speech Coding and Synthesis*, ed. by W.B. Kleijn, K.K. Paliwal (Elsevier, Amsterdam 1995) pp. 433–466
- 14.45** W.R. Gardner, B.D. Rao: Noncausal all-pole modeling of voiced speech, *IEEE Trans. Speech Audio Process.* **5**(1), 1–10 (1997)
- 14.46** M. Nilsson, W.B. Kleijn: Shannon entropy estimation based on high-rate quantization theory, *Proc. EUSIPCO* (2004) pp.1753–1756
- 14.47** M. Nilsson: *Entropy and Speech* (Royal Institute of Technology, Stockholm 2006), Ph.D. dissertation, KTH
- 14.48** C. Lamm: *Improved Spectral Estimation in Speech Coding* (Lund Institute of Technology (LTH), Lund 1998), Master's thesis
- 14.49** K.L.C. Chan: Split-dimension vector quantization of parcor coefficients for low bit rate speech cod-
- ing, *IEEE Trans. Speech Audio Process.* **2**(3), 443–446 (1994)
- 14.50** A. Subramaniam, B.D. Rao: PDF optimized parametric vector quantization of speech line spectral frequencies, *IEEE Speech Coding Workshop* (Delavan 2000) pp. 87–89
- 14.51** P. Hedelin, J. Skoglund: Vector quantization based on Gaussian mixture models, *IEEE Trans. Speech Audio Process.* **8**(4), 385–401 (2000)
- 14.52** S. Srinivasan, J. Samuelsson, W.B. Kleijn: Speech enhancement using a-priori information with classified noise codebooks, *Proc. EUSIPCO* (2004) pp.1461–1464
- 14.53** W.R. Gardner, B.D. Rao: Optimal distortion measures for the high rate vector quantization of LPC parameters, *Proc. IEEE ICASSP* (1995) pp. 752–755
- 14.54** M.Y. Kim, W.B. Kleijn: KLT-based adaptive classified vector quantization of the speech signal, *IEEE Trans. Speech Audio Process.* **12**(3), 277–289 (2004)
- 14.55** P. Kroon, E.F. Deprettere: A class of analysis-by-synthesis predictive coders for high quality speech coding at rates between 4.8 and 16 kbit/s, *IEEE J. Sel. Area. Commun.* **6**(2), 353–363 (1988)
- 14.56** J. Chen, A. Gershoff: Real-time vector APC speech coding at 4–800 bps with adaptive postfiltering, *Proc. IEEE ICASSP* (1987) pp. 2185–2188
- 14.57** J. Johnston: Transform coding of audio signals using perceptual noise criteria, *IEEE J. Sel. Area. Commun.* **6**(2), 314–323 (1988)
- 14.58** H. Malvar: Enhancing the performance of subband audio coders for speech signals, *Proc. IEEE Int. Symp. on Circ. Syst.*, Vol. 5 (1998) pp. 98–101
- 14.59** R. Veldhuis: Bit rates in audio source coding, *IEEE J. Sel. Area. Commun.* **10**(1), 86–96 (1992)
- 14.60** B.C.J. Moore: Masking in the human auditory system. In: *Collected papers on digital audio bit-rate reduction*, ed. by N. Gilchrist, C. Grewin (Audio Eng. Soc., New York 1996)
- 14.61** B.C.J. Moore: *An Introduction to the Psychology of Hearing* (Academic, London 1997)
- 14.62** E. Zwicker, H. Fastl: *Psychoacoustics* (Springer Verlag, Berlin, Heidelberg 1999)
- 14.63** T. Painter, A. Spanias: Perceptual coding of digital audio, *Proc. IEEE* **88**(4), 451–515 (2000)
- 14.64** J.H. Plasberg, W.B. Kleijn: The sensitivity matrix: Using advanced auditory models in speech and audio processing, *IEEE Trans. Speech Audio Process.* **15**, 310–319 (2007)
- 14.65** J.L. Hall: Auditory psychophysics for coding applications. In: *The Digital Signal Processing Handbook*, ed. by V.K. Madisetti, D. Williams (CRC, Boca Raton 1998) pp. 39.1–39.25
- 14.66** W. Jesteadt, S.P. Bacon, J.R. Lehman: Forward masking as a function of frequency, masker level and signal delay, *J. Acoust. Soc. Am.* **71**(4), 950–962 (1982)
- 14.67** D. Sinha, J.D. Johnston: Audio compression at low bit rates using a signal adaptive switched

- filterbank, Proc. IEEE ICASSP, Vol. 2 (1996) pp. 1053–1056
- 14.68 T. Verma, T. Meng: A 6 kbps to 85 kbps scalable audio coder, Proc. IEEE ICASSP, Vol. 2 (2000) pp. II877–II880
- 14.69 A.S. Scheuble, Z. Xiong: Scalable audio coding using the nonuniform modulated complex lapped transform, Proc. IEEE ICASSP, Vol. 5 (2001) pp. 3257–3260
- 14.70 R. Heusdens, R. Vafin, W.B. Kleijn: Sinusoidal modeling using psychoacoustic-adaptive matching pursuits, IEEE Signal Proc. Lett. **9**(8), 262–265 (2002)
- 14.71 M.Y. Kim, W.B. Kleijn: Resolution-constrained quantization with JND based perceptual-distortion measures, IEEE Signal Proc. Lett. **13**(5), 304–307 (2006)
- 14.72 O. Ghitza: Auditory nerve representation as a basis for speech processing. In: *Advances in Speech Signal Processing* (Dekker, New York 1992) pp. 453–485
- 14.73 T. Dau, D. Püschel, A. Kohlrausch: A quantitative model of the effective signal processing in the auditory system. I. Model structure, J. Acoust. Soc. Am. **99**(6), 3615–3622 (1996)
- 14.74 T. Dau, B. Kollmeier, A. Kohlrausch: Modeling auditory processing of amplitude modulation. I. detection and masking with narrowband car-
- riers, J. Acoust. Soc. Am. **102**(5), 2892–2905 (1997)
- 14.75 G. Kubin, W.B. Kleijn: On speech coding in a perceptual domain, Proc. IEEE ICASSP, Vol. I (1999) pp. 205–208
- 14.76 F. Baumgarte: *Ein psychophysiologisches Gehörmodell zur Nachbildung von Wahrnehmungsschwellen für die Audiocodierung* (Univ. Hannover, Hannover 2000), Ph.D. dissertation (in German)
- 14.77 S. van de Par, A. Kohlrausch, G. Charestan, R. Heusdens: A new psychoacoustical masking model for audio coding applications, Proc. IEEE ICASSP (2002) pp. 1805–1808
- 14.78 D. Sen, D. Irving, W. Holmes: Use of an auditory model to improve speech coders, Proc. IEEE ICASSP (1993) pp. II411–II414
- 14.79 J.H. Plasberg, D.Y. Zhao, W.B. Kleijn: The sensitivity matrix for a spectro-temporal auditory model, Proc. EUSIPCO (2004) pp. 1673–1676
- 14.80 X. Yang, K. Wang, S. Shamma: Auditory representation of acoustic signals, IEEE Trans. Inform. Theory **38**(2), 824–839 (1996)
- 14.81 T. Linder, R. Zamir, K. Zeger: High-resolution source coding for non-difference measures: the rate-distortion function, IEEE Trans. Inform. Theory **45**(2), 533–547 (1999)
- 14.82 I. Gerson, M. Jasiuk: Vector sum excited linear prediction (VSELP), Proc. IEEE ICASSP (1990) pp. 461–464

Chapter 1

How is speech processed in a cell phone conversation?

T. Dutoit([°]), N. Moreau(*), P. Kroon(+)

([°]) Faculté Polytechnique de Mons, Belgium

(*) Ecole Nationale Supérieure des Télécommunications, Paris, France

(+) LSI, Allentown, PA, USA

*Every cell phone solves 10 linear equations
in 10 unknowns every 20 milliseconds*

Although most people see the cell phone as an extension of conventional wired phone service or POTS (plain old telephone service), the truth is that cell phone technology is extremely complex and a marvel of technology. Very few people realize that these small devices perform hundreds of millions of operations per second to be able to maintain a phone conversation. If we take a closer look at the module that converts the electronic version of the speech signal into a sequence of bits, we see that for every 20 ms of input speech, a set of speech model parameters is computed and transmitted to the receiver. The receiver converts these parameters back into speech. In this chapter, we will see how linear predictive (LP) analysis–synthesis lies at the very heart of mobile phone transmission of speech. We first start with an introduction to linear predictive speech modeling and follow with a MATLAB-based proof of concept.

1.1 Background – Linear predictive processing of speech

Speech is produced by an excitation signal generated in our throat, which is modified by resonances produced by different shapes of our vocal, nasal,

and pharyngeal tracts. This excitation signal can be the glottal pulses produced by the periodic opening and closing of our vocal folds (which creates *voiced* speech such as the vowels in “voice”), or just some continuous air flow pushed by our lungs (which creates *unvoiced* speech such as the last sound in “voice”), or even a combination of both at the same time (such as the first sound in “voice”).

The periodic component of the glottal excitation is characterized by its fundamental frequency F_o (Hz) called *pitch*¹. The resonant frequencies of the vocal, oral, and pharyngeal tracts are called *formants*. On a spectral plot of a speech frame, pitch appears as narrow peaks for fundamental and harmonics; formants appear as wide peaks of the envelope of the spectrum (Fig. 1.1).

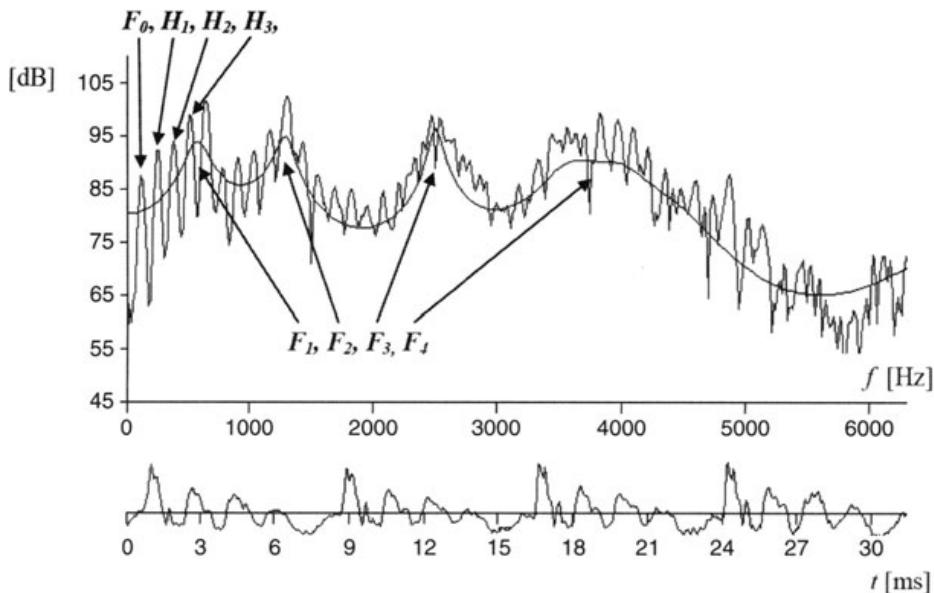


Fig. 1.1 A 30- ms frame of voiced speech (*bottom*) and its spectrum (shown here as the magnitude of its FFT). Harmonics are denoted as H_1, H_2, H_3 , etc.; formants are denoted as F_1, F_2, F_3 , etc. The spectral envelope is shown here for convenience; it implicitly appears only in the regular FFT

1.1.1 The LP model of speech

As early as 1960, Fant proposed a linear model of speech production (Fant 1960), termed as the *source-filter model*, based on the hypothesis that the

¹ Strictly speaking, pitch is defined as the *perceived* fundamental frequency.

glottis and the vocal tract are fully uncoupled. This model led to the well-known *autoregressive* (AR) or *linear predictive* (LP)² model of speech production (Rabiner and Shafer 1978), which describes speech $s(n)$ as the output $\tilde{s}(n)$ of an *all-pole* filter $1/A_p(z)$ excited by $\tilde{e}(n)$:

$$\tilde{S}(z) = \tilde{E}(z) \frac{1}{\sum_{i=0}^p a_i z^{-i}} = \tilde{E}(z) \frac{1}{A_p(z)} \quad (a_0 = 1) \quad (1.1)$$

where $\tilde{S}(z)$ and $\tilde{E}(z)$ are the Z transforms of the speech and excitation signals, respectively, and p is the *prediction order*. The excitation of the LP model (Fig. 1.2) is assumed to be either a sequence of regularly spaced pulses (whose period T_0 and amplitude σ can be adjusted) or white Gaussian noise (whose variance σ^2 can be adjusted), thereby implicitly defining the so-called voiced/unvoiced (V/UV) decision. The filter $1/A_p(z)$ is termed as the *synthesis filter* and $A_p(z)$ is called the *inverse filter*.

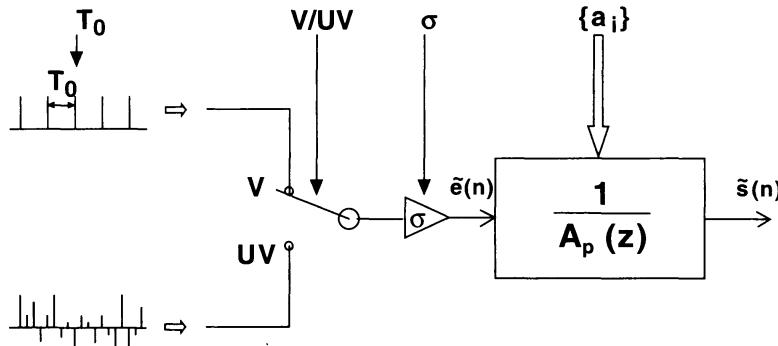


Fig. 1.2 The LP model of speech production

Equation (1.1) implicitly introduces the concept of linear predictability of speech (hence the name of the model), which states that each speech sample can be expressed as a weighted sum of the p previous samples, plus some excitation contribution:

$$\tilde{s}(n) = \tilde{e}(n) - \sum_{i=1}^p a_i \tilde{s}(n-i) \quad (1.2)$$

² Sometimes it is denoted as the *LPC model (linear predictive coding)* because it has been widely used for speech coding.

1.1.2 The LP estimation algorithm

From a given signal, a practical problem is to find the best set of prediction coefficients – that is, the set that minimizes modeling errors – by trying to minimize audible differences between the original signal and the one that is produced by the model of Fig. 1.2. This implies to estimate the value of the LP parameters: pitch period T_0 , gain σ , V/UV switch position, and prediction coefficients $\{a_i\}$.

Pitch and voicing (V/UV) determination is a difficult problem. Although speech seems periodic, it is never truly the case. Glottal cycle amplitude varies from period to period (*shimmer*) and its period itself is not constant (*jitter*). Moreover, the speech waveform reveals only filtered glottal pulses rather than glottal pulses themselves. This makes a realistic measure of T_0 even more complex. In addition, speech is rarely completely voiced; its additive noise components make pitch determination even harder. Many techniques have been developed to estimate T_0 (see Hess 1992; de la Cuadra 2007).

The estimation of σ and of the prediction coefficients can be performed simultaneously and fortunately independently of the estimation of T_0 .

For a given speech signal $s(n)$, imposing the value of the $\{a_i\}$ coefficients in the model results in the *prediction residual* signal, $e(n)$:

$$e(n) = s(n) + \sum_{i=1}^p a_i s(n-i) \quad (1.3)$$

which is simply the output of the inverse filter excited by the speech signal (Fig. 1.3).

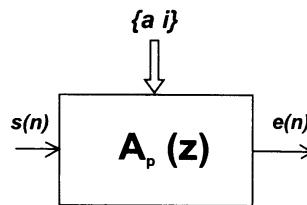


Fig. 1.3 Inverse filtering of speech

The principle of AR estimation is to choose the set $\{a_1, a_2, \dots, a_p\}$, which minimizes the expectation $E(e^2(n))$ of the residual energy:

$$\{a_i\}^{opt} = \arg \min_{a_i} (E(e^2(n))) \quad (1.4)$$

As a matter of fact, it can be shown that, if $s(n)$ is stationary, the synthetic speech $\tilde{s}(n)$ produced by the LP model (Fig. 1.2) using this specific set of prediction coefficients in Equation (1.2) will exhibit the same spectral envelope as $s(n)$. Since the excitation of the LP model (pulses or white noise) has a flat spectral envelope, this means that the frequency response of the synthesis filter will approximately match the spectral envelope of $s(n)$ and that the spectral envelope of the LP residual will be approximately flat. In a word, inverse filtering decorrelates speech.

Developing the LMSE (least mean squared error) criterion (1.4) easily leads to the so-called set of p Yule-Walker linear equations:

$$\begin{bmatrix} \phi_{xx}(0) & \phi_{xx}(1) & \dots & \phi_{xx}(p-1) \\ \phi_{xx}(1) & \phi_{xx}(0) & \dots & \phi_{xx}(p-2) \\ \dots & \dots & \dots & \dots \\ \phi_{xx}(p-1) & \phi_{xx}(p-2) & \dots & \phi_{xx}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_p \end{bmatrix} = - \begin{bmatrix} \phi_{xx}(1) \\ \phi_{xx}(2) \\ \dots \\ \phi_{xx}(p) \end{bmatrix} \quad (1.5)$$

in which $\phi_{xx}(k)$ ($k = 0 \dots p$) are the $p+1$ first autocorrelation coefficients of $s(n)$. After solving this set of equations, the optimal value of σ is then given by the following equation:

$$\sigma^2 = \sum_{i=0}^p a_i \phi_{xx}(i) \quad (1.6)$$

It should be noted that since Equations (1.5) are based only on the autocorrelation function of $s(n)$, the model does not try to imitate the exact speech waveform, but rather its spectral envelope (based on the idea that our ear is more sensitive to the amplitude spectrum than to the phase spectrum).

1.1.3 LP processing in practice

Since speech is nonstationary, the LP model is applied on speech *frames* (typically 30 ms long, with an overlap of 20 ms; Fig. 1.4) in which the

signal is assumed to be stationary given the inertia of the articulatory muscles³.

Speech samples are usually weighted using a *weighting window* (typically a 30-ms-long Hamming window). This prevents the first samples of each frame, which cannot be correctly predicted, from having too much weight in Equation (1.4) by producing higher values of $e^2(n)$.

The $\phi_{xx}(k)$ ($k=0\dots p$) autocorrelation coefficients are then estimated on a limited number of samples (typically 240 samples, for 30 ms of speech with a sampling frequency of 8 kHz). The prediction order p (which is also the number of poles in the all-pole synthesis filter) is chosen such that the resulting synthesis filter has enough degrees of freedom to copy the spectral envelope of the input speech. Since there is approximately one formant per kilohertz of bandwidth of speech, at least $2B$ poles are required (where B is the signal bandwidth in kHz, i.e., half the sampling frequency). Two more poles are usually added for modeling the glottal cycle waveform (and also empirically, because the resulting LPC speech sounds better). For telephone-based applications, working with a sampling frequency of 8 kHz, this leads to $p=10$.

Although Equation (1.5) can be solved with any classical matrix inversion algorithm, the so-called *Levinson–Durbin* algorithm is preferred for its speed, as it takes into account the special structure of the matrix (all elements on diagonals parallel to the principal diagonal are equal; this characterizes a *Toeplitz* matrix). See Rabiner and Schafer (1978) or Quatieri (2002) for details.

The *prediction coefficients* $\{a_i\}$ are finally computed for every frame (i.e., typically every 10–20 ms).

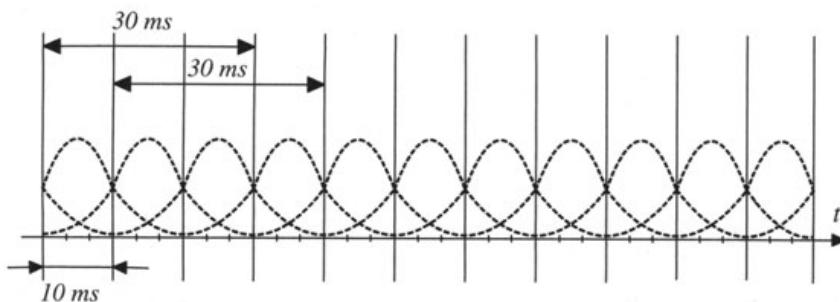


Fig. 1.4 Frame-based processing of speech (shown here with a frame length of 30 ms and a shift of 10 ms)

³ In practice, this is only an approximation, which tends to be very loose for plosives, for instance.

The complete block diagram of an LPC speech analysis–synthesis system is given in Fig. 1.5.

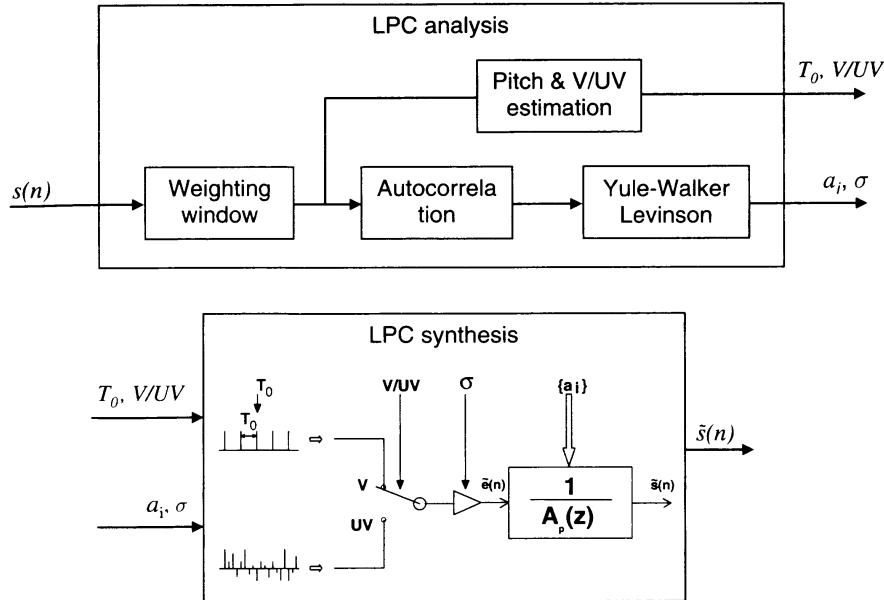


Fig. 1.5 A linear predictive speech analysis–synthesis system

1.1.4 Linear predictive coders

The LPC analysis–synthesis system, which has been described above, is not exactly the one embedded in cell phones.

It is, however, implemented in the so-called NATO LPC10 standard (NATO, 1984), which was used for satellite transmission of speech communications until 1996. This norm makes it possible to encode speech with a bit rate as low as 2,400 bits/s (frames are 22.5 ms long, and each frame is coded with 54 bits: 7 bits for pitch and V/UV decision, 5 bits for the gain, and 42 bits for the prediction coefficients⁴). In practice, prediction coefficients are actually not used as such; the related *reflection coefficients* or *log area ratios* are preferred, since they have better quantization properties. Quantization of prediction coefficients can result in unstable filters.

The number of bits in LPC10 was chosen such that it does not bring audible artifacts to the LPC speech. The example LPC speech produced in Section 1.2 is therefore a realistic example of typical LPC10 speech.

⁴ Advanced LP coders, such as CELP, have enhanced prediction coefficients coding down to 30 bits.

Clearly this speech coder suffers from the limitations of the poor (and binary!) excitation model. Voiced fricatives, for instance, cannot be adequately modeled since they exhibit voiced *and* unvoiced features simultaneously. Moreover, the LPC10 coder is very sensitive to the efficiency of its voiced/unvoiced detection and F_0 estimation algorithms. Female voices, whose higher F_0 frequency sometimes results in a second harmonic at the center of the first formant, often lead to F_0 errors (the second harmonic being mistaken for F_0).

One way of enhancing the quality of LPC speech is obviously to reduce the constraints on the LPC excitation so as to allow for a better modeling of the prediction residual $e(n)$ by the excitation $\tilde{e}(n)$. As a matter of fact, passing this residual through the synthesis filter $1/A(z)$ produces the original speech (Fig. 1.6, which is the inverse of Fig. 1.3).

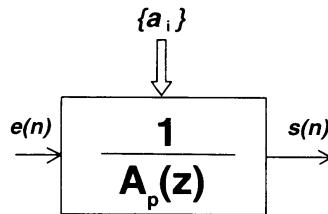


Fig. 1.6 Passing the prediction residual through the synthesis filter produces the original speech signal

The *multipulse excited* (MPE; Atal and Remde 1982) was an important step in this direction, as it was the first approach to implement an analysis-by-synthesis process (i.e., a closed loop) for the estimation of the excitation features. The MPE excitation is characterized by the positions and amplitudes of a limited number of pulses per frame (typically 10 pulses per 10 ms frame; Fig. 1.7). Pitch estimation and voiced/unvoiced decision are no longer required. Pulse positions and amplitudes are chosen iteratively (Fig. 1.8) so as to minimize the energy of the modeling error (the difference between the original speech and the synthetic speech). The error is filtered by a *perceptual filter* before its energy is computed:

$$P(z) = \frac{A(z)}{A(z/\gamma)} \quad (1.7)$$

The role of this filter, whose frequency response can be set to any intermediate between all pass response ($\gamma=1$) and the response of the

inverse filter ($\gamma=0$), is to reduce the contributions of the formants to the estimation of the error. The value of γ is typically set to 0.8.

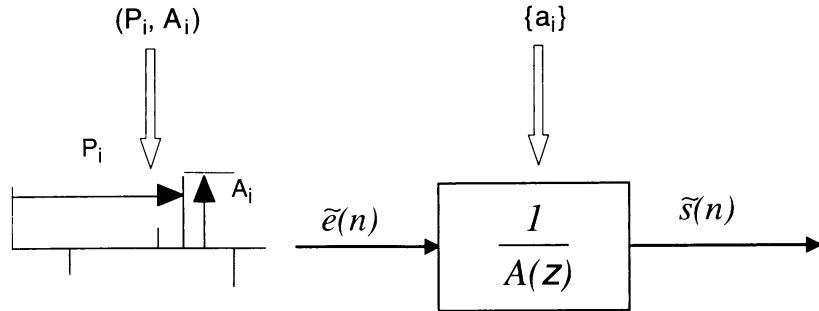


Fig. 1.7 The MPE decoder

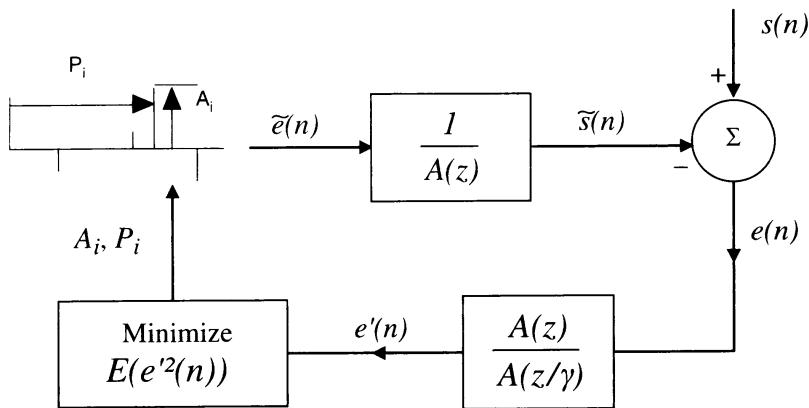


Fig. 1.8 Estimation of the MPE excitation by an analysis-by-synthesis loop in the MPE encoder

The *code-excited linear prediction* (CELP) coder (Schroeder and Atal, 1985) further extended the idea of analysis-by-synthesis speech coding by using the concept of *vector quantization* (VQ) for the excitation sequence. In this approach, the encoder selects one excitation sequence from a predefined *stochastic codebook* of possible sequences (Fig. 1.9) and sends only the index of the selected sequence to the decoder, which has a similar codebook. Although the lowest quantization rate for scalar quantization is 1 bit per sample, VQ allows fractional bit rates. For example, quantizing two samples simultaneously using a 1-bit codebook will result in 0.5 bits per sample. More typical values are a 10-bit codebook with codebook vectors of dimension 40, resulting in 0.25 bits per sample. Given the very high variability of speech frames, however (due to changes in glottal excitation *and* vocal tract), vector-quantized speech frames would be

possible only with a very large codebook. The great idea of CELP is precisely to perform VQ on LP residual sequences: as we have seen in Section 1.1.2, the LP residual has a flat spectral envelope, which makes it easier to produce a small but somehow exhaustive codebook of LP residual sequences. CELP can thus be seen as an *adaptive vector quantization* scheme of speech frames (adaptation being performed by the synthesis filter).

CELP additionally takes advantage of the periodicity of voiced sounds to further improve predictor efficiency. A so-called *long-term predictor* filter is cascaded with the synthesis filter, which enhances the efficiency of the codebook. The simplest long-term predictor consists of a simple variable delay with adjustable gain (Fig. 1.10).

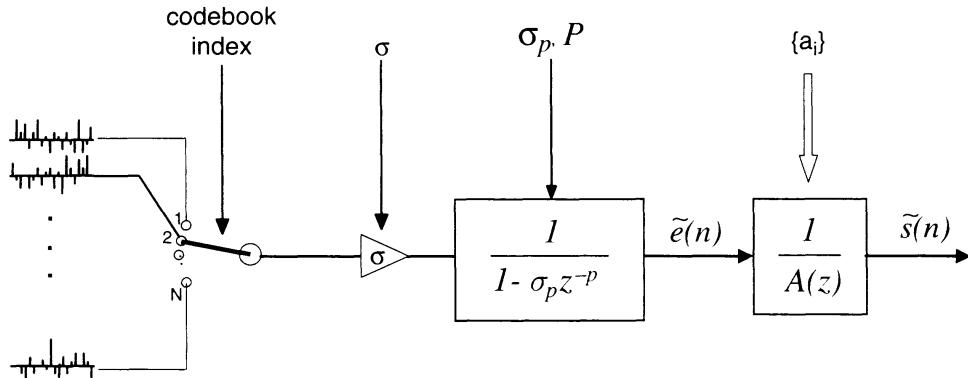


Fig. 1.9 The CELP decoder

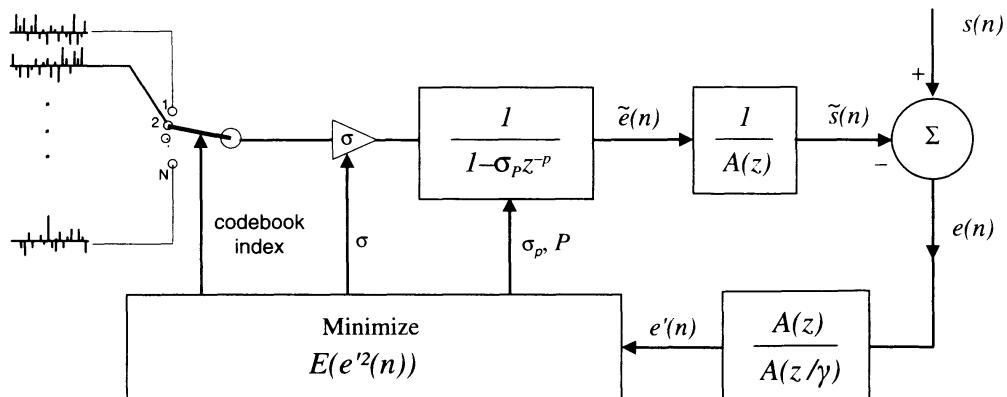


Fig. 1.10 Estimation of the CELP excitation by an analysis-by-synthesis loop in the CELP encoder

Various coders have been developed after MPE and CELP using the same analysis-by-synthesis principle with the goal of enhancing CELP quality while further reducing bit rate, among which are the mixed-excitation linear prediction (MELP; McCree and Barnwell, 1995) and the harmonic and vector excitation coding (HVXC; Matsumoto et al. 1997). In 1996, LPC-10 was replaced by MELP to be the United States Federal Standard for coding at 2.4 kbps.

From 1992 to 1996, GSM (global system for mobile communication) phones embedded a particular form of MPE, the *RPE-LPC* (regular pulse excited; Kroon et al. 1986) coder, with additional constraints on the positions of the pulses: the RPE pulses were evenly spaced (but their amplitude, as well as the position of the first pulse, is left open). Speech is divided into 20 ms frames, each of which is encoded as 260 bits, giving a total bit rate of 13 kbps. In 1996, this so-called *full-rate* (FR) codec was replaced by the *enhanced full-rate* (EFR) codec, implementing a variant of CELP termed as algebraic-CELP (ACELP, Salami et al. 1998). The ACELP codebook structure allows efficient searching of the optimal codebook index thereby eliminating one of the main drawbacks of CELP which is its complexity. The EFR coder operates at 11.2 kbps and produces better speech quality than the FR coder at 13 kb/s. A variant of the ACELP coder has been standardized by ITU-T as G.729 for operation at a bit rate of 8 kbps. Newer generations of coders that are used in cell phones are all based on the CELP principle and can operate at bit rates as low as 4.75 – 11.2 kbps.

1.2 MATLAB proof of concept : **ASP_cell_phone.m**

We will first examine the contents of a speech file (Section 1.2.1) and perform LP analysis and synthesis on a voiced (Section 1.2.2) and an unvoiced frame (Section 1.2.3). We will then generalize this approach to the complete speech file by first synthesizing all frames as voiced and imposing a constant pitch (Section 1.2.4), then by synthesizing all frames as unvoiced (Section 1.2.5), and finally by using the original pitch⁵ and voicing information as in LPC10 (Section 1.2.6). We will conclude this section by changing LPC10 into CELP (Section 1.2.7).

⁵ By “original pitch,” we mean the pitch that can be measured on the original signal.

1.2.1 Examining a speech file

Let us load file “speech.wav,” listen to it, and plot its samples (Fig. 1.11). This file contains the sentence “Paint the circuits” sampled at 8 kHz, with 16 bits.⁶

```
speech=wavread('speech.wav');
plot(speech)
xlabel('Time (samples)');
ylabel('Amplitude');
sound(speech,8000);
```

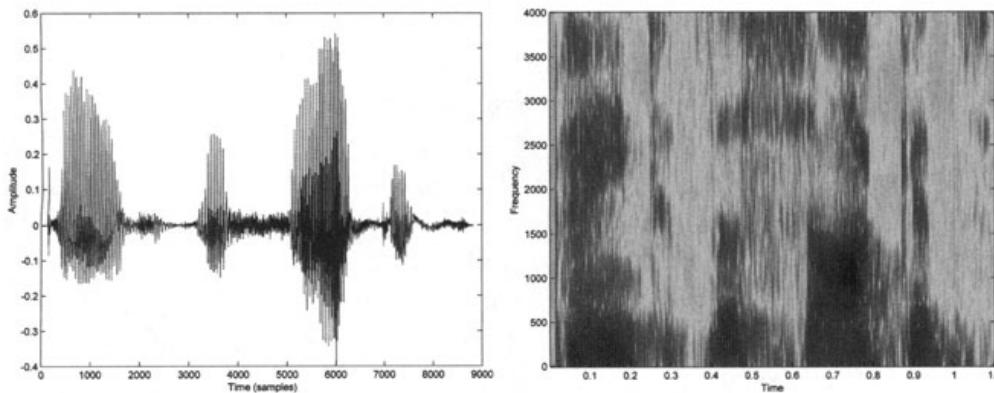


Fig. 1.11 Input speech: the speech.wav file (*left*: waveform; *right*: spectrogram)

The file is about 1.1 s long (9,000 samples). One can easily spot the position of the four vowels appearing in this plot, since vowels usually have higher amplitude than other sounds. The vowel “e” in “the”, for instance, is approximately centered on sample 3,500.

As such, however, the speech waveform is not “readable,” even by an expert phonetician. Its information content is hidden. In order to reveal it to the eyes, let us plot a spectrogram of the signal (Fig. 1.11). We then choose a *wideband spectrogram*⁷ by imposing the length of each frame to be approximately 5 ms long (40 samples) and a hamming weighting window.

```
specgram(speech,512,8000,hamming(40))
```

In this plot, pitch periods appear as vertical lines. As a matter of fact, since the length of analysis frames is very small, some frames fall on the

⁶ This sentence was taken from the Open Speech Repository on the web.

⁷ A wideband spectrogram uses a small amount of samples (typically less than the local pitch period) so as to better reveal formants.

peaks (resp., on the valleys) of pitch periods and thus appear as a darker (resp., lighter) vertical lines.

In contrast, formants (resonant frequencies of the vocal tract) appear as dark (and rather wide) horizontal traces. Although their frequency is not easy to measure with precision, experts looking at such a spectrogram can actually often read it (i.e., guess the corresponding words). This clearly shows that formants are a good indicator of the underlying speech sounds.

1.2.2 Linear prediction synthesis of 30 ms of voiced speech

Let us extract a 30-ms frame from a voiced part (i.e., 240 samples) of the speech file and plot its samples (Fig. 1.12).

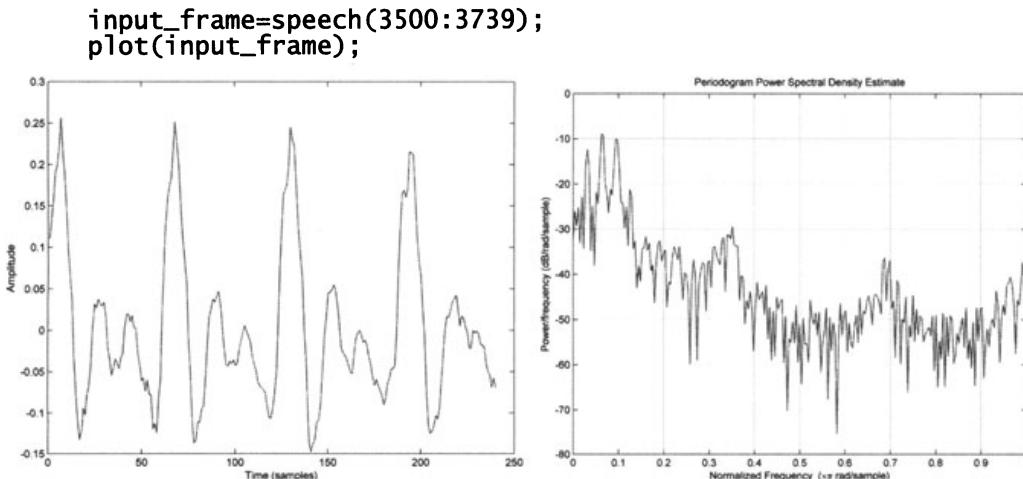


Fig. 1.12 A 30-ms-long voiced speech frame taken from a vowel (left: waveform; right: periodogram)

As expected this sound is approximately periodic (period=65 samples, i.e., 80 ms; fundamental frequency = 125 Hz). Note, though, that this is only apparent; in practice, no sequence of samples can be found more than once in the frame.

Now let us see the spectral content of this speech frame (Fig. 1.12) by plotting its *periodogram* on 512 points (using a normalized frequency axis; remember π corresponds to $F_s/2$, i.e., to 4,000 Hz here).

```
periodogram(input_frame, [], 512);
```

The fundamental frequency appears again at around 125 Hz. One can also roughly estimate the position of formants (peaks in the spectral envelope) at ± 300 , 1,400, and 2,700 Hz.

Let us now fit an LP model of order 10 to our voiced frame.⁸ We obtain the prediction coefficients (`ai`) and the variance of the residual signal (`sigma_square`).

```
[ai, sigma_square]=lpc(input_frame,10);
sigma=sqrt(sigma_square);
```

The estimation parameter inside LPC is called the Levinson–Durbin algorithm. It chooses the coefficients of an FIR filter $A(z)$ so that when passing the input frame into $A(z)$, the output, termed as the prediction residual, has minimum energy. It can be shown that this leads to a filter which has anti-resonances wherever the input frame has a formant. For this reason, the $A(z)$ filter is termed as the “inverse” filter. Let us plot its frequency response (on 512 points) and superimpose it to that of the “synthesis” filter $1/A(z)$ (Fig. 1.13).

```
[HI,WI]=freqz(ai, 1, 512);
[H,W]=freqz(1,ai, 512);
plot(W,20*log10(abs(H)),'-',WI,20*log10(abs(HI)), '--');
```

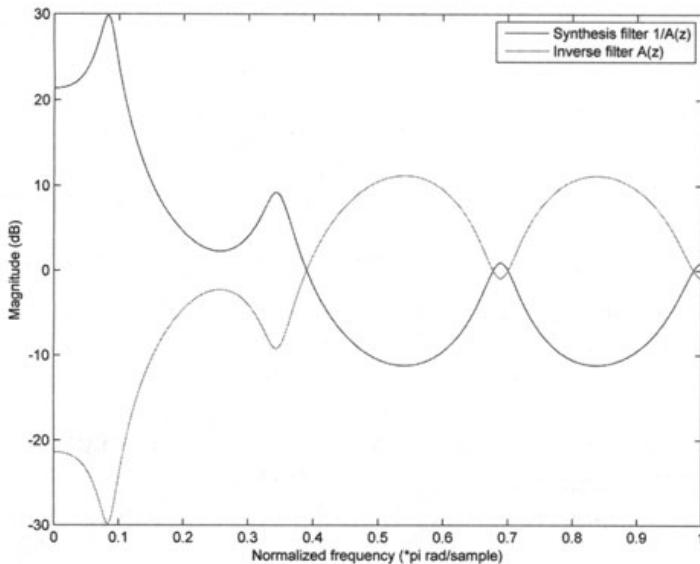


Fig. 1.13 Frequency responses of the inverse and synthesis filters

⁸ We do not apply windowing prior to LP analysis now, as it has no tutorial benefit. We will add it in subsequent sections.

In other words, the frequency response of the filter $1/A(z)$ matches the spectral amplitude envelope of the frame. Let us superimpose this frequency response to the periodogram of the vowel (Fig. 1.14).⁹

```
periodogram(input_frame,[],512,2)
hold on;
plot(w/pi,20*log10(sigma*abs(H)));
hold off;
```

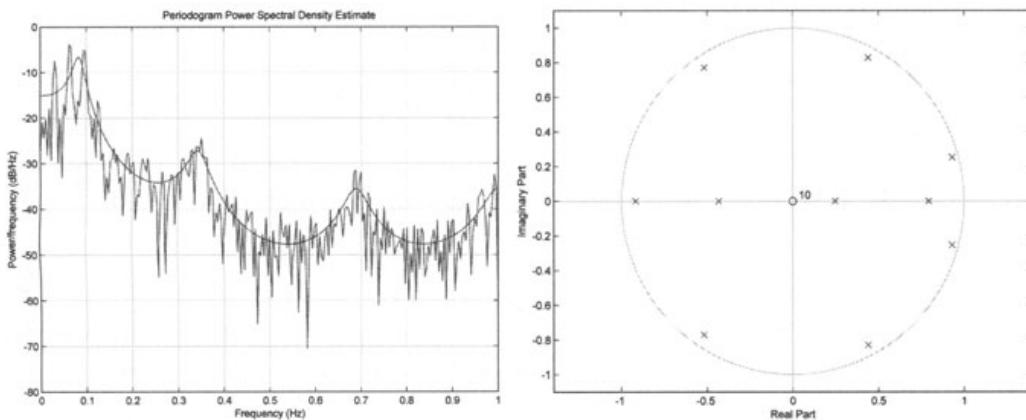


Fig. 1.14 *Left:* Frequency response of the synthesis filter superimposed with the periodogram of the frame; *right:* poles and zeros of the filter

In other words, the LPC fit has automatically adjusted the poles of the synthesis filter close to the unit circle at angular positions chosen to imitate formant resonances (Fig. 1.14).

```
zplane(1,ai);
```

If we apply the inverse of this filter to the input frame, we obtain the prediction residual (Fig. 1.15).

```
LP_residual=filter(ai,1,input_frame);
plot(LP_residual)
periodogram(LP_residual,[],512);
```

⁹ The `periodogram` function of MATLAB actually shows the so-called *one-sided periodogram*, which has twice the value of the two-sided periodogram in $[0, F_s/2]$. In order to force MATLAB to show the real value of the two-sided periodogram in $[0, F_s/2]$, we claim $F_s = 2$.

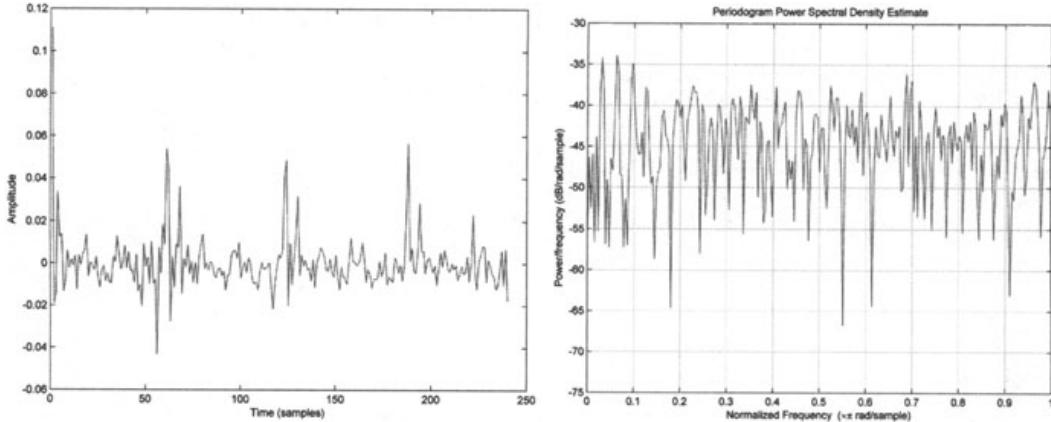


Fig. 1.15 The prediction residual (*left*: waveform; *right*: periodogram)

Let us compare the spectrum of this residual to the original spectrum. The new spectrum is approximately flat; its fine spectral details, however, are the same as those of the analysis frame. In particular, its pitch and harmonics are preserved.

For obvious reasons, applying the synthesis filter to this prediction residual results in the analysis frame itself (since the synthesis filter is the inverse of the inverse filter).

```
output_frame=filter(1, ai,LP_residual);
plot(output_frame);
```

The LPC model actually models the prediction residual of voiced speech as an impulse train with adjustable pitch period and amplitude. For the speech frame considered, for instance, the LPC ideal excitation is a sequence of pulses separated by 64 zeros (so as to impose a period of 65 samples; Fig. 1.16). Note we multiply the excitation by some gain so that its variance matches that of the residual signal.

```
excitation = [1;zeros(64,1);1;zeros(64,1);1;zeros(64,1);...
              1;zeros(44,1)];
gain=sigma/sqrt(1/65);
plot(gain*excitation);
periodogram(gain*excitation,[],512);
```

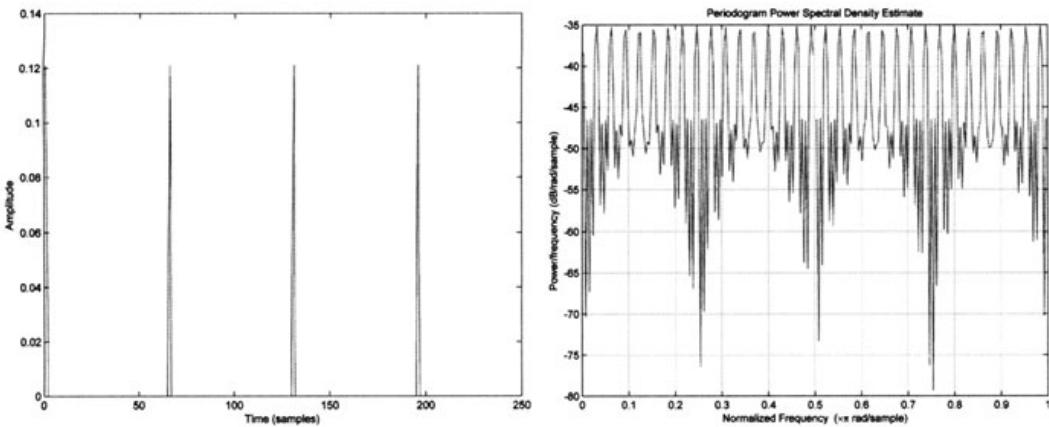


Fig. 1.16 The LPC excitation (*left*: waveform; *right*: periodogram)

Clearly, as far as the waveform is concerned, the LPC excitation is far from similar to the prediction residual. Its spectrum (Fig. 1.16), however, has the same broad features as that of the residual: flat envelope and harmonic content corresponding to F_0 . The main difference is that the excitation spectrum is “over-harmonic” compared to the residual spectrum.

Let us now use the synthesis filter to produce an artificial “e.”

```
synt_frame=filter(gain,ai,excitation);
plot(synt_frame);
periodogram(synt_frame,[],512);
```

Although the resulting waveform is obviously different from the original one (this is due to the fact that the LP model does not account for the phase spectrum of the original signal), its spectral envelope is identical. Its fine harmonic details, though, also widely differ: the synthetic frame is actually “over-harmonic” compared to the analysis frame (Fig. 1.17).

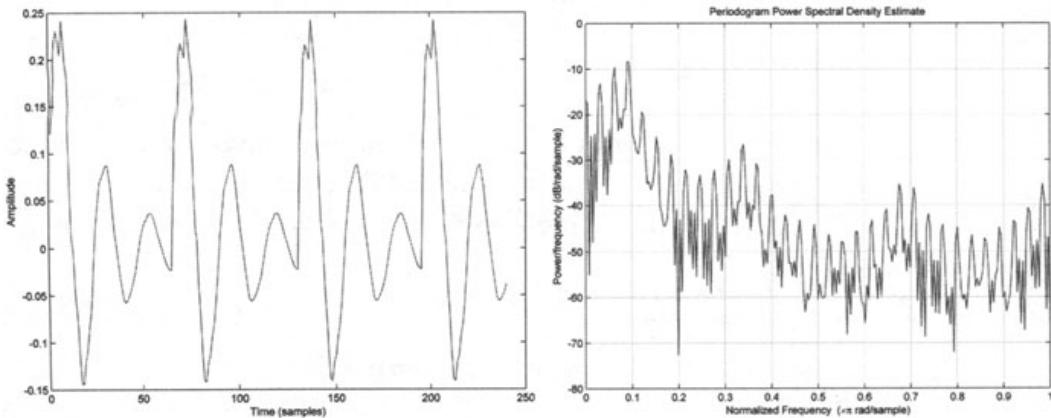


Fig. 1.17 Voiced LPC speech (*left*: waveform; *right*: periodogram)

1.2.3 Linear prediction synthesis of 30 ms of unvoiced speech

It is easy to apply the same process to an unvoiced frame and compare the final spectra again. Let us first extract an unvoiced frame and plot it (Fig. 1.18). As expected, no clear periodicity appears.

```
input_frame=speech_HF(4500:4739);
plot(input_frame);
```

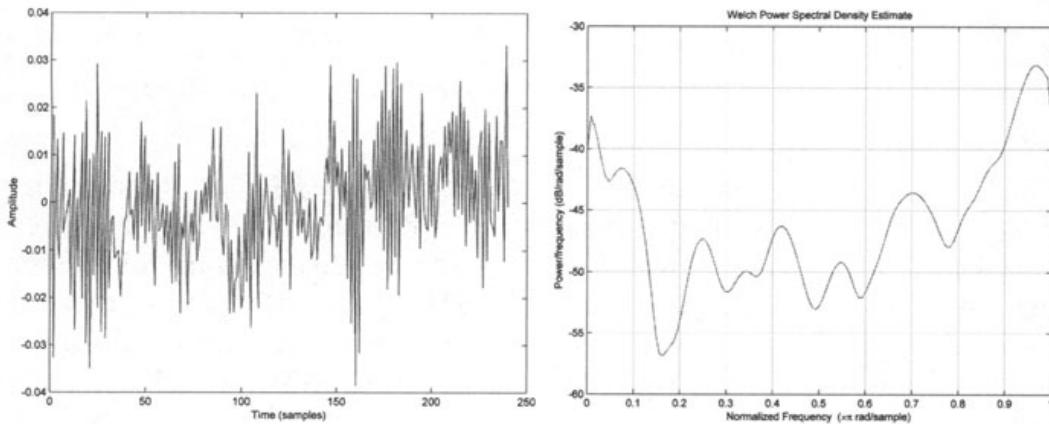


Fig. 1.18 A 30-ms-long frame of unvoiced speech (*left*: waveform; *right*: power spectral density)

Now let us see the spectral content of this speech frame. Note that, since we are dealing with noisy signals, we use the *averaged periodogram* to estimate power spectral densities, although with less-frequency resolution than using a simple periodogram. The MATLAB `pwelch` function does this with eight subframes by default and 50% overlap.

```
pwelch(input_frame);
```

Let us now apply an LP model of order 10 and synthesize a new frame. Synthesis is performed by all-pole filtering a Gaussian white noise frame with standard deviation set to the prediction residual standard deviation, σ .

```
[ai, sigma_square]=lpc(input_frame,10);
sigma=sqrt(sigma_square);
excitation=randn(240,1);
synt_frame=filter(sigma,ai,excitation);
plot(synt_frame);

pwelch(synt_frame);
```

The synthetic waveform (Fig. 1.19) has no sample in common with the original waveform. The spectral envelope of this frame, however, is still similar to the original one, enough at least for both the original and synthetic signals to be perceived as the same colored noise.¹⁰

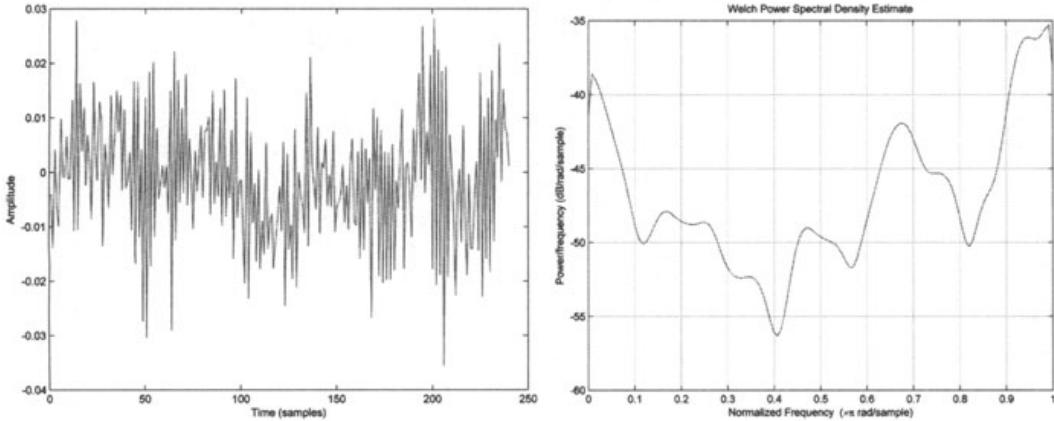


Fig. 1.19 Unvoiced LPC speech (*left*: waveform; *right*: psd)

1.2.4 Linear prediction synthesis of a speech file, with fixed F_0

We will now loop the previous operations for the complete speech file using 30 ms analysis frames overlapping by 20 ms. Frames are now weighted with a Hamming window. At synthesis time, we simply synthesize 10 ms of speech and concatenate the resulting synthetic frames to obtain the output speech file. Let us choose 200 Hz as synthesis F_0 , for convenience: this way each 10-ms excitation frame contains exactly two pulses.

```

for i=1:(length(speech)-160)/80; % number of frames
    % Extracting the analysis frame
    input_frame=speech_HF((i-1)*80+1:(i-1)*80+240);

    % Hamming window weighting and LPC analysis
    [ai, sigma_square]=lpc(input_frame.*hamming(240),10);
    sigma=sqrt(sigma_square);

    % Generating 10 ms of excitation
    % = 2 pitch periods at 200 Hz
    excitation=[1;zeros(39,1);1;zeros(39,1)];
    gain=sigma/sqrt(1/40);

    % Applying the synthesis filter

```

¹⁰ Although both power spectral densities have identical spectral slopes, one should not expect them to exhibit a close match in terms of their details, since only LPC modeling reproduces the smooth spectral envelope of the original signal.

```

synt_frame=filter(gain, ai, excitation);

% Concatenating synthesis frames
synt_speech_HF=[synt_speech_HF;synt_frame];

end

```

The output waveform basically contains a sequence of LP filter impulse responses. Let us zoom on 30 ms of LPC speech (Fig. 1.20).

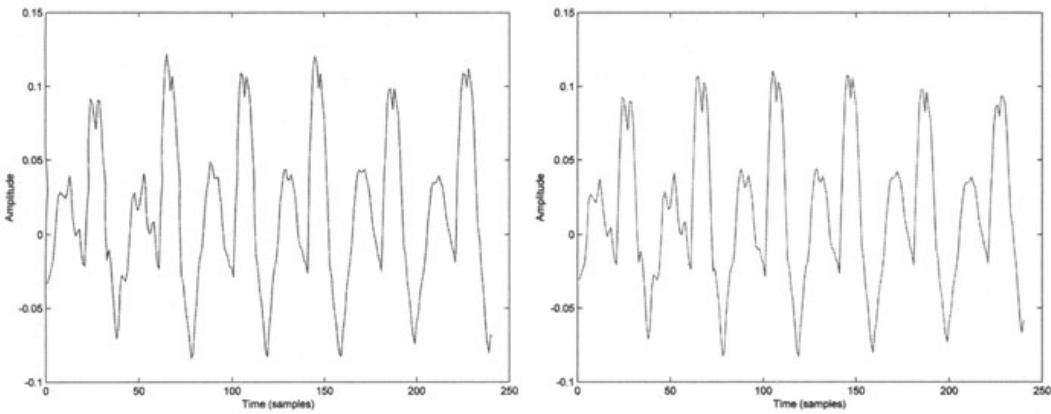


Fig. 1.20 Zoom on 30 ms of LPC speech (*left*: with internal variable reset; *right*: with internal variable memory)

It appears that in many cases the impulse responses have been cropped to the 10-ms synthetic frame size. As a matter of fact, since each synthesis frame was composed of two identical impulses, one should expect our LPC speech to exhibit pairs of identical pitch periods. This is not the case, because for producing each new synthetic frame, the internal variables of the synthesis filter are implicitly reset to zero. We can avoid this problem by maintaining the internal variables of the filter from the end of each frame to the beginning of the next one.

We initialize a vector z with 10 zeros and change the synthesis code into

```

% Applying the synthesis filter
% Taking care of the internal variables of the filter
gain=sigma/sqrt(1/40);
[synt_frame,z]=filter(gain, ai, excitation, z);

```

This time the end of each impulse response is properly added to the beginning of the next one, which results in more smoothly evolving periods (Fig. 1.20).

If we want to synthesize speech with constant pitch period length different from a submultiple of 80 samples (say, $N0=65$ samples), we additionally need to take care of a possible pitch period offset in the

excitation signal. After initializing this `offset` to zero, we simply change the excitation code into

```
% Generating 10 ms of excitation
% taking a possible offset into account

% if pitch period length > excitation frame length
if offset>=80
    excitation=zeros(80,1);
    offset=offset-80;
else
    % complete the previously unfinished pitch period
    excitation=zeros(offset,1);
    % for all pitch periods in the remaining of the frame
    for j=1:floor((80-offset)/N0)
        % add one excitation period
        excitation=[excitation;1;zeros(N0-1,1)];
    end;
    % number of samples left in the excitation frame
    flush=80-length(excitation);
    if flush~=0
        % fill the frame with a partial pitch period
        excitation=[excitation;1;zeros(flush-1,1)];
        % remember to fill the remaining of the period in
        % next frame
        offset=N0-flush;
    else offset=0;
    end
end
gain=sigma/sqrt(1/N0);
```

1.2.5 Unvoiced linear prediction synthesis of a speech file

Synthesizing the complete speech file as LPC unvoiced speech is easy. Periodic pulses are simply replaced by white noise, as in Section 1.2.3.

```
% Generating 10 ms of excitation
excitation=randn(80,1); % white Gaussian noise
gain=sigma;
```

As expected, the resulting speech sounds like whisper.

1.2.6 Linear prediction synthesis of a speech file, with original F_0

We will now synthesize the same speech using the original F_0 . We will thus have to deal with the additional problems of pitch estimation (on a frame-by-frame basis), including voiced/unvoiced decision. This approach is similar to that of the LPC10 coder (except that we do not quantize coefficients here). We change the excitation generation code into

```
% local synthesis pitch period (in samples)
N0=pitch(input_frame);
```

```

% Generating 10 ms of excitation
if N0~=0 % voiced frame
    % Generate 10 ms of voiced excitation
    % taking a possible offset into account
        (same code as in Section 1.2.4)

else
    % Generate 10 ms of unvoiced voiced excitation
        (same code as in Section 1.2.5)

    offset=0; % reset for subsequent voiced frames
end;

```

MATLAB function involved:

- `T0=pitch(speech_frame)` returns the pitch period T_0 (in samples) of a speech frame (T_0 is set to zero when the frame is detected as unvoiced). T_0 is obtained from the maximum of the (estimated) autocorrelation of the LPC residual. Voiced/unvoiced decision is based on the ratio of this maximum to the variance of the residual. This simple algorithm is not optimal but will do the job for this proof of concept.

The resulting synthetic speech (Fig. 1.21) is intelligible. It shows the same formants as the original speech. It is therefore acoustically similar to the original except for the additional buzziness that has been added by the LP model.

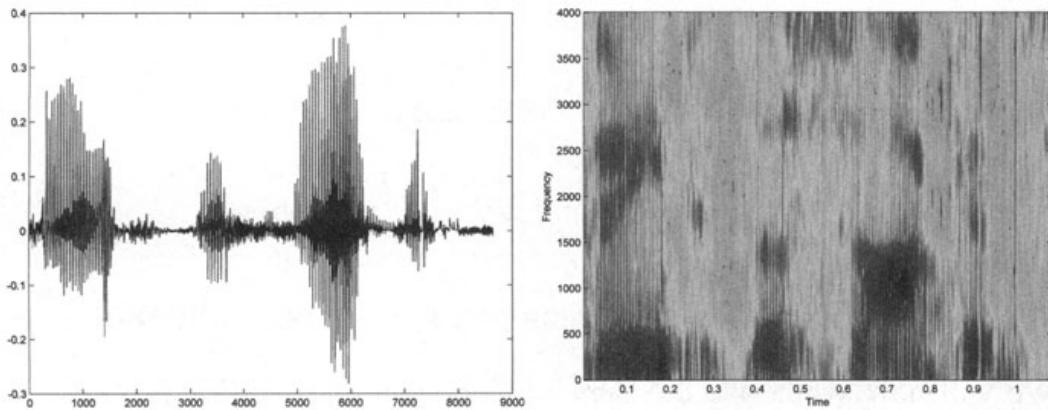


Fig. 1.21 LPC10 speech

1.2.7 CELP analysis-synthesis of a speech file

Our last step will be to replace the LPC10 excitation by a more realistic code-excited linear prediction (CELP) excitation obtained by selecting the best linear combination of excitation components from a codebook. Component selection is performed in a closed loop so as to minimize the difference between the synthetic and original signals.

We start with 30-ms LP analysis frames, shifted every 5 ms, and a codebook size of 512 vectors from which 10 components are chosen for every 5-ms synthesis frame.¹¹

MATLAB function involved:

- `[gains, indices] = find_Nbest_components(signal, ...
codebook_vectors, codebook_norms , N)`

This function finds the `N` best components of `signal` from the vectors in `codebook_vectors`, so that the residual error
`error = signal - codebook_vectors(indices)*gains`
is minimized. Components are found one-by-one using a greedy algorithm. When components in `codebook_vectors` are not orthogonal, the search is therefore suboptimal.

```
frame_length=240; % length of the LPC analysis frame
frame_shift=40; % length of the synthesis frames
codebook_size = 512; % number of vectors in the codebook
N_components= 10; % number of codebook components per frame
speech=wavread('speech.wav');

% Initializing internal variables
z_inv=zeros(10,1); % inverse filter
z_synt=zeros(10,1); % synthesis filter
synt_speech_CELP=[];

% Generating the stochastic excitation codebook
codebook = randn(frame_shift,codebook_size);

for i=1:(length(speech)-frame_length+frame_shift)/frame_shift;
    input_frame=speech((i-1)*frame_shift+1:...
                      (i-1)*frame_shift+frame_length);

    % LPC analysis of order 10
    ai = lpc(input_frame.*hamming(frame_length), 10);

    % Extracting frame_shift samples from the LPC analysis frame
    speech_frame = input_frame((frame_length-frame_shift)/2+1:...
```

¹¹ These values actually correspond to a rather high bit rate, but we will show in the next paragraphs how to lower the bit rate while maintaining the quality of synthetic speech.

```

(frame_length-frame_shift)/2+frame_shift);

% Filtering the codebook (all column vectors)
codebook_filt = filter(1, ai, codebook);

% Finding speech_frame components in the filtered codebook
% taking into account the transient stored in the internal
% variables of the synthesis filter
ringing = filter(1, ai, zeros(frame_shift,1), z_synt);
signal = speech_frame - ringing;
[gains, indices] = find_Nbest_components(signal, ...
    codebook_filt, N_components);

% Generating the corresponding excitation as a weighted sum
% of codebook vectors
excitation = codebook(:,indices)*gains;

% Synthesizing CELP speech, and keeping track of the
% synthesis filter internal variables
[synt_frame, z_synt] = filter(1, ai, excitation, z_synt);
synt_speech_CELP=[synt_speech_CELP;synt_frame];

end

```

Note that this analysis–synthesis simulation is implemented as mentioned in Section 1.2.4 as an adaptive vector quantization system. This is done by passing the whole codebook through the synthesis filter, for each new frame, and searching for the best linear decomposition of the speech frame in terms of filtered codebook sequences.

Also note our use of `ringing`, which stores the natural response of the synthesis filter due to its nonzero internal variables. This response should not be taken into account in the adaptive VQ.

The resulting synthetic speech sounds more natural than in LPC10. Plosives are much better rendered, and voiced sounds are no longer buzzy, but speech sounds a bit noisy. Note that pitch and V/UV estimation are no longer required.

One can see that the closed-loop optimization leads to excitation frames, which can somehow differ from the LP residual, while the resulting synthetic speech is similar to its original counterpart (Fig. 1.22).

In the above script, though, each new frame was processed independently of past frames. Since voiced speech is strongly self-correlated, it makes sense to incorporate a long-term prediction filter in cascade with the LPC (short-term) prediction filter. In the example below, we can reduce the number of stochastic components from 10 to 5 while still increasing speech quality, thanks to long-term prediction.

```
N_components= 5; % number of codebook components per frame
```

Since CELP excitation frames are only 5 ms long, we store them in a 256 samples circular buffer (i.e., a bit more than 30 ms of speech) for finding the best long-term prediction delay in the range [0–256] samples.

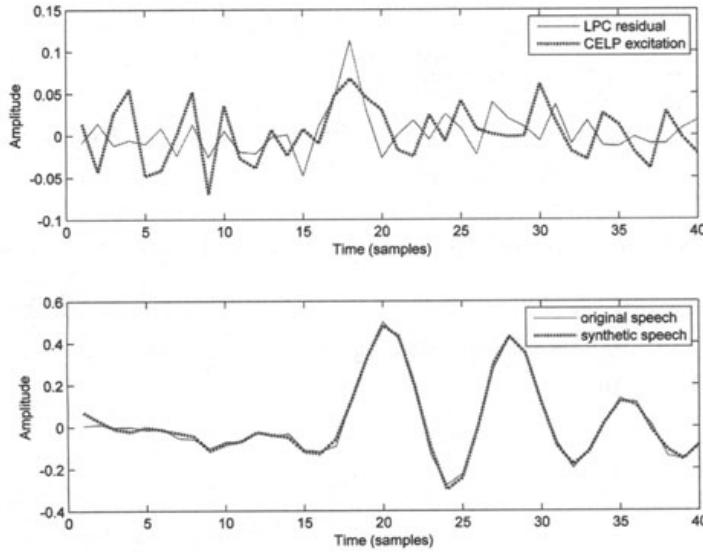


Fig. 1.22 CELP analysis–synthesis of frame #140. *Top*: CELP excitation compared to linear prediction residual. *Bottom*: CELP synthetic speech compared to original speech

```
LTP_max_delay=256; % maximum long-term prediction delay
excitation_buffer=zeros(LTP_max_delay+frame_shift,1);
```

Finding the delay itself (inside the frame-based loops) is achieved in a way very similar to finding the N best stochastic components in our previous example: we create a long-term prediction codebook, pass it through the synthesis filter, and search for *the* best excitation component in this filtered codebook.

```
% Building the long-term prediction codebook and filtering it
for j = 1:LTP_max_delay
    LTP_codebook(:,j) = excitation_buffer(j:j+frame_shift-1);
end
LTP_codebook_filt = filter(1, ai, LTP_codebook);

% Finding the best predictor in the LTP codebook
ringing = filter(1, ai, zeros(frame_shift,1), z_synt);
signal = speech_frame - ringing;
[LTP_gain, LTP_index] = find_Nbest_components(signal, ...
    LTP_codebook_filt, 1);

% Generating the corresponding prediction
LT_prediction= LTP_codebook(:,LTP_index)*LTP_gain;
```

Stochastic components are then searched *in the remaining signal* (i.e., the original signal minus the long-term predicted *signal*).

```
% Finding speech_frame components in the filtered codebook
% taking long term prediction into account
signal = signal - LTP_codebook_filt(:,LTP_index)*LTP_gain;
[gains, indices] = find_Nbest_components(signal, ...
    codebook_filt, N_components);
```

The final excitation is computed as the sum of the long-term predicted *excitation*.

```
excitation = LT_prediction + codebook(:,indices)*gains;
```

As can be seen in Fig. 1.23, the resulting synthetic speech is still similar to the original one, notwithstanding the reduction of the number of stochastic components.

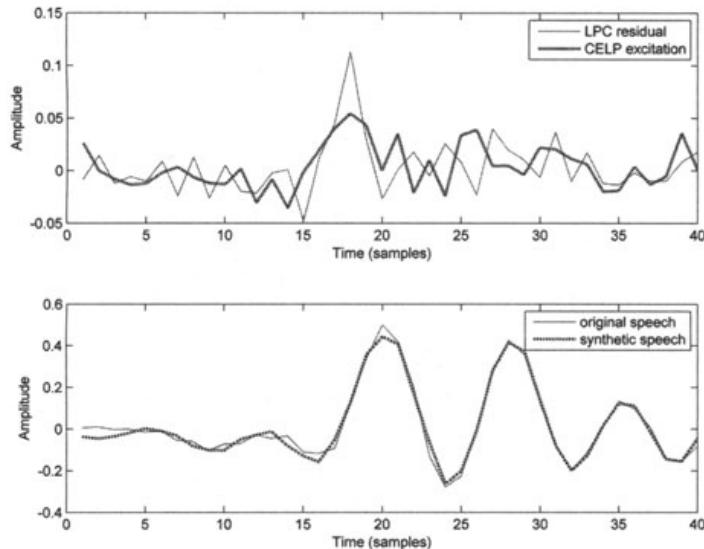


Fig. 1.23 CELP analysis–synthesis of frame #140 with long-term prediction and only five stochastic components. *Top:* CELP excitation compared to linear prediction residual. *Bottom:* CELP synthetic speech compared to original speech

Although the search for the best components in the previous scripts aims at minimizing the energy of the difference between original and synthetic speech samples, it makes sense to use the fact that the ear will be more tolerant to this difference in parts of the spectrum that are louder and vice versa. This can be achieved by applying a perceptual filter to the error,

which enhances spectral components of the error in frequency bands with less energy and vice versa (Fig. 1.24).

In the following example, we still decrease the number of components from 5 to 2, with the same overall synthetic speech quality.

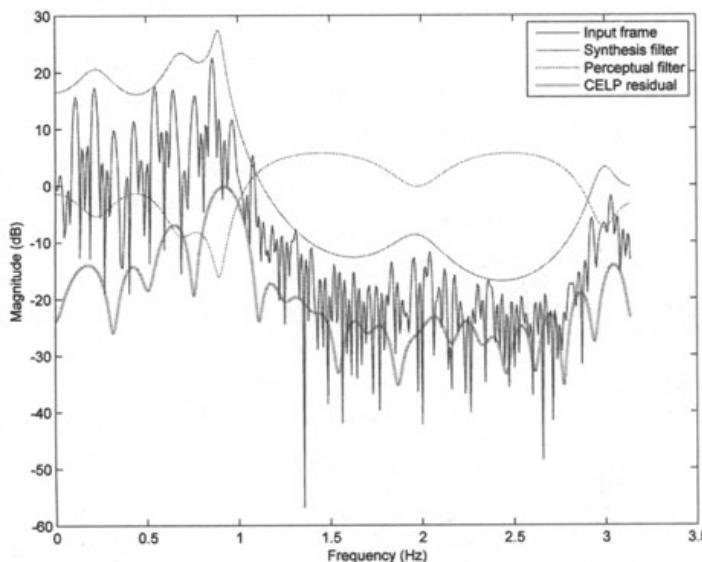


Fig. 1.24 CELP analysis–synthesis of frame #140: the frequency response of the perceptual filter approaches the inverse of that of the synthesis filter. As a result, the spectrum of the CELP residual somehow follows that of the speech frame

```
N_components= 2; % number of codebook components per frame
```

We will apply perceptual filter $A(z)/A(z/\gamma)$ to the input frame, and filter $1/A(z/\gamma)$ to the stochastic and long-term prediction codebook vectors.¹² We will therefore need to handle their internal variables.

```
gamma = 0.8; % perceptual factor
z_inv=zeros(10,1); % inverse filter
z_synt=zeros(10,1); % synthesis filter
z_gamma_s=zeros(10,1); % perceptual filter for speech
z_gamma_e=zeros(10,1); % perceptual filter for excitation
```

Finding the coefficients of $A(z/\gamma)$ is easy.

```
ai_perceptual = ai.* (gamma.^ (0:(length(ai)-1)) );
```

One can then filter the input frame and each codebook.

¹² In the previous examples, the input frame was not perceptually filtered, and codebooks were passed through the synthesis filter $1/A(z)$.

```
% Passing the central 5ms of the input frame through
% A(z)/A(z/gamma)
[LP_residual, z_inv] = filter(ai, 1, speech_frame, z_inv);
[perceptual_speech, z_gamma_s] = filter(1, ...
    ai_perceptual, LP_residual, z_gamma_s);

% Filtering both codebooks
LTP_codebook_filt = filter(1, ai_perceptual, LTP_codebook);
codebook_filt = filter(1, ai_perceptual, codebook);
```

The search for the best long-term predictor is performed as before, except that the perceptually filtered speech input is used as the reference from which to find codebook components.

```
% Finding the best predictor in the LTP codebook
ringing = filter(1, ai_perceptual, ...
    zeros(frame_shift,1), z_gamma_e);
signal = perceptual_speech - ringing;
[LTP_gain, LTP_index] = find_Nbest_components(signal, ...
    LTP_codebook_filt, 1);

% Generating the corresponding prediction
LT_prediction= LTP_codebook(:,LTP_index)*LTP_gain;

% Finding speech_frame components in the filtered codebook
% taking long term prediction into account
signal = signal - LTP_codebook_filt(:,LTP_index)*LTP_gain;
[gains, indices] = find_Nbest_components(signal, ...
    codebook_filt, N_components);
```

Last but not least, one should not forget to update the internal variables of the perceptual filter applied to the excitation.

```
[ans, z_gamma_e] = filter(1, ai_perceptual, excitation, ...
    z_gamma_e);
```

While using less stochastic components than in the previous example, synthetic speech quality is maintained, as revealed by listening. The synthetic speech waveform also looks much more similar to original speech than its LPC10 counterpart (Fig. 1.25).

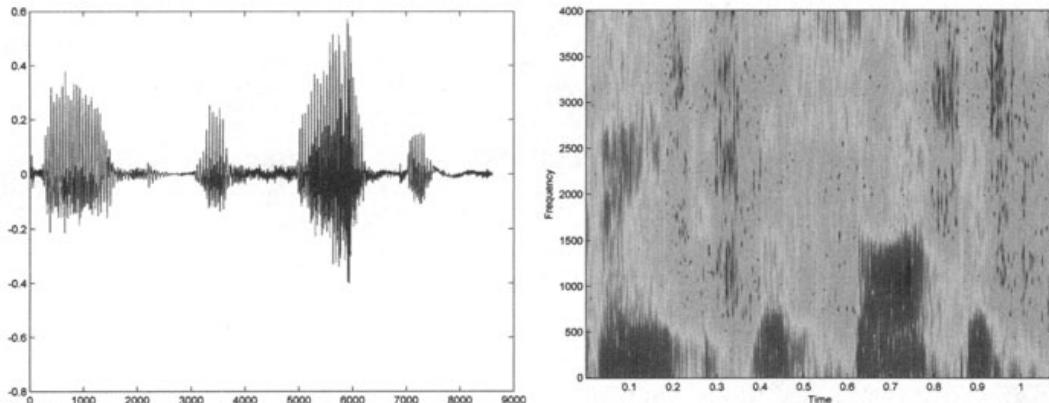


Fig. 1.25 CELP speech

One can roughly estimate the corresponding bit rate. Assuming 30 bits are enough for the prediction coefficients and each gain factor is quantized on 5 bits, we have to send for each frame: 30 bits [ai] + 7 bits [LTP index] + 5 bits [LTP gain] + 2 [stochastic components] *(9 bits [index] + 5 bits [gain]) = 70 bits every 5 ms, i.e., 14 kbps.

Note that G729 reaches a bit rate as low as 8 kbps by sending prediction coefficients only once every four frame.

1.3 Going further

Various tools and interactive tutorials on LP modeling of speech are available on the web (see Fellbaum 2007, for instance).

MATLAB code by A. Spanias for the LPC10e coder can be found on the web (Spanias and Painter 2002).

Another interesting MATLAB-based project on LPC coding, applied to wideband speech this time, can be found on the dspexperts.com website (Khan and Kashif 2003).

D. Ellis provides interesting MATLAB-based audio processing examples on his web pages (Ellis 2006), among which are a sinewave speech analysis/synthesis demo (including LPC) and a spectral warping of LPC demo.

For a broader view of speech coding standards, one might refer to Woodard (2007) or to the excellent book by Goldberg and Riek (2000).

1.4 Conclusion

We now understand how every cell phone solves a linear system of 10 equations in 10 unknowns every 20 ms which is the basis of the estimation of the LP model through Yule-Walker equations. The parameters that are actually sent from one cell phone to another are vocal tract coefficients related to the frequency response of the vocal tract and source coefficients related to the residual signal.

The fact that the vocal tract coefficients are very much related to the geometric configuration of the vocal tract for each frame of 10 ms of speech calls for an important conclusion: cell phones, in a way, transmit a picture of our vocal tract rather than the speech it produces.

In fact, the reach of LP speech modeling goes far beyond the development of cell phones. As shown by Gray (2006), its history is intermixed with that of Arpanet, the ancestor of Internet.

References

- Atal BS, Remde JR (1982) A New Model LPC Excitation for Producing Natural Sounding Speech at Low Bit Rates. In: Proc. ICASSP'82, pp 614–617
- de la Cuadra P (2007) Pitch Detection Methods Review [online] Available: <http://www-ccrma.stanford.edu/~pdelac/154/m154paper.htm> [20/2/1007]
- Ellis D (2006) Matlab Audio Processing Examples [online] Available: <http://www.ee.columbia.edu/%7Edpwe/resources/matlab/> [20/2/2007]
- Fant G (1960) Acoustic Theory of Speech Production. The Hague: Mouton
- Fellbaum K (2007) Human Speech Production Based on a Linear Predictive Vocoder [online] Available: <http://www.kt.tu-cottbus.de/speech-analysis/> [20/2/2007]
- Goldberg RG, Riek L (2000) Speech Coders. CRC Press: Boca Raton, FL
- Gray RM (2006) Packet speech on the Arpanet: A history of early LPC speech and its accidental impact on the Internet Protocol [online] Available: http://www.ieee.org/organizations/society/sp/ Packet_Speech.pdf [20/2/2007]
- Hess W (1992) Pitch and Voicing Determination. In: Advances in Speech Signal Processing, S. Furui, M. Sondhi, eds., Dekker, New York, pp 3–48
- Khan A, Kashif F (2003) Speech Coding with Linear Predictive Coding (LPC) [online] Available: <http://www.dspxperts.com/dsp/projects/lpc> [20/2/2007]
- Kroon P, Deprettere E, Sluyter R (1986) Regular-pulse excitation – A novel approach to effective and efficient multipulse coding of speech. IEEE Transactions on Acoustics, Speech, and Signal Processing 34(5): 1054–1063
- Matsumoto J, Nishiguchi M, Iijima K (1997) Harmonic Vector Excitation Coding at 2.0 kbps. In: Proc. IEEE Workshop on Speech Coding, pp 39–40
- McCree AV, Barnwell TP (1995) A mixed excitation LPC vocoder model for low bit rate speech coding. IEEE Transactions on Speech and Audio Processing, 3(4):242–250
- NATO (1984) Parameters and coding characteristics that must be common to assure interoperability of 2400 bps linear predictive encoded speech. NATO Standard STANAG-4198-Ed1
- Quatieri T (2002) Discrete-Time Speech Signal Processing: Principles and Practice. Prentice-Hall, Inc.: Upper Saddle River, NJ
- Rabiner LR, Schafer RW (1978) Digital Processing of Speech Signals. Prentice-Hall, Inc.: Englewood Cliffs, NJ
- Salami R, Laflamme C, Adoul J-P, Kataoka A, Hayashi S, Moriya T, Lamblin C, Massaloux D, Proust S, Kroon P, Shoham, Y (1998) Design and description of CS-ACELP: A toll quality 8 kb/s speech coder, IEEE Transactions on Speech and Audio Processing 6(2): 116–130
- Schroeder MR, Atal B (1985) Code-Excited Linear Prediction(CELP): High Quality Speech at Very Low Bit Rates. In: Proc. IEEE ICASSP-85, pp 937–940

- Spanias A, Painter T (2002) Matlab simulation of LPC10e vocoder [online]
Available: http://www.cysip.net/lpc10e_FORM.htm [19/2/2007]
- Woodard J (2007) Speech coding [online] Available: http://www-mobile.ecs.soton.ac.uk/speech_codecs/ [20/2/2007]

Chapitre 12

Application au codage d'images

12

Image Compression: The JPEG Standard

Presenting the JPEG standard at the level of detail contained in this chapter will require about four hours. To fit within this amount of time, you will have to skip Section 12.4; this section proves the orthogonality of the matrix C and can be seen as the advanced part of this chapter. It is necessary, however, to discuss the relationship between the matrices f and α and to present the 64 basis elements A_{ij} . The central idea underlying the JPEG standard is a change of basis in a 64-dimensional space; this chapter provides the perfect occasion to review this portion of linear algebra.

12.1 Introduction: Lossless and Lossy Compression

Data compression is at the very heart of computer science, and the Internet has made its use an everyday occurrence for most. Many of us may not even know we are using compression, or at least have little knowledge of how the underlying algorithms work. Even so, many compression algorithms have names that are familiar to general computer users (*WinZip*, *gzip*, and, in the UNIX world, *compress*), to music lovers and Internet users (*GIF*, *JPG*, *PNG*, *MP3*, *AAC*, etc). If not for the common use of compression algorithms, the Internet would be completely paralyzed by the volume of uncompressed data being transferred.

The goal of this chapter is to study a commonly used algorithm for the compression of black-and-white or color still images (“still” as opposed to “moving” images). This method of compression is commonly known as JPEG, the acronym of *Joint Photographic Experts Group*, the consortium of companies and researchers that developed and popularized it. The group started its work in June 1987, and the first draft of the standard was published in 1991. Internet users will no doubt associate this compression method with the “jpg” suffix that is a part of the names of many images and photographs transmitted over the Internet. The JPEG algorithm is the most commonly used compression method in digital cameras.

Before diving into the details of this algorithm and the underlying mathematics it is good to have a basic knowledge of data compression in general. There are two broad families of data compression algorithms: those that actually degrade the original information to some extent (called *lossy* algorithms) and those that allow for the reconstruction of the original with perfect accuracy (called *lossless* algorithms). Two simple observations can be made.

The first is that it is impossible to compress *without loss* all files of a given size using the same algorithm. Suppose that such a technique exists for files of exactly N bits in length. Each of these bits can take on 2 different values (0 or 1) and thus there are 2^N distinct N -bit files. If the algorithm compresses each of these files, then each one of them will be represented by some new file containing at most $N - 1$ bits. There are 2^{N-1} distinct files of $N - 1$ bits, 2^{N-2} distinct files of $N - 2$ bits, \dots , 2^1 distinct files of 1 bit and a single one with 0 bit. Thus, the number of distinct files containing at most $N - 1$ bits is

$$1 + 2^1 + 2^2 + \dots + 2^{N-2} + 2^{N-1} = \sum_{n=0}^{N-1} 2^n = \frac{2^N - 1}{2 - 1} = 2^N - 1.$$

Thus the algorithms we are using must compress at least two of the original N -bit files to some identical file containing fewer than N bits. These two compressed files will then be indistinguishable, and it is impossible to determine which original file they should decompress to. Again: *it is impossible to losslessly compress all files of a given size!*

The second observation is a consequence of the first: when developing a compression algorithm, the person charged with this task must decide whether the information must be preserved perfectly or whether a slight loss (or transformation) is tolerable. Two examples can help make this choice clear while also demonstrating different approaches once this decision has been made.

Webster's Ninth New Collegiate Dictionary has 1592 pages, most being typeset in two columns, each column having around 100 lines, each line having about 70 characters, spaces, or punctuation marks. This amounts to a total of about 22 million characters. These characters can be represented by an alphabet of 256 characters, each being coded by 8 bits, or 1 byte (see Section 12.2). About 22 MB are therefore needed to hold *Webster's*. If one recalls that compact disks store approximately 750 MB, a single CD can carry 34 copies of the whole of *Webster's* (without the figures and drawings, however). No author of a dictionary, an encyclopedia, or a textbook (or *any* book for that matter!) would tolerate the changing of a single character. Thus, in compressing such material it is extremely important to use a lossless compression algorithm allowing for a perfect reconstruction of the original document.

A simple approach to such an algorithm assigns variable length codes to each letter of the alphabet.¹ The most common characters in English are the “ ” (space) character

¹This approach is common to text compression. Different algorithms may assign codes to “words” rather than “letters,” and more complicated algorithms may change the assigned codes based on context.

and the letter “e” followed by the letters “t”, “a”, “o”, “i”, “n”, “h”, “s”, “r” (see Table 12.1). The most uncommonly used letters are “x”, “z”, “j”, and “q”. The actual frequencies depend on the author and the text. They may vary significantly if the text is short. It is natural to try to assign short codes to more frequently occurring characters (such as “l” and “e”) and longer codes to less frequently occurring ones (such as “j” and “q”). In this manner, characters are represented by a variable number of bits rather than always requiring a single byte. Does this approach violate our first observation? No, since in order for each assigned code to be uniquely decodable the codes for rarely occurring letters will be *longer* than 8 bits. Thus, files containing an unusually high percentage of such characters will actually be longer than the original uncompressed file. The idea of assigning variable length codes to individual symbols as a function of their frequency of use is the main idea underlying Huffman codes.

letter	frequency	letter	frequency	letter	frequency
e	0.125	d	0.047	p	0.018
t	0.088	l	0.041	b	0.016
a	0.080	u	0.027	v	0.010
o	0.077	m	0.026	k	0.0090
i	0.069	w	0.025	j	0.0014
n	0.068	c	0.023	x	0.0014
h	0.066	g	0.022	q	0.0010
s	0.060	f	0.021	z	0.0002
r	0.059	y	0.021		

Table 12.1. Frequencies of letters in Dickens’s *Oliver Twist*. (Spaces and punctuation marks have been ignored. Capital letters have been mapped to the corresponding lowercase letters. *Oliver Twist* contains a little over 680,000 letters.)

Our second example lies a little closer to the subject of this chapter. All computer screens have a finite resolution. Usually, this is measured by counting the number of pixels that it can display. Each pixel may be illuminated to take on any color and intensity.² Early screens could display $640 \times 480 = 307,200$ pixels.³ (Resolution is

²This is not exactly true. Computer screens are able to reproduce only a portion of the visible color gamut, broken down into a finite set of discrete colors that are roughly uniformly close to each other. As such, they can generally reproduce a large number of colors but not the entire visible spectrum.

³It is now common to have displays capable of displaying many millions of pixels, with the largest surpassing four million.

normally reported as “number of pixels per horizontal line \times number of lines.”) Suppose that the Louvre decided to digitize its entire collection of painted works. The museum would ideally like to do this with sufficient quality so as to please art experts. However, at the same time they would like to have lower-quality versions for transmission over the Internet and display on typical computer screens. In this case, it doesn’t make any sense for the image to be of a higher resolution than a typical computer monitor. Thus, the image satisfying art experts and that for display on a typical computer monitor are going to be of very different resolutions and sizes. The latter will contain significantly less detail but will be entirely satisfactory for displaying on a monitor. In fact, transmitting the higher-quality image would be a complete waste of time given the limited resolution of the display! The decision about the number of pixels to send is then a fairly obvious one. But suppose that Louvre technical people want to further reduce the size of the transmitted files. They argue that mathematicians often approximate functions around a given point by a straight line, and if one looks at the graph of the function and the approximating line they usually agree fairly well, at least locally. If we imagine the pale tones of a picture as the peaks and ridges of a function graph and the dark ones as its valleys, could we use the mathematical idea of approximation to this “function”?

This last question is more physiological than mathematical: can one fool the user by sending a picture that has been “mathematically approximated”? If the answer is yes, it will mean that a certain loss of quality is acceptable depending on the use of the data. Other criteria (such as human physiology) therefore play an equally important role in deciding how to compress. For example, in digitizing music it is useful to know that the (average) human ear is unable to perceive sounds above 20,000 Hz. In fact, the standard used for recording compact discs ignores frequencies over 22,000 Hz and is capable of accurately reproducing only those frequencies below this threshold, a loss that would bother only dogs, bats, or other animals with a keener sense of hearing than our own. For images are there limits to the variations in colors and intensities of light that may be perceived by the human eye? Are our eyes and mind content with receiving less than an exact reproduction of an image? Should photographic images and cartoons be compressed in the same manner? The JPEG compression standard, through its successes and its limits, answers these questions.

12.2 Zooming in on a JPEG Compressed Digital Image

A photograph can be digitized in a variety of ways. In the JPEG method the photograph is first divided into very small elements, called *pixels*, each one associated with a uniform color or gray tone. The photograph of a cat in Figure 12.1 has been subdivided into 640×640 pixels. Each of these $640 \times 640 = 409,600$ pixels has been associated with a uniform tone of gray between black and white. This particular photograph has been digitized using a scale of 256 gray tones where 0 represents black and 255 represents white. Since $256 = 2^8$, each of these values may be stored using 8 bits (a single byte).

Without compression we would require 409,600 bytes to store the photo of the cat, which equates to roughly 410 KB. (Here we are using the metric convention: a KB represents 1000 bytes, a MB represents 10^6 bytes, etc.) To encode a color image, each pixel is associated with three color values (red, green, and blue) each encoded using an 8-bit value between 0 and 255. An image of this size would require over 1.2 MB to store uncompressed. However, as frequent users of the Internet will know, large color JPEG-compressed images (files with a “jpg” suffix) rarely exceed 100 KB. The JPEG method is thus able to efficiently store the information in the image. The JPEG algorithm’s utility is not strictly confined to the Internet. It is the principal standard used in digital photography. Nearly all digital cameras will compress images to JPEG format by default; the compression occurs at the instant the photo is taken, and therefore a part of the information is lost forever. As we will see in this chapter, this loss is usually acceptable, but sometimes it is not. Depending on the specific use of the camera, it is up to the photographer to decide. (Exercise: As of 2006, many digital cameras offer resolutions exceeding 10 million pixels (megapixels). What is the space that would be required by such a color image in an uncompressed form?)

Rather than processing the entire photograph at once, the JPEG standard divides the image into little tiles of 8×8 pixels. Figure 12.1 shows two closeups of the image of the cat. In the bottom left, a 32×32 pixel region has been shown. The bottom right shows a further closeup of an 8×8 region of this closeup. The closeups focus on a small region depicting the intersection of two of the cat’s whiskers close to the edge of the table. This particular block of the image is unique in that it contains fine details and high contrast. This is not typical of most 8×8 tiles! In most of the image we see that the changes in color and texture are quite gradual. The surface under the table, the table itself, and even the cat’s fur consist largely of smooth gradients when looked at as 8×8 blocks. This is the case with most photographs; just think of any landscape photo containing open regions of land, water or sky. The JPEG standard was built on this uniformity; it tries to represent a nearly uniform 8×8 block using as little information as possible. When such a block contains significant detail (such as is the case in our closeup), the use of more space is accepted.

12.3 The Case of 2×2 Blocks

It is simpler to characterize 2×2 blocks than 8×8 blocks, so we will start with that.

We have seen that gray tones are typically represented using a scale with 256 increments. We could equally imagine a scale with infinitely fine increments that covers all of $[-1, 1]$ or any interval $[-L, L]$ of \mathbb{R} . In this case, we may associate negative values with dark grays tending to black and positive values to lighter grays tending to white. The origin would then correspond to a gray between levels 127 and 128 on the scale with 256 levels. Even though this change of scale and origin may be perfectly natural in some ways, it is not necessary for our discussion. We will, however, ignore the fact that our gray tones are integers between 0 and 255 and instead treat them as real numbers in

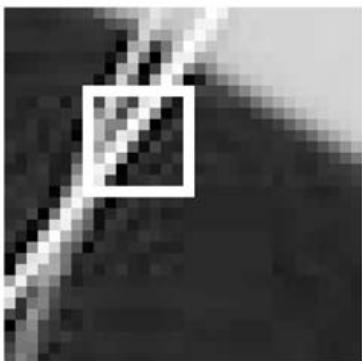
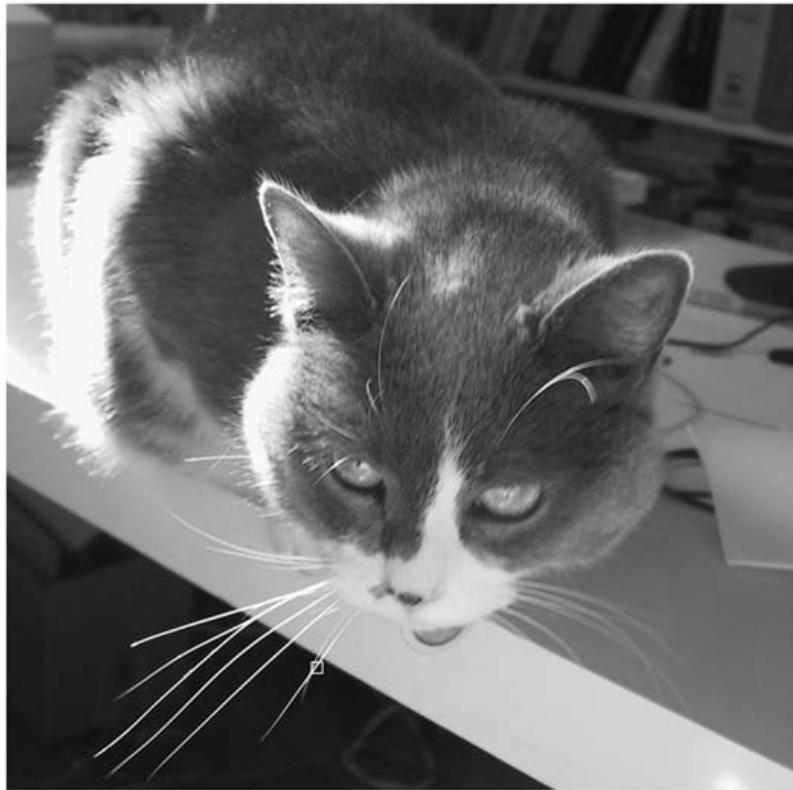


Fig. 12.1. Two successive closeups are made of the original photo (top), which contains 640×640 pixels. The first closeup (bottom left) contains 32×32 pixels. The second closeup (bottom right) contains 8×8 pixels. The white frames on the first and second images denotes the boundaries of the 8×8 closeup in the last image.

this same range. The tone of each pixel will therefore be represented by a real number, and a 2×2 block will require four such values, or equivalently, a point in \mathbb{R}^4 . (When we are dealing with an $N \times N$ block, we can consider it as a vector in \mathbb{R}^{N^2} .)

Given that we perceive the blocks in two dimensions, it is more natural to number the individual pixels using two indices i and j from the set $\{0, 1\}$ (or the set $\{0, 1, \dots, N-1\}$ when we are dealing with $N \times N$ blocks). The first index will indicate the row, while the second will indicate the column, as is typical in linear algebra. For example, the values of the function f giving the gray tones on the 2×2 square of Figure 12.2 are

$$f = \begin{pmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{pmatrix} = \begin{pmatrix} 191 & 207 \\ 191 & 175 \end{pmatrix}.$$

Many of the functions that we will study naturally take their values in the range $[-1, 1]$. When representing them as gray tones we will use the obvious affine transformation to map them to the range $[0, 255]$. This transformation can be

$$\text{aff}_1(x) = 255(x + 1)/2 \quad (12.1)$$

or

$$\text{aff}_2(x) = [255(x + 1)/2], \quad (12.2)$$

where $[x]$ denotes the integer part of x . (This last transformation will be used when the values need to be constrained to integers in the range $[0, 255]$. See Exercise 1.) We will use f to denote a function defined in the range $[0, 255]$ and g to denote functions defined in the range $[-1, 1]$. The following box summarizes this notation and specifies the translation we will use. Using this method, the function g associated with the above function f is

$$g = \begin{pmatrix} g_{00} & g_{01} \\ g_{10} & g_{11} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{5}{8} \\ \frac{1}{2} & \frac{3}{8} \end{pmatrix} :$$

$$f_{ij} \in [0, 255] \subset \mathbb{Z} \quad \longleftrightarrow \quad g_{ij} \in [-1, 1] \subset \mathbb{R}$$

$$f_{ij} = \text{aff}_2(g_{ij}), \quad \text{where} \quad \text{aff}_2(x) = \left[\frac{255}{2}(x + 1) \right].$$

We will graphically represent a 2×2 block in two different manners. The first will be simply to draw it using the associated gray tones that would appear in a photograph. The second is to interpret the values g_{ij} as a two-dimensional function of the variables i and j , $i, j \in \{0, 1\}$. Figure 12.2 represents the function $g = (g_{00}, g_{01}, g_{10}, g_{11}) = (\frac{1}{2}, \frac{5}{8}, \frac{1}{2}, \frac{3}{8})$ in these two manners. The coefficients giving the gray values for both the top left g_{00} and bottom left g_{10} pixels are identical. Those of the right column are g_{01} (the paler of the two) and g_{11} . In other words, if we use the matrix notation

$$g = \begin{pmatrix} g_{00} & g_{01} \\ g_{10} & g_{11} \end{pmatrix},$$

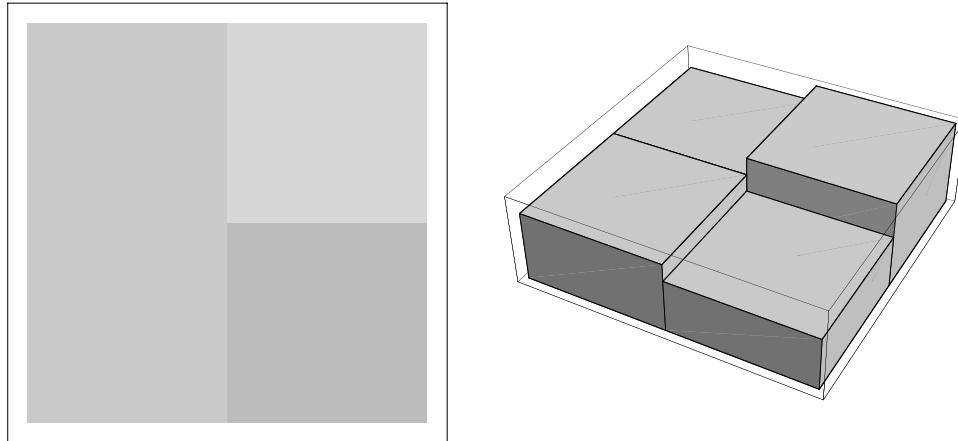


Fig. 12.2. Two graphical representations of the function $g = (g_{00}, g_{01}, g_{10}, g_{11}) = (\frac{1}{2}, \frac{5}{8}, \frac{1}{2}, \frac{3}{8})$.

then the elements of the matrix g are in the same positions as the pixels of Figure 12.2. The second image interprets these same values but displays them as a histogram in two variables i and j , with darker colors being associated to lesser heights. This particular 2×2 block was chosen because all of the pixels are closely related gray tones, as is typical of most 2×2 blocks in a photograph. (In fact, the higher the resolution of the photo, the gentler the gradients become.)

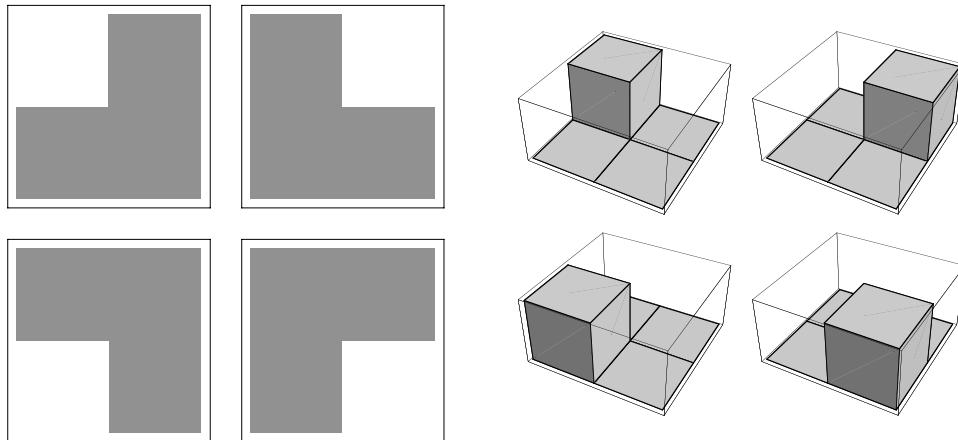


Fig. 12.3. The four elements of the usual basis \mathcal{B} of \mathbb{R}^4 represented graphically.

The coordinates $(g_{00}, g_{01}, g_{10}, g_{11})$ (or equivalently $(f_{00}, f_{01}, f_{10}, f_{11})$) represent the small 2×2 block without any loss. (In other words, no compression has yet been done.) These coordinates are expressed in the usual basis \mathcal{B} of \mathbb{R}^4 , where each element of the basis contains a single nonzero entry with value 1. This basis is depicted graphically in

Figure 12.3. If we were to apply a change of basis

$$[g]_{\mathcal{B}} = \begin{pmatrix} g_{00} \\ g_{01} \\ g_{10} \\ g_{11} \end{pmatrix} \mapsto [g]_{\mathcal{B}'} = \begin{pmatrix} \beta_{00} \\ \beta_{01} \\ \beta_{10} \\ \beta_{11} \end{pmatrix} = [P]_{\mathcal{B}'\mathcal{B}}[g]_{\mathcal{B}},$$

the new coordinates β_{ij} would also accurately represent the contents of the block. The coordinates g_{ij} are not appropriate to our end goal. In fact, we would like to easily recognize blocks where all of the pixels are nearly the same color or gray tone. To do this, it is useful to construct a basis in which completely uniform blocks are represented by a single nonzero coefficient. Similarly, we would like a cursory inspection of the coordinates to reveal when the block is far from being uniform.

The JPEG standard proposes using another basis $\mathcal{B}' = \{A_{00}, A_{01}, A_{10}, A_{11}\}$. Each element A_{ij} of this basis can be expressed using the standard basis shown in Figure 12.3. In the standard basis \mathcal{B} their coefficients are

$$[A_{00}]_{\mathcal{B}} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}, \quad [A_{01}]_{\mathcal{B}} = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}, \quad [A_{10}]_{\mathcal{B}} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}, \quad [A_{11}]_{\mathcal{B}} = \begin{pmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \end{pmatrix}. \quad (12.3)$$

The elements of this new basis are represented graphically in Figure 12.4. The first element A_{00} represents a uniform block. If the 2×2 block is completely uniform, only the coefficient of A_{00} will be nonzero. The two elements A_{01} and A_{10} represent left/right and top/bottom contrasts, respectively. The last element A_{11} represents a mixture of these two, where each pixel is in contrast with its neighbor along both directions, much like a checkerboard.

Knowing the A_{ij} in the standard basis, it is easy to obtain the change of basis matrix $[P]_{\mathcal{B}\mathcal{B}'}$ from \mathcal{B}' to \mathcal{B} . In fact, its columns are given by the coordinates of the elements of \mathcal{B}' expressed in the basis \mathcal{B} . It is therefore given by

$$[P]_{\mathcal{B}\mathcal{B}'} = [P]_{\mathcal{B}'\mathcal{B}}^{-1} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}. \quad (12.4)$$

To calculate $[g]_{\mathcal{B}'}$ we will need to use $[P]_{\mathcal{B}'\mathcal{B}}$, that is, the inverse of $[P]_{\mathcal{B}\mathcal{B}'}$. Here the matrix $[P]_{\mathcal{B}\mathcal{B}'}$ is orthogonal. (Exercise: A matrix A is orthogonal if $A^t A = A A^t = I$. Verify that $P_{\mathcal{B}\mathcal{B}'}$ is orthogonal.) The computation is therefore easy:

$$[P]_{\mathcal{B}'\mathcal{B}} = [P]_{\mathcal{B}\mathcal{B}'}^{-1} = [P]_{\mathcal{B}\mathcal{B}'}^t = [P]_{\mathcal{B}\mathcal{B}'}.$$

The last equality comes from the fact that the matrix $[P]_{\mathcal{B}\mathcal{B}'}$ is symmetric. The coefficients of g in this basis are simply

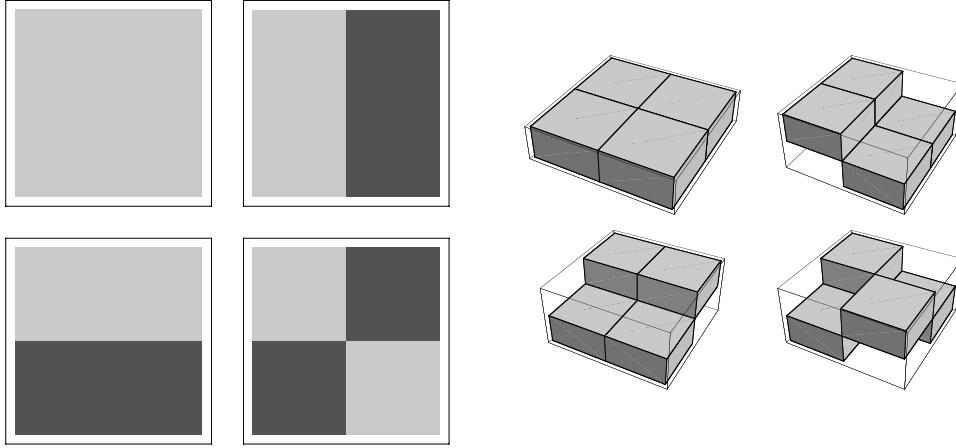


Fig. 12.4. The four elements of the proposed basis \mathcal{B}' . (Element A_{00} is at the upper left and element A_{01} is at the upper right.)

$$[g]_{\mathcal{B}'} = \begin{pmatrix} \beta_{00} \\ \beta_{01} \\ \beta_{10} \\ \beta_{11} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ \frac{5}{8} \\ \frac{1}{2} \\ \frac{3}{8} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \frac{1}{8} \\ -\frac{1}{8} \end{pmatrix}.$$

In this basis the largest coefficient is $\beta_{00} = 1$. This is the weight of the element A_{00} that gives an equal importance to each of the four pixels; in other words, this element of the new basis assigns them all the same gray tone. The two remaining nonzero coefficients, both much smaller in magnitude ($\beta_{10} = -\beta_{11} = \frac{1}{8}$), contain information regarding the small amount of contrast between the left and the right columns, and between the two pixels in the right column. The careful choice of the basis highlights spatial contrast information rather than giving individual pixel information. This is the heart of the JPEG standard. To make this technique lossy, one needs only to decide what coefficients correspond to visible contrasts for each of the elements of the basis. The rest of the coefficients may simply be thrown away.

12.4 The Case of $N \times N$ Blocks

The JPEG standard divides the image into 8×8 blocks. The definition of the basis that puts the focus on contrast information rather than individual pixels can equally be defined for arbitrary $N \times N$ blocks. The basis \mathcal{B}' that we introduced in the previous section ($N = 2$) and that used in the JPEG standard ($N = 8$) are particular cases.

The *discrete cosine transform*⁴ replaces the function $\{f_{ij}, i, j = 0, 1, 2, \dots, N - 1\}$ defined over an $N \times N$ square grid by a set of coefficients $\alpha_{kl}, k, l = 0, 1, \dots, N - 1$.

⁴The discrete cosine transform is a particular instance of a more general mathematical technique called *Fourier analysis*. Introduced at the beginning of the nineteenth century by

The coefficients α_{kl} are given by

$$\alpha_{kl} = \sum_{i,j=0}^{N-1} c_{ki} c_{lj} f_{ij}, \quad 0 \leq k, l \leq N-1, \quad (12.5)$$

where the c_{ij} are defined as

$$c_{ij} = \frac{\delta_i}{\sqrt{N}} \cos \frac{i(2j+1)\pi}{2N}, \quad i, j = 0, 1, \dots, N-1, \quad (12.6)$$

with

$$\delta_i = \begin{cases} 1, & \text{if } i = 0, \\ \sqrt{2}, & \text{otherwise.} \end{cases} \quad (12.7)$$

(Exercise: For the case $N = 2$, show that the coefficients c_{ij} are given by

$$C = \begin{pmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}.$$

Is it possible for the transformation (12.5) to be equivalent to the change of basis embodied by the matrix $[P]_{\mathcal{B}\mathcal{B}'}$ of (12.4)? Explain.)

The transformation in (12.5) from the $\{f_{ij}\}$ to the $\{\alpha_{kl}\}$ is clearly linear. By writing

$$\alpha = \begin{pmatrix} \alpha_{00} & \alpha_{01} & \dots & \alpha_{0,N-1} \\ \alpha_{10} & \alpha_{11} & \dots & \alpha_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{N-1,0} & \alpha_{N-1,1} & \dots & \alpha_{N-1,N-1} \end{pmatrix}, \quad f = \begin{pmatrix} f_{00} & f_{01} & \dots & f_{0,N-1} \\ f_{10} & f_{11} & \dots & f_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{N-1,0} & f_{N-1,1} & \dots & f_{N-1,N-1} \end{pmatrix},$$

and

$$C = \begin{pmatrix} \sqrt{\frac{1}{N}} & \sqrt{\frac{1}{N}} & \dots & \sqrt{\frac{1}{N}} \\ \sqrt{\frac{2}{N}} \cos \frac{\pi}{2N} & \sqrt{\frac{2}{N}} \cos \frac{3\pi}{2N} & \dots & \sqrt{\frac{2}{N}} \cos \frac{(2N-1)\pi}{2N} \\ \sqrt{\frac{2}{N}} \cos \frac{2\pi}{2N} & \sqrt{\frac{2}{N}} \cos \frac{6\pi}{2N} & \dots & \sqrt{\frac{2}{N}} \cos \frac{2(2N-1)\pi}{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{\frac{2}{N}} \cos \frac{(N-1)\pi}{2N} & \sqrt{\frac{2}{N}} \cos \frac{3(N-1)\pi}{2N} & \dots & \sqrt{\frac{2}{N}} \cos \frac{(2N-1)(N-1)\pi}{2N} \end{pmatrix},$$

we see that the transformation of (12.5) takes on the matrix form

$$\alpha = C f C^t, \quad (12.8)$$

Jean Baptiste Joseph Fourier for studying the propagation of heat, this technique has since invaded the world of engineering. It also plays an important role in Chapter 10.

where C^t denotes the transpose of the matrix C . In fact,

$$\alpha_{kl} = [\alpha]_{kl} = [CfC^t]_{kl} = \sum_{i,j=0}^{N-1} [C]_{ki}[f]_{ij}[C^t]_{jl} = \sum_{i,j=0}^{N-1} c_{ki}f_{ij}c_{lj},$$

which is the same as (12.5).

This transformation is an isomorphism if the matrix C is invertible. (That this is the case will be shown later.) If it is so, we are able to write

$$f = C^{-1}\alpha(C^t)^{-1}$$

and recover the values f_{ij} , $i, j = 0, 1, \dots, N - 1$, from the α_{kl} , $k, l = 0, 1, \dots, N - 1$. The transformation $f \mapsto \alpha$ given by (12.8) is also a linear transformation. Indeed, suppose that f and g are related to α and β through (12.8) (namely $\alpha = CfC^t$ and $\beta = CgC^t$). Then

$$C(f + g)C^t = CfC^t + CgC^t = \alpha + \beta$$

follows from the distributivity of matrix multiplication. And if $c \in \mathbb{R}$ then

$$C(cf)C^t = c(CfC^t) = c\alpha.$$

The two previous identities are the defining properties of linear transformations. Since this linear transformation is an isomorphism, it is a *change of basis!* Note that the passage from f to α is not expressed through a matrix $[P]_{\mathcal{B}'\mathcal{B}}$ as in the previous section. But linear algebra assures us that the transformation $f \mapsto \alpha$ could be written with such a matrix. (If the two indices of f run through $\{0, 1, \dots, N - 1\}$, then there are N^2 coordinates f_{ij} , and the matrix $[P]_{\mathcal{B}'\mathcal{B}}$ doing the change of basis is of size $N^2 \times N^2$. The form (12.8) has the advantage of using only $N \times N$ matrices.)

The proof of the invertibility of C rests on the observation that C is orthogonal:

$$C^t = C^{-1}. \tag{12.9}$$

This observation simplifies the calculations because the above expression for f becomes

$$f = C^t\alpha C. \tag{12.10}$$

We will give a proof of this property at the end of the section.

For the moment we will accept this fact and give an example of the transformation $f \mapsto \alpha$. To do this we will use the gray tones defined over the 8×8 block of Figure 12.1. The f_{ij} , $0 \leq i, j \leq 7$, are given in Table 12.2. The positions of pixels in the picture correspond to positions of entries in the table, and the entries are the gray intensities with 0 = black and 255 = white. The large numbers (> 150) correspond to the two white whiskers. The principal characteristic of this 8×8 block is the presence of diagonal stripes with high contrast. We will see how this contrast influences the coefficients α of this function.

The α_{kl} of the function f from Table 12.2 are given in Table 12.3. They are presented in the same order as previously, with α_{00} in the upper left and α_{07} in the upper right. None of the entries are exactly zero-valued, but we see that the largest coefficients (in terms of absolute value) are α_{00} , α_{01} , α_{12} , α_{23} , To interpret these numbers we need to have a better “visual” understanding of the elements of the basis \mathcal{B}' .

Consider once again the change of basis expressions

$$\alpha = CfC^t \quad \text{and} \quad f = C^t \alpha C.$$

In terms of the coefficients themselves, the relationship giving f from α is

$$f_{ij} = \sum_{k,l=0}^{N-1} \alpha_{kl} (c_{ki} c_{lj}).$$

Let A_{kl} be the $N \times N$ matrix whose elements are $[A_{kl}]_{ij} = c_{ki} c_{lj}$. We see that f is a linear combination of the matrices A_{kl} with weights α_{kl} . The set of N^2 matrices $\{A_{kl}, 0 \leq k, l \leq N - 1\}$ forms a basis in terms of which the function f is described. The 64 basis matrices A_{kl} of this example ($N = 8$) are shown in Figure 12.5. Matrix

40	193	89	37	209	236	41	14
102	165	36	150	247	104	7	19
157	92	88	251	156	3	20	35
153	75	220	193	29	13	34	22
116	173	240	54	11	38	20	19
162	255	109	9	26	22	20	29
237	182	5	28	20	15	28	20
222	33	8	23	24	29	23	23

Table 12.2. The 64 values of the function f .

681.63	351.77	-8.671	54.194	27.63	-55.11	-23.87	-15.74
144.58	-94.65	-264.52	5.864	7.660	-89.93	-24.28	-12.13
-31.78	-109.77	9.861	216.16	29.88	-108.14	-36.07	-24.40
23.34	12.04	53.83	21.91	-203.72	-167.39	0.197	0.389
-18.13	-40.35	-19.88	-35.83	-96.63	47.27	119.58	36.12
11.26	9.743	24.22	-0.618	0.0879	47.44	-0.0967	-23.99
0.0393	-12.14	0.182	-11.78	-0.0625	0.540	0.139	0.197
0.572	-0.361	0.138	-0.547	-0.520	-0.268	-0.565	0.305

Table 12.3. The 64 coefficients α_{kl} of the function f .

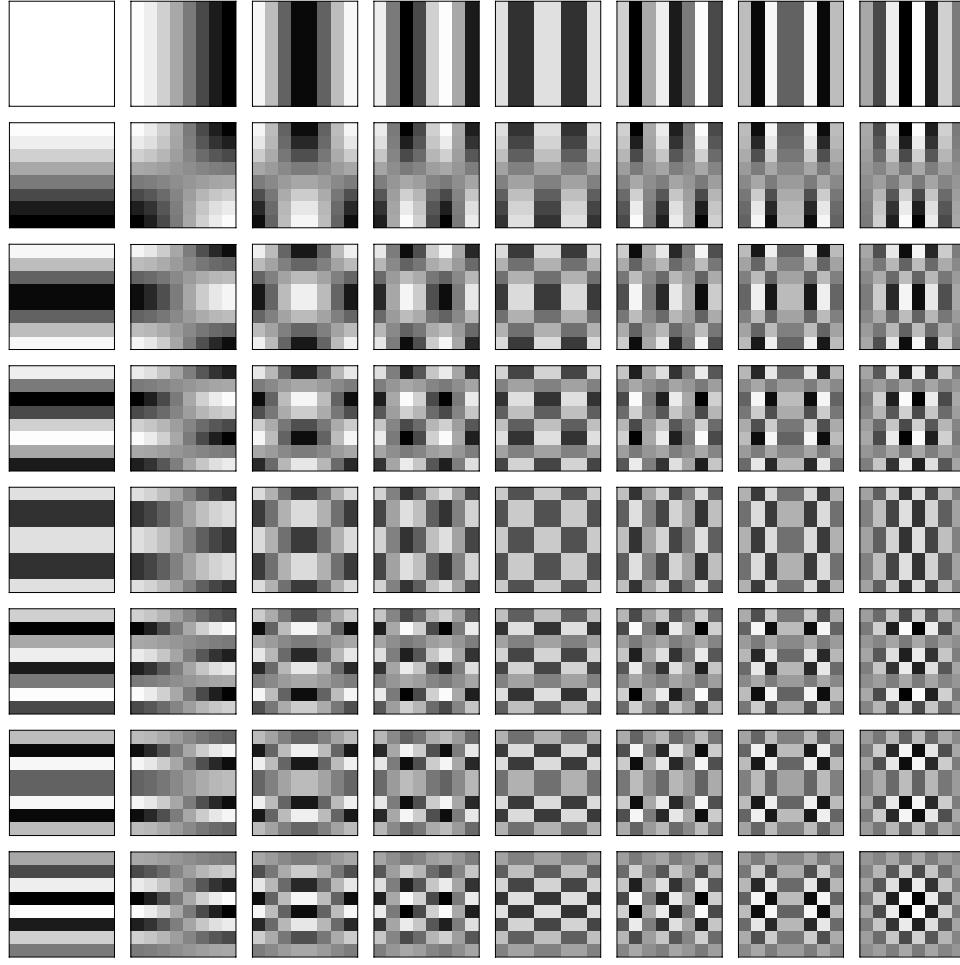


Fig. 12.5. The 64 elements A_{kl} of the basis \mathcal{B}' . Element A_{00} is at the upper left and element A_{07} is at the upper right.

A_{00} is in the upper left corner of the image, while A_{07} is found in the upper right. To graphically represent each basis matrix we needed to have their coefficients mapped to gray tones in the range 0 to 255. This was done by first replacing the $[A_{kl}]_{ij}$ by

$$[\tilde{A}_{kl}]_{ij} = \frac{N}{\delta_k \delta_l} [A_{kl}]_{ij},$$

where δ_k and δ_l are given by (12.7). This transformation ensures that $[\tilde{A}_{kl}]_{ij} \in [-1, 1]$. Next, the transformation aff_2 of (12.2) was applied to each scaled coefficient to obtain

$$[B_{kl}]_{ij} = \text{aff}_2([\tilde{A}_{kl}]_{ij}) = \left[\frac{255}{2} ([\tilde{A}_{kl}]_{ij} + 1) \right].$$

The $[B_{kl}]_{ij}$ can be directly interpreted as gray tones, since $0 \leq [B_{kl}]_{ij} \leq 255$. These are the values represented in Figure 12.5.

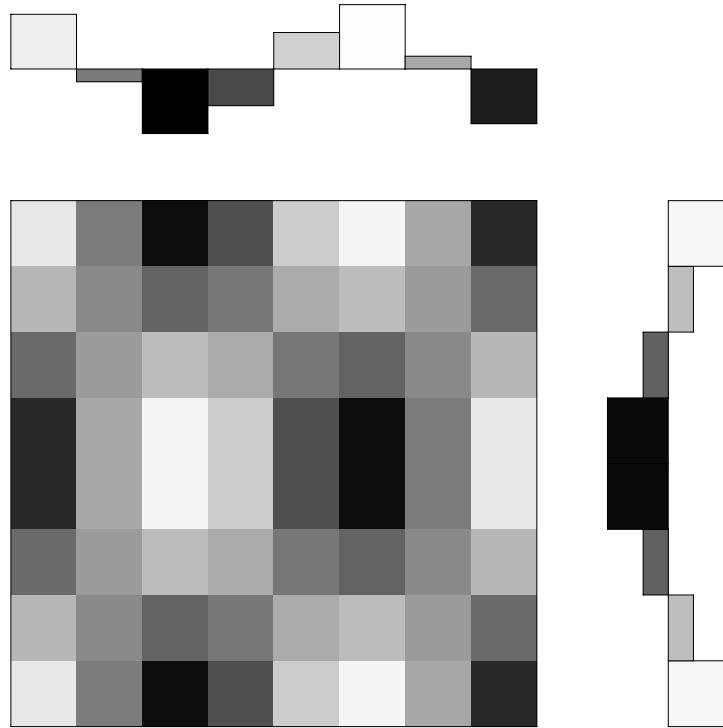


Fig. 12.6. Constructing the graphic representation of A_{23} .

It is possible to understand the graphic representations of the A_{kl} directly from their definitions. Here we consider the details of the construction of the element A_{23} , given by

$$[A_{23}]_{ij} = \frac{2}{N} \cos \frac{2(2i+1)\pi}{2N} \cos \frac{3(2j+1)\pi}{2N}.$$

The upper portion of Figure 12.6 shows the function

$$\cos \frac{3(2j+1)\pi}{16},$$

and at right, vertically, the function

$$\cos \frac{2(2i+1)\pi}{16}$$

has been shown. Since j varies from 0 to $N - 1 = 7$, the argument of the cosine of the first function passes from $3\pi/16$ to $3 \cdot 15\pi/16 = 45\pi/16 = 2\pi + 13\pi/16$ and the figure therefore shows roughly one and one-half cycles of the cosine. Each rectangle of the histogram has been assigned the gray tone corresponding to

$$\frac{255}{2} \left(\cos \left(\frac{3(2j+1)\pi}{16} \right) + 1 \right).$$

The same process has been repeated for the second function, $\cos 2(2i+1)\pi/16$, and the results of this shown vertically at the right of the figure. The function A_{23} is obtained by multiplying these two functions. This multiplication is between two cosine functions, thus between values in the range $[-1, 1]$. The result of this multiplication can be interpreted visually from the image. Multiplying two very light rectangles (corresponding to values near +1) or two very dark rectangles (corresponding to values near -1) results in light values. The 8×8 “product” of the two histograms is the matrix of basis element A_{23} .

We return to the 8×8 block depicting the two cat whiskers. What coefficients α_{kl} will be the most important? A coefficient α_{kl} will have larger magnitude if the extrema of the basis matrix correspond roughly to those of f . For example, the basis A_{77} (bottom right corner of Figure 12.5) alternates rapidly between black and white in both directions. It has many extrema, while f depicts only a diagonal pattern. As can be predicted, the associated coefficient is quite small at $\alpha_{77} = 0.305$. On the other hand, the coefficient α_{01} will be quite large. The basis matrix A_{01} (second from the left in the top row of Figure 12.5) contains a bright left half and a dark right half. Even though the two white whiskers of f extend into the right half of the 8×8 block, the left half is significantly paler than the right one. The actual coefficient is $\alpha_{01} = 351.77$.

How should we interpret a negative coefficient α_{kl} ? The coefficient $\alpha_{12} = -264.52$ is negative, and a closer inspection yields an answer. The basis matrix A_{12} is roughly divided into six contrasting bright and dark regions, three at the top and three at the bottom. Observe that two of the dark regions are roughly aligned with the brightest region of f , the whiskers. Multiplying this basis matrix by -1 would make these dark regions light, indicating that $-A_{12}$ describes the contrast between the whiskers and the background relatively well, thus the importance of this (negative) coefficient. We can easily repeat this “visual calculation” for each of the basis matrices, but it quickly becomes tedious. In fact, it is faster to program a computer to perform the calculations of (12.5). Regardless, this discussion has demonstrated the following intuitive rule: *the coefficient α_{kl} associated with a function f will have a significant magnitude if the extrema of A_{kl} are similar to those of f . A negative coefficient indicates that the bright spots of f matched dark spots of the basis element and vice versa.* As such, the nearly constant basis matrices A_{00} , A_{01} , and A_{10} are likely to have large factors α_{kl} for nearly constant functions f . At the other extreme, the basis matrices A_{67} , A_{76} , and A_{77} will be important for representing rapidly varying functions.

PROOF OF THE ORTHOGONALITY OF C (12.11): To show this somewhat surprising fact, we rewrite the identity $C^t C = I$ in terms of its coefficients:

$$[C^t C]_{jk} = \sum_{i=0}^{N-1} [C^t]_{ji} [C]_{ik} = \sum_{i=0}^{N-1} [C]_{ij} [C]_{ik} = \delta_{jk} = \begin{cases} 1, & \text{if } j = k, \\ 0, & \text{otherwise,} \end{cases}$$

or equivalently

$$[C^t C]_{jk} = \sum_{i=0}^{N-1} \frac{\delta_i^2}{N} \cos \frac{i(2j+1)\pi}{2N} \cos \frac{i(2k+1)\pi}{2N} = \delta_{jk}. \quad (12.11)$$

Proving (12.11) is equivalent to proving (12.9), the orthogonality of C , which implies the invertibility of (12.5). The proof that follows is not that difficult, but it contains several cases and subcases that must be carefully considered.

We expand the product of cosines from (12.11) using the trigonometric identity

$$\cos \alpha \cos \beta = \frac{1}{2} \cos(\alpha + \beta) + \frac{1}{2} \cos(\alpha - \beta).$$

Let $S_{jk} = [C^t C]_{jk}$. Then we have that

$$\begin{aligned} S_{jk} &= \sum_{i=0}^{N-1} \frac{\delta_i^2}{N} \cos \frac{i(2j+1)\pi}{2N} \cos \frac{i(2k+1)\pi}{2N} \\ &= \sum_{i=0}^{N-1} \frac{\delta_i^2}{2N} \left(\cos \frac{i(2j+2k+2)\pi}{2N} + \cos \frac{i(2j-2k)\pi}{2N} \right) \\ &= \sum_{i=0}^{N-1} \frac{\delta_i^2}{2N} \left(\cos \frac{2\pi i(j+k+1)}{2N} + \cos \frac{2\pi i(j-k)}{2N} \right). \end{aligned}$$

Since $\delta_i^2 = 1$ if $i = 0$ and $\delta_i^2 = 2$ otherwise, we can add the $i = 0$ term and subtract it to obtain

$$S_{jk} = \frac{1}{N} \sum_{i=0}^{N-1} \left(\cos \frac{2\pi i(j+k+1)}{2N} + \cos \frac{2\pi i(j-k)}{2N} \right) - \frac{1}{N}.$$

We split the proof into the following three cases: $j = k$, $j - k$ is even but nonzero, $j - k$ is odd. Observe that exactly one of $(j - k)$ and $(j + k + 1)$ is even, while the other is odd. We consider each of these cases by separating the sum and the term $-\frac{1}{N}$ as follows:

| $j = k$ We write $S_{jk} = S_1 + S_2$ with

$$S_1 = -\frac{1}{N} + \frac{1}{N} \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N}, \quad S_2 = \frac{1}{N} \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N},$$

where $l = j + k + 1$ is odd,

where $l = j - k = 0$.

| $j - k$ even and $j \neq k$ Write $S_{jk} = S_1 + S_2$ with

$$S_1 = -\frac{1}{N} + \frac{1}{N} \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N}, \quad S_2 = \frac{1}{N} \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N},$$

where $l = j + k + 1$ is odd,

where $l = j - k$ is even and nonzero.

$j - k$ odd Write $S_{jk} = S_1 + S_2$ with

$$S_1 = \frac{1}{N} \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N}, \quad S_2 = -\frac{1}{N} + \frac{1}{N} \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N},$$

where $l = j + k + 1$ is even,
nonzero, and $< 2N$,

where $l = j - k$ is odd.

There are three distinct sums to be studied:

$$\frac{1}{N} \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N}, \quad \text{where } l = 0, \quad (12.12)$$

$$\frac{1}{N} \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N}, \quad \text{where } l \text{ even, nonzero, and } < 2N, \quad (12.13)$$

$$-\frac{1}{N} + \frac{1}{N} \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N}, \quad \text{where } l \text{ odd.} \quad (12.14)$$

The first case is simple, since if $l = 0$ it follows that

$$\frac{1}{N} \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N} = \frac{1}{N} \sum_{i=0}^{N-1} 1 = \frac{N}{N} = 1.$$

Since we wish to show that S_{jk} is zero unless $j = k$ (otherwise, $S_{jj} = 1$), the proof is finished if we can show that (12.13) and (12.14) are both zero. For (12.13) recall that

$$\sum_{i=0}^{2N-1} e^{2\pi il\sqrt{-1}/2N} = \frac{e^{2\pi l \cdot 2N\sqrt{-1}/2N} - 1}{e^{2\pi l\sqrt{-1}/2N} - 1} = 0 \quad (12.15)$$

if $e^{2\pi l\sqrt{-1}/2N} \neq 1$. If $l < 2N$ this inequality is always satisfied. By taking the real part of (12.15) we find that

$$\sum_{i=0}^{2N-1} \cos \frac{2\pi il}{2N} = 0.$$

The sum contains twice as many terms as (12.13). However, we can rewrite it as

$$\begin{aligned}
0 &= \sum_{i=0}^{2N-1} \cos \frac{2\pi il}{2N} \\
&= \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N} + \sum_{i=N}^{2N-1} \cos \frac{2\pi il}{2N} \\
&= \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N} + \sum_{j=0}^{N-1} \cos \frac{2\pi(j+N)l}{2N}, \quad \text{for } i = j + N, \\
&= \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N} + \sum_{j=0}^{N-1} \cos \left(\frac{2\pi jl}{2N} + \frac{2\pi Nl}{2N} \right).
\end{aligned}$$

If l is even, the phase $\frac{2\pi Nl}{2N} = \pi l$ is an even multiple of π and can therefore be dropped, since the cosine is periodic with period 2π . Thus

$$0 = \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N} + \sum_{j=0}^{N-1} \cos \frac{2\pi jl}{2N} = 2 \sum_{i=0}^{N-1} \cos \frac{2\pi il}{2N},$$

and hence the sum of (12.13) is zero-valued.

Observe that the first term $i = 0$ of the sum from (12.14) is

$$\frac{1}{N} \cos \frac{2\pi \cdot 0 \cdot l}{2N} = \frac{1}{N},$$

which cancels the term $-\frac{1}{N}$. As such, the sum from (12.14) simplifies to

$$\sum_{i=1}^{N-1} \cos \frac{2\pi il}{2N}.$$

We must now divide case (12.14) into two subcases, N even and N odd. We divide the sum $\sum_{i=1}^{N-1} \cos \frac{2\pi il}{2N}$ as follows:

N odd

$$\sum_{i=1}^{\frac{N-1}{2}} \cos \frac{2\pi il}{2N} \quad \text{and} \quad \sum_{i=\frac{N-1}{2}+1}^{N-1} \cos \frac{2\pi il}{2N}$$

and

N even

$$\text{the term } i = \frac{N}{2}, \quad \sum_{i=1}^{\frac{N-1}{2}} \cos \frac{2\pi il}{2N}, \quad \text{and} \quad \sum_{i=\frac{N}{2}+1}^{N-1} \cos \frac{2\pi il}{2N}.$$

We start with this last subcase. If N is even, then for $i = N/2$ we have

$$\cos \frac{2\pi}{2N} \cdot \frac{N}{2} \cdot l = \cos \frac{\pi}{2} l = 0,$$

since l is odd. Rewrite the second sum by letting $j = N - i$; since $\frac{N}{2} + 1 \leq i \leq N - 1$, the domain of j is $1 \leq j \leq \frac{N}{2} - 1$:

$$\sum_{i=\frac{N}{2}+1}^{N-1} \cos \frac{2\pi il}{2N} = \sum_{j=1}^{\frac{N}{2}-1} \cos \frac{2\pi(N-j)l}{2N} = \sum_{j=1}^{\frac{N}{2}-1} \cos \left(\pi l - \frac{2\pi jl}{2N} \right).$$

And since l is odd, the phase πl is always an odd multiple of π , and

$$\sum_{i=\frac{N}{2}+1}^{N-1} \cos \frac{2\pi il}{2N} = \sum_{j=1}^{\frac{N}{2}-1} -\cos \left(-\frac{2\pi jl}{2N} \right).$$

Since the cosine function is even, we have finally that

$$\sum_{i=\frac{N}{2}+1}^{N-1} \cos \frac{2\pi il}{2N} = - \sum_{j=1}^{\frac{N}{2}-1} \cos \frac{2\pi jl}{2N},$$

and the two sums of the subcase cancel each other. The subcase of (12.14) where N is odd is left as an exercise to the reader. \square

12.5 The JPEG Standard

As discussed in the introduction, a good compression method will be tailored to the specific use and type of the object being compressed. The JPEG standard is intended for use in compressing images, more specifically photorealistic ones. As such, the compression technique is based on the fact that most photographs consist primarily of gentle gradients and transitions, while rapid variations are relatively rare. With what we have just learned about the discrete cosine transform and the coefficients α_{kl} , it seems natural to let the low-frequency components (with small l and k) play a large role, while letting high-frequency components (with l and k near N) play a small role. The following rule serves as a guide: all loss of information that is imperceptible to the human visual system (eyes and brain) is acceptable.

The compression algorithm can be broken down into the following major steps:

- translation of the image function,
- application of the discrete cosine transform to each 8×8 block,
- quantization of the transformed coefficients,

- zigzag ordering and encoding of the quantized coefficients.

We will describe each of these steps as applied to the image of a cat from Figure 12.1. This photo was taken by a digital camera that natively compressed the image in JPEG format. A 640×640 crop of the image was taken and subsequently converted to grayscale, with each pixel taking an integer value between 0 and 255. Recall that each pixel requires one byte of raw storage and therefore that the image requires $409,600 \text{ B} = 409.6 \text{ KB} = 0.4096 \text{ MB}$ to store uncompressed.

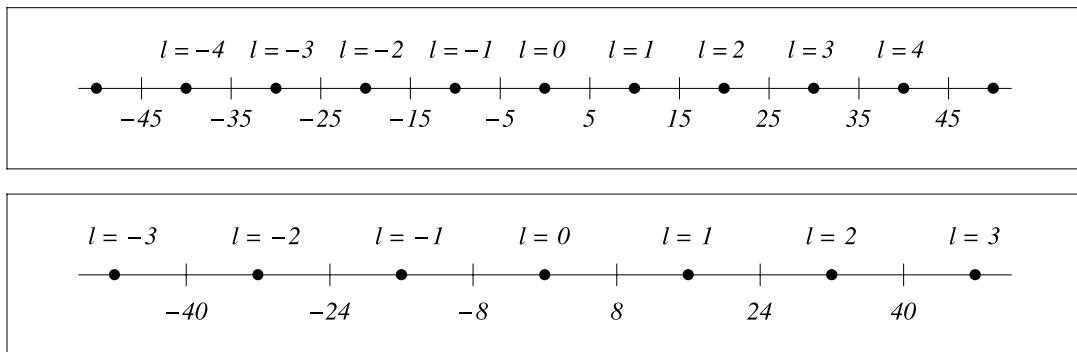
Translation of the image function. The first step is the *translation* of the values of f by the quantity 2^{b-1} , where b is the number of bits (or *bit depth*) used to represent each pixel. In our case we are using $b = 8$, and we therefore subtract $2^{b-1} = 2^7 = 128$ from each pixel. This first step produces a function \tilde{f} whose values are in the interval $[-2^{b-1}, 2^{b-1} - 1]$, which is (nearly) symmetric with respect to the origin, like the range of the cosine functions that form the basis matrices A_{kl} . We will follow the details of the algorithm on the 8×8 block shown in Table 12.2. The values of the translated function $\tilde{f}_{ij} = f_{ij} - 128$ are shown in Table 12.4, while the original values of the function f may be found in Table 12.2.

-88	65	-39	-91	81	108	-87	-114
-26	37	-92	22	119	-24	-121	-109
29	-36	-40	123	28	-125	-108	-93
25	-53	92	65	-99	-115	-94	-106
-12	45	112	-74	-117	-90	-108	-109
34	127	-19	-119	-102	-106	-108	-99
109	54	-123	-100	-108	-113	-100	-108
94	-95	-120	-105	-104	-99	-105	-105

Table 12.4. The 64 values of the function $\tilde{f}_{ij} = f_{ij} - 128$.

Discrete cosine transformation of each 8×8 block. The second step consists in partitioning the image into nonoverlapping blocks of 8×8 pixels. (If the image width is not a multiple of 8, then columns are added to the right until it is. The pixels in these additional columns are assigned the same gray tone as the rightmost pixel in each row of the original image. A similar treatment is applied to the bottom of the picture if the height is not a multiple of 8.) After *partitioning* the image into 8×8 blocks the *discrete cosine transform* is applied to each block. The result of this second step as applied to \tilde{f} is given in Table 12.5. If we compare these coefficients to the α_{kl} of f shown in Table 12.3, we see that only the coefficient α_{00} has changed. This is no coincidence and is a direct result of the fact that \tilde{f} is obtained from f by a translation. Exercise 11 (b) investigates why this happens.

-342.38	351.77	-8.671	54.194	27.63	-55.11	-23.87	-15.74
144.58	-94.65	-264.52	5.864	7.660	-89.93	-24.28	-12.13
-31.78	-109.77	9.861	216.16	29.88	-108.14	-36.07	-24.40
23.34	12.04	53.83	21.91	-203.72	-167.39	0.197	0.389
-18.13	-40.35	-19.88	-35.83	-96.63	47.27	119.58	36.12
11.26	9.743	24.22	-0.618	0.0879	47.44	-0.0967	-23.99
0.0393	-12.14	0.182	-11.78	-0.0625	0.540	0.139	0.197
0.572	-0.361	0.138	-0.547	-0.520	-0.268	-0.565	0.305

Table 12.5. The 64 coefficients α_{kl} of the function \tilde{f} .**Fig. 12.7.** The discrete scales used to measure α_{00} (top) and both α_{01} and α_{10} (bottom).

Quantization. The third step is called *quantization*: it consists in transforming the real-valued coefficients α_{kl} into integers ℓ_{kl} . The integer ℓ_{kl} is obtained from α_{kl} and q_{kl} by the formula

$$\ell_{kl} = \left[\frac{\alpha_{kl}}{q_{kl}} + \frac{1}{2} \right], \quad (12.16)$$

where $[x]$ is the integer part of x .

We explain the origins of this formula. Since the set of real numbers that can be represented on a computer is finite, the mathematical concept of the real line is not natural on computers. These numbers must be discretized, but must it be to the full precision that the computer is capable of representing? Could we not discretize them at a coarser scale? The JPEG standard gives a large amount of flexibility at this step: each coefficient α_{kl} is discretized with an individually chosen quantization step. The size of the step is encoded in the *quantization table*, which is fixed across all 8×8 blocks in a single image. The quantization table that we will use is shown in Table 12.6. For this table the step size for α_{00} will be 10, while already for α_{01} and α_{10} it will be 16. Figure 12.7 shows the effects of these step sizes for these three coefficients. Observe that all α_{00} from 5 up to but not including 15 will be mapped to the value $\ell_{00} = 1$; in

10	16	22	28	34	40	46	52
16	22	28	34	40	46	52	58
22	28	34	40	46	52	58	64
28	34	40	46	52	58	64	70
34	40	46	52	58	64	70	76
40	46	52	58	64	70	76	82
46	52	58	64	70	76	82	88
52	58	64	70	76	82	88	94

Table 12.6. The quantization table q_{kl} used in this example.

fact, from

$$\ell_{00}(5) = \left[\frac{5}{10} + \frac{1}{2} \right] = [1] = 1$$

and

$$\ell_{00}(15 - \epsilon) = \left[\frac{15 - \epsilon}{10} + \frac{1}{2} \right] = \left[2 - \frac{\epsilon}{10} \right] = 1$$

for an arbitrarily small positive number ϵ . Figure 12.7 shows the window of values that are mapped to the same quantized coefficient, each delimited by a small vertical bar. Any values of α_{kl} between two numbers below the axis will share the same ℓ at the moment of reconstruction, the ℓ noted above the central dot. These dots indicate the middle of each region, and the value $\ell_{kl} \times q_{kl}$ will be assigned to the coefficient when they are uncompressed. The fraction $\frac{1}{2}$ in (12.16) ensures that $\ell_{kl} \times q_{kl}$ falls in the middle of each window. The second axis of Figure 12.7 depicts the situation for α_{01} and α_{10} , whose quantification factor is larger, namely $q_{01} = q_{10} = 16$. More values of α_{01} (and α_{10}) will be identified to the same ℓ_{01} (and ℓ_{10}) due to this wider window. As can be seen, the larger the value of q_{kl} , the rougher the approximation of the reconstructed α_{kl} and the more information that is lost. The largest step size in our quantification table is $q_{77} = 94$. All coefficients α_{77} whose values lie in the range $[-47, 47]$ will map to the value $\ell_{77} = 0$. The precise value of the original coefficient in this interval will be irrevocably lost during the compression process.

Having chosen the quantization table shown in Table 12.6, we can quantify the transform coefficients of the original block f ; they are shown in Table 12.7.

Most digital cameras offer a way to save images at various quality levels (basic, normal, and fine, for example). Most software packages for manipulating digital images offer similar functionality. Once a given quality level has been chosen, the image is compressed using a quantization table that has been predetermined by the makers of the hardware or software. The same quantization table is used for *all* 8×8 blocks

-34	22	0	2	1	-1	-1	0
9	-4	-9	0	0	-2	0	0
-1	-4	0	5	1	-2	-1	0
1	0	1	0	-4	-3	0	0
-1	-1	0	-1	-2	1	2	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Table 12.7. The quantization ℓ_{kl} of the transformed coefficients α_{kl} .

in the image. It is transmitted once from the header of the JPEG file, followed by the transformed, quantized, and compressed block coefficients. Even though the JPEG standard suggests a family of quantization tables, any one may be used. As such, the quantization table offers a large amount of flexibility to the end user.

Zigzag ordering and encoding. The last step of the compression algorithm is the *encoding* of the table of quantized coefficients ℓ_{kl} . We will not delve too far into the details of this step. We will say only that the coefficient ℓ_{00} is encoded slightly differently from the rest and that the encoding uses the ideas discussed in the introduction: the values of ℓ_{kl} occurring more frequently are assigned shorter code words and vice versa. What are the most likely values? The JPEG standard prefers coefficients with a small absolute value: the smaller $|\ell_{kl}|$, the smaller the code word for ℓ_{kl} . Is it surprising that many coefficients ℓ_{kl} are nearly zero-valued? No, it is not if we recall that the α_{kl} (and hence the ℓ_{kl}) typically measure changes that are relatively small in scope with respect to the actual size of the image.

Thanks to the quantization step, many ℓ_{kl} with large k and l are zero-valued. The encoding makes use of this fact by ordering the coefficients such that long strings of zero-valued coefficients are more likely. The precise ordering defined by the JPEG standard is shown in Figure 12.8: $\ell_{01}, \ell_{10}, \ell_{20}, \ell_{11}, \ell_{02}, \ell_{03}, \dots$. Given that most of the nonzero coefficients tend to be clustered in the upper left corner, it often happens that the coefficients ordered in this manner are terminated by a long run of zero values. Rather than encoding each of these zero values, the encoder sends a single special code word indicating the “end of block.” When the decoder encounters this symbol it knows that the rest of the 64 symbols are to be filled in with zeros. Looking at Table 12.7, note that $\ell_{46} = 2$ is the last nonzero coefficient in the proposed zigzag ordering. The eleven remaining coefficients ($\ell_{37}, \ell_{47}, \ell_{56}, \ell_{65}, \ell_{74}, \ell_{75}, \ell_{66}, \ell_{57}, \ell_{67}, \ell_{76}, \ell_{77}$) are all zero-valued and will not be explicitly transmitted. As we will see in the example of the image of the cat, this provides an enormous gain to the compression ratio.

Reconstruction. A computer can quickly reconstruct a photo from the information in a JPEG file. The quantification table is first read from the file header. Then the

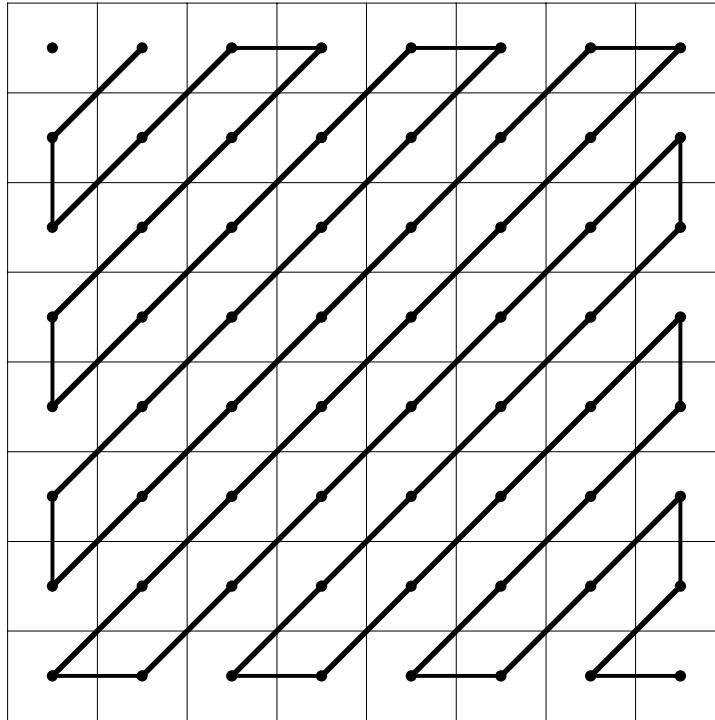


Fig. 12.8. The order in which the coefficients ℓ_{kl} are transmitted: $\ell_{01}, \ell_{1,0}, \dots, \ell_{77}$.

following steps are performed for each 8×8 block: the information for a block is read until the “end of block” signal is encountered. If fewer than 64 coefficients were read, the missing ones are set to zero. The computer then multiplies each ℓ_{kl} by the corresponding q_{kl} . The coefficient $\beta_{kl} = \ell_{kl} \times q_{kl}$ is therefore chosen in the middle of the quantification window where the original α_{kl} lay. The inverse of the discrete cosine transformation (12.10) is then applied to the β ’s to get the new gray tones \bar{f} :

$$\bar{f} = C^t \beta C.$$

After correcting for the translation of the original image, the gray tones for this 8×8 block are ready to be shown on screen.

Figure 12.9 shows the visual results of JPEG compression, applied to the entire image as described in this section. Recall that the original photo contains 640×640 pixels and therefore $80 \times 80 = 6400$ blocks of 8×8 pixels. The four steps (translation, transformation, quantization, and encoding) are thus performed 6400 times. The left column of Figure 12.9 contains the original image plus two successive closeups.⁵ The right column contains the same image after being JPEG compressed and decompressed using the quantization table of Table 12.6.

⁵Recall that the original photo was obtained from a digital camera that itself stores the image in JPEG form.

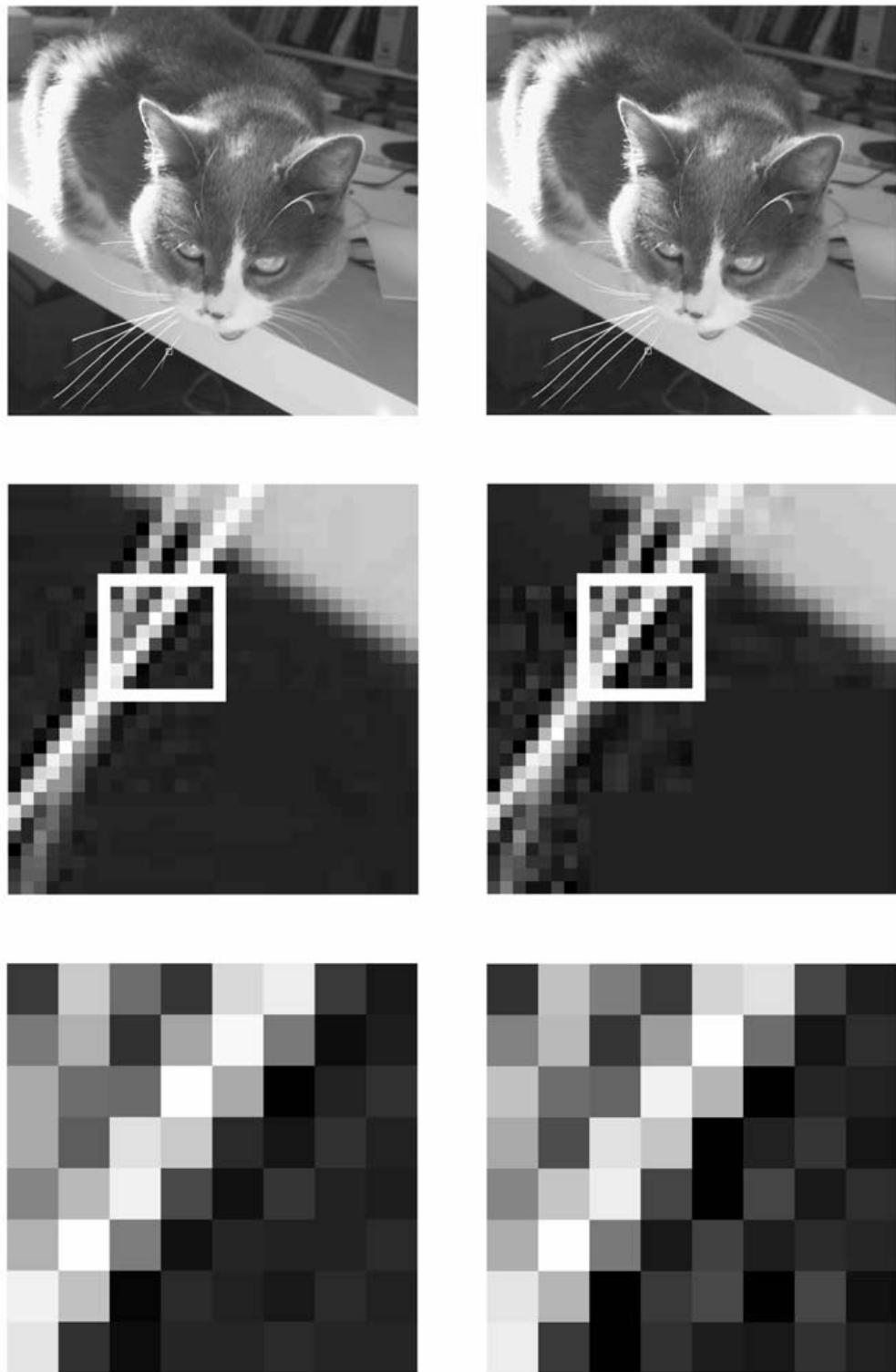


Fig. 12.9. The three images at the left are the same as those of Figure 12.1. Those at the right have been obtained from this image after being heavily JPEG compressed. The middle blocks are 32×32 pixels, while those at the bottom are 8×8 pixels.

The 8×8 block containing the crossing of two whiskers has been chosen because it is a block with high contrast. These are the types of blocks that are the least well compressed by the JPEG standard. By comparing the closeups we can see the effect of the aggressive compression. Close to the border between the highly contrasting regions the effect is most noticeable. Since this block contains high-contrast quickly varying data, we would have had to store the coefficients α_{kl} with more precision in order to reproduce them clearly. The aggressive zeroing of many of these coefficients in the quantization step has introduced a certain “noise” close to the whiskers. Note that a certain amount of noise was already present in this region in the original photograph, a clear sign that the camera was using JPEG compression. Another clear sign that JPEG compression has been used is the often visible boundaries of 8×8 blocks, specifically blocks containing high contrast next to smooth blocks, as is the case in the region of the whiskers. Notice the 8×8 block second from the bottom and third from the left of the 32×32 blocks in Figure 12.9. This block is completely “under the table” and has been compressed to a uniform gray. As such, it is not surprising that after quantization it contains only two nonzero coefficients (ℓ_{00} and ℓ_{10}). The encoding of this block omits 62 coefficients, and the compression is very good!

Is this block the rule or the exception? There are $640 \times 640 = 409,600$ pixels in the entire image. After transforming and quantizing these coefficients, the image is encoded by a series of 409,600 coefficients ℓ_{kl} . By ordering them in zigzag order and omitting the trailing runs of zeros, we are able to avoid storing over 352,000 zeros, roughly $\frac{7}{8}$ of the coefficients! It is not surprising that the compression achieved by the JPEG standard is so good.⁶

The ultimate test is the comparison of the two images with the naked eye. It is up to the user to judge whether the compression (in this case, the zeroing of roughly $\frac{7}{8}$ of the Fourier coefficients α_{kl}) has damaged the photograph. It is important to note that this comparison should be performed under the same conditions in which the compressed photograph will be used. Recall the example of the digitized works from the Louvre. If the image is going to be looked at using a low-resolution screen, then the compression can be relatively aggressive. However, if the image is to be closely studied by art historians, is to be printed at high resolution, or is to be viewed through software that allows zooming in, then a higher resolution and a less-aggressive compression should be used.

The JPEG standard offers an enormous amount of flexibility through its quantization tables. In certain cases we can imagine that using even higher values in this table will lead to better compression and acceptable quality. However, the weaknesses of the JPEG standard are made apparent in areas of high contrast and detail, especially when the quantization table contains overly large values. This is why the JPEG standard performs so poorly at compressing line art and cartoons, which consist largely of black lines on a white background. These lines become marred (with a characteristic JPEG

⁶Through careful choice of the quantization table this photo can be compressed to less than 30 KB in size (compared to 410 KB uncompressed) without the degradation being intolerable.

“speckle”) after aggressive compression. It would be equally inappropriate to take a picture of a page of text and compress it using the JPEG standard; the letters are in high contrast with the page and would become blurred. The JPEG standard was created with the goal of compressing photographs and photorealistic images and it excels at this task.

What about color images? It is well known that colors can be described using three dimensions. For example, the color of a pixel on a computer screen is normally described as a ratio of the three (additive) primary colors: red, green, and blue. The JPEG standard uses a different set of coordinates (or *color space*). It is based on recommendations made by the *Commission internationale de l'éclairage* (International Commission on Illumination), which in the 1930s developed the first standards in this domain. The three dimensions of this color space are separated, leading to three independent images. These images, each corresponding to one coordinate, are then individually treated in the same manner as discussed in this chapter for gray tones. (For those who want to learn more, the book [2] contains a self-contained description of the standard with enough information to fully implement the standard, a discussion of the science underlying the various mathematical tools used in it, and the necessary knowledge on the human visual system. References [3, 4] are good entry points in the field of data compression.)

12.6 Exercises

1. (a) Verify that if $x \in [-1, 1] \subset \mathbb{R}$, then $\text{aff}_1(x) = 255(x+1)/2$ is an element of $[0, 255]$.
 (b) Is aff_1 the ideal transformation? For which x will $\text{aff}_1(x) = 255$? Can you propose a function aff' such that all integers in $\{0, 1, 2, \dots, 255\}$ will be images of equal-length subintervals of $[-1, 1]$?
 (c) Give the inverse of aff_1 . The function aff' cannot have an inverse. Why? Despite this, can you propose a rule that would allow you to construct a function g starting from a function f as in Section 12.3?
2. (a) Verify that the four vectors A_{00} , A_{01} , A_{10} , and A_{11} of (12.3) (expressed in the usual basis \mathcal{B}) are orthonormal, that is, they have length 1 and are pairwise orthogonal.
 (b) Let v be the vector whose coefficients in the basis \mathcal{B} are

$$[v]_{\mathcal{B}} = \begin{pmatrix} -\frac{3}{8} \\ \frac{5}{8} \\ \frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}.$$

Give the coefficients of this vector in the basis $\mathcal{B}' = \{A_{00}, A_{01}, A_{10}, A_{11}\}$. What is the largest coefficient of $[v]_{\mathcal{B}'}$ in terms of absolute value? Could you have guessed which one it was going to be without explicitly calculating them? How?

3. (a) Show that the $N \times N$ matrix C used in the discrete cosine transform for $N = 4$ is given by

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \gamma & \delta & -\delta & -\gamma \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \delta & -\gamma & \gamma & -\delta \end{pmatrix}.$$

Express the two unknowns γ and δ in terms of the cosine function.

- (b) Using the trigonometric identity $\cos 2\theta = 2\cos^2 \theta - 1$, explicitly give the numbers γ and δ . (Here “explicitly” means as an algebraic expression with integer numbers and radicals *but* without the cosine function.) Using these expressions, show that the second line of C represents a vector with unit norm as is required by the orthogonality of C .

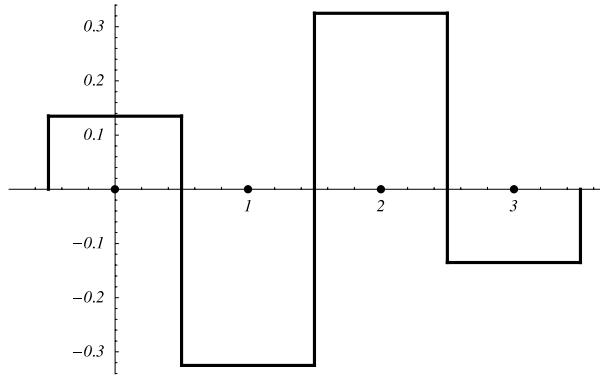


Fig. 12.10. The discrete function g of Exercise 4 (b).

4. (a) The discrete cosine transformation allows the expression of discrete functions $g : \{0, \dots, N-1\} \rightarrow \mathbb{R}$ (given by $g(i) = g_i$) as linear combinations of the N discrete basis vectors C_k , where $C_k(i) = (C_k)_i = c_{ki}$, $k = 0, 1, 2, \dots, N-1$. This transformation expresses g in the form $g = \sum_{k=0}^{N-1} \beta_k C_k$, which yields

$$g_i = \sum_{k=0}^{N-1} \beta_k (C_k)_i.$$

For $N = 4$, represent the function $(C_2)_i$ by a histogram. (This exercise reuses results from Exercise 3, but the reader is not required to have completed that exercise.)

- (b) Knowing that the numeric values of γ and δ of the previous exercise are roughly 0.65 and 0.27 respectively, what will be the coefficient β_k with the largest magnitude for the function g represented in Figure 12.10?

5. Complete the calculation of (12.14) for the subcase in which N is odd.



Fig. 12.11. The function f of Exercise 6.

6. A function

$$f : \{0, 1, 2, 3, 4, 5, 6, 7\} \times \{0, 1, 2, 3, 4, 5, 6, 7\} \rightarrow \{0, 1, 2, \dots, 255\}$$

is represented graphically by the gray tones of Figure 12.11. The values f_{ij} are constant along a given row; in other words, $f_{ij} = f_{ik}$ for all $j, k \in \{0, 1, 2, \dots, 7\}$.

- (a) If $f_{0j} = 0, f_{1j} = 64, f_{2j} = 128, f_{3j} = 192, f_{4j} = 192, f_{5j} = 128, f_{6j} = 64, f_{7j} = 0$ for all j , calculate α_{00} as defined by the JPEG standard, but without doing the translation of f as described in the first step of Section 12.5.
 - (b) If the discrete cosine transform is carried out as suggested by the JPEG standard, several of the coefficients α_{kl} will be zero-valued. Determine which elements of α_{kl} will be zero-valued and explain why.
7. Let C be the matrix representing the discrete cosine transform. Its elements $[C]_{ij} = c_{ij}, 0 \leq i, j \leq N - 1$, are given by (12.6). Let N be even. Show that each of the elements of rows i of C where i is odd is one of the following N values:

$$\pm \sqrt{\frac{2}{N}} \cos \frac{k\pi}{2N}, \quad \text{with } k \in \{1, 3, 5, \dots, N - 1\}.$$

8. Figure 12.12 displays an 8×8 block of gray tones. Which coefficient α_{ij} will have the largest magnitude (ignoring α_{00})? What will its sign be?

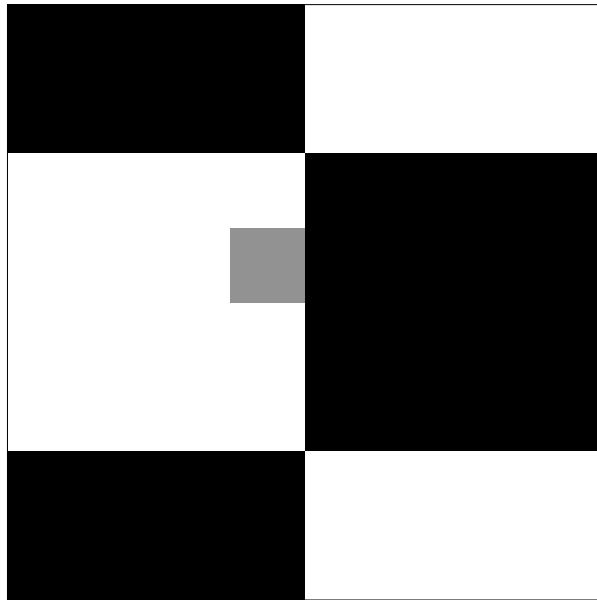


Fig. 12.12. An 8×8 block of gray tones for Exercise 8.

9. With the rising popularity of digital photography, programs allowing for the manipulation and retouching of photographs have become increasingly popular. Among other things, they allow images to be reframed (or *cropped*) by removing rows or columns from the outer edges. If an image is JPEG compressed, explain why it is better to remove groups of rows or columns that are multiples of 8.

10. (a) Two copies of the same photograph are independently compressed using distinct quantization tables q_{ij} and q'_{ij} . If $q_{ij} > q'_{ij}$ for all i and j , what will be, in general, the larger file, the second or the first? Which quantization table will lead to a larger loss of quality in the photograph?
 (b) If the quantization table from Table 12.6 is used and if $\alpha_{34} = 87.2$, what will be the value of ℓ_{34} ? What if $\alpha_{34} = -87.2$?
 (c) What is the smallest value of q_{34} that will lead to a zero-valued ℓ_{34} for the values of α_{34} in the preceding question?
 (d) Does $\ell_{kj}(-\alpha_{kj}) = -\ell_{kj}(\alpha_{kj})$? Explain.

Note: Another slightly different problem is raised by technology. Suppose a photo is already in the JPEG format and is available through the Internet. If the file remains large, it could be useful to recompress the file using a more aggressive quantification table for users having slower Internet connections. The choice of the new quantification table would then depend on the speed of the connection and perhaps on the use of the photo. It turns out that the choice of this second table is delicate, since the degradation of the picture does not increase monotonically with the size of its coefficients. See, for example, [1].

11. (a) Calculate the difference between the α_{00} of the function f given in Table 12.2 and that of the function \tilde{f} obtained through translation.
 (b) Show that a translation of f by any constant (for example 128) changes only the coefficient α_{00} .
 (c) Using the definition of the discrete cosine transform, predict the difference between the two coefficients α_{00} calculated in (a).
 (d) Show that α_{00} is N times the average gray tone of the block.
12. Let g be a step function representing a checkerboard: the upper left corner $(0, 0)$ has value +1, and the rest of the squares are filled in such a way that they have the opposite sign to their horizontal and vertical neighbors.
- (a) Show that the step function g_{ij} can be described by the formula

$$g_{ij} = \sin(i + \frac{1}{2})\pi \cdot \sin(j + \frac{1}{2})\pi.$$

- (b) Calculate the eight numbers

$$\lambda_i = \sum_{j=0}^7 c_{ij} \sin(j + \frac{1}{2})\pi, \quad \text{for } i = 0, \dots, 7,$$

where c_{ij} is given by (12.6). (If this exercise is taking too long to perform by hand, consider using a computer!)

- (c) Calculate the coefficients β_{kl} of the checkboard function g given by $\beta_{kl} = \sum_{i,j=0}^{N-1} c_{ki} c_{lj} g_{ij}$ (calculating the values λ_i is helpful). Could you have guessed exactly which coefficients would be zero-valued? Is the position of the largest nonzero coefficient β_{kl} surprising?

References

- [1] H.H. Bauschke, C.H. Hamilton, M.S. Macklem, J.S. McMichael, and N.R. Swart. Recompression of JPEG images by requantization. *IEEE Transactions on Image Processing*, 12:843–849, 2003.
- [2] W.B. Pennebaker and J.L. Mitchell. *JPEG Still Image Data Compression Standard*. Springer, New York, 1996.
- [3] D. Salomon. *Data Compression: The Complete Reference*. Springer, New York, 2nd edition, 2000.
- [4] K. Sayood. *Introduction to Data Compression*. Morgan Kaufmann, San Francisco, 1996.

Bibliographie

- [1] P. Brémaud, “Signaux aléatoires pour le traitement du signal et les communications”, *Collection: Cours de l’Ecole Polytechnique, Ellipses*, 1993.
- [2] P. Brémaud, “Introduction aux probabilités”, Springer Verlag, Berlin.
- [3] J.M. Bony, “Cours d’Analyse”, *Ecole Polytechnique*, 1994.
- [4] J.M. Bony, “Méthodes mathématiques pour les sciences physiques,” *Ecole Polytechnique*, 1995.
- [5] J.F. Genat, “Synthèse et traitement des sons en temps réel”, *Ecole Polytechnique*, 1994.
- [6] J.F Genat et A. Karar, “Introduction à l’analyse et synthèse de la parole”, *Ecole Polytechnique*, 1994.
- [7] S. Haykin, “Adaptive filter theory”, *Prentice Hall*, 1991.
- [8] A. Oppenheim et R. Schafer, “Discrete-time signal processing”, *Prentice Hall*, 1989.
- [9] T. Parsons, “Voice and speech processing”, *Mc-Graw Hill*, 1987.
- [10] A. Papoulis, “Signal analysis”, *Mc Graw Hill*, 1977.
- [11] M.B. Priestley, “Spectral analysis and time series”, *Academic Press*, 1981.
- [12] Y. Thomas, “Signaux et systèmes linéaires”, *Masson*, 1994.
- [13] B. Torrésani, “Analyse continue par ondelettes”, *CNRS editions*, 1995.
- [14] M. Vetterli and J. Kovacevic, “Wavelets and subband coding”, *Prentice Hall*, 1995.

Part V

Compléments mathématiques

Appendix A

Bases d'analyse Hilbertienne

Ce chapitre n'est pas un cours d'analyse Hilbertienne, mais rassemble les notions essentielles que nous aurons à manipuler dans ce cours. La plupart des résultats sont élémentaires et sont démontrés.

A.1 Définitions

Définition 157 (Espace pré-hilbertien). Soit \mathcal{H} un espace vectoriel sur l'ensemble des nombres complexes \mathbb{C} . L'espace \mathcal{H} est appelé pré-hilbertien si \mathcal{H} est muni d'un produit scalaire :

$$\langle \cdot, \cdot \rangle : x, y \in \mathcal{H} \times \mathcal{H} \mapsto \langle x, y \rangle \in \mathbb{C}$$

qui vérifie les propriétés suivantes :

- a) pour tout $(x, y) \in \mathcal{H} \times \mathcal{H}$, $\langle x, y \rangle = \overline{\langle y, x \rangle}$
- b) pour tout $(x, y) \in \mathcal{H} \times \mathcal{H}$ et tout $(\alpha, \beta) \in \mathbb{C} \times \mathbb{C}$, $\langle \alpha x + \beta y, z \rangle = \alpha \langle x, z \rangle + \beta \langle y, z \rangle$
- c) pour tout $x \in \mathcal{H}$, $\langle x, x \rangle \geq 0$, et $\langle x, x \rangle = 0$ si et seulement si $x = 0$.

L'application :

$$\| \cdot \| : x \in \mathcal{H} \mapsto \sqrt{\langle x, x \rangle} \geq 0$$

définit alors une norme sur \mathcal{H} .

Exemple 158 (Espace \mathbb{C}^n). L'ensemble des vecteurs colonnes $x = [x_1 \quad \cdots \quad x_n]^T$, où $x_k \in \mathbb{C}$, est un espace vectoriel dans lequel la relation :

$$\langle x, y \rangle = y^H x = \sum_{k=1}^n x_k \overline{y_k}$$

définit un produit scalaire.

Exemple 159 (Espace $\ell^2(\mathbb{Z})$). L'ensemble des suites numériques complexes $\{x_k\}_{k \in \mathbb{N}}$ vérifiant $\sum_{k=0}^{\infty} |x_k|^2 < \infty$ est un espace vectoriel sur \mathbb{C} . On définit pour tout x et y de cet espace :

$$\langle x, y \rangle = \sum_{k=0}^{\infty} x_k \overline{y_k}.$$

Cette somme est bien définie puisque $|x_k \overline{y_k}| \leq (|x_k|^2 + |y_k|^2)/2$. De plus, on vérifie aisément les propriétés (i-iii) de la définition 157. L'espace ainsi défini est donc un espace pré-Hilbertien, que l'on note $\ell^2(\mathbb{Z})$.

Exemple 160 (Fonctions de carré intégrable). L'ensemble $\mathscr{L}^2(T)$ des fonctions boréliennes définies sur un intervalle T de \mathbb{R} , à valeurs complexes et de module de carré intégrable par rapport à la mesure de Lebesgue ($\int_T |f(t)|^2 dt < \infty$) est un espace vectoriel. Considérons alors le produit intérieur :

$$(f, g) \in \mathscr{L}^2(T) \times \mathscr{L}^2(T) \mapsto \langle f, g \rangle = \int_T f(t) \overline{g(t)} dt$$

Cette intégrale est bien définie puisque $|f(t)\overline{g(t)}| \leq (|f(t)|^2 + |g(t)|^2)/2$ et l'on montre aisément que les propriétés (i) et (ii) de la définition 157. Par contre la propriété (iii) n'est pas vérifiée puisque :

$$\langle f, f \rangle = 0 \not\Rightarrow \forall t \in T \ f(t) = 0$$

En effet une fonction f qui est nulle sauf sur un ensemble de mesure nulle pour la mesure de Lebesgue, vérifie $\langle f, f \rangle = 0$. L'espace \mathcal{H} muni du produit (f, g) n'est donc pas un espace pré-Hilbertien. C'est pourquoi on définit l'ensemble $L_2(T)$ des classes d'équivalence de $\mathcal{L}^2(T)$ pour la relation d'équivalence définie par l'égalité presque partout entre deux fonctions. Par construction, $L^2(T)$ est alors un espace pré-Hilbertien.

Exemple 161 (Variables aléatoires de variance finie). De façon similaire à l'exemple 160, pour tout espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$, on définit $\mathcal{H} = \mathcal{L}^2(\Omega, \mathcal{F}, \mathbb{P})$ (noté $\mathcal{L}^2(\Omega)$ s'il n'y a pas de confusion possible) comme l'ensemble des v.a. X définies sur $(\Omega, \mathcal{F}, \mathbb{P})$ à valeurs complexes telles que

$$\mathbb{E}[|X|^2] < \infty.$$

Sur cet ensemble, on définit

$$(X, Y) \in \mathcal{L}^2(\Omega) \times \mathcal{L}^2(\Omega) \mapsto \langle X, Y \rangle = \mathbb{E}[X\bar{Y}] .$$

Pour les mêmes raisons que dans l'exemple 160, on définit l'espace pré-Hilbertien $L^2(\Omega, \mathcal{F}, \mathbb{P})$ (ou $L^2(\Omega)$) comme l'ensemble des classes d'équivalences de $\mathcal{L}^2(\Omega)$ pour la relation d'équivalence définie par l'égalité presque sûre entre deux v.a. Cet exemple et l'exemple 160 se généralisent en fait à tout espace mesuré $(\Omega, \mathcal{F}, \mu)$ en posant

$$(f, g) \in \mathcal{L}^2(\Omega, \mathcal{F}, \mu) \times \mathcal{L}^2(\Omega, \mathcal{F}, \mu) \mapsto \langle f, g \rangle = \int f \bar{g} \, d\mu .$$

On montre aisément les propriétés suivantes :

Théorème 162. Pour tout $x, y \in \mathcal{H} \times \mathcal{H}$, nous avons :

- a) Inégalité de Cauchy-Schwarz: $|\langle x, y \rangle| \leq \|x\| \|y\|$,
- b) Inégalité triangulaire: $\|x\| - \|y\| \leq \|x - y\| \leq \|x\| + \|y\|$,
- c) Identité du parallélogramme:

$$\|x + y\|^2 + \|x - y\|^2 = 2\|x\|^2 + 2\|y\|^2$$

Définition 163 (Convergence forte dans \mathcal{H}). Soit (x_n) une suite de vecteurs et x un vecteur d'un espace préhilbertien \mathcal{H} . On dit que (x_n) tend fortement vers x si et seulement si $\|x_n - x\| \rightarrow 0$ quand $n \rightarrow +\infty$. On note $x_n \rightarrow x$.

Proposition 164. Si dans un espace de Hilbert la suite $x_n \rightarrow x$, alors (x_n) est bornée.

Proof. D'après l'inégalité triangulaire, on a :

$$\|x_n\| = \|(x_n - x) + x\| \leq \|x_n - x\| + \|x\|$$

□

Définition 165 (Convergence faible dans \mathcal{H}). Soit (x_n) une suite de vecteurs et x un vecteur d'un espace préhilbertien \mathcal{H} . On dit que (x_n) tend faiblement vers x si et seulement si, pour tout $y \in \mathcal{H}$, $\langle x_n, y \rangle \rightarrow \langle x, y \rangle$ quand $n \rightarrow \infty$. On note $x_n \rightsquigarrow x$.

Théorème 166. Une suite (x_n) fortement convergente converge aussi faiblement vers la même limite.

Proof. Supposons que (x_n) converge fortement vers x , $\lim_{n \rightarrow \infty} \|x_n - x\| = 0$. Alors, pour tout $y \in \mathcal{H}$,

$$|\langle x_n - x, y \rangle| \leq \|x_n - x\| \|y\| \rightarrow 0, \quad \text{quand } n \rightarrow \infty.$$

□

En général, la convergence faible n'entraîne pas la convergence forte. Pour tout $y \in \mathcal{H}$, l'application $\langle \cdot, y \rangle : \mathcal{H} \rightarrow \mathbb{C}, x \mapsto \langle x, y \rangle$ est une forme linéaire. Le théorème 166 montre que cette forme linéaire est continue.

Théorème 167 (Continuité du produit scalaire). Soit $x_n \rightarrow x$ et $y_n \rightarrow y$ deux suites convergentes de vecteurs d'un espace pré-hilbertien \mathcal{H} . Alors quand $n \rightarrow +\infty$: $\langle x_n, y_n \rangle \rightarrow \langle x, y \rangle$. En particulier, si $x_n \rightarrow x$, $\|x_n\| \rightarrow \|x\|$.

Proof. D'après l'inégalité triangulaire puis l'inégalité de Cauchy-Schwarz, nous avons :

$$\begin{aligned} \langle x, y \rangle - \langle x_n, y_n \rangle &= \langle (x - x_n) + x_n, (y - y_n) + y_n \rangle - \langle x_n, y_n \rangle \\ &= \langle x - x_n, y - y_n \rangle + \langle x - x_n, y_n \rangle + \langle x_n, y - y_n \rangle \\ &\leq \|x_n - x\| \|y_n - y\| + \|x_n - x\| \|y_n\| + \|y_n - y\| \|x_n\| \end{aligned}$$

On conclut en utilisant la proposition 164 qui montre que les suites (x_n) et (y_n) sont bornées.

□

Définition 168 (Suite de Cauchy). Soit (x_n) une suite de vecteurs d'un espace vectoriel normé $(\mathcal{H}, \|\cdot\|)$. On dit que (x_n) est une suite de Cauchy si et seulement si :

$$\|x_n - x_m\| \rightarrow 0$$

quand $n, m \rightarrow +\infty$.

Notons qu'en vertu de l'inégalité triangulaire toute suite convergente est une suite de Cauchy. La réciproque est fausse : une suite de Cauchy peut ne pas être convergente. Un contre-exemple est donné par l'exemple 174.

Définition 169 (Espace complet, espace de Hilbert). *On dit qu'un espace vectoriel normé $(\mathcal{H}, \|\cdot\|)$ est complet si toute suite de Cauchy d'éléments de cet espace converge dans cet espace. On dit \mathcal{H} est un espace de Hilbert si \mathcal{H} est pré-hilbertien et complet.*

Proposition 170 (Suites normalement convergentes). *Un espace vectoriel normé $(\mathcal{H}, \|\cdot\|)$ est complet si et seulement si toute série normalement convergente est convergente.*

Ce résultat classique, voir [?, proposition 5 du chapitre 6, page 124], est utile pour montrer que les espaces L^p sont complets.

Exemple 171 (Espace de suite). *L'espace $\ell^2(\mathbb{Z})$ est un espace de Hilbert. Soit (a_n) une suite de Cauchy dans $\ell^2(\mathbb{Z})$. Si nous notons*

$$a_n = (a_{n,1}, a_{n,2}, \dots),$$

alors, pour tout $\varepsilon > 0$, il existe N tel que, pour tout $n, m \geq N$,

$$\sum_{k=1}^{\infty} |a_{m,k} - a_{n,k}| \leq \varepsilon^2, \quad (\text{A.1})$$

pour tout $n, m \geq N$. Fixons tout d'abord k . La relation précédente montre que la suite $(a_{n,k})$ est une suite de Cauchy dans \mathbb{C} . Cette suite converge donc vers a_k . Nous notons $a = (a_k)$. Nous allons montrer que $a \in \ell^2(\mathbb{Z})$ et que $\lim_{n \rightarrow \infty} \|a_n - a\| = 0$. Comme l'espace $\ell^2(\mathbb{Z})$ est stable par différence, nous allons montrer que pour tout n , $a_n - a \in \ell^2(\mathbb{Z})$. Comme $a = a_n - (a_n - a)$ et $a_n \in \ell^2(\mathbb{Z})$, cette propriété implique donc que $a \in \ell^2(\mathbb{Z})$.

En utilisant (A.1), nous avons pour tout $p \in \mathbb{N}$, et tout $m, n \geq N$,

$$\sum_{k=1}^p |a_{m,k} - a_{n,k}|^2 \leq \sum_{k=1}^{\infty} |a_{m,k} - a_{n,k}|^2 \leq \varepsilon^2.$$

Par conséquent, pour tout $p \in \mathbb{N}$ et tout $n \geq N$, $\lim_{m \rightarrow \infty} \sum_{k=1}^p |a_{m,k} - a_{n,k}|^2 = \sum_{k=1}^p |a_k - a_{n,k}|^2 \leq \varepsilon^2$. En prenant, la limite en p , nous obtenons donc, pour tout $n \geq N$,

$$\|a - a_n\|^2 = \sum_{k=1}^{\infty} |a_k - a_{n,k}|^2 \leq \varepsilon^2,$$

ce qui montre que $(a - a_n) \in \ell^2(\mathbb{Z})$. Comme ε est arbitraire, nous avons aussi $\lim_{n \rightarrow \infty} \|a - a_n\| = 0$.

Proposition 172 (Espaces L^2). *Pour tout espace mesurable $(\Omega, \mathcal{F}, \mu)$, L'espace $L^2(\Omega, \mathcal{F}, \mu)$ (voir l'exemple 161) des fonctions de carré intégrable pour la mesure μ est un espace de Hilbert.*

Un résultat plus général sur les espaces L^p est fourni par [?, proposition 6 du chapitre 6, page 126].

Définition 173 (Sous-espace fermé). *Soit \mathcal{E} un sous-espace d'un espace de Hilbert \mathcal{H} . On dit que \mathcal{E} est fermé, si toute suite (x_n) de \mathcal{E} , qui converge, converge dans \mathcal{E} .*

Exemple 174 (Sous-espace non-fermé, espace non-complet). Soit $\mathcal{C}([-\pi, \pi])$ l'espace des fonctions continues sur $[-\pi, \pi]$. Cet espace est un sous-espace de l'espace de Hilbert $L^2([-\pi, \pi])$. Considérons la suite de fonctions :

$$f_n(x) = \sum_{k=1}^n \frac{1}{k} \cos(kx)$$

Les fonctions $f_n(x)$, qui sont indéfiniment continûment différentiables, appartiennent à $\mathcal{C}(-\pi, \pi)$. Montrons que cette suite est une suite de Cauchy. En effet, pour $m > n$, on a :

$$\|f_n - f_m\|^2 = \pi \sum_{k=n+1}^m \frac{1}{k^2} \rightarrow 0 \quad \text{quand } (n, m) \rightarrow \infty$$

D'autre part on montre aisément que la limite de cette suite $f_\infty(x) = \sum_{k=1}^\infty k^{-1} \cos(kx) = \log|\sin(x/2)|$ n'est pas continue et n'appartient donc pas à $\mathcal{C}([-\pi, \pi])$. Le sous-espace $\mathcal{C}([-\pi, \pi])$ est donc un sous-espace vectoriel de $L^2([-\pi, \pi])$ mais n'est pas fermé. C'est donc aussi un espace pré-hilbertien qui n'est pas complet.

Définition 175 (Sous espace engendré par un sous-ensemble). Soit \mathcal{X} un sous-ensemble de \mathcal{H} . Nous notons $\text{Vect}(\mathcal{X})$ le sous-espace vectoriel des combinaisons linéaires finies d'éléments de \mathcal{X} et $\overline{\text{Vect}(\mathcal{X})}$ l'adhérence de $\text{Vect}(\mathcal{X})$ dans \mathcal{H} , c'est-à-dire le plus petit sous-ensemble fermé de \mathcal{H} contenant $\text{Vect}(\mathcal{X})$, ou encore l'espace obtenu par l'ensemble $\text{Vect}(\mathcal{X})$ complété de toute les limites de suite d'éléments de $\text{Vect}(\mathcal{X})$.

Définition 176 (Orthogonalité). Deux vecteurs $x, y \in \mathcal{H}$ sont dit orthogonaux, si $\langle x, y \rangle = 0$, ce que nous notons $x \perp y$. Si \mathcal{S} est un sous-ensemble de \mathcal{H} , la notation $x \perp \mathcal{S}$, signifie que $x \perp s$ pour tout $s \in \mathcal{S}$. Nous notons $\mathcal{S} \perp \mathcal{T}$ si tout élément de \mathcal{S} est orthogonal à tout élément de \mathcal{T} .

Supposons qu'il existe deux sous-espaces \mathcal{A} et \mathcal{B} tels que $\mathcal{H} = \mathcal{A} \oplus \mathcal{B}$, dans le sens où, pour tout vecteur $h \in \mathcal{H}$, il existe $a \in \mathcal{A}$ et $b \in \mathcal{B}$, tel que $h = a + b$. Si en plus $\mathcal{A} \perp \mathcal{B}$ nous dirons que \mathcal{H} est la somme directe de \mathcal{A} et \mathcal{B} , ce que nous notons $\mathcal{H} = \mathcal{A} \overset{\perp}{\oplus} \mathcal{B}$.

Définition 177 (Complément orthogonal). Soit \mathcal{E} un sous-ensemble d'un espace de Hilbert \mathcal{H} . On appelle ensemble orthogonal de \mathcal{E} , l'ensemble défini par :

$$\mathcal{E}^\perp = \{x \in \mathcal{H} : \forall y \in \mathcal{E} \quad \langle x, y \rangle = 0\}$$

A.2 Famille orthogonale et orthonormal

Définition 178 (Famille orthogonale, orthonormale). Soit E un sous ensemble de \mathcal{H} . On dit que E est une famille orthogonale si et seulement si pour tout $(x, y) \in E \times E$, $x \neq y$, $\langle x, y \rangle = 0$. Si de plus $\|x\| = 1$ pour tout $x \in E$, on dira que E est une famille orthonormale.

Une famille orthogonale a la propriété remarquable que dans le développement de la norme quadratique des combinaisons linéaires finies, les termes de doubles produits sont nuls. Soit E une famille orthogonale, $(x_1, \dots, x_n) \in E^n$ et $(\alpha_1, \dots, \alpha_n) \in \mathbb{C}^n$. Alors

$$\left\| \sum_{k=1}^n \alpha_k x_k \right\|^2 = \sum_{k=1}^n |\alpha_k|^2 \|x_k\|^2. \quad (\text{A.2})$$

Il s'en suit donc le théorème élémentaire suivant.

Théorème 179. *Toute famille orthogonale d'éléments non-nuls est libre.*

La relation (A.2) est bien connue en géométrie euclidienne. L'avantage du cadre hilbertien est qu'il permet d'obtenir des résultats pour une somme infinie.

Théorème 180. *Soit $(e_i)_{i \geq 1}$ une suite orthonormale d'un espace de Hilbert \mathcal{H} et soit $(\alpha_i)_{i \geq 1}$ une suite de nombre complexes. La série*

$$\sum_{i=1}^{\infty} \alpha_i e_i \quad (\text{A.3})$$

converge dans \mathcal{H} si et seulement si $\sum_i |\alpha_i|^2 < \infty$ auquel cas

$$\left\| \sum_{i=1}^{\infty} \alpha_i e_i \right\|^2 = \sum_{i=1}^{\infty} |\alpha_i|^2. \quad (\text{A.4})$$

Proof. Pour tout $m > k > 0$, de même que pour l'équation (A.2), nous avons

$$\left\| \sum_{i=k}^m \alpha_i e_i \right\|^2 = \sum_{i=k}^m |\alpha_i|^2.$$

Comme $\sum_{i=1}^{\infty} |\alpha_i|^2 < \infty$, la suite $s_m = \sum_{i=1}^m \alpha_i e_i$ est une suite de Cauchy dans \mathcal{H} . Comme \mathcal{H} est complet, cette suite de Cauchy converge. L'identité (A.4) est obtenue par passage à la limite.

Réciproquement, si la série $\sum_{i=1}^{\infty} \alpha_i e_i$ converge, ce passage à la limite reste valide et (A.4) montre que la série de terme $(|\alpha_i|^2)_{i \geq 1}$ est convergente.

□

La question se pose aussi de savoir comment approcher un élément x de \mathcal{H} par une décomposition en série de la forme (A.3). Pour cela le résultat suivant pour une famille finie sera utile.

Proposition 181. *Si x est un vecteur d'un espace de Hilbert \mathcal{H} et si $E = \{e_1, \dots, e_n\}$ est une famille orthonormale finie, alors :*

$$\left\| x - \sum_{k=1}^n \langle x, e_k \rangle e_k \right\|^2 = \|x\|^2 - \sum_{k=1}^n |\langle x, e_k \rangle|^2. \quad (\text{A.5})$$

De plus $\sum_{k=1}^n \langle x, e_k \rangle e_k$ est l'élément de $\text{Vect}(e_1, \dots, e_n)$ le plus proche de x : la quantité (A.5) est aussi égale à

$$\inf \{ \|x - y\|^2 : y \in \text{Vect}(e_1, \dots, e_n) \}.$$

Proof. On remarque que pour tout $j = 1, \dots, n$,

$$\left\langle x - \sum_{k=1}^n \langle x, e_k \rangle e_k, e_j \right\rangle = \langle x, e_j \rangle - \langle x, e_j \rangle = 0.$$

Il s'en suit que la décomposition

$$x = \left(x - \sum_{k=1}^n \langle x, e_k \rangle e_k \right) + \sum_{i=1}^n \langle x, e_k \rangle e_k$$

est une décomposition en la somme de deux termes orthogonaux. L'identité de Pythagore et l'égalité (A.2) avec $x_k = e_k$ et $\alpha_k = \langle x, e_k \rangle$

On montre de même que, pour tout $(\alpha_1, \dots, \alpha_n) \in \mathbb{C}^n$,

$$\left\| x - \sum_{k=1}^n \alpha_k e_k \right\|^2 = \left\| x - \sum_{k=1}^n \langle x, e_k \rangle e_k \right\|^2 + \sum_{k=1}^n |\langle x, e_k \rangle - \alpha_k|^2,$$

et donc que $\sum_{k=1}^n \langle x, e_k \rangle e_k$ est la meilleure approximation de x par une combinaison linéaire des vecteurs e_1, \dots, e_n .

□

Cette propriété d'approximation des familles orthonormales joue un rôle essentiel.

Exemple 182 (Procédé de Gram-Schmidt). *Soit $(y_i)_{i \geq 1}$ une famille d'éléments d'un espace de Hilbert \mathcal{H} . Le procédé de Gram-Schmidt est un procédé par récurrence qui permet alors de construire une famille orthogonale qui vérifie la propriété $\text{Vect}(e_1, \dots, e_n) = \text{Vect}(y_1, \dots, y_n)$ pour tout $n \geq 1$. Nous donnons ici la construction de la suite $(e_i)_{i \geq 1}$. La preuve de ses propriétés est laissée à titre d'exercice.*

En partant d'une suite orthonormale (e_i) et en appliquant la proposition 181 pour toute sous-suite finie, on obtient aussi le résultat suivant.

Corollaire 183 (Inégalité de Bessel). *Soit $(e_i)_{i \geq 1}$ une suite orthonormale d'un espace de Hilbert \mathcal{H} . Alors*

$$\sum_{i=1}^{\infty} |\langle x, e_i \rangle|^2 \leq \|x\|^2.$$

L'inégalité de Bessel implique que pour tout $x \in \mathcal{H}$, $\lim_{n \rightarrow \infty} \langle x, e_n \rangle = 0$ (la suite de vecteurs (e_n) converge faiblement vers 0) mais aussi que la suite $(\langle x, e_i \rangle)_{i \geq 1}$ est un élément de l'espace $\ell^2(\mathbb{Z})$ des suites de carrés sommables. En appliquant le théorème 180, on obtient que le développement en série

$$\sum_{i=1}^{\infty} \langle x, e_i \rangle e_i \tag{A.6}$$

est toujours convergent. On l'appelle le *développement de Fourier généralisé* de x ; les coefficients $\langle x, e_i \rangle$ sont appelés *coefficients de Fourier généralisés* par rapport à la suite orthonormale (e_i) . Il faut prendre garde toutefois au fait que si la série $\sum_{i=1}^{\infty} \langle x, e_i \rangle e_i$ converge, sa limite n'est pas nécessairement égale à x .

Exemple 184. Considérons $\mathcal{H} = L_2(\mathbb{T})$ et soit $e_n(t) = \pi^{-1/2} \sin(nt)$ pour $n = 1, 2, \dots$. La suite (e_n) est orthonormale dans \mathcal{H} , mais pour $x(t) = \cos(t)$, nous avons

$$\begin{aligned} \sum_{n=1}^{\infty} \langle x, e_n \rangle e_n(t) &= \sum_{n=1}^{\infty} \left[\pi^{-1/2} \int_{\mathbb{T}} \cos(t) \sin(nt) dt \right] \pi^{-1/2} \sin(nt) \\ &= \sum_{n=1}^{\infty} 0 \cdot \sin(nt) = 0 \neq \cos t. \end{aligned}$$

En fait, pour obtenir x , il faut une propriété supplémentaire.

Définition 185 (Famille complète, base hilbertienne). *Une famille E d'éléments d'un espace de Hilbert \mathcal{H} est dite complète si $\overline{\text{Vect}(E)} = \mathcal{H}$. Une suite orthonormale complète s'appelle une base hilbertienne.*

La complétude signifie donc n'importe quel élément de \mathcal{H} s'écrit comme la limite d'une suite de combinaisons linéaires finies de la famille considérées. Pour les bases hilbertiennes, cette suite se construit aisément sous la forme de la série (A.6), comme l'indique le résultat suivant.

Théorème 186. *Soit $(e_i)_{i \geq 1}$ une base hilbertienne de l'espace de Hilbert \mathcal{H} . Alors pour tout $x \in \mathcal{H}$,*

$$x = \sum_{i=1}^{\infty} \langle x, e_i \rangle e_i. \quad (\text{A.7})$$

Proof. Nous savons que la série (A.6) converge. D'autre part la suite étant complète, il existe un tableau $(\alpha_{p,n})_{1 \leq i \leq n}$ tel que

$$\lim_{n \rightarrow \infty} \sum_{i=1}^n \alpha_{i,n} e_i = x.$$

Or d'après la proposition 181, on a

$$\left\| x - \sum_{i=1}^n \sum_{i=1}^n \langle x, e_i \rangle e_i \right\| \leq \left\| x - \sum_{i=1}^n \sum_{i=1}^n \alpha_{i,n} e_i \right\|.$$

On a donc aussi convergence du développement de Fourier généralisé de x vers x .

□

Le théorème 186 montre en particulier qu'une famille orthonormale est une base hilbertienne si et seulement si l'identité (A.7) entre un élément et son développement de Fourier est vérifié pour tout élément. Ceci implique aisément le résultat suivant dont la preuve est laissée à titre d'exercice.

Théorème 187. *Soit $(e_i)_{i \geq 1}$ une suite orthonormale d'un espace de Hilbert \mathcal{H} . Les trois propositions suivantes sont équivalentes.*

- a) $(e_i)_{i \geq 1}$ est une base hilbertienne.

b) L'élément nul est l'unique élément qui satisfait

$$\langle x, e_i \rangle = 0 \quad \text{pour tout } i \geq 1.$$

c) Pour tout $x \in \mathcal{H}$,

$$\|x\|^2 = \sum_{i=1}^{\infty} |\langle x, e_i \rangle|^2. \quad (\text{A.8})$$

Exemple 188 (Base de Fourier). *Le système de fonctions*

$$e_n(x) = (2\pi)^{-1/2} e^{inx}, n \in \mathbb{Z}$$

est une suite orthonormale complète de $L_2(\mathbb{T})$. La preuve de l'orthogonalité est élémentaire, mais la preuve de la complétude est plus délicate. Nous l'établirons dans le paragraphe A.3.

Définition 189 (Espace de Hilbert séparable). *On dit qu'un espace de Hilbert est séparable s'il contient un sous-ensemble dénombrable dense.*

L'intérêt d'un espace de Hilbert séparable est qu'il existe une base hilbertienne. C'est d'ailleurs une condition suffisante.

Théorème 190. *Un espace de Hilbert \mathcal{H} est séparable si et seulement si il existe une base hilbertienne.*

Proof. Soit (e_i) une base hilbertienne de \mathcal{H} . L'ensemble $S = \bigcup_{n=1}^{\infty} S_n$ avec, pour $n \in \mathbb{N}$,

$$S_n \stackrel{\text{def}}{=} \left\{ \sum_{k=1}^n (\alpha_k + i\beta_k) e_k, (\alpha_k, \beta_k) \in \mathbb{Q} \times \mathbb{Q}, k = 1, \dots, n \right\}$$

est dénombrable (comme union dénombrable d'ensembles dénombrables). Comme, pour $x \in \mathcal{H}$,

$$\lim_{n \rightarrow \infty} \left\| \sum_{k=1}^n \langle x, e_k \rangle e_k - x \right\| = 0,$$

l'ensemble S est dense dans \mathcal{H} .

Si \mathcal{H} est séparable alors il existe une suite $(y_i)_{i \geq 1}$ dense et donc aussi complète. Le procédé de Gram-Schmidt décrit à l'exemple 182 permet alors de construire une famille orthogonale $(e_i)_{i \geq 1}$ telle que $\text{Vect}(e_1, \dots, e_n) = \text{Vect}(y_1, \dots, y_n)$ pour tout n . En retirant les éléments nuls de cette suite et en renormalisant ses termes non-nuls pour qu'ils soient de norme 1, on obtient alors une base hilbertienne.

□

A.3 Séries de Fourier

Dans cette partie, nous allons établir que

$$\phi_n(x) = (2\pi)^{-1/2} e^{inx}, \quad n \in \mathbb{Z}, \quad (\text{A.9})$$

est une famille complète de $L_2(\mathbb{T}, \mathcal{B}(\mathbb{T}), \mu)$ quelque soit la mesure finie μ sur les boréliens du tore \mathbb{T} . En particulier si μ est proportionnelle à la mesure de Lebesgue, on obtient que cette suite forme une famille orthogonale complète.

Notons $L_1(\mathbb{T})$ l'ensemble des fonctions 2π -périodiques localement intégrables par rapport à la mesure de Lebesgue \mathbb{R} . Pour $f \in L_1(\mathbb{T})$, posons

$$f_n = \sum_{k=-n}^n \langle g, \phi_k \rangle \phi_k, \quad n = 0, 1, 2, \dots$$

Nous avons

$$f_n(x) = \sum_{k=-n}^n \frac{1}{2\pi} \int_{\mathbb{T}} f(t) e^{-ikt} dt = \sum_{k=-n}^n \frac{1}{2\pi} \int_{\mathbb{T}} f(t) e^{ik(x-t)} dt.$$

Nous allons établir que, pour tout $f \in L_1(\mathbb{T})$,

$$\frac{1}{n+1} \sum_{k=0}^n f_k,$$

qui est dans $\text{Vect}(\phi_n, n \in \mathbb{Z})$ est une *bonne approximation* de f , en précisant la notion d'approximation utilisée, suivant les hypothèses supplémentaires sur f . Remarquons que pour tout $x \in \mathbb{R}$,

$$\begin{aligned} \frac{1}{n+1} \sum_{k=0}^n f_k(x) &= \sum_{k=-n}^n \left(1 - \frac{|k|}{n+1}\right) \langle f, \phi_k \rangle \phi_k(x) \\ &= \frac{1}{2\pi} \int_{\mathbb{T}} f(t) \left[\sum_{k=-n}^n \left(1 - \frac{|k|}{n+1}\right) e^{ik(x-t)} \right] dt. \end{aligned}$$

On note la fonction entre crochets $K_n(x-t)$, d'où finalement, pour tout $x \in \mathbb{R}$,

$$\frac{1}{n+1} \sum_{k=0}^n f_k(x) = \frac{1}{2\pi} \int_{\mathbb{T}} f(t) K_n(x-t) dt. \quad (\text{A.10})$$

Un calcul élémentaire donne

$$K_n(u) = \frac{1}{n+1} \frac{\sin^2 \frac{(n+1)u}{2}}{\sin^2 \frac{u}{2}}. \quad (\text{A.11})$$

Définition 191 (Noyau de sommabilité). *Nous dirons qu'une suite de fonctions (κ_n) de fonctions 2π -périodique continue est un noyau de sommabilité si, pour tout $n \in \mathbb{N}$*

$$\int_{\mathbb{T}} \kappa_n(t) dt = 2\pi \quad (\text{A.12})$$

$$\int_{\mathbb{T}} |\kappa_n(t)| dt \leq M, \quad \text{où } M \text{ est une constante} \quad (\text{A.13})$$

$$\lim_{n \rightarrow \infty} \int_{-\delta}^{2\pi-\delta} |\kappa_n(t)| dt = 0, \quad \text{pour tout } \delta \in [0, \pi]. \quad (\text{A.14})$$

Lemme 192. *La fonction $x \rightarrow K_n(x)$ donnée par (A.11) est un noyau de sommabilité, appelé noyau de Fejer.*

Proof. Comme $\int_{\mathbb{T}} e^{ikt} dt = 0$ si $k \neq 0$, nous avons $\int_{\mathbb{T}} K_n(t) dt = 2\pi$. Comme $K_n(t) \geq 0$ pour tout $t \in \mathbb{T}$, nous avons de même $\int_{\mathbb{T}} |K_n(t)| dt = 2\pi$. Finalement, soit $\delta \in]0, \pi[$. Pour tout $t \in]\delta, 2\pi - \delta[$, $\sin t / 2 \geq \sin \delta / 2$, ce qui implique

$$K_n(t) \leq \frac{1}{(n+1) \sin^2 \delta / 2}.$$

Par conséquent

$$\int_{\delta}^{2\pi-\delta} K_n(t) dt \leq \frac{2\pi}{(n+1) \sin^2 \delta / 2} \xrightarrow{n \rightarrow \infty} 0.$$

□

Le résultat suivant montre que la convolution avec un noyau de sommabilité fournit une approximation uniforme d'une fonction continue. On appelle ce procédé d'approximation une *régularisation*.

Lemme 193. *Soient $f : \mathbb{R} \rightarrow \mathbb{C}$ une fonction 2π -périodique continue sur \mathbb{R} et (κ_n) un noyau de sommabilité. Alors*

$$\sup_{x \in \mathbb{R}} \left| \frac{1}{2\pi} \int_{\mathbb{T}} f(t) \kappa_n(x-t) dt - f(x) \right| \rightarrow 0 \quad \text{quand } n \rightarrow \infty.$$

Proof. En utilisant (A.12), on a, pour tout x ,

$$\left| \frac{1}{2\pi} \int_{\mathbb{T}} f(t) \kappa_n(x-t) dt - f(x) \right| = \left| \frac{1}{2\pi} \int_{\mathbb{T}} [f(t) - f(x)] \kappa_n(x-t) dt \right|.$$

Il suffit donc de montrer que

$$\sup_{x \in \mathbb{R}} \left| \int_{-\pi}^{\pi} [f(x-u) - f(x)] \kappa_n(u) du \right| \rightarrow 0. \quad (\text{A.15})$$

Comme f est continue sur \mathbb{R} et périodique, f est uniformément continue. Soit $\varepsilon > 0$. Il existe alors $\delta \in (0, \pi)$ tel que $|f(x) - f(t)| \leq \varepsilon$ pour $|x-t| \leq \delta$. Il s'en suit donc en séparant l'intégrale en 2 parties suivant que $|u| \leq \delta$ ou l'inverse:

$$\left| \int_{-\pi}^{\pi} [f(x-u) - f(x)] \kappa_n(u) du \right| \leq \int_{|\delta| < |u| < \pi} |\kappa_n(u)| du + \varepsilon \int_{-\delta}^{\delta} |\kappa_n(u)| du.$$

On remarque alors que cette majoration ne dépend pas de x , que son premier terme tend vers 0 quand $n \rightarrow \infty$ en vertu de (A.14) et que son deuxième terme est borné par $M\varepsilon$ en utilisant (A.13). Comme ε peut être pris arbitrairement petit, on en conclut (A.15).

□

Corollaire 194. *Soit μ une mesure finie sur les boréliens du tore. La suite $(\phi_n)_{n \in \mathbb{Z}}$ définie par (A.9) génère l'espace de Hilbert $L_2(\mathbb{T}, \mathcal{B}(\mathbb{T}), \mu)$, ce qu'on écrit $\overline{\text{Vect}(\phi_n, n \in \mathbb{Z})} = L_2(\mathbb{T}, \mathcal{B}(\mathbb{T}), \mu)$.*

Proof. D'après [?, proposition 8 du chapitre 6, page 128], pour tout $\varepsilon > 0$ et toute $f \in L_2(\mathbb{T}, \mathcal{B}(\mathbb{T}), \mu)$, il existe une fonction f_ε continue sur $[0, 2\pi]$ telle que

$$\int_{[0,2\pi)} |f - f_\varepsilon|^2 d\mu \leq \varepsilon^2.$$

Soit $\varepsilon' > 0$ arbitrairement petit. On note $g_{\varepsilon'}$ la fonction continue et 2π -périodique qui coïncide avec f_ε sur $[\varepsilon', 2\pi - \varepsilon']$ et qui est linéaire sur $[-\varepsilon', \varepsilon']$. Alors

$$\int_{[0,2\pi)} |f_\varepsilon - g_{\varepsilon'}|^2 d\mu \leq 2\mu([-\varepsilon', \varepsilon']) \sup_{t \in [0,2\pi]} |f_\varepsilon(t)|^2.$$

On remarque que ce majorant tend vers 0 quand $\varepsilon' \rightarrow 0$, en particulier, il existe $\varepsilon' > 0$ tel que ce majorant est inférieur à ε^2 et donc tel que

$$\left(\int_{\mathbb{T}} |f - g_{\varepsilon'}|^2 d\mu \right)^{1/2} \leq 2\varepsilon.$$

Autrement dit les fonctions 2π -périodiques continues sur \mathbb{R} approchent n'importe quelle fonction de $L_2(\mathbb{T}, \mathcal{B}(\mathbb{T}), \mu)$. D'après (A.10) les lemmes 193 et 192, on a d'autre part que toute fonction continue sur le tore peut être arbitrairement approchée par une fonction de $\text{Vect}(\phi_n, n \in \mathbb{Z})$ en norme sup, et donc aussi en norme $L_2(\mathbb{T}, \mathcal{B}(\mathbb{T}), \mu)$, ce qui conclut la preuve. \square

Dans le cas où μ est la mesure de Lebesgue sur le tore, on obtient le résultat suivant.

Corollaire 195. *La suite $(\phi_n)_{n \in \mathbb{Z}}$ définie par (A.9) forme une famille orthogonale complète de $L_2(\mathbb{T})$.*

Le corollaire 195 montre que pour tout $f \in L_2(\mathbb{T})$,

$$f = \sum_{k=-\infty}^{\infty} \alpha_k \phi_k \quad \text{avec} \quad \alpha_k = (2\pi)^{-1/2} \int_{\mathbb{T}} f(x) e^{-ikx} dx,$$

où la somme infinie converge au sens $L_2(\mathbb{T})$. En général, cette condition n'implique pas la convergence ponctuelle. Les coefficients (α_k) sont appelés les coefficients de Fourier de f . L'identité de Parseval s'écrit dans ce cas

$$\int_{\mathbb{T}} |f(x)|^2 dx = \sum_{k=-\infty}^{\infty} |\alpha_k|^2.$$

A.4 Projection et principe d'orthogonalité

Le théorème suivant, appelé *théorème de projection*, joue un rôle central en analyse Hilbertienne.

Théorème 196. *Soit \mathcal{E} est un sous-ensemble convexe fermé d'un espace de Hilbert \mathcal{H} et soit x un élément quelconque de \mathcal{H} , alors :*

a) il existe un unique élément noté $\text{proj}(x|\mathcal{E}) \in \mathcal{E}$ tel que :

$$\|x - \text{proj}(x|\mathcal{E})\| = \inf_{w \in \mathcal{E}} \|x - w\|$$

b) Si de plus \mathcal{E} est un espace vectoriel, $\text{proj}(x|\mathcal{E})$ est l'unique élément $\hat{x} \in \mathcal{E}$ tel que $x - \hat{x} \in \mathcal{E}^\perp$.

On appelle $\text{proj}(x|\mathcal{E})$ la projection orthogonale de x sur \mathcal{E} .

Proof. a) Soit $x \in \mathcal{H}$. On note $h = \inf_{w \in \mathcal{E}} \|x - w\| \geq 0$. Alors il existe une suite w_1, w_2, \dots , de vecteurs de \mathcal{E} tels que :

$$\lim_{m \rightarrow +\infty} \|x - w_m\|^2 = h^2 \geq 0 \quad (\text{A.16})$$

L'identité du parallélogramme, $\|a - b\|^2 + \|a + b\|^2 = 2\|a\|^2 + 2\|b\|^2$ avec $a = w_m - x$ et $b = w_n - x$, montre que :

$$\|w_m - w_n\|^2 + \|w_m + w_n - 2x\|^2 = 2\|w_m - x\|^2 + 2\|w_n - x\|^2$$

Comme $(w_m + w_n)/2 \in \mathcal{E}$, nous avons $\|w_m + w_n - 2x\|^2 = 4\|(w_m + w_n)/2 - x\|^2 \geq 4h^2$. D'après A.16, pour tout $\varepsilon > 0$, il existe N tel que et $\forall m, n > N$:

$$\|w_m - w_n\|^2 \leq 2(h^2 + \varepsilon) + 2(h^2 + \varepsilon) - 4h^2 = 4\varepsilon.$$

qui montre que $\{w_n, n \in \mathbb{N}\}$ est une suite de Cauchy et donc que la suite $\{w_n, n \in \mathbb{N}\}$ tend vers une limite dans \mathcal{E} , puisque l'espace \mathcal{E} est fermé. On note y cette limite. On en déduit, par continuité de la norme, que $\|y - x\| = h$. Montrons que cet élément est unique. Supposons qu'il existe un autre élément $z \in \mathcal{E}$ tel que $\|x - z\|^2 = \|x - y\|^2 = h^2$. Alors l'identité du parallélogramme donne :

$$\begin{aligned} 0 \leq \|y - z\|^2 &= -4\|(y + z)/2 - x\|^2 + 2\|x - y\|^2 + 2\|x - z\|^2 \\ &\leq -4h^2 + 2h^2 + 2h^2 = 0 \end{aligned}$$

où nous avons utilisé que $(y + z)/2 \in \mathcal{E}$ par hypothèse de convexité et donc $\|(y + z)/2 - x\|^2 \geq h^2$. Il s'en suit que $y = z$, d'où l'unicité.

b) Soit \hat{x} la projection orthogonale de x sur \mathcal{E} . Alors, si il existe $u \in \mathcal{E}$ tel que $x - u \perp \mathcal{E}$, on peut écrire :

$$\begin{aligned} \|x - \hat{x}\|^2 &= \langle x - u + u - \hat{x}, x - u + u - \hat{x} \rangle \\ &= \|x - u\|^2 + \|u - \hat{x}\|^2 + 2\langle u - \hat{x}, x - u \rangle \\ &= \|x - u\|^2 + \|u - \hat{x}\|^2 + 0 \geq \|x - u\|^2 \end{aligned}$$

et donc $u = \hat{x}$. Réciproquement supposons que $u \in \mathcal{E}$ et $x - u \notin \mathcal{E}$. Alors choisissons $y \in \mathcal{E}$ tel que $\|y\| = 1$ et tel que $c = \langle x - u, y \rangle \neq 0$ et notons $\tilde{x} = u + cy \in \mathcal{E}$. On a :

$$\begin{aligned} \|x - \tilde{x}\|^2 &= \langle x - u + u - \tilde{x}, x - u + u - \tilde{x} \rangle \\ &= \|x - u\|^2 + \|u - \tilde{x}\|^2 + 2\langle u - \tilde{x}, x - u \rangle \\ &= \|x - u\|^2 + c^2 - 2c\langle y, x - u \rangle = \|x - u\|^2 - c^2 < \|x - u\|^2 \end{aligned}$$

Par conséquent $\tilde{x} \in \mathcal{E}$ est strictement plus proche de x que ne l'est u . \square

Le point (b)) est particulièrement important pour calculer la projection puisqu'il permet de remplacer un problème de minimisation par la résolution d'une équation linéaire.

Exemple 197 (Projection sur un vecteur). *Soit \mathcal{H} un espace de Hilbert, $\mathcal{C} = \text{Vect}(v)$ le sous-espace engendré par un vecteur $v \in \mathcal{H}$ et x un vecteur quelconque de \mathcal{H} . On a alors $\text{proj}(x|\mathcal{C}) = \alpha v$ avec $\alpha = \langle x, v \rangle / \|v\|^2$. Si on note $\varepsilon = x - \text{proj}(x|\mathcal{C})$, on a :*

$$\|\varepsilon\|^2 = \|x\|^2 (1 - \|\rho\|^2) \quad \text{où} \quad \rho = \frac{\langle x, v \rangle}{\|x\| \|v\|} \quad \text{avec} \quad |\rho| \leq 1$$

La projection définie par la proposition 198 satisfait les propriétés intéressantes suivantes.

Proposition 198. *Soit \mathcal{H} un espace de Hilbert et $\text{proj}(\cdot|\mathcal{E})$ la projection orthogonale sur le sous-espace fermé \mathcal{E} . On a :*

1. *l'application $x \in \mathcal{H} \mapsto \text{proj}(x|\mathcal{E}) \in \mathcal{E}$ est linéaire :*

$$\forall (\alpha, \beta) \in \mathbb{C} \times \mathbb{C}, \quad \text{proj}(\alpha x + \beta y|\mathcal{E}) = \alpha \text{proj}(x|\mathcal{E}) + \beta \text{proj}(y|\mathcal{E}).$$

2. $\|x\|^2 = \|\text{proj}(x|\mathcal{E})\|^2 + \|x - \text{proj}(x|\mathcal{E})\|^2$ (Pythagore),

3. *La fonction $\text{proj}(\cdot|\mathcal{E}) : \mathcal{H} \rightarrow \mathcal{H}$ est continue,*

4. *$x \in \mathcal{E}$ si et seulement si $\text{proj}(x|\mathcal{E}) = x$,*

5. *$x \in \mathcal{E}^\perp$ si et seulement si $\text{proj}(x|\mathcal{E}) = 0$,*

6. *Soient \mathcal{E}_1 et \mathcal{E}_2 deux sous espaces vectoriels fermés de \mathcal{H} , tels que $\mathcal{E}_1 \subset \mathcal{E}_2$.*

Alors :

$$\forall x \in \mathcal{H}, \quad \text{proj}(\text{proj}(x|\mathcal{E}_2)|\mathcal{E}_1) = \text{proj}(x|\mathcal{E}_1).$$

7. *Soient \mathcal{E}_1 et \mathcal{E}_2 deux sous-espaces vectoriels fermés de \mathcal{H} , tels que $\mathcal{E}_1 \perp \mathcal{E}_2$.*

Alors :

$$\forall x \in \mathcal{H}, \quad \text{proj}\left(x|\mathcal{E}_1 \overset{\perp}{\oplus} \mathcal{E}_2\right) = \text{proj}(x|\mathcal{E}_1) + \text{proj}(x|\mathcal{E}_2).$$

Théorème 199. *Si \mathcal{E} est un sous-ensemble d'un espace de Hilbert \mathcal{H} , alors \mathcal{E}^\perp est un sous-espace fermé.*

Proof. Soit $(x_n)_{n \geq 0}$ une suite convergente d'éléments de \mathcal{E}^\perp . Notons x la limite de cette suite. Par continuité du produit scalaire nous avons, pour tout $y \in \mathcal{E}$,

$$\langle x, y \rangle = \lim_{n \rightarrow \infty} \langle x_n, y \rangle = 0$$

et donc $x \in \mathcal{E}^\perp$. \square

Théorème 200. Soit $(\mathcal{M}_n)_{n \in \mathbb{Z}}$ une suite croissante de sous-espaces vectoriels (s.e.v.) fermés d'un espace de Hilbert \mathcal{H} .

a) Soit $\mathcal{M}_{-\infty} = \bigcap_n \mathcal{M}_n$. Alors, pour tout $h \in \mathcal{H}$, nous avons

$$\text{proj}(h|\mathcal{M}_{-\infty}) = \lim_{n \rightarrow -\infty} \text{proj}(h|\mathcal{M}_n)$$

b) Soit $\mathcal{M}_{\infty} = \overline{\bigcup_{n \in \mathbb{Z}} \mathcal{M}_n}$. Alors, pour tout $h \in \mathcal{H}$,

$$\text{proj}(h|\mathcal{M}_{\infty}) = \lim_{n \rightarrow \infty} \text{proj}(h|\mathcal{M}_n).$$

Proof. Remarquons tout d'abord (b)) se déduit aisément en appliquant (a)) et en remarquant que

$$\mathcal{M}_{\infty}^{\perp} = \bigcap_n \mathcal{M}_n^{\perp},$$

et donc, comme \mathcal{M}_{∞} et les \mathcal{M}_n sont fermés, on a d'après la propriété 7 de la proposition 198: $\text{proj}(h|\mathcal{M}_{\infty}) = h - \text{proj}(h|\mathcal{M}_{\infty}^{\perp})$ et de même pour \mathcal{M}_n . De plus comme les \mathcal{M}_n^{\perp} sont fermés d'après le théorème 199, on peut bien appliquer (a)).

Il reste donc à montrer (a)). Comme \mathcal{M}_n est un s.e.v. fermé de \mathcal{H} , $\mathcal{M}_{-\infty}$ est un s.e.v. fermé de \mathcal{H} . Le théorème de projection 196 prouve que $\text{proj}(h|\mathcal{M}_{-\infty})$ existe. Pour $m < n$, définissons $\mathcal{M}_n \ominus \mathcal{M}_m$ le complément orthogonal de \mathcal{M}_m dans \mathcal{M}_n , c'est à dire $\mathcal{M}_m^{\perp} \cap \mathcal{M}_n$. Cet ensemble est un s.e.v fermé de \mathcal{H} d'après le théorème 199. Notons que d'après la propriété 7 de la proposition 198,

$$\text{proj}(h|\mathcal{M}_n \ominus \mathcal{M}_m) = \text{proj}(h|\mathcal{M}_n) - \text{proj}(h|\mathcal{M}_m).$$

Il s'en suit que, pour tout $m \geq 1$,

$$\sum_{n=-m+1}^0 \|\text{proj}(h|\mathcal{M}_n \ominus \mathcal{M}_{n-1})\|^2 = \|\text{proj}(h|\mathcal{M}_0 \ominus \mathcal{M}_{-m})\|^2 \leq \|h\|^2 < \infty.$$

On obtient que la série de termes positifs $(\|\text{proj}(h|\mathcal{M}_n \ominus \mathcal{M}_{n-1})\|^2)_{n \leq 0}$ est convergente et comme pour tout $m \leq p \leq 0$,

$$\|\text{proj}(h|\mathcal{M}_p) - \text{proj}(h|\mathcal{M}_m)\|^2 = \sum_{n=-m+1}^p \|\text{proj}(h|\mathcal{M}_n \ominus \mathcal{M}_{n-1})\|^2,$$

on voit que la suite $\{\text{proj}(h|\mathcal{M}_n), n = 0, -1, -2, \dots\}$ est une suite de Cauchy. Comme \mathcal{H} est complet, $\text{proj}(h|\mathcal{M}_n)$ converge dans \mathcal{H} . Notons z sa limite. Il reste à prouver que $z = \text{proj}(h|\mathcal{M}_{-\infty})$. En appliquant le théorème de projection 196, nous devons donc démontrer que $z \in \mathcal{M}_{-\infty}$ et $h - z \perp \mathcal{M}_{-\infty}$. Comme $\text{proj}(h|\mathcal{M}_n) \in \mathcal{M}_p$ pour tout $n \leq p$, nous avons donc $z \in \mathcal{M}_p$ pour tout p et donc $z \in \mathcal{M}_{-\infty}$. Prenons maintenant $p \in \mathcal{M}_{-\infty}$. Alors $p \in \mathcal{M}_n$ pour tout $n \in \mathbb{Z}$, et donc, pour tout $n \in \mathbb{Z}$, $\langle h - \text{proj}(h|\mathcal{M}_n), p \rangle = 0$ et $\langle h - z, p \rangle = 0$ en passant à la limite, ce qui conclut la preuve. \square

Corollaire 201. Soit $\{e_k, k \in \mathbb{N}\}$ une famille orthonormale d'un espace de Hilbert \mathcal{H} . On note $\mathcal{E}_{\infty} = \overline{\text{Vect}(e_k, k \in \mathbb{N})}$. Alors

$$\text{proj}(h|\mathcal{E}_{\infty}) = \sum_{k=0}^{\infty} \langle h, e_k \rangle e_k.$$

A.5 Isométries et isomorphismes d'espaces de Hilbert

Définition 202 (Isométrie). Soient \mathcal{H} et \mathcal{I} deux espaces de Hilbert complexes et \mathcal{G} un sous-espace vectoriel de \mathcal{H} . Une isométrie S de \mathcal{G} dans \mathcal{I} est une application linéaire $S : \mathcal{G} \rightarrow \mathcal{I}$ telle que $\langle Sv, Sw \rangle_{\mathcal{I}} = \langle v, w \rangle_{\mathcal{H}}$ pour tout $(v, w) \in \mathcal{G}$.

On peut montrer qu'une isométrie entre deux espaces de Hilbert est nécessairement linéaire. On remarque aussi qu'une isométrie est toujours une application continue.

Définition 203 (isomorphisme d'espaces de Hilbert). Un espace de Hilbert \mathcal{H} est isomorphe à un espace de Hilbert \mathcal{I} s'il existe une isométrie bijective T de \mathcal{H} dans \mathcal{I} .

Théorème 204. Soit \mathcal{H} un espace de Hilbert séparable.

- a) Si \mathcal{H} est infini dimensionnel, alors il est isomorphe à $\ell^2(\mathbb{Z})$.
- b) Si \mathcal{H} est de dimension finie, alors il est isomorphe à \mathbb{C}^n .

Proof. Soit (e_i) une suite orthonormale complète de \mathcal{H} . Si \mathcal{H} est infini-dimensionnel, alors (e_i) est une suite infinie. Soit $x \in \mathcal{H}$. Pour $x \in \mathcal{H}$, définissons $Tx = (\alpha_i)$, où $\alpha_i = \langle x, e_i \rangle$. Le théorème 180 montre que T est une isométrie de \mathcal{H} dans $\ell^2(\mathbb{Z})$. \square

Comme tous les espaces de Hilbert infini-dimensionnels sont isomorphes à l'espace des suites $\ell^2(\mathbb{Z})$, deux espaces de Hilbert infini-dimensionnels séparables quelconques sont isomorphes.

Le résultat suivant permet de construire facilement des isométries.

Théorème 205. Soit $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ et $(\mathcal{I}, \langle \cdot, \cdot \rangle_{\mathcal{I}})$ deux espaces de Hilbert complexes. Soit \mathcal{G} un sous-espace vectoriel de \mathcal{H} .

- a) soit $S : \mathcal{G} \rightarrow \mathcal{I}$ une isométrie de \mathcal{G} dans \mathcal{I} . Alors, S se prolonge de façon unique en une isométrie $\bar{S} : \overline{\mathcal{G}} \rightarrow \mathcal{I}$ et $\bar{S}(\overline{\mathcal{G}})$ est l'adhérence de $S(\mathcal{G})$ dans \mathcal{I} .
- b) Soit $(v_t, t \in T)$ et $(w_t, t \in T)$ deux familles de vecteurs de \mathcal{H} et \mathcal{I} indexées par un ensemble d'indices T quelconque. Supposons que pour tout $(s, t) \in T \times T$, $\langle v_t, v_s \rangle_{\mathcal{H}} = \langle w_t, w_s \rangle_{\mathcal{I}}$. Alors, il existe une unique isométrie $S : \overline{\text{Vect}(v_t, t \in T)} \rightarrow \overline{\text{Vect}(w_t, t \in T)}$ telle que pour tout $t \in T$, $Sv_t = w_t$. De plus, on a $S\left(\overline{\text{Vect}(v_t, t \in T)}\right) = \overline{\text{Vect}(w_t, t \in T)}$.

Dans la suite, nous utiliserons la même notation pour S et son prolongement \bar{S} .

Proof. Soit $v \in \mathcal{G}$. Pour toute suite $(v_n) \in \mathcal{G}$ convergeant vers v , la suite (Sv_n) est une suite de Cauchy dans \mathcal{I} (car (v_n) est une suite de Cauchy dans \mathcal{G} et S est isométrique). Il existe donc $w \in \mathcal{I}$ telle que $w = \lim_{n \rightarrow \infty} Sv_n$. Si (v'_n) est une autre suite convergeant vers v , nous avons $\|v'_n - v_n\|_{\mathcal{H}} \rightarrow 0$ et, comme S est isométrique, $\|Sv_n - Sv'_n\|_{\mathcal{I}} \rightarrow 0$, ce qui montre que la limite w ne dépend pas du choix de la suite. Posons $\bar{S}v = w$. Les propriétés de linéarité et de conservation du produit scalaire sont conservées par passage à la limite et $\bar{S} : \overline{\mathcal{G}} \rightarrow \mathcal{I}$ est une isométrie prolongeant S .

Par construction $\bar{S}(\bar{\mathcal{G}})$ est inclus dans l'adhérence de $S(\mathcal{G})$. Inversement, soit $w \in \overline{S(\mathcal{G})}$. Il existe une suite $(v_n) \in \mathcal{G}$ telle que $w = \lim_{n \rightarrow \infty} Sv_n$. La suite (Sv_n) est de Cauchy et comme S est une isométrie, la suite (v_n) est aussi de Cauchy dans \mathcal{G} . Soit $v \in \bar{\mathcal{G}}$ sa limite. Nous avons $\bar{S}v = \lim_{n \rightarrow \infty} Sv_n$ et donc $\bar{S}v = w$, ce qui montre $\overline{S(\mathcal{G})} \subseteq \bar{S}(\bar{\mathcal{G}})$. Ceci établit le point (a)) de la proposition.

Pour toute partie finie J de T et pour tous coefficients complexes $(a_t)_{t \in J}$ et $(b_t)_{t \in J}$, nous avons

$$\sum_{t \in J} a_t v_t = \sum_{t \in J} b_t v_t \Rightarrow \sum_{t \in J} a_t w_t = \sum_{t \in J} b_t w_t$$

puisque en posant $c_t = a_t - b_t$,

$$\left\| \sum_{t \in J} c_t v_t \right\|_{\mathcal{H}}^2 = \sum_{t \in J} \sum_{t' \in J} c_t \overline{c_{t'}} \langle v_t, v'_{t'} \rangle_{\mathcal{H}} = \sum_{t \in J} \sum_{t' \in J} c_t \overline{c_{t'}} \langle w_t, w'_{t'} \rangle_{\mathcal{J}} = \left\| \sum_{t \in J} c_t w_t \right\|_{\mathcal{J}}^2,$$

par linéarité et conservation du produit scalaire. Ceci permet de définir $Sf = \sum_{t \in I} a_t w_t$ pour tout f tel que $f = \sum_{t \in I} a_t v_t$ avec I partie finie de T . Nous avons donc défini S sur $\mathcal{G} = \text{Vect}(v_t, t \in T)$ et c'est une isométrie. Cette isométrie, en vertu de (a)) se prolonge de façon unique en une isométrie $\bar{S} : \bar{\mathcal{G}} \rightarrow \mathcal{J}$ telle que $\bar{S}(\bar{\mathcal{G}}) = \overline{S(\mathcal{G})}$. Par construction, $\bar{\mathcal{G}} = \text{Vect}(v_t, t \in T)$ et $S(\mathcal{G}) = \text{Vect}(w_t, t \in T)$.

□

Appendix B

Une autre approche de la prédition

Linear Prediction

J. Benesty, J. Chen, Y. Huang

Linear prediction plays a fundamental role in all aspects of speech. Its use seems natural and obvious in this context since for a speech signal the value of its current sample can be well modeled as a linear combination of its past values. In this chapter, we attempt to present the most important ideas on linear prediction. We derive the principal results, widely recognized by speech experts, in a very intuitive way without sacrificing mathematical rigor.

7.1 Fundamentals	121
7.2 Forward Linear Prediction	122

7.3 Backward Linear Prediction	123
7.4 Levinson–Durbin Algorithm	124
7.5 Lattice Predictor	126
7.6 Spectral Representation	127
7.7 Linear Interpolation	128
7.8 Line Spectrum Pair Representation	129
7.9 Multichannel Linear Prediction	130
7.10 Conclusions	133
References	133

7.1 Fundamentals

Linear prediction (**LP**) is a fundamental tool in many diverse areas such as adaptive filtering, system identification, economics, geophysics, spectral estimation, and speech. Recently, a nice history of **LP** in the context of speech coding was written by Atal [7.1]. Readers are invited to consult this reference for more information about this topic and how it has evolved.

Linear prediction is widely used in speech applications (recognition, compression, modeling, etc.) [7.2, 3]. This is due to the fact that the speech production process is well modeled with **LP**. Indeed, it is well recognized that a speech signal can be written in the following form [7.4, 5],

$$x(k) = \sum_{l=1}^L a_l x(k-l) + Gu(k), \quad (7.1)$$

where k is the time index, L represents the number of coefficients in the model (the order of the predictor), a_l , $l = 1, \dots, L$, are defined as the linear prediction coefficients, G is the gain of the system, and $u(k)$ is the excitation signal, which can be either a quasiperiodic train of impulses or a random noise source (also a combination of both signals for voiced fricatives such as ‘v’, ‘z’, and ‘zh’). The periodic source produces voiced sounds such as vowels and nasals, and the noise

source produces unvoiced or fricated sounds such as the fricatives. The parameters, a_l , determine the spectral characteristics of the particular sound for each of the two types of excitation and are widely used directly in many speech coding schemes and automatic speech recognition systems [7.4].

Equation (7.1) can be rewritten in the frequency domain, by using the z -transform. If $H(z)$ is the transfer function of the system, we have:

$$\begin{aligned} H(z) &= \frac{G}{1 - \sum_{l=1}^L a_l z^{-l}} \\ &= \frac{G}{A(z)}, \end{aligned} \quad (7.2)$$

which is an all-pole transfer function. This filter [$H(z)$] is a good model of the human vocal tract [7.2]. Our main concern is to determine the predictor coefficients, a_l , $l = 1, 2, \dots, L$, and to study the properties of the filter $A(z)$.

The applications of **LP** are numerous. Before addressing the estimation of **LP** coefficients, we give some examples to show the importance of **LP**. In many aspects of speech processing (noise reduction, speech separation, speech dereverberation, speech coding, etc.), it is of great interest to compare the closeness of the spectral envelope of two speech signals (the desired and

the processed ones) [7.6, 7]. One way of doing this is through comparing their LP coefficients. Consider the two speech signals $x(k)$ (desired) and $\hat{x}(k)$ (processed). Without entering too much into the details, one possible measure to evaluate the closeness of these two signals is the Itakura distance:

$$\text{ID}_{x\hat{x}} = \ln \frac{E_x}{E_{\hat{x}}}, \quad (7.3)$$

where E_x and $E_{\hat{x}}$ are the prediction-error powers of the signals $x(k)$ and $\hat{x}(k)$, respectively (see the following sections for more details). Note that the Itakura distance is not symmetric, i. e.,

$$\text{ID}_{x\hat{x}} \neq \text{ID}_{\hat{x}x}, \quad (7.4)$$

therefore, it is not a distance metric. However, asymmetry is usually not a problem for applications such as speech quality evaluation.

A more-powerful distance was proposed by Itakura and Saito in their formulation of linear prediction as an approximate maximum-likelihood estimation [7.8]. This distance between the two signals $x(k)$ and $\hat{x}(k)$ is defined as,

$$\text{ISD}_{x\hat{x}} = \frac{E_{\hat{x}}}{E_x} - \ln \frac{E_{\hat{x}}}{E_x} - 1. \quad (7.5)$$

Like the Itakura distance, this measure is not symmetric either; therefore, it is not a true metric.

The Itakura–Saito distance has many interesting properties. It has been shown that this measure is highly correlated with subjective quality judgements [7.6]. For example, a recent report on speech codec evaluation reveals that, if the Itakura–Saito measure between two speech signals is less than 0.5, the difference in their mean opinion score would be less than 1.6 [7.9]. Many other reported experiments also confirmed that when the Itakura–Saito distance between two speech signals is below 0.1, they would be perceived nearly identically by human ears. As a result, the Itakura–Saito distance, which is based on LP, is often used as an objective measure of speech quality. It is probably the most widely used measure of similarity between speech signals.

The two previous examples of the vocal-tract filter and the speech quality measure clearly show the importance of LP in speech applications.

In this chapter, we study the theory of linear prediction and derive the most important LP techniques that are often encountered in many speech applications. We assume here that all signals of interest are real, stationary, and zero mean.

7.2 Forward Linear Prediction

Consider a stationary random signal $x(k)$. The objective of the forward linear prediction is to predict the value of the sample $x(k)$ from its past values, i. e., $x(k-1)$, $x(k-2)$, etc. We define the forward prediction error as [7.10, 11],

$$\begin{aligned} e_{f,L}(k) &= x(k) - \hat{x}(k) \\ &= x(k) - \sum_{l=1}^L a_{L,l} x(k-l) \\ &= x(k) - \mathbf{a}_L^T \mathbf{x}(k-1), \end{aligned} \quad (7.6)$$

where the superscript ‘T’ denotes transposition, $\hat{x}(k)$ is the predicted sample,

$$\mathbf{a}_L = [a_{L,1} \ a_{L,2} \ \dots \ a_{L,L}]^T$$

is the forward predictor of length L , and

$$\mathbf{x}(k-1) = [x(k-1) \ x(k-2) \ \dots \ x(k-L)]^T$$

is a vector containing the L most recent samples starting with and including $x(k-1)$.

We would like to find the optimal Wiener predictor. For that, we seek to minimize the mean-square error (MSE):

$$J_f(\mathbf{a}_L) = E\{e_{f,L}^2(k)\}, \quad (7.7)$$

where $E\{\cdot\}$ denotes mathematical expectation. Taking the gradient of $J_f(\mathbf{a}_L)$ with respect to \mathbf{a}_L and equating to $\mathbf{0}_{L \times 1}$ (a vector of length L containing only zeroes), we easily find the Wiener–Hopf equations:

$$\mathbf{R}_L \mathbf{a}_{o,L} = \mathbf{r}_{f,L}, \quad (7.8)$$

where the subscript ‘o’ in $\mathbf{a}_{o,L}$ stands for optimal,

$$\begin{aligned} \mathbf{R}_L &= E\{\mathbf{x}(k-1)\mathbf{x}^T(k-1)\} \\ &= E\{\mathbf{x}(k)\mathbf{x}^T(k)\} \\ &= \begin{pmatrix} r(0) & r(1) & \cdots & r(L-1) \\ r(1) & r(0) & \cdots & r(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(L-1) & r(L-2) & \cdots & r(0) \end{pmatrix} \end{aligned} \quad (7.9)$$

is the correlation matrix, and

$$\mathbf{r}_{f,L} = E\{\mathbf{x}(k-1)\mathbf{x}(k)\} = [r(1) \ r(2) \ \cdots \ r(L)]^T \quad (7.10)$$

is the correlation vector. The matrix \mathbf{R}_L has a Toeplitz structure (i.e., all the entries along the diagonals are the same); assuming that it is nonsingular, we deduce the optimal forward predictor:

$$\mathbf{a}_{o,L} = \mathbf{R}_L^{-1} \mathbf{r}_{f,L}. \quad (7.11)$$

Expanding $e_{f,L}^2(k)$ in (7.7) and using (7.8) shows that the minimum mean-square error (MMSE),

$$J_{f,\min} = J_f(\mathbf{a}_{o,L}) = r(0) - \mathbf{r}_{f,L}^T \mathbf{a}_{o,L} = E_{f,L}. \quad (7.12)$$

This is also called the forward prediction-error power.

Define the augmented correlation matrix:

$$\mathbf{R}_{L+1} = \begin{pmatrix} r(0) & \mathbf{r}_{f,L}^T \\ \mathbf{r}_{f,L} & \mathbf{R}_L \end{pmatrix}, \quad (7.13)$$

equations (7.8) and (7.12) may be combined in a convenient way:

$$\mathbf{R}_{L+1} \begin{pmatrix} 1 \\ -\mathbf{a}_{o,L} \end{pmatrix} = \begin{pmatrix} E_{f,L} \\ \mathbf{0}_{L \times 1} \end{pmatrix}. \quad (7.14)$$

We refer to (7.14) as the augmented Wiener–Hopf equations of a forward predictor of order L . From (7.13) we derive that,

$$\det(\mathbf{R}_{L+1}) = E_{f,L} \det(\mathbf{R}_L), \quad (7.15)$$

7.3 Backward Linear Prediction

The aim of the backward linear prediction is to predict the value of the sample $x(k-L)$ from its future values, i.e., $x(k), x(k-1), \dots, x(k-L+1)$. We define the backward prediction error as,

$$\begin{aligned} e_{b,L}(k) &= x(k-L) - \hat{x}(k-L) \\ &= x(k-L) - \sum_{l=1}^L b_{L,l} x(k-l+1) \\ &= x(k-L) - \mathbf{b}_L^T \mathbf{x}(k), \end{aligned} \quad (7.21)$$

where $\hat{x}(k-L)$ is the predicted sample,

$$\mathbf{b}_L = [b_{L,1} \ b_{L,2} \ \cdots \ b_{L,L}]^T$$

is the backward predictor of order L , and

$$\mathbf{x}(k) = [x(k) \ x(k-1) \ \cdots \ x(k-L+1)]^T.$$

The minimization of the MSE,

$$J_b(\mathbf{b}_L) = E\{e_{b,L}^2(k)\}, \quad (7.22)$$

where ‘det’ stands for determinant.

Let us now write the forward prediction errors for the optimal predictors of orders L and $L-i$:

$$e_{f,o,L}(k) = x(k) - \sum_{l=1}^L a_{o,L,l} x(k-l), \quad (7.16)$$

$$e_{f,o,L-i}(k) = x(k) - \sum_{l=1}^{L-i} a_{o,L-i,l} x(k-l). \quad (7.17)$$

From the principle of orthogonality [7.11], we know that:

$$E\{e_{f,o,L}(k)\mathbf{x}(k-1)\} = \mathbf{0}_{L \times 1}. \quad (7.18)$$

For $1 \leq i \leq L$, we can verify by using (7.18), that:

$$E\{e_{f,o,L}(k)e_{f,o,L-i}(k-i)\} = 0. \quad (7.19)$$

As a result,

$$\begin{aligned} \lim_{L \rightarrow \infty} E\{e_{f,o,L}(k)e_{f,o,L-i}(k-i)\} \\ = E\{e_{f,o}(k)e_{f,o}(k-i)\} = 0. \end{aligned} \quad (7.20)$$

This indicates that the signal $e_{f,o}(k)$ is a white noise. So the optimal forward predictor has this important property of being able to whiten a stationary random process, provided that the order of the predictor is high enough.

leads to the Wiener–Hopf equations:

$$\mathbf{R}_L \mathbf{b}_{o,L} = \mathbf{r}_{b,L}, \quad (7.23)$$

where

$$\begin{aligned} \mathbf{r}_{b,L} &= E\{\mathbf{x}(k)\mathbf{x}(k-L)\} \\ &= [r(L) \ r(L-1) \ \cdots \ r(1)]^T. \end{aligned} \quad (7.24)$$

Therefore, the optimal backward predictor is:

$$\mathbf{b}_{o,L} = \mathbf{R}_L^{-1} \mathbf{r}_{b,L}. \quad (7.25)$$

The MMSE for backward prediction,

$$\begin{aligned} J_{b,\min} &= J_b(\mathbf{b}_{o,L}) \\ &= r(0) - \mathbf{r}_{b,L}^T \mathbf{b}_{o,L} = E_{b,L}, \end{aligned} \quad (7.26)$$

is also called the backward prediction-error power.

Define the augmented correlation matrix:

$$\mathbf{R}_{L+1} = \begin{pmatrix} \mathbf{R}_L & \mathbf{r}_{b,L} \\ \mathbf{r}_{b,L}^T & r(0) \end{pmatrix}, \quad (7.27)$$

equations (7.23) and (7.26) may be combined in a convenient way:

$$\mathbf{R}_{L+1} \begin{pmatrix} -\mathbf{b}_{o,L} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{L \times 1} \\ E_{b,L} \end{pmatrix}. \quad (7.28)$$

We refer to this expression as the augmented Wiener–Hopf equations of a backward predictor of order L .

One important property of backward prediction is that the error signals of different orders with the optimal predictors are uncorrelated, i.e., $E\{\mathbf{e}_{b,o,i}(k)\mathbf{e}_{b,o,l}(k)\} = 0, i \neq l, i, l = 0, 1, \dots, L-1$. To prove this, let us rewrite the error signal in vector form:

$$\mathbf{e}_{b,o}(k) = \mathbf{L}\mathbf{x}(k), \quad (7.29)$$

where

$$\mathbf{e}_{b,o}(k) = [e_{b,o,0}(k) \ e_{b,o,1}(k) \ \dots \ e_{b,o,L-1}(k)]^T \quad (7.30)$$

and

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -\mathbf{b}_{o,1}^T & 1 & 0 & \dots & 0 \\ -\mathbf{b}_{o,2}^T & & 1 & \dots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ -\mathbf{b}_{o,L-1}^T & & & & 1 \end{pmatrix} \quad (7.31)$$

is a lower triangular matrix with 1s along its main diagonal. The covariance matrix corresponding to the vector signal $\mathbf{e}_{b,o}(k)$ is:

7.4 Levinson–Durbin Algorithm

The Levinson–Durbin algorithm is an efficient way to solve the Wiener–Hopf equations for the forward and backward prediction coefficients. This efficient method can be derived thanks to the Toeplitz structure of the correlation matrix \mathbf{R}_L . This algorithm was first invented by *Levinson* [7.13] and independently reformulated at a later date by *Durbin* [7.14, 15]. *Burg* gave a more-elegant presentation [7.16]. Before describing this algorithm, we first need to show some important relations between the forward and backward predictors.

We define the co-identity matrix as:

$$\mathbf{J}_L = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{pmatrix}.$$

$$E\{\mathbf{e}_{b,o}(k)\mathbf{e}_{b,o}^T(k)\} = \mathbf{L}\mathbf{R}_L\mathbf{L}^T. \quad (7.32)$$

By definition, the previous matrix is symmetric. The matrix product $\mathbf{R}_L\mathbf{L}^T$ is a lower triangular matrix because of (7.28) and the main diagonal contains the backward prediction-error powers $E_{b,l}$ ($0 \leq l \leq L-1$). Since \mathbf{L} is also a lower triangular matrix, the product between the two matrices \mathbf{L} and $\mathbf{R}_L\mathbf{L}^T$ should have the same structure and, since it has to be symmetric, the only possibility is that this resulting matrix is diagonal:

$$E\{\mathbf{e}_{b,o}(k)\mathbf{e}_{b,o}^T(k)\} = \text{diag}[E_{b,0}, E_{b,1}, \dots, E_{b,L-1}], \quad (7.33)$$

and hence the prediction errors are uncorrelated.

Furthermore,

$$\mathbf{L}\mathbf{R}_L\mathbf{L}^T = \text{diag}[E_{b,0}, E_{b,1}, \dots, E_{b,L-1}], \quad (7.34)$$

taking the inverse of the previous equation,

$$\mathbf{L}^{-T}\mathbf{R}_L^{-1}\mathbf{L}^{-1} = \text{diag}[E_{b,0}^{-1}, E_{b,1}^{-1}, \dots, E_{b,L-1}^{-1}], \quad (7.35)$$

we finally get:

$$\mathbf{R}_L^{-1} = \mathbf{L}^T \text{diag}[E_{b,0}^{-1}, E_{b,1}^{-1}, \dots, E_{b,L-1}^{-1}] \mathbf{L}. \quad (7.36)$$

Expression (7.36) defines the Cholesky factorization of the inverse matrix \mathbf{R}_L^{-1} [7.10, 12].

We can easily check that:

$$\mathbf{R}_L \mathbf{J}_L = \mathbf{J}_L \mathbf{R}_L. \quad (7.37)$$

The matrix \mathbf{R}_L is said to be persymmetric. We also have, $\mathbf{r}_{f,L} = \mathbf{J}_L \mathbf{r}_{b,L}$. If we left-multiply both sides of the Wiener–Hopf equations (7.23) by \mathbf{J}_L , we get:

$$\begin{aligned} \mathbf{J}_L \mathbf{R}_L \mathbf{b}_{o,L} &= \mathbf{J}_L \mathbf{r}_{b,L} = \mathbf{r}_{f,L} \\ &= \mathbf{R}_L \mathbf{J}_L \mathbf{b}_{o,L} = \mathbf{R}_L \mathbf{a}_{o,L}, \end{aligned} \quad (7.38)$$

and, assuming that \mathbf{R}_L is nonsingular, we see that:

$$\mathbf{a}_{o,L} = \mathbf{J}_L \mathbf{b}_{o,L}. \quad (7.39)$$

Furthermore,

$$\begin{aligned} E_{b,L} &= r(0) - \mathbf{r}_{b,L}^T \mathbf{b}_{o,L} \\ &= r(0) - \mathbf{r}_{b,L}^T \mathbf{J}_L \mathbf{J}_L \mathbf{b}_{o,L} \\ &= r(0) - \mathbf{r}_{f,L}^T \mathbf{a}_{o,L} \\ &= E_{f,L} = E_L. \end{aligned} \quad (7.40)$$

Therefore, for a stationary process, the forward and backward prediction-error powers are equal and the coefficients of the optimal forward predictor are the same as those of the optimal backward predictor, but in a reverse order.

The Levinson–Durbin algorithm is based on recursions of the orders of the prediction equations. Consider the following expression,

$$\begin{pmatrix} \mathbf{R}_L & \mathbf{r}_{b,L} \\ \mathbf{r}_{b,L}^T & r(0) \end{pmatrix} \begin{pmatrix} 1 \\ -\mathbf{a}_{o,L-1} \\ 0 \end{pmatrix} = \begin{pmatrix} E_{L-1} \\ \mathbf{o}_{(L-1) \times 1} \\ K_L \end{pmatrix}, \quad (7.41)$$

where

$$\begin{aligned} K_L &= r(L) - \mathbf{a}_{o,L-1}^T \mathbf{r}_{b,L-1} \\ &= r(L) - \mathbf{a}_{o,L-1}^T \mathbf{J}_{L-1} \mathbf{r}_{f,L-1}. \end{aligned} \quad (7.42)$$

We define the reflection coefficient as,

$$\kappa_L = \frac{K_L}{E_{L-1}}. \quad (7.43)$$

From backward linear prediction, we have:

$$\begin{pmatrix} r(0) & \mathbf{r}_{f,L}^T \\ \mathbf{r}_{f,L} & \mathbf{R}_L \end{pmatrix} \begin{pmatrix} 0 \\ -\mathbf{b}_{o,L-1} \\ 1 \end{pmatrix} = \begin{pmatrix} K_L \\ E_{L-1} \end{pmatrix}. \quad (7.44)$$

Multiplying both sides of the previous equation by κ_L , we get,

$$\mathbf{R}_{L+1} \begin{pmatrix} 0 \\ -\kappa_L \mathbf{b}_{o,L-1} \\ \kappa_L \end{pmatrix} = \begin{pmatrix} \kappa_L^2 E_{L-1} \\ \mathbf{o}_{(L-1) \times 1} \\ K_L \end{pmatrix}. \quad (7.45)$$

If we now subtract (7.45) from (7.41), we obtain,

$$\mathbf{R}_{L+1} \begin{pmatrix} 1 \\ \kappa_L \mathbf{b}_{o,L-1} - \mathbf{a}_{o,L-1} \\ -\kappa_L \end{pmatrix} = \begin{pmatrix} E_{L-1}(1 - \kappa_L^2) \\ \mathbf{o}_{L \times 1} \\ K_L \end{pmatrix}. \quad (7.46)$$

Assuming that \mathbf{R}_{L+1} is nonsingular and identifying (7.46) with (7.14), we can deduce the recursive equa-

tions:

$$\mathbf{a}_{o,L} = \begin{pmatrix} \mathbf{a}_{o,L-1} \\ 0 \end{pmatrix} - \kappa_L \begin{pmatrix} \mathbf{b}_{o,L-1} \\ -1 \end{pmatrix}, \quad (7.47)$$

$$E_L = E_{L-1}(1 - \kappa_L^2), \quad (7.48)$$

$$a_{o,L,L} = \kappa_L. \quad (7.49)$$

Iterating on the prediction-error power given in (7.48), we find that,

$$E_L = r(0) \prod_{l=1}^L (1 - \kappa_l^2), \quad (7.50)$$

and since $E_L \geq 0$, this implies that,

$$|\kappa_l| \leq 1, \quad \forall l \geq 1. \quad (7.51)$$

Also, from (7.48) we see that we have,

$$0 \leq E_l \leq E_{l-1}, \quad \forall l \geq 1, \quad (7.52)$$

so, as the order of the predictors increases, the prediction-error power decreases.

Table 7.1 summarizes the Levinson–Durbin algorithm, whose arithmetic complexity is proportional to L^2 . This algorithm is much more efficient than standard methods such as the Gauss elimination technique, whose complexity is on the order of L^3 . The saving in number of operations to find the optimal Wiener predictor can be very important, especially when L is large. The other advantage of the Levinson–Durbin algorithm is that it gives the predictors of all orders and the algorithm can be stopped if the prediction-error power is under a threshold, which can be very useful in practice when the choice of the predictor order is not easy to get in advance. A slightly more-efficient approach, called the split Levinson algorithm, can be found in [7.17]. This algorithm requires roughly half the number of multiplications and the same number of additions as the classical Levinson–Durbin algorithm. Even more-efficient algorithms have been proposed (see, for example, [7.18]) but they are numerically unstable, which is not acceptable in most speech applications.

Table 7.1 Levinson–Durbin algorithm

Initialization: $E_0 = r(0)$ For $1 \leq l \leq L$ $\kappa_l = \frac{1}{E_{l-1}} [r(l) - \mathbf{a}_{o,l-1}^T \mathbf{J}_{l-1} \mathbf{r}_{f,l-1}]$ $\mathbf{a}_{o,l} = \begin{pmatrix} \mathbf{a}_{o,l-1} \\ 0 \end{pmatrix} - \kappa_l \mathbf{J}_l \begin{pmatrix} -1 \\ \mathbf{a}_{o,l-1} \end{pmatrix}$ $E_l = E_{l-1}(1 - \kappa_l^2)$

7.5 Lattice Predictor

In this section, we will show that the order-recursive structure of the forward and backward prediction errors has the form of a ladder, which is called a lattice predictor.

Inserting (7.47) into the forward prediction error for the optimal predictor of order L ,

$$e_{f,o,L}(k) = x(k) - \mathbf{a}_{o,L}^T \mathbf{x}(k-1), \quad (7.53)$$

we obtain,

$$\begin{aligned} e_{f,o,L}(k) &= e_{f,o,L-1}(k) \\ &\quad - \kappa_L \left(-\mathbf{b}_{o,L-1}^T \mathbf{1} \right) \mathbf{x}(k-1). \end{aligned} \quad (7.54)$$

The second term (without the reflection coefficient) on the right-hand side of (7.54) is the backward prediction error, at time $k-1$, for the optimal predictor of order $L-1$. Therefore, (7.54) can be rewritten

$$e_{f,o,L}(k) = e_{f,o,L-1}(k) - \kappa_L e_{b,o,L-1}(k-1). \quad (7.55)$$

If we insert (7.47) again into the backward prediction error for the optimal predictor of order L ,

$$\begin{aligned} e_{b,o,L}(k) &= x(k-L) - \mathbf{b}_{o,L}^T \mathbf{x}(k) \\ &= x(k-L) - \mathbf{a}_{o,L}^T \mathbf{J}_L \mathbf{x}(k), \end{aligned} \quad (7.56)$$

we get,

$$e_{b,o,L}(k) = e_{b,o,L-1}(k-1) - \kappa_L e_{f,o,L-1}(k). \quad (7.57)$$

If we put (7.55) and (7.57) into a matrix form, we have,

$$\begin{pmatrix} e_{f,o,L}(k) \\ e_{b,o,L}(k) \end{pmatrix} = \begin{pmatrix} 1 & -\kappa_L \\ -\kappa_L & 1 \end{pmatrix} \begin{pmatrix} e_{f,o,L-1}(k) \\ e_{b,o,L-1}(k-1) \end{pmatrix} = \prod_{l=1}^L \begin{pmatrix} 1 & -\kappa_l \\ -\kappa_l & 1 \end{pmatrix} \begin{pmatrix} x(k) \\ x(k-1) \end{pmatrix}, \quad (7.58)$$

where we have taken for initial conditions (order 0), $e_{f,o,0}(k) = x(k)$ and $e_{b,o,0}(k-1) = x(k-1)$. Figure 7.1

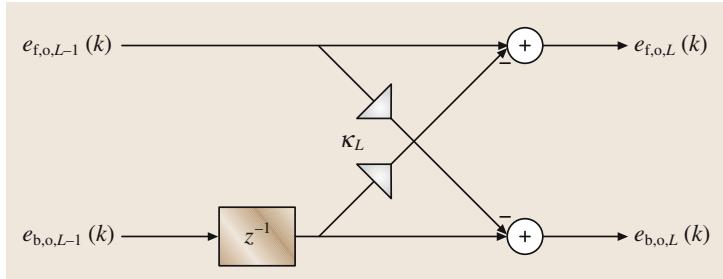


Fig. 7.1 Stage l of a lattice predictor

depicts the l -th stage of a lattice predictor. For the whole lattice predictor, L of these stages are needed and are connected in cascade, one to each other, starting from order 0 to order L .

Now let us compute the variance of $e_{b,o,L}(k)$ from (7.57),

$$\begin{aligned} E\{e_{b,o,L}^2(k)\} &= E_L \\ &= E_{L-1} + \kappa_L^2 E_{L-1} \\ &\quad - 2\kappa_L E\{e_{f,o,L-1}(k)e_{b,o,L-1}(k-1)\} \\ &= E_{L-1}(1 - \kappa_L^2). \end{aligned} \quad (7.59)$$

Developing the previous expression, we obtain,

$$\begin{aligned} \kappa_L &= \frac{E\{e_{f,o,L-1}(k)e_{b,o,L-1}(k-1)\}}{E_{L-1}} \\ &= \frac{E\{e_{f,o,L-1}(k)e_{b,o,L-1}(k-1)\}}{\sqrt{E\{e_{f,o,L-1}^2(k)\} E\{e_{b,o,L-1}^2(k-1)\}}}. \end{aligned} \quad (7.60)$$

We see from (7.60) that the reflection coefficients are also the normalized cross-correlation coefficients between the forward and backward prediction errors, which is why they are also often called partial correlation (PARCOR) coefficients [7.3, 19, 20]. These coefficients are linked to the zeroes of the forward prediction-error FIR filter of order L , whose transfer function is

$$A_{o,L}(z) = 1 - \sum_{l=1}^L a_{o,L,l} z^{-l} = \prod_{l=1}^L (1 - z_{o,l} z^{-1}), \quad (7.61)$$

where $z_{o,l}$ are the roots of $A_{o,L}(z)$. Since $\kappa_L = a_{o,L,L}$, we have,

$$\kappa_L = (-1)^{L+1} \prod_{l=1}^L z_{o,l}. \quad (7.62)$$

The filter $A_{o,L}(z)$ can be shown to be minimum phase, i.e., $|z_{o,l}| \leq 1$, $\forall l$. As a result [because of the relation (7.39)], the filter $B_{o,L}(z)$ corresponding to the backward predictor is maximum phase. We will now show this very important property that the forward predictor is minimum phase. As far as we know, this simple and elegant proof was first shown by M. Mohan Sondhi but was never been published. A similar proof can be found in [7.21] and [7.22].

To avoid cumbersome notation, redefine the coefficients $w_l = -a_{o,L,l}$, with $w_0 = 1$, so that the polynomial

becomes,

$$A_{0,L}(z) = \sum_{l=0}^L w_l z^{-l}. \quad (7.63)$$

Also, define the vector,

$$\mathbf{w} = [w_0 \ w_1 \ \dots \ w_L]^T.$$

We know that,

$$\mathbf{R}_{L+1}\mathbf{w} = \begin{pmatrix} E_L \\ \mathbf{0}_{L \times 1} \end{pmatrix}. \quad (7.64)$$

If λ is a root of the polynomial, it follows that,

$$A_{0,L}(z) = (1 - \lambda z^{-1}) \sum_{l=0}^{L-1} g_l z^{-l}, \text{ with } g_0 = 1. \quad (7.65)$$

(Note that since λ can be complex, the coefficients g_l are, in general, complex.) Thus the vector \mathbf{w} can be written

$$\mathbf{w} = \mathbf{g} - \lambda \tilde{\mathbf{g}}, \quad (7.66)$$

where

$$\mathbf{g} = [1 \ g_1 \ g_2 \ \dots \ g_{L-1} \ 0]^T = [\mathbf{g}'^T \ 0]^T,$$

$$\tilde{\mathbf{g}} = [0 \ 1 \ g_1 \ g_2 \ \dots \ g_{L-1}]^T = [0 \ \mathbf{g}'^T]^T.$$

Substituting (7.66) in (7.64), we obtain,

$$\mathbf{R}_{L+1}\mathbf{g} = \lambda \mathbf{R}_{L+1}\tilde{\mathbf{g}} + \begin{pmatrix} E_L \\ \mathbf{0}_{L \times 1} \end{pmatrix}. \quad (7.67)$$

Now, premultiplying by $\tilde{\mathbf{g}}^H$ (where the superscript H denotes conjugate transpose) gives,

$$\tilde{\mathbf{g}}^H \mathbf{R}_{L+1}\mathbf{g} = \lambda \tilde{\mathbf{g}}^H \mathbf{R}_{L+1}\tilde{\mathbf{g}}. \quad (7.68)$$

Thus,

$$|\tilde{\mathbf{g}}^H \mathbf{R}_{L+1}\mathbf{g}|^2 = |\lambda|^2 (\tilde{\mathbf{g}}^H \mathbf{R}_{L+1}\tilde{\mathbf{g}})^2. \quad (7.69)$$

Using the Schwartz inequality,

$$|\tilde{\mathbf{g}}^H \mathbf{R}_{L+1}\mathbf{g}|^2 \leq (\tilde{\mathbf{g}}^H \mathbf{R}_{L+1}\tilde{\mathbf{g}})(\mathbf{g}^H \mathbf{R}_{L+1}\mathbf{g}). \quad (7.70)$$

However,

$$\begin{aligned} \tilde{\mathbf{g}}^H \mathbf{R}_{L+1}\tilde{\mathbf{g}} &= \begin{pmatrix} 0 & \mathbf{g}'^H \end{pmatrix} \begin{pmatrix} r(0) & \mathbf{r}_{f,L}^T \\ \mathbf{r}_{f,L} & \mathbf{R}_L \end{pmatrix} \begin{pmatrix} 0 \\ \mathbf{g}' \end{pmatrix} \\ &= \mathbf{g}'^H \mathbf{R}_L \mathbf{g}', \end{aligned} \quad (7.71)$$

Similarly,

$$\begin{aligned} \mathbf{g}^H \mathbf{R}_{L+1}\mathbf{g} &= \begin{pmatrix} \mathbf{g}'^H & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R}_L & \mathbf{r}_{b,L} \\ \mathbf{r}_{b,L}^T & r(0) \end{pmatrix} \begin{pmatrix} \mathbf{g}' \\ 0 \end{pmatrix} \\ &= \mathbf{g}'^H \mathbf{R}_L \mathbf{g}'. \end{aligned} \quad (7.72)$$

Therefore, $\tilde{\mathbf{g}}^H \mathbf{R}_{L+1}\tilde{\mathbf{g}} = \mathbf{g}^H \mathbf{R}_{L+1}\mathbf{g}$, and the Schwartz inequality becomes,

$$|\tilde{\mathbf{g}}^H \mathbf{R}_{L+1}\mathbf{g}|^2 \leq (\tilde{\mathbf{g}}^H \mathbf{R}_{L+1}\tilde{\mathbf{g}})^2. \quad (7.73)$$

From (7.69) we see that $|\lambda|^2 \leq 1$. This completes the proof.

This property allows one easily to ensure that the all-pole system in (7.2) is stable (when the correlation matrix is positive definite) by simply imposing the constraint that the PARCOR coefficients are less than 1 in magnitude. As a result, in speech communication, transmitting PARCOR coefficients is more advantageous than directly transmitting linear predication coefficients.

7.6 Spectral Representation

It is important to understand the link between the spectrum of a speech signal and its prediction coefficients. Let us again take the speech model given in Sect. 7.1,

$$x(k) = \sum_{l=1}^L a_l x(k-l) + G u(k), \quad (7.74)$$

where we now assume that $u(k)$ is a white random signal with variance $\sigma_u^2 = 1$. Since $x(k)$ is the output of the filter $H(z)$ (see Sect. 7.1), whose input is $u(k)$, its spectrum is [7.11],

$$S_x(\omega) = |H(e^{i\omega})|^2 S_u(\omega), \quad (7.75)$$

where ω is the angular frequency, $H(e^{i\omega})$ is the frequency response of the filter $H(z)$, and $S_u(\omega)$ is the spectrum of u . We have $S_u(\omega) = 1$ (u is white). Using (7.2), we deduce the spectrum of x ,

$$\begin{aligned} S_x(\omega) &= \frac{G^2}{|A(e^{i\omega})|^2} \\ &= \frac{G^2}{\left|1 - \sum_{l=1}^L a_l e^{-il\omega}\right|^2}. \end{aligned} \quad (7.76)$$

Therefore, the spectrum of a speech signal can be modeled by the frequency response of an all-pole filter,

whose elements are the prediction coefficients [7.23–25].

Consider the prediction error signal,

$$e_L(k) = x(k) - \mathbf{a}_L^T \mathbf{x}(k-1). \quad (7.77)$$

Taking the z -transform of (7.77) and setting $z = e^{i\omega}$, we obtain:

$$S_{x,L}(\omega) = \frac{|E_L(e^{i\omega})|^2}{|A_L(e^{i\omega})|^2}, \quad (7.78)$$

where $|E_L(e^{i\omega})|^2$ is the spectrum of $e_L(k)$. From Sect. 7.2, we know that, for a large order L , linear prediction tends to whiten the signal, so the power spectrum

$|E_L(e^{i\omega})|^2$ of the error signal, $e_L(k)$, will tend to be flat. Hence,

$$\lim_{L \rightarrow \infty} |E_L(e^{i\omega})|^2 = G^2. \quad (7.79)$$

As a result,

$$\lim_{L \rightarrow \infty} S_{x,L}(\omega) = \frac{G^2}{|1 - \sum_{l=1}^{\infty} a_l e^{-il\omega}|^2}. \quad (7.80)$$

This confirms that (7.76) can be a very good approximation of the spectrum of a speech signal, as long as the order of the predictor is large enough.

7.7 Linear Interpolation

Linear interpolation can be seen as a straightforward generalization of forward and backward linear predictions. Indeed, in linear interpolation, we try to predict the value of the sample $x(k-i)$ from its past and future values [7.26, 27]. We define the interpolation error as

$$\begin{aligned} e_i(k) &= x(k-i) - \hat{x}(k-i) \\ &= x(k-i) - \sum_{l=0, l \neq i}^L c_{i,l} x(k-l) \\ &= \mathbf{c}_i^T \mathbf{x}_{L+1}(k), \quad i = 0, 1, \dots, L, \end{aligned} \quad (7.81)$$

where $\hat{x}(k-i)$ is the interpolated sample,

$$\mathbf{c}_i = [-c_{i,0} \ -c_{i,1} \ \dots \ c_{i,i} \ \dots \ -c_{i,L}]^T$$

is a vector of length $L+1$ containing the interpolation coefficients, with $c_{i,i} = 1$, and

$$\mathbf{x}_{L+1}(k) = [x(k) \ x(k-1) \ \dots \ x(k-L)]^T.$$

The special cases $i = 0$ and $i = L$ are the forward and backward prediction errors, respectively.

To find the optimal Wiener interpolator, we need to minimize the cost function,

$$\begin{aligned} J_i(\mathbf{c}_i) &= E\{e_i^2(k)\} \\ &= \mathbf{c}_i^T \mathbf{R}_{L+1} \mathbf{c}_i, \end{aligned} \quad (7.82)$$

subject to the constraint

$$\mathbf{c}_i^T \mathbf{v}_i = c_{i,i} = 1, \quad (7.83)$$

where

$$\mathbf{v}_i = [0 \ 0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0]^T$$

is a vector of length $L+1$ with its i -th component equal to one and all others equal to zero. By using a Lagrange multiplier, it is easy to see that the solution to this optimization problem is

$$\mathbf{R}_{L+1} \mathbf{c}_{o,i} = E_i \mathbf{v}_i, \quad (7.84)$$

where

$$\begin{aligned} E_i &= \mathbf{c}_{o,i}^T \mathbf{R}_{L+1} \mathbf{c}_{o,i} \\ &= \frac{1}{\mathbf{v}_i^T \mathbf{R}_{L+1}^{-1} \mathbf{v}_i} \end{aligned} \quad (7.85)$$

is the interpolation-error power.

From (7.84) we find,

$$\frac{\mathbf{c}_{o,i}}{E_i} = \mathbf{R}_{L+1}^{-1} \mathbf{v}_i, \quad (7.86)$$

hence the i -th column of \mathbf{R}_{L+1}^{-1} is $\mathbf{c}_{o,i}/E_i$. We can now see that \mathbf{R}_{L+1}^{-1} can be factorized as follows [7.28]:

$$\begin{aligned} \mathbf{R}_{L+1}^{-1} &= \begin{pmatrix} 1 & -c_{o,1,0} & \dots & -c_{o,L,0} \\ -c_{o,0,1} & 1 & \dots & -c_{o,L,1} \\ \vdots & \vdots & \ddots & \vdots \\ -c_{o,0,L} & -c_{o,1,L} & \dots & 1 \end{pmatrix} \\ &\cdot \begin{pmatrix} 1/E_0 & 0 & \dots & 0 \\ 0 & 1/E_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/E_{L-1} \end{pmatrix} \\ &= \mathbf{C}_o^T \mathbf{D}_e^{-1}. \end{aligned} \quad (7.87)$$

Furthermore, since \mathbf{R}_{L+1}^{-1} is a symmetric matrix, (7.87) can be written as,

$$\begin{aligned}\mathbf{R}_{L+1}^{-1} &= \begin{pmatrix} 1/E_0 & 0 & \cdots & 0 \\ 0 & 1/E_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/E_{L-1} \end{pmatrix} \\ &\cdot \begin{pmatrix} 1 & -c_{o,0,1} & \cdots & -c_{o,0,L} \\ -c_{o,1,0} & 1 & \cdots & -c_{o,1,L} \\ \vdots & \vdots & \ddots & \vdots \\ -c_{o,L,0} & -c_{o,L,1} & \cdots & 1 \end{pmatrix} \\ &= \mathbf{D}_e^{-1} \mathbf{C}_o.\end{aligned}\quad (7.88)$$

Therefore, we deduce that,

$$\frac{c_{o,i,l}}{E_i} = \frac{c_{o,l,i}}{E_l}, \quad i, l = 0, 1, \dots, L.\quad (7.89)$$

The first and last columns of \mathbf{R}_{L+1}^{-1} contain, respectively, the normalized forward and backward predictors and all the columns between contain the normalized interpolators.

We are now going to show how the condition number of the correlation matrix depends on the interpolators. The condition number of the matrix \mathbf{R}_{L+1} is defined as [7.29]:

$$\chi(\mathbf{R}_{L+1}) = \|\mathbf{R}_{L+1}\| \|\mathbf{R}_{L+1}^{-1}\|,\quad (7.90)$$

where $\|\cdot\|$ can be any matrix norm. Note that $\chi(\mathbf{R})$ depends on the underlying norm. Let us compute $\chi(\mathbf{R}_{L+1})$ using the Frobenius norm:

$$\begin{aligned}\|\mathbf{R}_{L+1}\|_F &= [\text{tr}(\mathbf{R}_{L+1}^T \mathbf{R}_{L+1})]^{1/2} \\ &= [\text{tr}(\mathbf{R}_{L+1}^2)]^{1/2}\end{aligned}\quad (7.91)$$

7.8 Line Spectrum Pair Representation

Line spectrum pair (LSP) representation, first introduced by Itakura [7.33], is a more-robust way to represent the coefficients of linear predictive models. The LSP polynomials have some very interesting properties shown in [7.34].

A polynomial $P(z)$ of order L is said to be symmetric if

$$P(z) = z^{-L} P(z^{-1})\quad (7.97)$$

and a polynomial $Q(z)$ is antisymmetric if

$$Q(z) = -z^{-L} Q(z^{-1}).\quad (7.98)$$

and

$$\|\mathbf{R}_{L+1}^{-1}\|_F = [\text{tr}(\mathbf{R}_{L+1}^{-2})]^{1/2}.\quad (7.92)$$

From (7.86), we have,

$$\frac{\mathbf{c}_{o,i}^T \mathbf{c}_{o,i}}{E_i^2} = \mathbf{v}_i^T \mathbf{R}_{L+1}^{-2} \mathbf{v}_i,\quad (7.93)$$

which implies that,

$$\begin{aligned}\sum_{i=0}^L \frac{\mathbf{c}_{o,i}^T \mathbf{c}_{o,i}}{E_i^2} &= \sum_{i=0}^L \mathbf{v}_i^T \mathbf{R}_{L+1}^{-2} \mathbf{v}_i \\ &= \text{tr}(\mathbf{R}_{L+1}^{-2}).\end{aligned}\quad (7.94)$$

Also, we can easily check that,

$$\text{tr}(\mathbf{R}_{L+1}^2) = (L+1)r^2(0) + 2 \sum_{l=1}^L (L+1-l)r^2(l).\quad (7.95)$$

Therefore, the square of the condition number of the correlation matrix associated with the Frobenius norm is

$$\begin{aligned}\chi_F^2(\mathbf{R}_{L+1}) &= \left[(L+1)r^2(0) + 2 \sum_{l=1}^L (L+1-l)r^2(l) \right] \\ &\times \sum_{i=0}^L \frac{\mathbf{c}_{o,i}^T \mathbf{c}_{o,i}}{E_i^2}.\end{aligned}\quad (7.96)$$

Some other interesting relations between the forward predictors and the condition number can be found in [7.30].

To conclude this section, we would like to let readers know that several algorithms exist to compute the optimal predictors efficiently, see for example, [7.31] and [7.32]. All these algorithms are based on Levinson–Durbin recursions.

Let

$$A(z) = 1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_L z^{-L}\quad (7.99)$$

be the optimal polynomial predictor of order L . It is well known that in speech compression the coefficients of this polynomial are inappropriate for quantization because of their relatively large dynamic range and also because, as stated earlier, quantization can change a stable **LPC** filter into an unstable one [7.35]. From (7.99), we can construct two artificial $(L+1)$ -th-order (symmetric and antisymmetric) polynomials by setting the $(L+1)$ -th

reflection coefficient, κ_{L+1} , to be +1 and -1. These two cases correspond, respectively, to an entirely closed or to an entirely open end at the last section of an acoustic tube of $L+1$ piecewise-uniform sections [7.35],

$$P(z) = A(z) + z^{-L} A(z^{-1}), \quad (7.100)$$

$$Q(z) = A(z) - z^{-L} A(z^{-1}). \quad (7.101)$$

The polynomial $A(z)$ can be easily reconstructed from $P(z)$ and $Q(z)$ by

$$A(z) = \frac{1}{2}[P(z) + Q(z)]. \quad (7.102)$$

It was proved in [7.36] and [7.34] that the LSP polynomials, $P(z)$ and $Q(z)$, have the following important properties:

- all zeros of LSP polynomials are on the unit circle,
- the zeros of $P(z)$ and $Q(z)$ are interlaced, and
- the minimum-phase property of $A(z)$ can be easily preserved if the first two properties are intact after quantization.

Now, define the two prediction error signals:

$$e^+(k) = x(n) - \frac{1}{2}[x(n-1) + x(n)]^T \mathbf{a}^+, \quad (7.103)$$

$$e^-(k) = x(n) - \frac{1}{2}[x(n-1) - x(n)]^T \mathbf{a}^-. \quad (7.104)$$

It is shown in [7.37] and [7.38] that the LSP polynomials, whose trivial zeroes have been removed, are equivalent

to the two optimal Wiener predictors \mathbf{a}_0^+ and \mathbf{a}_0^- . This is easy to see if we rewrite, $e^+(k)$ for example as

$$\begin{aligned} e^+(k) &= \left[1 \ - \frac{\mathbf{a}^{+T}}{2} \right] \mathbf{x}_{L+1}(n) \\ &\quad + \left[-\frac{\mathbf{a}^{+T}}{2} \ 0 \right] \mathbf{x}_{L+1}(n) \\ &= \mathbf{g}^T \mathbf{I}_d^T \mathbf{x}_{L+1}(n), \end{aligned} \quad (7.105)$$

where

$$\mathbf{I}_d = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 1 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \quad (7.106)$$

and

$$\mathbf{g} = \begin{pmatrix} 1 \\ -\frac{\mathbf{a}^+}{2} \end{pmatrix}. \quad (7.107)$$

By minimizing the MSE, $E\{e^{+2}(k)\}$, with respect to \mathbf{g} , with the constraint $\mathbf{g}^T \mathbf{v}_1 = 1$, one can find the most important results. For readers who are interested in more details on the properties of LSP polynomials, we recommend the paper by Bäckström and Magi [7.39].

7.9 Multichannel Linear Prediction

Multichannel linear prediction can be very useful in stereo or multichannel speech compression. In an increasing number of speech or audio applications, we have at least two channels available, which are often highly correlated with each other. Therefore, it makes sense to take this interchannel correlation into account in order to obtain more-efficient compression schemes. Multichannel linear prediction is the best way to do this.

Let

$$\chi(k) = \begin{bmatrix} x_1(k) & x_2(k) & \cdots & x_M(k) \end{bmatrix}^T$$

be a real, zero-mean, stationary M -channel time series. We define the multichannel forward prediction error vector as,

$$\begin{aligned} e_{f,L}(k) &= \chi(k) - \hat{\chi}(k) \\ &= \chi(k) - \sum_{l=1}^L \mathbf{A}_{L,l} \chi(k-l) \\ &= \chi(k) - \mathbf{A}_L^T \chi(k-1), \end{aligned} \quad (7.108)$$

where

$$\mathbf{A}_L = [\mathbf{A}_{L,1} \ \mathbf{A}_{L,2} \ \cdots \ \mathbf{A}_{L,L}]^T$$

is the forward predictor matrix of size $ML \times M$, each one of the square matrices $\mathbf{A}_{L,l}$ is of size $M \times M$, and

$$\chi(k-1) = [\chi^T(k-1) \ \chi^T(k-2) \ \cdots \ \chi^T(k-L)]^T$$

is a vector of length ML . (For convenience, some of the notation used in this section is the same as that in the previous sections.)

To derive the optimal Wiener forward predictors, we need to minimize the MSE,

$$J_f(\mathbf{A}_L) = E\{\mathbf{e}_{f,L}^T(k)\mathbf{e}_{f,L}(k)\}. \quad (7.109)$$

We find the multichannel Wiener–Hopf equations:

$$\mathbf{R}_L \mathbf{A}_{o,L} = \mathbf{R}_f(1/L), \quad (7.110)$$

where

$$\begin{aligned} \mathbf{R}_L &= E\{\mathbf{x}(k-1)\mathbf{x}^T(k-1)\} \\ &= E\{\mathbf{x}(k)\mathbf{x}^T(k)\} \end{aligned} \quad (7.111)$$

$$= \begin{pmatrix} \mathbf{R}(0) & \mathbf{R}(1) & \cdots & \mathbf{R}(L-1) \\ \mathbf{R}^T(1) & \mathbf{R}(0) & \cdots & \mathbf{R}(L-2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}^T(L-1) & \mathbf{R}^T(L-2) & \cdots & \mathbf{R}(0) \end{pmatrix} \quad (7.112)$$

is the block-Toeplitz covariance matrix of size $\text{ML} \times \text{ML}$,

$$\begin{aligned} \mathbf{R}(l) &= E\{\chi(k)\chi^T(k-l)\}, \quad l = 0, 1, \dots, L-1, \\ \mathbf{R}(-l) &= E\{\chi(k-l)\chi^T(k)\} = \mathbf{R}^T(l), \end{aligned}$$

and

$$\begin{aligned} \mathbf{R}_f(1/L) &= [\mathbf{R}(1) \ \mathbf{R}(2) \ \cdots \ \mathbf{R}(L)]^T \\ &= E\{\mathbf{x}(k-1)\chi^T(k)\} \end{aligned}$$

is the intercorrelation matrix of size $\text{ML} \times M$.

Using the augmented block-Toeplitz covariance matrix of size $(\text{ML} + M) \times (\text{ML} + M)$:

$$\mathbf{R}_{L+1} = \begin{pmatrix} \mathbf{R}(0) & \mathbf{R}_f^T(1/L) \\ \mathbf{R}_f(1/L) & \mathbf{R}_L \end{pmatrix}, \quad (7.113)$$

we deduce the augmented multichannel Wiener–Hopf equations:

$$\mathbf{R}_{L+1} \begin{pmatrix} \mathbf{I}_{M \times M} \\ -\mathbf{A}_{o,L} \end{pmatrix} = \begin{pmatrix} \mathbf{E}_{f,L} \\ \mathbf{o}_{\text{ML} \times M} \end{pmatrix}, \quad (7.114)$$

where $\mathbf{I}_{M \times M}$ is the identity matrix of size $M \times M$ and

$$\begin{aligned} \mathbf{E}_{f,L} &= E\{\mathbf{e}_{f,o,L}(k)\mathbf{e}_{f,o,L}^T(k)\} \\ &= \mathbf{R}(0) - \mathbf{R}_f^T(1/L)\mathbf{A}_{o,L} \end{aligned} \quad (7.115)$$

is the forward error covariance matrix of size $M \times M$, with

$$\mathbf{e}_{f,o,L}(k) = \chi(k) - \mathbf{A}_{o,L}^T \mathbf{x}(k-1). \quad (7.116)$$

We will proceed with the same philosophy to derive important equations for the multichannel backward prediction. We define the multichannel backward prediction error vector as

$$\begin{aligned} \mathbf{e}_{b,L}(k) &= \chi(k-L) - \hat{\chi}(k-L) \\ &= \chi(k-L) - \sum_{l=1}^L \mathbf{B}_{L,l} \chi(k-l+1) \\ &= \chi(k-L) - \mathbf{B}_L^T \mathbf{x}(k), \end{aligned} \quad (7.117)$$

where

$$\mathbf{B}_L = [\mathbf{B}_{L,1} \ \mathbf{B}_{L,2} \ \cdots \ \mathbf{B}_{L,L}]^T$$

is the backward predictor matrix of size $\text{ML} \times M$ with each one of the square submatrices $\mathbf{B}_{L,l}$ being of size $M \times M$.

The minimization of the MSE,

$$J_b(\mathbf{B}_L) = E\{\mathbf{e}_{b,L}^T(k)\mathbf{e}_{b,L}(k)\}, \quad (7.118)$$

leads to the multichannel Wiener–Hopf equations for the backward prediction:

$$\mathbf{R}_L \mathbf{B}_{o,L} = \mathbf{R}_b(1/L), \quad (7.119)$$

where

$$\begin{aligned} \mathbf{R}_b(1/L) &= E\{\mathbf{x}(k)\chi^T(k-L)\} \\ &= [\mathbf{R}^T(L) \ \mathbf{R}^T(L-1) \ \cdots \ \mathbf{R}^T(1)]^T. \end{aligned} \quad (7.120)$$

By using the augmented block-Toeplitz covariance matrix:

$$\mathbf{R}_{L+1} = \begin{pmatrix} \mathbf{R}_L & \mathbf{R}_b(1/L) \\ \mathbf{R}_b^T(1/L) & \mathbf{R}(0) \end{pmatrix}, \quad (7.121)$$

we find the augmented multichannel Wiener–Hopf equations:

$$\mathbf{R}_{L+1} \begin{pmatrix} -\mathbf{B}_{o,L} \\ \mathbf{I}_{M \times M} \end{pmatrix} = \begin{pmatrix} \mathbf{o}_{\text{ML} \times M} \\ \mathbf{E}_{b,L} \end{pmatrix}, \quad (7.122)$$

where

$$\begin{aligned} \mathbf{E}_{b,L} &= E\{\mathbf{e}_{b,o,L}(k)\mathbf{e}_{b,o,L}^T(k)\} \\ &= \mathbf{R}(0) - \mathbf{R}_b^T(1/L)\mathbf{B}_{o,L} \end{aligned} \quad (7.123)$$

is the backward error covariance matrix of size $M \times M$, with

$$\mathbf{e}_{b,o,L}(k) = \chi(k-L) - \mathbf{B}_{o,L}^T \mathbf{x}(k). \quad (7.124)$$

To solve the multichannel Wiener–Hopf equations efficiently, we need to derive some important relations [7.40]. Consider the following system,

$$\begin{pmatrix} \mathbf{R}_L & \mathbf{R}_b(1/L) \\ \mathbf{R}_b^T(1/L) & \mathbf{R}(0) \end{pmatrix} \begin{pmatrix} \mathbf{I}_{M \times M} \\ -\mathbf{A}_{o,L-1} \\ \mathbf{0}_{M \times M} \end{pmatrix} = \begin{pmatrix} \mathbf{E}_{f,L-1} \\ \mathbf{0}_{(ML-M) \times M} \\ \mathbf{K}_{f,L} \end{pmatrix}, \quad (7.125)$$

where

$$\mathbf{K}_{f,L} = \mathbf{R}^T(L) - \mathbf{R}_b^T(1/L-1)\mathbf{A}_{o,L-1}. \quad (7.126)$$

Consider the other system,

$$\begin{pmatrix} \mathbf{R}(0) & \mathbf{R}_f^T(1/L) \\ \mathbf{R}_f(1/L) & \mathbf{R}_L \end{pmatrix} \begin{pmatrix} \mathbf{0}_{M \times M} \\ -\mathbf{B}_{o,L-1} \\ \mathbf{I}_{M \times M} \end{pmatrix} = \begin{pmatrix} \mathbf{K}_{b,L} \\ \mathbf{0}_{(ML-M) \times M} \\ \mathbf{E}_{b,L-1} \end{pmatrix}, \quad (7.127)$$

where

$$\mathbf{K}_{b,L} = \mathbf{R}(L) - \mathbf{R}_f^T(1/L-1)\mathbf{B}_{o,L-1}. \quad (7.128)$$

If we post-multiply both sides of (7.127) by $\mathbf{E}_{b,L-1}^{-1}\mathbf{K}_{f,L}$, we get:

$$\begin{pmatrix} \mathbf{R}_{L+1} & \mathbf{0}_{M \times M} \\ -\mathbf{B}_{o,L-1} & \mathbf{I}_{M \times M} \end{pmatrix} \mathbf{E}_{b,L-1}^{-1} \mathbf{K}_{f,L} = \begin{pmatrix} \mathbf{K}_{b,L} \mathbf{E}_{b,L-1}^{-1} \mathbf{K}_{f,L} \\ \mathbf{0}_{(ML-M) \times M} \\ \mathbf{K}_{f,L} \end{pmatrix}. \quad (7.129)$$

Subtracting (7.129) from (7.125) and identifying the resulting system with the augmented multichannel Wiener–Hopf equations for forward prediction [eq. (7.114)], we deduce the two recursions:

$$\mathbf{E}_{f,L} = \mathbf{E}_{f,L-1} - \mathbf{K}_{b,L} \mathbf{E}_{b,L-1}^{-1} \mathbf{K}_{f,L}, \quad (7.130)$$

$$\begin{aligned} \mathbf{A}_{o,L} &= \begin{pmatrix} \mathbf{A}_{o,L-1} \\ \mathbf{0}_{M \times M} \end{pmatrix} \\ &\quad - \begin{pmatrix} \mathbf{B}_{o,L-1} \\ -\mathbf{I}_{M \times M} \end{pmatrix} \mathbf{E}_{b,L-1}^{-1} \mathbf{K}_{f,L}. \end{aligned} \quad (7.131)$$

Similarly, if we post-multiply both sides of (7.125) by $\mathbf{E}_{f,L-1}^{-1}\mathbf{K}_{b,L}$, we obtain:

$$\begin{aligned} \mathbf{R}_{L+1} &\begin{pmatrix} \mathbf{I}_{M \times M} \\ -\mathbf{A}_{o,L-1} \\ \mathbf{0}_{M \times M} \end{pmatrix} \mathbf{E}_{f,L-1}^{-1} \mathbf{K}_{b,L} \\ &= \begin{pmatrix} \mathbf{K}_{b,L} \\ \mathbf{0}_{(ML-M) \times M} \\ \mathbf{K}_{f,L} \mathbf{E}_{f,L-1}^{-1} \mathbf{K}_{b,L} \end{pmatrix}. \end{aligned} \quad (7.132)$$

Subtracting (7.132) from (7.127) and identifying the resulting system with the augmented multichannel Wiener–Hopf equations for backward prediction (7.122), we deduce the two recursions:

$$\mathbf{E}_{b,L} = \mathbf{E}_{b,L-1} - \mathbf{K}_{f,L} \mathbf{E}_{f,L-1}^{-1} \mathbf{K}_{b,L}, \quad (7.133)$$

$$\begin{aligned} \mathbf{B}_{o,L} &= \begin{pmatrix} \mathbf{0}_{M \times M} \\ \mathbf{B}_{o,L-1} \end{pmatrix} \\ &\quad - \begin{pmatrix} -\mathbf{I}_{M \times M} \\ \mathbf{A}_{o,L-1} \end{pmatrix} \mathbf{E}_{f,L-1}^{-1} \mathbf{K}_{b,L}. \end{aligned} \quad (7.134)$$

Relations (7.130), (7.131), (7.133), and (7.134) were independently discovered by Whittle [7.41] and Wiggins and Robinson [7.42].

Another important relation needs to be found. Indeed, using (7.116) and (7.124), we can easily verify,

$$E\{\mathbf{e}_{f,o,L-1}(k)\mathbf{e}_{b,o,L-1}^T(k-1)\} = \mathbf{K}_{b,L}, \quad (7.135)$$

$$E\{\mathbf{e}_{b,o,L-1}(k-1)\mathbf{e}_{f,o,L-1}^T(k)\} = \mathbf{K}_{f,L}, \quad (7.136)$$

which implies that,

$$\mathbf{K}_{b,L} = \mathbf{K}_{f,L}^T. \quad (7.137)$$

Table 7.2 summarizes the Levinson–Wiggins–Robinson algorithm [7.42–44], which is a generalization of the Levinson–Durbin algorithm to the multichannel case.

Table 7.2 Levinson–Wiggins–Robinson algorithm

Initialization: $\mathbf{E}_{f,0} = \mathbf{E}_{b,0} = \mathbf{R}(0)$

For $1 \leq l \leq L$

$$\mathbf{K}_{b,l} = \mathbf{R}(l) - \mathbf{R}_f^T(1:l-1)\mathbf{B}_{o,l-1}$$

$$\mathbf{A}_{o,l} = \begin{bmatrix} \mathbf{A}_{o,l-1} \\ \mathbf{0}_{M \times M} \end{bmatrix} - \begin{bmatrix} \mathbf{B}_{o,l-1} \\ -\mathbf{I}_{M \times M} \end{bmatrix} \mathbf{E}_{b,l-1}^{-1} \mathbf{K}_{b,l}^T$$

$$\mathbf{B}_{o,l} = \begin{bmatrix} \mathbf{0}_{M \times M} \\ \mathbf{B}_{o,l-1} \end{bmatrix} - \begin{bmatrix} -\mathbf{I}_{M \times M} \\ \mathbf{A}_{o,l-1} \end{bmatrix} \mathbf{E}_{f,l-1}^{-1} \mathbf{K}_{b,l}$$

$$\mathbf{E}_{f,l} = \mathbf{E}_{f,l-1} - \mathbf{K}_{b,l} \mathbf{E}_{b,l-1}^{-1} \mathbf{K}_{b,l}^T$$

$$\mathbf{E}_{b,l} = \mathbf{E}_{b,l-1} - \mathbf{K}_{b,l}^T \mathbf{E}_{f,l-1}^{-1} \mathbf{K}_{b,l}$$

7.10 Conclusions

In this chapter, we have tried to present the most important results in linear prediction for speech. We have explained the principle of forward linear prediction and have shown that the optimal prediction error signal tends to be a white signal. We have extended the principle of forward linear prediction to backward linear prediction and derived the Cholesky factorization of the inverse correlation matrix. We have developed the classical Levinson–Durbin algorithm, which is a very efficient way to solve the Wiener–Hopf equations for the forward

and backward prediction coefficients. We have explained the idea behind the lattice predictor. We have shown how the spectrum of a speech signal can easily be estimated thanks to the prediction coefficients. We have given some notions of linear interpolation and have demonstrated how the condition number of the correlation matrix is related to the optimal interpolators. We have also presented some notions of line spectrum pair polynomials. Finally, in the last section, we have generalized some of these ideas to the multichannel case.

References

- 7.1 B.S. Atal: The history of linear prediction, *IEEE Signal Proc. Mag.* **23**, 154–161 (2006)
- 7.2 L.R. Rabiner, R.W. Schaffer: *Digital Processing of Speech Signals* (Prentice–Hall, Englewood Cliffs 1976)
- 7.3 J.D. Markel, A.H. Gray Jr.: *Linear Prediction of Speech* (Springer, New York 1976)
- 7.4 J. Makhoul: Linear prediction: a tutorial review, *Proc. IEEE* **63**, 561–580 (1975)
- 7.5 J.W. Picone: Signal modeling techniques in speech recognition, *Proc. IEEE* **81**, 1215–1247 (1993)
- 7.6 S.R. Quackenbush, T.P. Barnwell, M.A. Clements: *Objective Measures of Speech Quality* (Prentice–Hall, Englewood Cliffs 1988)
- 7.7 Y. Huang, J. Benesty, J. Chen: *Acoustic MIMO Signal Processing* (Springer, New York 2006)
- 7.8 F. Itakura, S. Saito: A statistical method for estimation of speech spectral density and formant frequencies, *Electron. Commun. Jpn.* **53**(A), 36–43 (1970)
- 7.9 G. Chen, S.N. Koh, I.Y. Soon: Enhanced Itakura measure incorporating masking properties of human auditory system, *Signal Process.* **83**, 1445–1456 (2003)
- 7.10 M.G. Bellanger: *Adaptive Digital Filters and Signal analysis* (Marcel Dekker, New York 1987)
- 7.11 S. Haykin: *Adaptive Filter Theory*, 4th edn. (Prentice-Hall, Upper Saddle River 2002)
- 7.12 C.W. Therrien: On the relation between triangular matrix decomposition and linear prediction, *Proc. IEEE* **71**, 1459–1460 (1983)
- 7.13 N. Levinson: The Wiener RMS (root mean square) error criterion in filter design and prediction, *J. Math. Phys.* **25**(4), 261–278 (1947), Also Appendix B, in N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series* (MIT, Cambridge 1949)
- 7.14 J. Durbin: Efficient estimation of parameters in moving-average models, *Biometrika* **46**, 306–316 (1959), Parts 1 and 2
- 7.15 J. Durbin: The fitting of time-series models, *Rev. Inst. Int. Stat.* **28**(3), 233–243 (1960)
- 7.16 J.P. Burg: *Maximum Entropy Spectral Analysis Ph.D. Dissertation* (Stanford Univ., Stanford 1975)
- 7.17 P. Delsarte, Y.V. Genin: The split Levinson algorithm, *IEEE Trans. Acoust. Speech ASSP-34*, 470–478 (1986)
- 7.18 R. Kumar: A fast algorithm for solving a Toeplitz system of equations, *IEEE Trans. Acoust. Speech ASSP-33*, 254–265 (1985)
- 7.19 J. Makhoul: Stable and efficient lattice methods for linear prediction, *IEEE Trans. Acoust. Speech ASSP-25*, 423–428 (1977)
- 7.20 F. Itakura, S. Saito: Digital filtering techniques for speech analysis and synthesis, *Proc. 7th Int. Conf. Acoust.* **25**(C-1), 261–264 (1971)
- 7.21 S.W. Lang, J.H. McClellan: A simple proof of stability for all-pole linear prediction models, *Proc. IEEE* **67**, 860–861 (1979)
- 7.22 M.H. Hayes: *Statistical Digital Signal Processing and Modeling* (Wiley, New York 1996)
- 7.23 J. Makhoul: Spectral analysis of speech by linear prediction, *IEEE Trans. Acoust. Speech AU-21*(3), 140–148 (1973)
- 7.24 S.M. Kay, S.L. Marple Jr.: Spectrum analysis – a modern perspective, *Proc. IEEE* **69**, 1380–1419 (1981)
- 7.25 J. M. Cadzow: Spectral estimation: an overdetermined rational model equation approach, *Proc. IEEE* **70**, 907–939 (1982)
- 7.26 S. Kay: Some results in linear interpolation theory, *IEEE Trans. Acoust. Speech ASSP-31*, 746–749 (1983)
- 7.27 B. Picinbono, J.-M. Kerilis: Some properties of prediction and interpolation errors, *IEEE Trans. Acoust. Speech ASSP-36*, 525–531 (1988)
- 7.28 J. Benesty, T. Gaensler: New insights into the RLS algorithm, *EURASIP J. Appl. Si. Pr.* **2004**, 331–339 (2004)
- 7.29 G.H. Golub, C.F. Van Loan: *Matrix Computations* (Johns Hopkins Univ. Press, Baltimore 1996)
- 7.30 J. Benesty, T. Gaensler: Computation of the condition number of a non-singular symmetric Toeplitz matrix with the Levinson–Durbin algorithm, *IEEE Trans. Signal Proces.* **54**, 2362–2364 (2006)

- 7.31 C.K. Coursey, J.A. Stuller: Linear interpolation lattice, *IEEE Trans. Signal Proces.* **39**, 965–967 (1991)
- 7.32 M.R.K. Khansari, A. Leon-Garcia: A fast algorithm for optimal linear interpolation, *IEEE Trans. Signal Process.* **41**, 2934–2937 (1993)
- 7.33 F. Itakura: Line spectrum representation of linear predictive coefficients of speech signal, *J. Acoust. Soc. Am.* **57**(1), 35 (1975)
- 7.34 F.K. Soong, B.-H. Juang: Line spectrum pair (LSP) and speech data compression, *Proc. ICASSP* (1984) pp. 1.10.1–1.10.4
- 7.35 F.K. Soong, B.-H. Juang: Optimal quantization of LSP parameters, *IEEE Trans. Speech Audio Process.* **1**, 15–24 (1993)
- 7.36 S. Sagayama: Stability condition of LSP speech synthesis digital filter, *Proc. Acoust. Soc. Jpn.* (1982) pp. 153–154, (in Japanese)
- 7.37 B. Kleijn, T. Bäckström, P. Alku: On line spectral frequencies, *IEEE Signal Proc. Lett.* **10**, 75–77 (2003)
- 7.38 T. Bäckström, P. Alku, T. Paatero, B. Kleijn: A time-domain interpretation for the LSP decomposi-
tion, *IEEE Trans. Speech Audio Process.* **12**, 554–560 (2004)
- 7.39 T. Bäckström, C. Magi: Properties of line spectrum pair polynomials – A review, *Elsevier Signal Process.* **86**, 3286–3298 (2006)
- 7.40 T. Kailath: A view of three decades of linear filtering theory, *IEEE Trans. Inf. Theory* **IT-20**, 146–181 (1974)
- 7.41 P. Whittle: On the fitting of multivariate autoregressions and the approximate canonical factorization of a spectral density matrix, *Biometrika* **50**, 129–134 (1963)
- 7.42 R.A. Wiggins, E.A. Robinson: Recursive solution to the multichannel filtering problem, *J. Geophys. Res.* **70**, 1885–1891 (1965)
- 7.43 O.N. Strand: Multichannel complex maximum entropy (autoregressive) spectral analysis, *IEEE T. Automat. Contr.* **AC-22**, 634–640 (1977)
- 7.44 P. Delsarte, Y.V. Genin: Multichannel singular predictor polynomials, *IEEE Trans. Circuits Syst.* **35**, 190–200 (1988)

Appendix C

Transformée de Fourier rapide

Computing the vector $(Y_0, Y_1, \dots, Y_{N-1})$ using formula (8.5) requires

$$(N - 1)^2 \text{ complex multiplications,}$$

$$N(N - 1) \text{ complex additions,}$$

assuming that the values of ω_N^j , the sines and cosines of the given angles, have already been computed and stored.

A typical value of N is of the order of 1000, which implies about a million operations of each kind. Considering the frequency of this computation, it was natural to seek to lower the cost. In 1965, two American scientists, J. W. Cooley and J. W. Tukey, developed a much more efficient algorithm that has since been known as the *fast Fourier transform* (FFT). This algorithm takes into consideration the special form of the transformation matrix, which is constructed from the roots of unity. From the beginning, the FFT, including its many extensions, has enjoyed enormous success. In fact, it is safe to say that it has been the backbone of signal and image processing in the last half of the twentieth century. Furthermore, it has been the inspiration for numerous investigations in algebra independently of its intensive use in signal processing. It was indeed a marvelous discovery. The fast Fourier transform marked an important step in the theory of the *complexity of algorithm*. This field of research is concerned with determining and minimizing the cost of a given computation or class of computations, where the cost is measured by the number of numerical operations. For example, we will see that the cost of the FFT is of the order $N \log N$.

9.1 The Cooley–Tukey algorithm

Assume that N is even, $N = 2m$, and rearrange the terms of (8.5) into two groups—those with even indices and those with odd indices. Then

$$Y_k = \frac{1}{2} (P_k + \omega_N^{-k} I_k),$$

where The P_k and I_k are given by the formulas

$$\begin{aligned} P_k &= \frac{1}{m} (y_0 + y_2 \omega_N^{-2k} + \cdots + y_{N-2} \omega_N^{-(N-2)k}), \\ I_k &= \frac{1}{m} (y_1 + y_3 \omega_N^{-2k} + \cdots + y_{N-1} \omega_N^{-(N-2)k}). \end{aligned}$$

Note that we have the relations

$$P_{k+m} = P_k, \quad I_{k+m} = I_k, \quad \omega_N^{-(k+m)} = -\omega_N^{-k}$$

for $k = 0, 1, \dots, m-1$. These identities provide the key to the algorithm, whose essential idea is this:

- Step 1: Compute P_k and $\omega_N^{-k} I_k$;
- Step 2: Form $Y_k = \frac{1}{2}(P_k + \omega_N^{-k} I_k)$;
- Step 3: Deduce $Y_{k+m} = \frac{1}{2}(P_k - \omega_N^{-k} I_k)$.

These computations are done successively only for $k = 0, 1, \dots, m-1$. This scheme is illustrated schematically in Figure 9.1, where the arrows indicate dependent relations in the calculations.

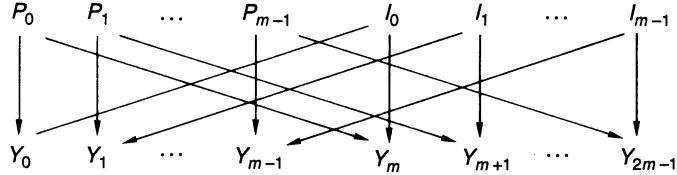


FIGURE 9.1.

The cost of Step 1 is $2(m-1)^2 + m - 1$ (or roughly $N^2/2$) multiplications. Steps 2 and 3 cost nothing in complex multiplications. Thus one obtains the same result for about half the work. (Note we have neglected the divisions by 2 and m . In practice, m is a power of 2, and these are binomial shifts.)

One could consider that this saving is sufficient and stop here. But most readers probably have noticed that P_k and I_k are themselves two independent discrete Fourier transforms of order $m = N/2$. In any case, it takes only a moment to be convinced that

$$\begin{aligned} (y_0, y_2, \dots, y_{2m-2}) &\xrightarrow{\mathcal{F}_{N/2}} (P_0, P_1, \dots, P_{m-1}), \\ (y_1, y_3, \dots, y_{2m-1}) &\xrightarrow{\mathcal{F}_{N/2}} (I_0, I_1, \dots, I_{m-1}). \end{aligned}$$

An obvious strategy is to repeat this clever decomposition, provided that m is even. The best case is where N is a power of 2, $N = 2^p$. We can then iterate the process until we arrive at discrete Fourier transforms of order 2. These are particularly simple computations, since they are of the form

$$\begin{aligned} Y &= (y + z)/2, \\ Z &= (y - z)/2. \end{aligned}$$

We illustrate the algorithm for $N = 8$. The first step is to rearrange the sequence (y_1, y_2, \dots, y_8) into two sequences of length 4, the first having the odd indices and the second the even indices. The process is repeated, and we obtain the four vectors of length 2 shown in Figure 9.2. The computation begins with the vectors of length 2. As in Figure 9.1, the arrows indicate

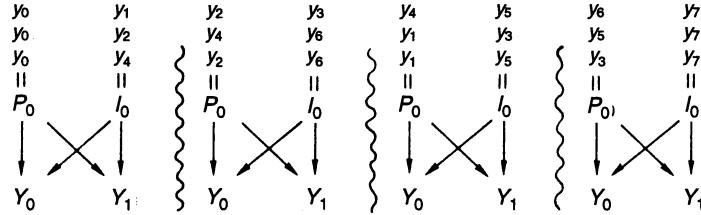


FIGURE 9.2. Rearrangement of the data.

the dependencies of the Y -vectors on the data. To simplify the notation, we have written P_0, I_0, Y_0, Y_1 four times, but they are clearly not the same values. The wiggly lines separate independent computations. Going from one level (vectors of length m) to the next (vectors of length $2m$) is done using the formulas

$$\begin{aligned} Y_k &= \frac{1}{2}(P_k + \omega_N^{-k} I_k), \\ Y_{k+m} &= \frac{1}{2}(P_k - \omega_N^{-k} I_k), \end{aligned}$$

for $k = 0, 1, \dots, m-1$. Figure 9.3 illustrates the complete algorithm for $N = 2^3$. The wiggly lines separate the independent computations. In an actual program, a single vector is used. This is ultimately the output vector $(Y_0, Y_1, \dots, Y_{N-1})$; it is the result of successively transforming the vector obtained by appropriately rearranging the original data.

9.2 Evaluating the cost of the algorithm

The only arithmetic operations that appear in the FFT are multiplications and additions of complex numbers. (We neglect the successive divisions by 2; these reduce to a single division by $N = 2^p$, at the outset, for example.) We denote the cost of r complex multiplications and of s complex additions by $[r; s]$.

For $N = 2^p$, let M_p be the number of multiplications used in the algorithm and let A_p be the number of additions. Formulas (9.1) are used to evaluate the cost for $N = 2^p$ in terms of the cost for $N = 2^{p-1}$:

Cost of computing the P_k : $[M_{p-1}; A_{p-1}]$;
Cost of computing the I_k : $[M_{p-1}; A_{p-1}]$;

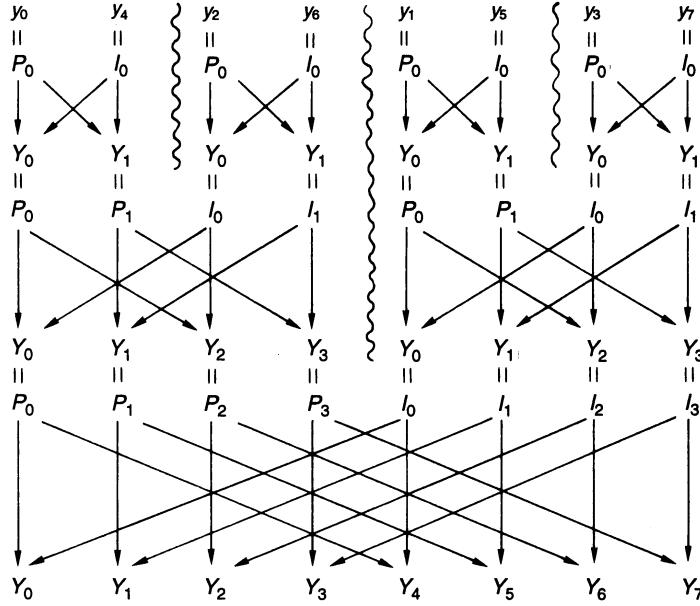


FIGURE 9.3. The FFT algorithm of order 8.

Multiplications by ω_N^{-k} ($k \geq 1$): $[2^{p-1} - 1; 0]$;
Additions: $[0; 2^p]$.

From these relations we have

$$\begin{aligned} M_1 &= 0, & A_1 &= 2, \\ M_p &= 2M_{p-1} + 2^{p-1} - 1, & A_p &= 2A_{p-1} + 2^p. \end{aligned}$$

A computation, which is left as an exercise, provides an explicit expressions for M_p and A_p , namely,

$$\begin{aligned} M_p &= (p-2)2^{p-1} + 1, \\ A_p &= p2^p. \end{aligned}$$

We see from this that the global cost, as a function of N , is

$$\left[\frac{1}{2}N(\log_2 N - 2) + 1; N \log_2 N \right]. \quad (9.1)$$

Table 9.1 compares the FFT with the “old” method. It shows the savings for the two operations as a function of N . For $N = 1024$, we see that the FFT divides the cost by 250: a fantastic gain.

9.3 The mirror permutation

If we wish to obtain the values Y_0, Y_1, \dots, Y_{N-1} in this order, it is clear from Figure 9.3 that we must begin with a vector (y_n) , $n = p(k)$, where

N	Multiplications			Additions		
	Old Method	FFT	Ratio	Old Method	FFT	Ratio
2	0	0		2	2	1
4	0	0		12	8	1.5
8	49	5	10	56	24	2.3
16	225	17	13	240	64	3.8
32	961	49	20	992	160	6.2
64	3,969	129	31	4,032	384	10
128	16,129	321	50	16,256	896	18
256	65,025	769	85	65,280	2,048	32
512	261,121	1,793	145	261,632	4,608	57
1,024	1,046,529	4,097	255	1,047,552	10,240	102

TABLE 9.1.

p is a permutation of the indices $k = 0, 1, \dots, N - 1$. This permutation of the data at the outset is an important issue, particularly for programming the algorithm. There are a number of ways to do this, and it is a problem that generally stimulates much imagination from students. The only restriction is not to introduce so many operations that the gain realized by the algorithm is compromised.

For these consecutive even-odd permutations, one feels that the representation of the indices in base 2 must come into play. Take the case $N = 8$ and notice what happens:

$$\begin{array}{l}
 0 = 000 \dots \parallel \dots 0 = 000 \\
 1 = 001 \dots \parallel \dots 4 = 100 \\
 2 = 010 \dots \parallel \dots 2 = 010 \\
 3 = 011 \dots \parallel \dots 6 = 110 \\
 4 = 100 \dots \parallel \dots 1 = 001 \\
 5 = 101 \dots \parallel \dots 5 = 101 \\
 6 = 110 \dots \parallel \dots 3 = 011 \\
 7 = 111 \dots \parallel \dots 7 = 111
 \end{array}$$

Each number has been written in binary form using three places, which is possible, since we stop at 7 ($N = 2^3$). We notice a surprising property: The required permutation of an index is given by reversing the order of its binary representation. It is as if they were reflected in a mirror. One can verify that this holds for $N = 16$, and it is an excellent exercise to show that it is true in general.

This “mirror” permutation leads to a method for programming the initial permutation. For this, it is necessary to work with the binary representations of the indices. These, however, are not directly accessible in high-level languages like PASCAL; consequently, this is not the best method.

9.4 A recursive program

Here, to finish the chapter, is a program (written in a simplified pseudo-language) for computing the FFT of a vector y . It is taken from [Lip81]. We include it because it is astonishingly simple to program and because it follows step by step the approach we have taken. The particularity of this procedure is that it is *recursive*, which means that calls are made within the program to the program itself.

```

Procedure FFT(n,w,y,Y);
begin
    if n=1 then Y[0]:=y[0] else
    begin
        m:=n div 2;
        for k:=0 to m-1 do
        begin
            b[k]:=y[2*k];
            c[k]:=y[2*k+1]
        end; w2=w*w;
        TFR(m,w2,b,B);
        TFR(m,w2,c,C);
        wk:=1;
        for k:=0 to m-1 do
        begin
            X:=B[k]; T=wk*C[k];
            Y[k]:=(X+T)/2;
            Y[k+m]:=(X-T)/2;
            wk=wk*w
        end
    end
end.

```

We note that compilers deal with these recursions more or less well, particularly on microcomputers. While the program itself is concisely written, which is very attractive to the programmer, its execution, by contrast, requires a great deal of processing and a large amount of memory: At each call to FFT, the procedure is completely recopied with new parameters.

Finally, it is not obvious that this procedure does indeed compute the desired FFT. For example, the second call to FFT is executed only after many other such calls.

Part VI

Documents

Part VII

Production du signal de parole

2. Physiological Processes of Speech Production

K. Honda

Speech sound is a wave of air that originates from complex actions of the human body, supported by three functional units: generation of air pressure, regulation of vibration, and control of resonators. The lung air pressure for speech results from functions of the respiratory system during a prolonged phase of expiration after a short inhalation. Vibrations of air for voiced sounds are introduced by the vocal folds in the larynx; they are controlled by a set of laryngeal muscles and airflow from the lungs. The oscillation of the vocal folds converts the expiratory air into intermittent airflow pulses that result in a buzzing sound. The narrow constrictions of the airway along the tract above the larynx also generate transient source sounds; their pressure gives rise to an airstream with turbulence or burst sounds. The resonators are formed in the upper respiratory tract by the pharyngeal, oral, and nasal cavities. These cavities act as resonance chambers to transform the laryngeal buzz or turbulence sounds into the sounds with special linguistic functions. The main articulators are the tongue, lower jaw, lips, and velum. They generate patterned movements to alter the resonance characteristics of the supra-laryngeal airway. In this chapter, contemporary views on phonatory and

2.1 Overview of Speech Apparatus	7
2.2 Voice Production Mechanisms	8
2.2.1 Regulation of Respiration.....	8
2.2.2 Structure of the Larynx	9
2.2.3 Vocal Fold and its Oscillation	10
2.2.4 Regulation of Fundamental Frequency (F_0).....	12
2.2.5 Methods for Measuring Voice Production.....	13
2.3 Articulatory Mechanisms	14
2.3.1 Articulatory Organs.....	14
2.3.2 Vocal Tract and Nasal Cavity.....	18
2.3.3 Aspects of Articulation in Relation to Voicing	19
2.3.4 Articulators' Mobility and Coarticulation.....	22
2.3.5 Instruments for Observing Articulatory Dynamics	23
2.4 Summary	24
References	25

articulatory mechanisms are summarized to illustrate the physiological processes of speech production, with brief notes on their observation techniques.

2.1 Overview of Speech Apparatus

The speech production apparatus is a part of the motor system for respiration and alimentation. The form of the system can be characterized, when compared with those of other primates, by several unique features, such as small red lips, flat face, compact teeth, short oral cavity with a round tongue, and long pharynx with a low larynx position. The functions of the system are also uniquely advanced by the developed brain with the language areas, direct neural connections from the cortex to motor nuclei, and dense neural supply to each muscle. Independent control over phonation and articulation is a human-specific ability. These morphological and neural changes along human evolution reorganized the

original functions of each component into an integrated motor system for speech communication.

The speech apparatus is divided into the organs of phonation (voice production) and articulation (settings of the speech organs). The phonatory organs (lungs and larynx) create voice source sounds by setting the driving air pressure in the lungs and parameters for vocal fold vibration at the larynx. The two organs together adjust the pitch, loudness, and quality of the voice, and further generate prosodic patterns of speech. The articulatory organs give resonances or modulations to the voice source and generate additional sounds for some consonants. They consist of the lower jaw, tongue,

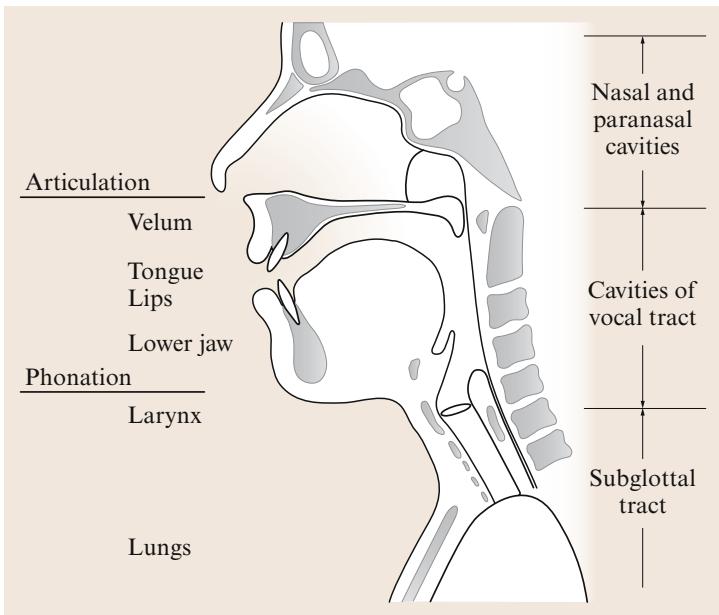


Fig. 2.1 Sketch of a speech production system. Physiological processes of speech production are realized by combined sequential actions of the speech organs for phonation and articulation. These activities result in sound propagation phenomena at the three levels: subglottal cavities, cavities of the vocal tract, and nasal and paranasal cavities

lips, and the velum. The larynx also takes a part in the articulation of voiced/voiceless distinctions. The tongue and lower lip attach to the lower jaw, while the velum is loosely combined with other articulators. The constrictor muscles of the pharynx and larynx also participate in articulation as well as in voice quality control. The phonatory and articulatory systems influence each other mutually, while changing the vocal tract shape for producing vowels and consonants. Figure 2.1 shows a schematic drawing of the speech production system.

2.2 Voice Production Mechanisms

Generation of voice source requires adequate configuration of the airflow from the lungs and vocal fold parameters for oscillation. The sources for voiced sounds are the airflow pulses generated at the larynx, while those for some consonants (i.e., stops and fricatives) are airflow noises made at a narrow constriction in the vocal tract. The expiratory and inspiratory muscles together regulate relatively constant pressure during speech. The laryngeal muscles adjust the onset/offset, amplitude, and frequency of vocal fold vibration.

2.2.1 Regulation of Respiration

The respiratory system is divided into two segments: the conduction airways for ventilation between the atmosphere and the lungs, and the respiratory tissue of the lungs for gas exchange. Ventilation (i.e., expiration and inhalation) is carried out by movements of the thorax, diaphragm, and abdomen. These movements involve actions of respiratory muscles and elastic recoil forces of the system. During quiet breathing, the lungs expand to inhale air by the actions of inspiratory muscles (diaphragm, external intercostal, etc.), and expel air by the elastic recoil force of the lung tissue, diaphragm, and cavities of the thorax and abdomen. In effort expiration, the expiratory muscles (internal intercostals, abdominal muscles, etc.) come into action.

The inspiratory and expiratory muscles work alternately, making the thorax expand and contract during deep breathing.

During speech production, the respiratory pattern changes to a longer expiratory phase with a shorter inspiratory phase during quiet breathing. Figure 2.2 shows a conventional view of the respiratory pattern during

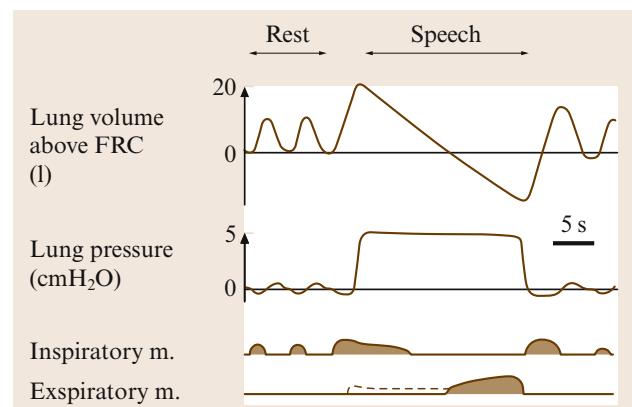


Fig. 2.2 Respiratory pattern during speech. Top two curves show the changes in the volume and pressure in the lungs. The bottom two curves show schematic activity patterns of the inspiratory and expiratory muscles (after [2.1]). The dashed line for the expiratory muscles indicates their predicted activity for expiration

speech [2.1]. The thorax is expanded by inspiration prior to initiation of speech, and then compressed by elastic recoil force by the tissues of the respiratory system to the level of the functional residual capacity (**FRC**). The lung pressure during speech is kept nearly constant except for the tendency of utterance initial rise and final lowering. In natural speech, stress and emphasis add local pressure increases. The constant lung pressure is due to the actions of the inspiratory muscles to prevent excessive airflow and maintain the long expiratory phase. As speech continues, the lung volume decreases gradually below the level of **FRC**, and the lung pressure is then maintained by the actions of the expiratory muscles that actively expel air from the lung. It has been argued whether the initiation of speech involves only the elastic recoil forces of the thorax to generate expiratory airflow. Indeed, a few studies have suggested that not only the thoracic system but also the abdominal system assists the regulation of expiration during speech [2.2, 3], as shown by the dashed line in Fig. 2.2. Thus, the contemporary view of speech respiration emphasizes that expiration of air during speech is not a passive process but a controlled one with co-activation of the inspiratory and expiratory muscles.

2.2.2 Structure of the Larynx

The larynx is a small cervical organ located at the top of the trachea making a junction to the pharyngeal cavity: it primarily functions to prevent foreign material from entering the lungs. The larynx contains several rigid structures such as the cricoid, thyroid, arytenoid, epiglottic, and other smaller cartilages. Figure 2.3a shows the arrangement of the major cartilages and the hyoid bone. The cricoid cartilage is ring-shaped and supports the lumen of the laryngeal cavity. It offers two bilateral articulations to the thyroid and arytenoid cartilages at the cricothyroid and cricoarytenoid joints, respectively. The thyroid cartilage is a shield-like structure that offers attachments to the vocal folds and the vestibular folds. The arytenoid cartilages are bilateral tetrahedral cartilages that change in location and orientation between phonation and respiration. The whole larynx is mechanically suspended from the hyoid bone by muscles and ligaments.

The gap between the free edges of the vocal folds is called the *glottis*. The space is divided into two portions by the vocal processes of the arytenoid cartilages: the membranous portion in front (essential for vibration) and cartilaginous portion in back (essential for respiration). The glottis changes its form in various ways

during speech: it narrows by adduction and widens by abduction of the vocal folds. Figure 2.3b shows that this movement is carried out by the actions of the intrinsic laryngeal muscles that attach to the arytenoid cartilages. These muscles are functionally divided into the adductor and abductor muscles. The adductor muscles include the thyroarytenoid muscles, lateral cricoarytenoid, and arytenoid muscles, and the abductor muscle is the posterior cricoarytenoid muscle. The glottis also changes in length according to the length of the vocal folds, which takes place mainly at the membranous portion. The length of the glottis shows a large developmental sexual variation. The membranous length on average is 10 mm in adult females and 16 mm in adult males, while the cartilaginous length is about 3 mm for both [2.4].

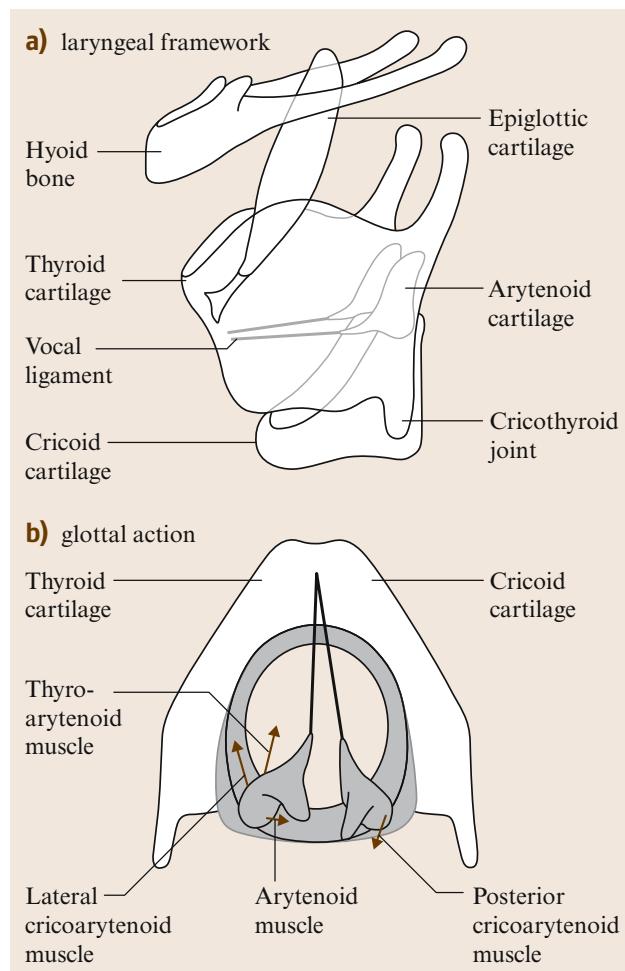


Fig. 2.3a,b Laryngeal framework and internal structures. (a) Oblique view of the laryngeal framework, which includes the hyoid bone and four major cartilages. (b) Adduction (left) and abduction (right) of the glottis and the effects of the intrinsic laryngeal muscles

2.2.3 Vocal Fold and its Oscillation

The larynx includes several structures such as the subglottic dome, vocal folds, ventricle, vestibular folds, epiglottis, and aryepiglottic folds, as shown in Fig. 2.4a. The vocal folds run anteroposteriorly from the vocal processes of the arytenoid cartilages to the internal surface of the thyroid cartilage. The vocal fold tissue consists of the thyroarytenoid muscle, vocal ligament, lamina propria, and mucous membrane. They form a special layer structure that yields to aerodynamic forces to oscillate, which is often described as the *body-cover* structure [2.5].

During voiced speech sounds, the vocal folds are set into vibration by pressurized air passing through the membranous portion of the narrowed glottis. The glottal airflow thus generated induces wave-like motion

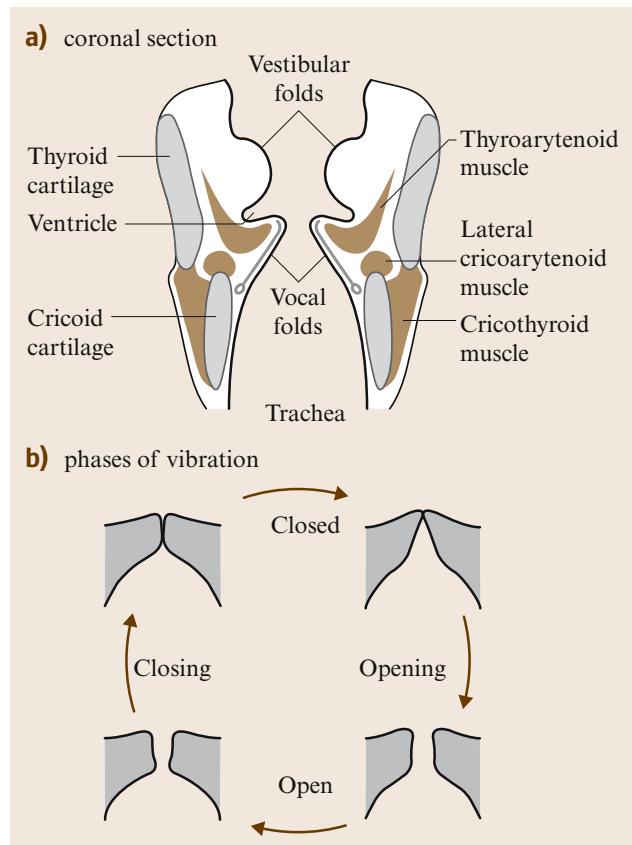


Fig. 2.4a,b Vocal folds and their vibration pattern. (a) Coronal section of the larynx, showing the tissues of the vocal and vestibular (false) folds. The cavity of the larynx includes supraglottic and subglottic regions. (b) Vocal-fold vibration pattern and glottal shapes in open phases. As the vocal-fold edge deforms in a glottal cycle, the glottis follows four phases: closed, opening, open and closing

of the vocal fold membrane, which appears to propagate from the bottom to the top of the vocal fold edges. When this oscillatory motion builds up, the vocal fold membranes on either side come into contact with each other, resulting in repetitive closing and opening of the glottis. Figure 2.4b shows that vocal fold vibration repeats four phases within a cycle: the closed phase, opening phase, open phase, and closing phase. The conditions that determine vocal fold vibration are the stiffness and mass of the vocal folds, the width of the glottis, and the pressure difference across the glottis.

The aerodynamic parameters that regulate vocal fold vibration are the transglottal pressure difference and glottal airflow. The former coincides with the measure of subglottal pressure during mid and low vowels, which is about $5\text{--}10\text{ cm H}_2\text{O}$ in comfortable loudness and pitch ($1\text{ cm H}_2\text{O} = 0.98\text{ hPa}$). The latter also coincides with the average measure of oral

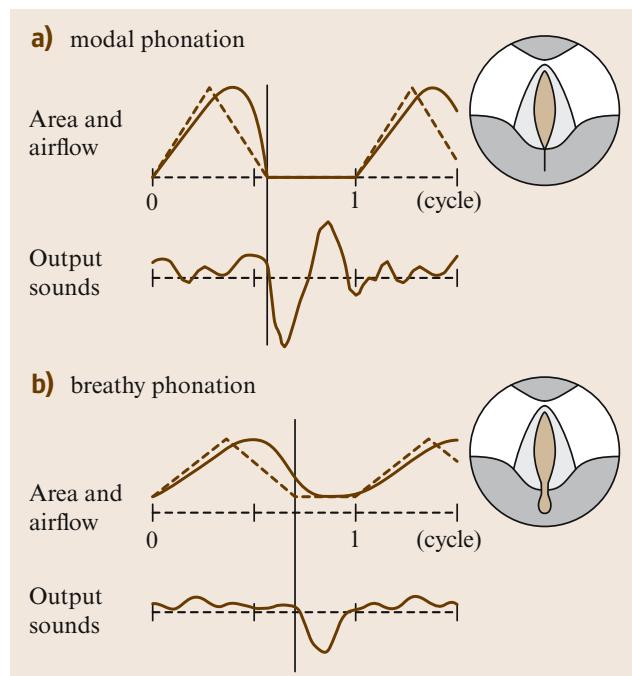


Fig. 2.5a,b Changes in glottal area and airflow in relation to output sounds during 1.5 glottal cycles from glottal opening, with glottal shapes at peak opening (in the circles). (a) In modal phonation with complete glottal closure in the closed phase, glottal closure causes abrupt shut-off of glottal airflow and strong excitation of the air in the vocal tract during the closed phase. (b) In breathy phonation, the glottal closure is incomplete, and the airflow wave includes a DC component, which results in weak excitation of the tract

airflow during vowel production, which is roughly 0.1–0.21/s. These values show a large individual variation: the pressure range is 4.2–9.6 cm H₂O in males and 4.4–7.6 cm H₂O in females, while the airflow rate ranges between 0.1–0.31/s in males and 0.09–0.21 l/s in females [2.6].

Figure 2.5 shows schematically the relationship between the glottal cycle and volumic airflow change in normal and breathy phonation. The airflow varies within each glottal cycle, reflecting the cyclic variation of the glottal area and subglottal pressure. The glottal area curve roughly shows a triangular pattern, while the airflow curve shows a skew of the peak to the right due to the inertia of the air mass within the glottis [2.7]. The closure of the glottis causes a discontinuous decrease of the glottal airflow to zero, which contributes the main source of vocal tract excitation, as shown in Fig. 2.5a. When the glottal closure is more abrupt, the output sounds are more intense with richer harmonic components [2.8]. When the glottal closure is incomplete in soft and breathy voices or the cartilaginous portion of the glottis is open to show the *glottal chink*, the airflow includes a direct-current (DC) component and exhibits a gradual decrease of airflow, which results in a more sinusoidal waveform and a lower intensity of the output sounds, as shown in Fig. 2.5b.

Laryngeal control of the oscillatory patterns of the vocal folds is one of the major factors in voice quality

control. In sharp voice, the open phase of the glottal cycle becomes shorter, while in soft voice, the open phase becomes longer. The ratio of the open phase within a glottal cycle is called the *open quotient (OQ)*, and the ratio of the closing slope to the opening slope in the glottal cycle is called the *speed quotient (SQ)*. These two parameters determine the slope of the spectral envelope. When the open phase is longer (high OQ) with a longer closing phase (low SQ), the glottal airflow becomes more sinusoidal, with weak harmonic components. Contrarily, when the open phase is shorter (low OQ), glottal airflow builds up to pulsating waves with rich harmonics. In modal voice, all the vocal fold layers are involved in vibration, and the membranous glottis is completely closed during the closed phase of each cycle. In falsetto, only the edges of the vocal folds vibrate, glottal closure becomes incomplete, and harmonic components reduce remarkably.

The oscillation of the vocal folds during natural speech is quasiperiodic, and cycle-to-cycle variation are observed in speech waveforms as two types of measures: *jitter* (frequency perturbation) and *shimmer* (amplitude perturbation). These irregularities appear to arise from combinations of biomechanical (vocal fold asymmetry), neurogenic (involuntary activities of laryngeal muscles), and aerodynamic (fluctuations of airflow and subglottal pressure) factors. In sustained phonation of normal voice, the jitter is about 1% in frequency, and the shimmer is about 6% in amplitude.

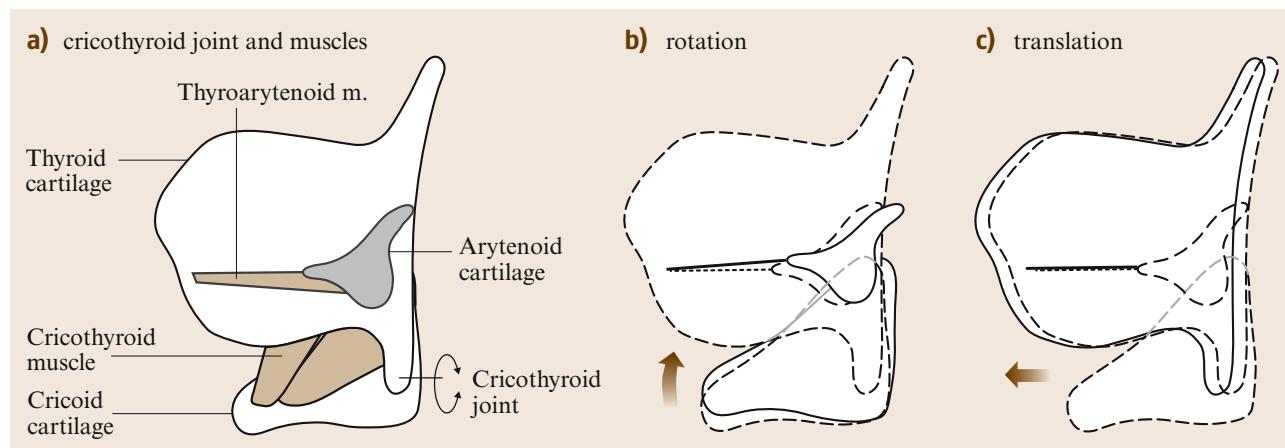


Fig. 2.6a–c Cricothyroid joint and F_0 regulation mechanism. (a) The cricothyroid joint is locally controlled by the thyroarytenoid and two parts of the cricothyroid muscles: Pars recta (anterior) and pars obliqua (posterior). As F_0 rises, the thyroid cartilage advances and cricoid cartilage rotates to the direction to stretch the vocal folds, which leads to the increases in the stiffness of vocal fold tissue and in the natural resonance frequency of the vocal folds. (b) Rotation of the cricothyroid joint is caused mainly by the action of the pars recta to raise the cricoid arch. (c) Translation of the joint is produced mainly by the pars obliqua

2.2.4 Regulation of Fundamental Frequency (F_0)

The fundamental frequency (F_0) of voice is the lowest harmonic component in voiced sounds, which conforms to the natural frequency of vocal fold vibration. F_0 changes depending on two factors: regulation of the length of the vocal folds and adjustment of aerodynamic factors that satisfy the conditions necessary for vocal fold vibration. In high F_0 , the vocal folds become thinner and longer; while in low F_0 , the vocal folds become shorter and thicker. As the vocal folds are stretched by separating their two attachments (the anterior commissure and vocal processes), the mass per unit length of the vocal fold tissue is reduced while the stiffness of the tissue layer involved in vibration increases. Thus, the mass is smaller and the stiffness is greater for higher F_0 than lower F_0 , and it follows that the characteristic frequency of vibrating tissue increases for higher F_0 . The length of the vocal folds is adjusted by relative movement of the cricoid and thyroid cartilages. Its natural length is a determinant factor of individual difference in F_0 . The possible range of F_0 in adult speakers is about 80–400 Hz in males, and about 120–800 Hz in females.

The thyroid and cricoid cartilages are articulated at the cricothyroid joint. Any external forces applied to this joint cause rotation and translation (sliding) of the joint, which alters the length of the vocal folds. It is well known that the two joint actions are brought about by the contraction of the cricothyroid muscle to approximate the two cartilages at their front edges. Figure 2.6 shows two possible actions of the cricothyroid muscle on the joint: rotation by the pars recta and translation of the pars obliqua [2.9]. Questions still remain as to whether each part of the cricothyroid conducts pure actions of rotation or translation, and as to which part is more responsible for determining F_0 .

The extrinsic laryngeal muscles can also apply external forces to this joint as a supplementary mechanism for regulating F_0 [2.10]. The most well known among the activities of the extrinsic muscles in this regulation is the transient action of the sternohyoid muscle observed as F_0 falls. Since this muscle pulls down the hyoid bone to lower the entire larynx, larynx lowering has long been thought to play a certain role in F_0 lowering. Figure 2.7 shows a possible mechanism of F_0 lowering by vertical larynx movement revealed by magnetic resonance imaging (MRI). As the cricoid cartilage descends along the anterior surface of the cervical spine, the cartilage rotates in a direction that

shortens the vocal folds because the cervical spine shows anterior convexity at the level of the cricoid cartilage [2.11].

Aerodynamic conditions are an additional factor that alters F_0 , as seen in the local rises of the subglottal pressure during speech at stress or emphasis. The increase of the subglottal air pressure results in a larger airflow rate and a wider opening of the glottis, which causes greater deformation of the vocal folds with larger average tissue stiffness. The rate of F_0 increase due to the subglottal pressure is reported to be about 2–5 Hz/cmH₂O when the chest cavity is compressed externally, and is observed to be 5–15 Hz/cmH₂O, when

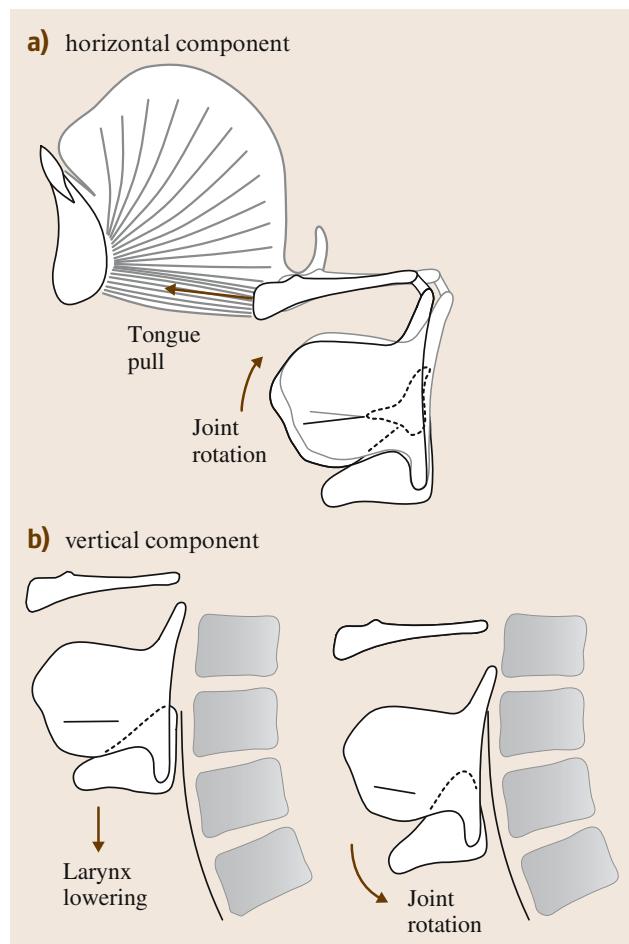


Fig. 2.7a,b Extrinsic control of F_0 . Actions of the cricothyroid joint are determined not only by the cricothyroid muscle but also by other laryngeal muscles. Any external forces applied to the joint can activate the actions of the joint. (a) In F_0 raising, advancement of the hyoid bone possibly apply a force to rotate the thyroid cartilage. (b) In F_0 lowering, the cricoid cartilage rotates as its posterior plate descends along the anterior convexity of the cervical spine

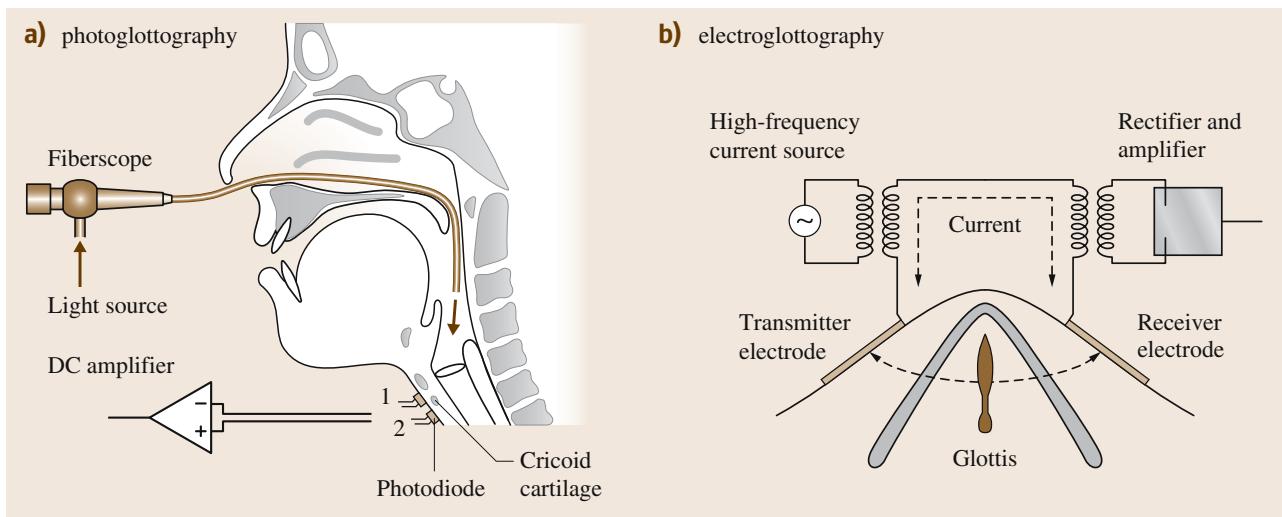


Fig. 2.8a,b Glottographic methods. (a) PGG with fiberscopy uses a photodetector attached near the cricothyroid cartilage in two locations: one attachment for measuring vibrations, and two attachment for glottal gestures. (b) EGG uses a pair of electrodes on the skin above the thyroid lamina to form a induction circuit to record electrical currents passed through the vocal-fold edges

it is measured between the beginning and end of speech utterances.

2.2.5 Methods for Measuring Voice Production

Speech production mechanisms arise from the functions of the internal organs of the human body that are mostly invisible. Therefore, better understanding of speech production processes relies on the development of observation techniques. The lung functions in speech can be assessed by the tools for aerodynamic measurements, while examination of the larynx functions during speech requires special techniques for imaging and signal recording.

Monitoring Respiratory Functions

Respiratory functions during speech are examined by recording aerodynamic measurements of lung volume, airflow, and pressure. Changes in lung volume are monitored with several types of plethysmography (e.g., whole-body, induction, and magnetic). The airflow from the mouth is measured with pneumotachography using a mask with pressure probes (differential-pressure anemometry) or thermal probes (hot-wire anemometry). Measurements of the subglottal pressure require a tracheal puncture of a needle with a pressure sensor or a thin catheter-type pressure transducer inserted from the nostril to the trachea via the cartilaginous part of the glottis.

Laryngeal Endoscopy

Imaging of the vocal folds during speech has been conducted with a combination of an endoscope and video camera. A solid-type endoscope is capable of observing vocal fold vibration with stroboscopic or real-time digital imaging techniques during sustained phonation. The flexible endoscope is beneficial for video recording of glottal movements during speech with a fiber optic bundle inserted into the pharynx through the nostril via the velopharyngeal port. Recently, an electronic type of flexible endoscope with a built-in image sensor has become available.

Glottography

Glottography is a technique to monitor vocal fold vibration as a waveform. Figure 2.8 shows two types of glottographic techniques. Photoglottography (PGG) detects light intensity modulated by the glottis using an optical sensor. The sensor is placed on the neck and a flexible endoscope is used as a light source. The signal from the sensor corresponds to the glottal aperture size, reflecting vocal fold vibration and glottal adduction–abduction movement. Electroglottography (EGG) records the contact of the left and right vocal fold edges during vibration. High-frequency current is applied to a pair of surface electrodes placed on the skin above the thyroid lamina, which detect a varying induction current that corresponds to the change in vocal fold contact area.

2.3 Articulatory Mechanisms

Speech articulation is the most complex motor activity in humans, producing concatenations of phonemes into syllables and syllables into words using movements of the speech organs. These articulatory processes are conducted within a phrase of a single expiratory phase with continuous changes of vocal fold vibration, which is one of the human-specific characteristics of sound production mechanisms.

2.3.1 Articulatory Organs

Articulatory organs are composed of the rigid organ of the lower jaw and soft-tissue organs of the tongue, lips, and velum, as illustrated in Fig. 2.9. These organs together alter the resonance of the vocal tract in various ways and generate sound sources for consonants in the vocal tract. The tongue is the most important articulatory organ, and changes the gross configuration of the vocal tract. Deformation of the whole tongue determines vowel quality and produces palatal, velar, and pharyngeal consonants. Movements of the tongue apex and blade contribute to the differentiation of dental and alveolar consonants and the realization of retroflex consonants. The lips deform the open end of the vocal tract by various types of gestures, assisting the production of vowels and labial consonants. Actions of these soft-tissue organs are essentially based on contractions of

the muscles within these organs, and their mechanism is often compared with the *muscular hydrostat*. Since the tongue and lips have attachments to the lower jaw, they are interlocked with the jaw to open the mouth. The velum controls opening and closing of the velopharyngeal port, and allows distinction between nasal and oral sounds. Additionally, the constrictor muscles of the pharynx adjust the lateral width of the pharyngeal cavity, and their actions also assist articulation for vowels and back consonants.

Upper Jaw

The upper jaw, or the maxilla with the upper teeth, is the structure fixed to the skull, forming the palatal dome on the arch of the alveolar process with the teeth. It forms a fixed wall of the vocal tract and does not belong to the articulatory organs: yet it is a critical structure for speech articulation because it provides the frame of reference for many articulatory gestures. The structures of the upper jaw offer the location for contact or approximation by many parts of the tongue such as the apex, blade, and dorsum. The phonetics literature describes the place of articulation as classified according to the locations of lingual approximation along the upper jaw for dental, alveolar, and palatal consonants. The hard palate is covered by the thick mucoperiosteum, which has several transverse lines of mucosal folds called the *palatine rugae*.

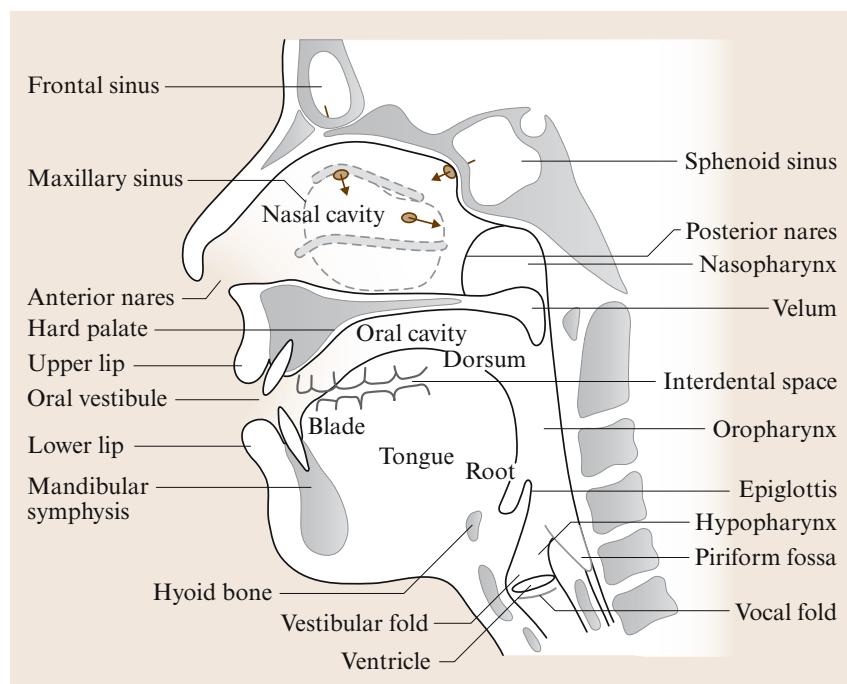


Fig. 2.9 Illustration of the articulatory system with names of articulators and cavities

Lower Jaw

The lower jaw, or the mandible with the lower teeth, is the largest rigid motor organ among the speech production apparatus. Its volume is about 100 cm^3 . As well as playing the major role in opening and closing the mouth, it provides attachments for many speech muscles and supports the tongue, lips, and hyoid bone.

Figure 2.10 shows the action of the jaw and the muscles used in speech articulation. The mandible articulates with the temporal bone at the temporomandibular joint (TMJ) and brings about jaw opening–closing actions by rotation and translation. The muscles that control jaw movements are generally called the masticatory muscles. The jaw opening muscles are the digastric and lateral pterygoid muscles. The strap muscles, such as the geniohyoid and sternohyoid, also assist jaw opening. The jaw closing muscles include the masseter, temporalis, and medial pterygoid muscles. While the larger muscles play major roles in biting and chewing, comparatively small muscles are used for speech articulation. The medial pterygoid is mainly used for jaw closing in articulation, and the elastic recoil force of the connective tissues surrounding the mandible is another factor for closing the jaw from its open position.

Tongue

The tongue is an organ of complex musculature [2.12]. It consists of a round body occupying its main mass and a short blade with an apex. Its volume is approximately 100 cm^3 , including the muscles in the tongue floor. The tongue body moves in the oral cavity by variously deforming its voluminous mass, while the tongue blade alters its shape and changes the angle of the tongue apex. Deformation of the tongue tissue is caused by contractions of the extrinsic and intrinsic tongue muscles, which are illustrated schematically in Fig. 2.11.

The extrinsic tongue muscles are those that arise outside of the tongue and end within the tongue tissue. This group includes four muscles, the genioglossus, hyoglossus, styloglossus, and palatoglossus muscles, although the former three muscles are thought to be involved in the articulation of the tongue. The palatoglossus muscle participates in the lowering of the velum as discussed later.

The genioglossus is the largest and strongest muscle in the tongue. It begins from the posterior aspect of the mandibular symphysis and runs along the midline of the tongue. Morphologically, it belongs to the triangular muscle, and its contraction effects differ across portions of the muscle. Therefore, the genioglossus is divided functionally into the anterior, middle, and posterior bun-

dles. The anterior and middle bundles run midsagittally, and their contraction makes the midline groove of the tongue for the production of front vowels. The anterior bundle often makes a shallow notch on the tongue surface called the *lingual fossa* and assists elevation of the tongue apex. The middle bundle runs obliquely, and advances the tongue body for front vowels. The posterior bundle of the genioglossus runs midsagittally and

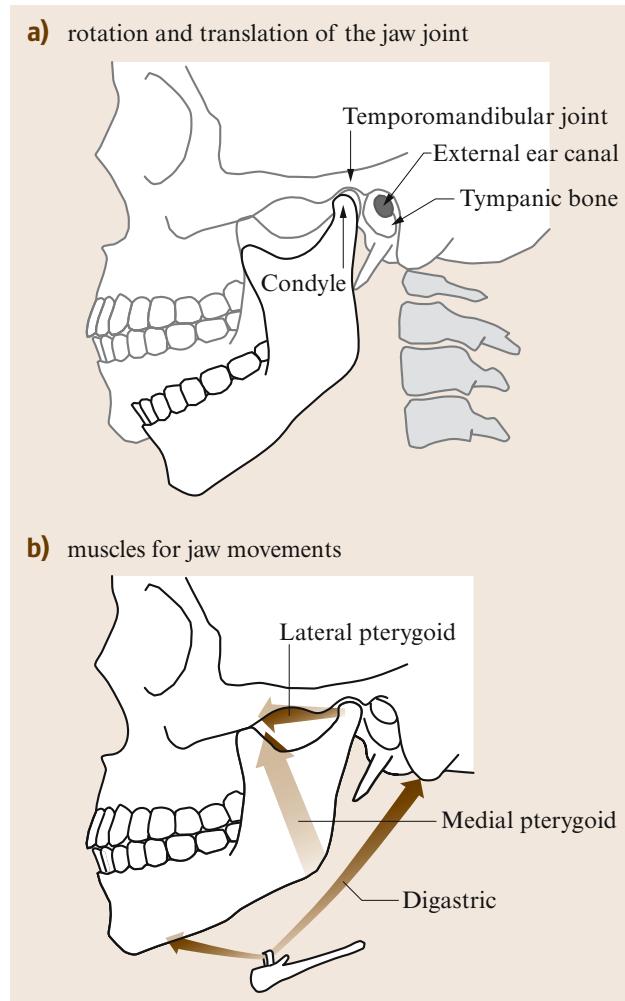


Fig. 2.10a,b Actions of the temporomandibular joint and muscles for jaw opening and closing. (a) The lower jaw opens by rotation and translation of the mandible at the temporomandibular joint. Jaw translation is needed for wide opening of the jaw because jaw rotation is limited by the narrow space between the condyle and tympanic bone. (b) Jaw opening in speech depends on the actions of the digastric and medial pterygoid muscles with support of the strap muscles. Jaw closing is carried out by the contraction of the lateral pterygoid muscle and elastic recoil forces of the tissues surrounding the jaw

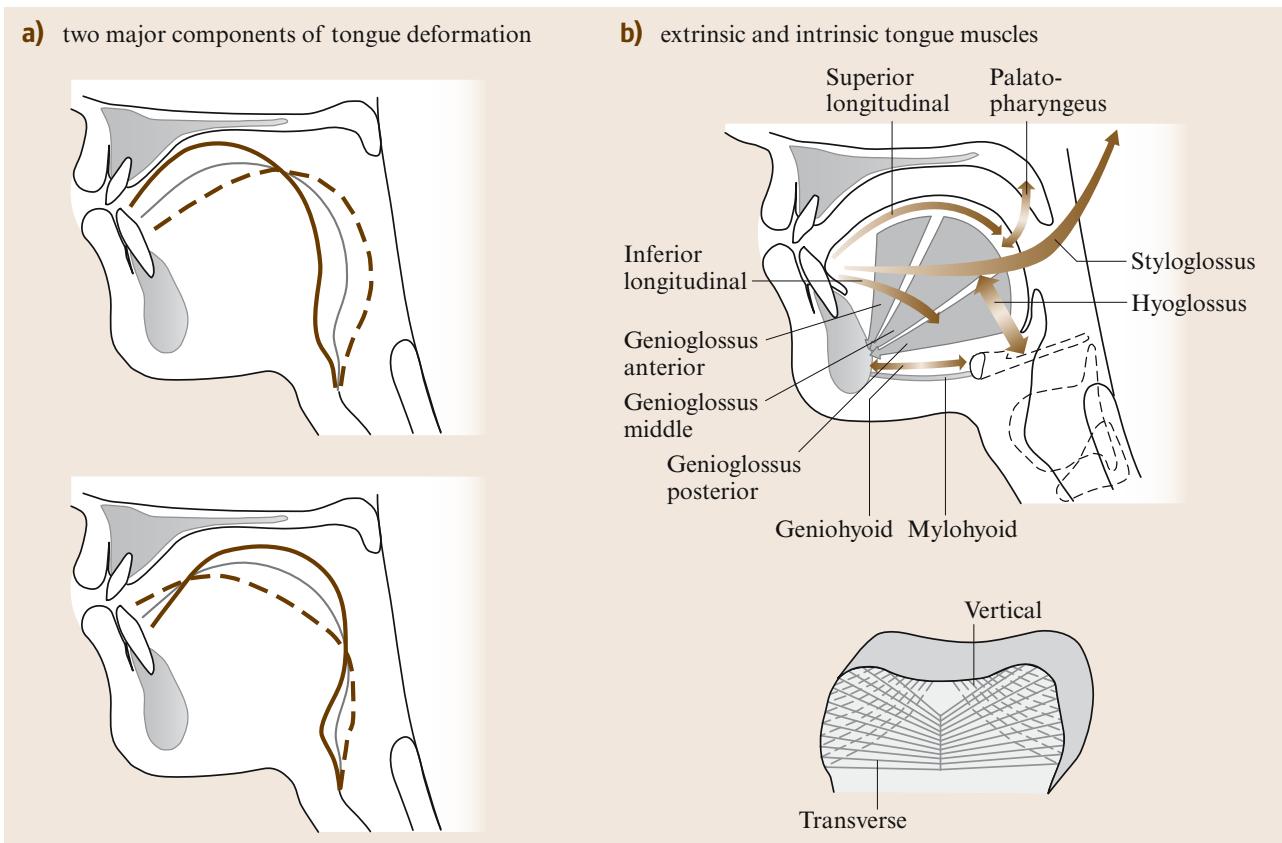


Fig. 2.11a,b Actions of the tongue and its musculature. (a) Major components of tongue deformation are high-front vs. low-back (top) and high back versus low front (bottom), (after [2.14]). (b) Lateral view (top) shows the extrinsic and intrinsic muscles of the tongue with two tongue floor muscles. Coronal section (bottom) shows additional intrinsic muscles

also spreads laterally, reaching a wide area of the tongue root. This bundle draws the tongue root forward and elevates the upper surface of the tongue for high vowels and anterior types of oral consonants. The hyoglossus is a bilateral thin-sheet muscle, which arises from the hyoid bone, runs upward along the sides of the tongue, and ends in the tongue tissue, intermingling with the styloglossus. Its contraction lowers the tongue dorsum and pushes the tongue root backward for the production of low vowels. The styloglossus is a bilateral long muscle originating from the styloid process on the skull base, running obliquely to enter the back sides of the tongue. Within the tongue, it runs forward to reach the apex of the tongue, while branching downward to the hyoid bone and medially toward the midline. Although the extra-lingual bundle of the styloglossus runs obliquely, it pulls the tongue body straight back at the insertion point because the bundle is surrounded by fatty and muscular tissues. The shortening of the intra-lingual bundle draws the tongue apex backward and causes an

upward bunching of the tongue body [2.13]. Each of the extrinsic tongue muscles has two functions: drawing of the relevant attachment point toward the origin, and deforming the tongue tissue in the orthogonal orientation. The resulting deformation of the tongue can be explained by two antagonistic pairs of extrinsic muscles: posterior genioglossus versus styloglossus, and anterior genioglossus versus hyoglossus. The muscle arrangement appears to be suitable for tongue body movements in the vertical and horizontal dimensions.

The intrinsic tongue muscle is a group of muscles that have both their origin and termination within the tongue tissue. They include four bilateral muscles: the superior longitudinal, inferior longitudinal, transverse, and vertical muscles. The superior and inferior longitudinal muscles operate on the tongue blade to produce vertical and horizontal movements of the tongue tip. The transverse and vertical muscles together compress the tongue tissue medially to change the cross-sectional shape of the tongue.

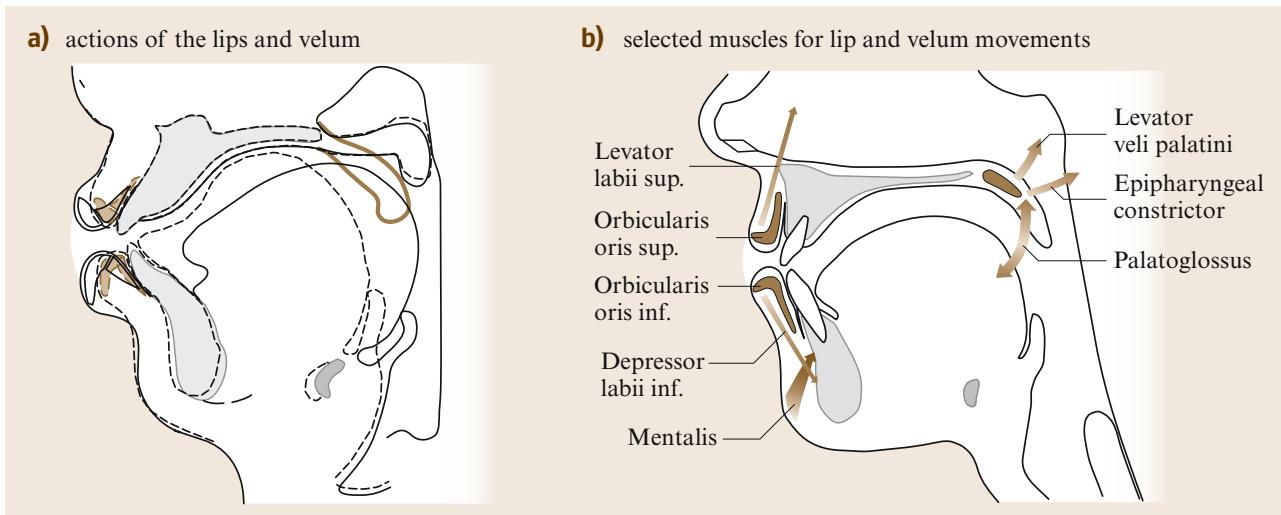


Fig. 2.12a,b Actions of the lips and velum, and their muscles. (a) Trace of MRI data in the production of /i/ and /u/ with lip protrusion show that two parts of the orbicularis oris, marginal (*front*) and peripheral (*back*) bundles demonstrate their geometrical changes within the vermillion tissue. The shapes of the velum also vary greatly between the rest position (thick gray line) and vowel articulation. (b) Five labial muscles are shown selectively from among many facial muscles. The velum shape is determined by the elevator, constrictor, and depressor (palatopharyngeus)

There are two muscles that support the tongue floor: the geniohyoid and mylohyoid muscles. The geniohyoid runs from the genial process of the mandibular symphysis to the body of the hyoid bone. This muscle has two functions: opening the jaw for open vowels and advancing the hyoid bone to help raise F_0 . The mylohyoid is a sheet-like muscle beneath the tongue body that stretches between the mandible and the hyoid bone to support the entire tongue floor. This muscle supports the tongue floor to assist articulation of high front vowels and oral consonants.

Lips and Velum

The lips are a pair of soft-tissue organs consisting of many muscles. Their functions resemble those of the tongue because they partly adhere to the mandible and partly run within the soft tissue of the lips. The vermillion, or the part of red skin, is the unique feature of the human lips, which transmits phonetic signals visually. The deformation of the lips in speech can be divided into three components. The first is opening/closing of the lip aperture, which is augmented by jaw movement. The second is rounding/spreading of the lip tissue, produced by the changes in their left-right dimension. The third is protraction/retraction of the lip gesture, generated by three-dimensional deformation of the entire lip tissue.

The muscles that cause deformation of the lips are numerous. Figure 2.12 shows only a few representative

muscles of the lips. The orbicularis oris is the muscle that surrounds the lips, consisting of two portions; the marginal and peripheral bundles. Contraction of the marginal bundles near the vermillion borders is thought to produce lip rounding without protrusion. Contraction of the peripheral bundles that run in the region around the marginal bundles compresses the lip tissue circumferentially to advance the vermillion in lip protrusion [2.15]. The mentalis arises from the mental part of the mandible to the lip surface, and its contraction elevates the lower lip by pulling the skin at the mental region. The levator labii superior elevates the upper lip, and the depressor labii inferior depresses the lower lip relative to the jaw. The superior and inferior anguli oris muscles move the lip corners up and down, respectively, which makes facial expressions rather than speech articulation.

The exact mechanism of lip protrusion is still in question. Tissue bunching by muscle shortening as a general rule for the organs of muscle does not fully apply to the phenomenon of lip protrusion. This is because, as the vermillion thickens in lip protrusion, it does not compress on the teeth; its dental surface often detaches from the teeth (Figure 2.12a). A certain three-dimensional stress distribution within the entire labial tissue must be considered to account for the causal factors of lip protrusion.

The velum, or the soft palate, works as a valve behind the hard palate to control the velopharyngeal port, as shown in Fig. 2.12a. Elevation of the velum is carried

out during the production of oral sounds, while lowering takes place during the production of nasal sounds. The action of the velum to close the velopharyngeal port is not a pure hinge motion but is accompanied by the deformation of the velum tissue with narrowing of the nasopharyngeal wall. In velopharyngeal closure, the levator veli palatine contracts to elevate the velum, and the superior pharyngeal constrictor muscle produces concentric narrowing of the port. In velopharyngeal opening, the palatoglossus muscle assists active lowering of the velum.

2.3.2 Vocal Tract and Nasal Cavity

The vocal tract is an acoustic space where source sounds for speech propagate. Vowels and consonants rely on strengthening or weakening of the spectral components of the source sound by resonance of the air column in the vocal tract. In the broad definition, the vocal tract includes all the air spaces where acoustic pressure variation takes place in speech production. In this sense, the vocal tract divides into three regions: the subglottal tract, the tract from the glottis to the lips, and the nasal cavities.

The subglottal tract is the lower respiratory tract below the glottis down to the lungs via the trachea and bronchial tubes. The length of the trachea from the glottis to the carina is 10–15 cm in adults, including the

length of the subglottic laryngeal cavity (about 2 cm). Vocal source sounds propagate from the glottis to the trachea, causing the subglottal resonance in speech spectra. The resonance frequencies of the subglottal airway are estimated to be 640, 1400, and 2100 Hz [2.16]. The second subglottal resonance is often observed below the second formant of high vowels.

The vocal tract, according to the conventional definition, is the passage of vocal sounds from the glottis to the lips, where source sounds propagate and give rise to the major resonances. The representative values for the length of the main vocal tract from the glottis to the lips are 15 cm in adult females and 17.5 cm in adult males. According to the measurement data based on the younger population, vocal tract lengths are 14 cm in females and 16.5 cm in males [2.17, 18], which are shorter than the above values. Considering the elongation of the vocal tract during a course of life, the above representative values appear reasonable. While the oral cavity length is maintained by the rigid structures of the skull and jaw, the pharyngeal cavity length increases due to larynx lowering before and after puberty. Thus, elongation of the pharyngeal cavity is the major factor in the developmental variation in vocal tract length.

The vocal tract anatomically divides into four segments: the hypopharyngeal cavities, the mesopharynx, the oral cavity, and the oral vestibule (lip tube). The hypopharyngeal part of the vocal tract consists of the supraglottic laryngeal cavity (2 cm long) and the bilateral conical cavities of the piriform fossa (2 cm long). The mesopharynx extends from the aryepiglottic fold to the anterior palatal arch. The oral cavity is the segment from the anterior palatal arch to the incisors. The oral vestibule extends from the incisors to the lip opening. The latter shows an anterior convexity, which often makes it difficult to measure the exact location of lip opening.

The vocal tract is not a simple uniaxial tube but has a complex three-dimensional construction. The immobile wall of the vocal tract includes the dental arch and the palatal dome. The posterior pharyngeal wall is almost rigid, but it allows subtle changes in convexity and orientation. The soft walls include the entire tongue surface, the velum with the uvula, the lateral pharyngeal wall, and the lip tube. The shape of the vocal tract varies individually due to a few factors. First, the lateral width of the upper and lower jaws relative to the pharyngeal cavity width affects tongue articulation and results in a large individual variation of vocal tract shape observed midsagittally. Second, the mobility of the jaw

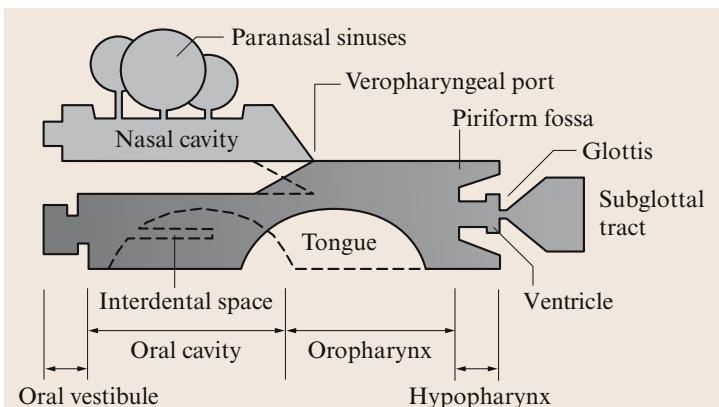


Fig. 2.13 Acoustic design of the vocal tract. Passages from the subglottal tract to two output ends at the lips and nares are shown with the effects of tongue and velar movements. The resonance of the vocal tract above the supraglottic laryngeal cavity determines major the vowel formants (F_1 , F_2 , and F_3). The resonance of the subglottal tract and interdental space interacts with the vowel formants, while the hypopharyngeal cavities and other small cavities cause local resonances and antiresonances in the higher-frequency region

depending on the location of the mandibular symphysis relative to the skull can vary the openness of vowels. Third, the size of the tongue relative to the oral and pharyngeal cavities varies individually; the larger the tongue size, the smaller the articulatory space for vowels.

Figure 2.13 shows a schematic drawing of the vocal tract and nasal cavity. The vocal tract has nearly constant branches such as the piriform fossa (entrance to the esophagus) and the vallecula (between the tongue root and epiglottis). The vocal tract also has controlled branches to the nasal cavity at the velopharyngeal port and to the *interdental space* (the space bounded by the upper and lower teeth and the lateral cheek wall). The latter forms a pair of side-branches when the tongue is in a higher position as in /i/ or /e/, while it is unified with the oral cavity when the tongue is in a lower position as in /a/.

The nasal cavity is an accessory channel to the main vocal tract. Its horizontal dimension from the anterior nares to the posterior wall of the epipharynx is approximately 10–11 cm. The nasal cavity can be divided into the single-tube segment (the velopharyngeal region and epipharynx) and the dual-tube segment (the nasal cavity proper and nasal vestibule). Each of the bilateral channels of the nasal cavity proper has a complex shape of walls with the three turbinates with thick mucous membrane, which makes a narrower cross section compared with the epipharyngeal area [2.19]. The nasal cavity has its own side-branches of the paranasal sinuses; the maxillary, sphenoid, ethmoid, and frontal sinuses.

The nasal cavity builds nasal resonance to accomplish phonetic features of nasal sounds and nasalized vowels. The paranasal sinuses also contribute to acoustic characteristics of the nasal sounds. The nasal murmur results from these characteristics: a Helmholtz resonance of the entire nasopharyngeal tract from the glottis to the anterior nares and regional Helmholtz resonances caused by the paranasal sinuses, together characterized by a resonance peak at 200–300 Hz and spectral flattening up to 2 kHz [2.20, 21]. The nasal resonance could take place even in oral vowels with a complete closure of the velopharyngeal port: the soft tissue of the velum transmits the pressure variation in the oral cavity to the nasal cavity, which would enhance sound radiation for close vowels and voiced stops.

2.3.3 Aspects of Articulation in Relation to Voicing

Here we consider a few phonetic evidences that can be considered as joint products of articulation and phona-

tion. Vowel production is the typical example for this topic, in view of its interaction with the larynx. Regulation of voice quality, which has been thought to be a laryngeal phenomenon, is largely affected by the lower part of the vocal tract. The voiced versus voiceless distinction is a pertinent issue of phonetics that involves both phonatory and articulatory mechanisms.

Production of Vowels

The production of vowels is the result of the joint action of phonatory and articulatory mechanisms. In this process, the larynx functions as a source generator, and the vocal tract plays the role of an acoustic filter to modulate the source sounds and radiate from the lip opening, as described by the *source-filter theory* [2.22, 23]. The quality of oral vowels is determined by a few peak frequencies of vocal tract resonance (formants). In vowel production, the vocal tract forms a *closed tube* with the closed end at the glottis and the open end at the lip opening. Multiple reflections of sound wave between the two ends of the vocal tract give rise to vowel formants (F_1 , F_2 , F_3). The source-filter theory has been supported by many studies as the fundamental concept explaining the acoustic process of speech production, which is further discussed in the next section.

Vowel articulation is the setup for the articulatory organs to determine vocal tract shape for each vowel. When the jaw is in a high position and the tongue is in a high front position, the vocal tract assumes the shape for /i/. Contrarily, when the jaw is in a low position and the tongue is in a low back position, the vocal tract takes the shape for /a/. The articulatory organ that greatly influences vocal tract shape for vowels is the tongue. When the vocal tract is modeled as a tube with two segments (front and back cavities), the movements of the tongue body between its low back and high front positions creates contrasting diverging and converging shapes of the main vocal tract. Jaw movement enhances these changes in the front cavity volume, while pharyngeal constriction assists in the back cavity volume. When the vocal tract is modeled as a tube with three segments, the movements of the tongue body between its high back and low front positions determine the constriction or widening of the vocal tract in its middle portion. The velum also contributes to the articulation of open vowels by decreasing the area of the vocal tract at the velum or making a narrow branch to the nasal cavity. The lip tube is another factor for vowel articulation that determines the vocal tract area near the open end.

Although muscular control for vowel articulation is complex, a simplified view can be drawn based

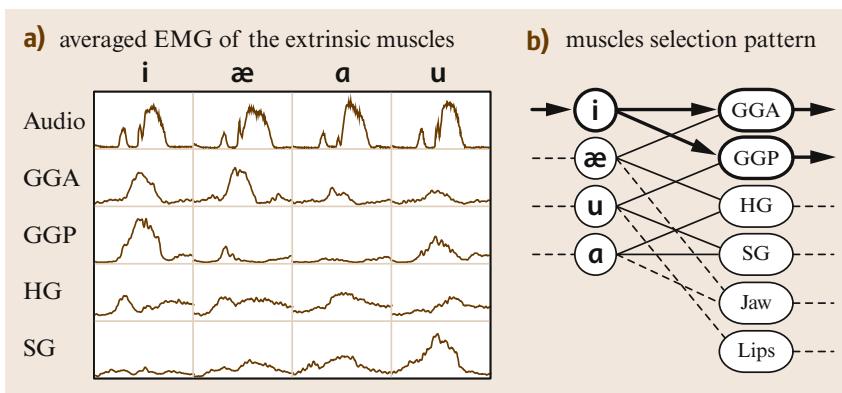


Fig. 2.14a,b Tongue EMG data during VCV utterances and muscle selection pattern in vowel articulation. (a) Averaged EMG data for four English corner vowels are shown for the major muscles of the tongue: the anterior genioglossus (GGA), posterior genioglossus (GGP), hyoglossus (HG), and styloglossus (SG). (b) The systematic variation observed in the muscle–vowel matrix suggests a muscle selection pattern

on electromyographic (EMG) data obtained from the tongue muscles [2.24]. Figure 2.14a shows a systematic pattern of muscle activities for CVC (consonant-vowel-consonant) utterances with /p/ and four English corner vowels. The anterior and posterior genioglossus are active for front vowels, while the styloglossus and hyoglossus are active for back vowels. These muscles also show a variation depending on vowel height. These observations are shown schematically in Fig. 2.14b: the basic control pattern for vowel articulation is the selection of two muscles among the four extrinsic muscles of the tongue [2.25].

As the tongue or jaw moves for vowel articulation, they apply forces to the surrounding organs and cause secondary effects on vowel sounds. For example, articulation of high vowels such as /i/ and /u/ is mainly produced by contraction of the posterior genioglossus, which is accompanied by forward movement of the hyoid bone. This action applies a force to rotate the thyroid

cartilage in a direction that stretches the vocal folds. In evidence, higher vowels tend to have a higher F_0 , known as the *intrinsic vowel F_0* [2.26, 27]. When the jaw opens to produce open vowels, jaw rotation compresses the tissue behind the mandibular symphysis, which applies a force to rotate the thyroid cartilage in the opposite direction, thereby shortening the vocal folds. Thus, the jaw opening has the secondary effect of lowering the intrinsic F_0 for lower vowels.

Supra-Laryngeal Control of Voice Quality

The laryngeal mechanisms controlling voice quality were described in an earlier section. In this section, the supra-laryngeal factors are discussed. Recent studies have shown evidence that the resonances of the hypopharyngeal cavities determine the spectral envelope in the higher frequencies above 2.5 kHz by causing an extra resonance and antiresonances [2.28–31]. The hypopharyngeal cavities include a pair of vocal-tract

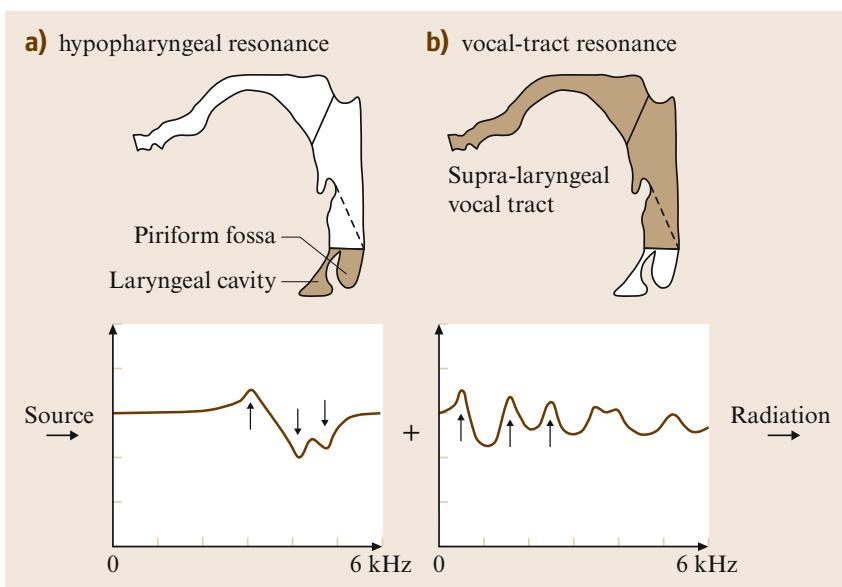


Fig. 2.15a,b Vocal-tract resonance with hypopharyngeal cavity coupling in vowel production. (a) The supra-glottal laryngeal cavity contributes a resonance peak at 3–3.5 kHz, and the bilateral cavities of the piriform fossa cause antiresonances at 4–5 kHz. (b) The main vocal tract above the laryngeal cavity determines the major vowel formants

side-branches formed by the piriform fossa. Each fossa maintains a relatively constant cavity during speech, which is collapsed only in deep inhalation by the wide abduction of the arytenoid cartilage. The piriform fossa causes one or two obvious antiresonances in the higher frequencies above 4 kHz [2.29] and affects the surrounding formants. The laryngeal cavity above the vocal folds also contributes to shaping the higher frequencies [2.28, 32]. The supraglottic laryngeal cavity, from the ventricles to the aryepiglottic folds via the ventricular folds, forms a type of Helmholtz resonator and gives rise to a resonance at higher frequencies of 3–3.5 kHz. This resonance can be counted as the fourth formant (F_4) but it is actually an *extraformant* to the resonance of the vocal tract above the laryngeal cavity [2.30]. When the glottis opens in the open phase of vocal fold vibration, the supraglottic laryngeal cavity no longer constitutes a typical Helmholtz resonator, and demonstrates a strong damping of the resonance, which is observed as the disappearance of the affiliated extra formant. Therefore, the laryngeal cavity resonance shows a cyclic nature during vocal fold vibration, and it is possibly absent in breathy phonation or pathological conditions with insufficient glottal closure [2.31]. Figure 2.15 shows an acoustic model of the vocal tract to illustrate this coupling of the hypopharyngeal cavities.

The hypopharyngeal cavities are not an entirely fixed structure but vary due to physiological efforts to control F_0 and voice quality. A typical case of the

hypopharyngeal adjustment of voice quality is found in the *singing formant* [2.28]. When high notes are produced by opera singers, the entire larynx is pulled forward due to the advanced position of the tongue, which widens the piriform fossa to deepen the fossa's antiresonances, resulting in a decrease of the frequency of the adjacent lower formant (F_5). When the supraglottic laryngeal cavity is constricted, its resonance (F_4) comes down towards the lower formant (F_3). Consequently, the third to fifth formants come closer to each other and generate a high resonance peak observed near 3 kHz.

Regulation of Voiced and Voiceless Sounds

Voiced and voiceless sounds are often attributed to the glottal state with and without vocal fold vibration, while their phonetic characteristics result from phonatory and articulatory controls over the speech production system. In voiced vowels, the vocal tract forms a closed tube with no significant constrictions except for the narrow laryngeal cavity. On the other hand, in whispered vowels, the membranous glottis is closed, and the supraglottic laryngeal cavity forms an extremely narrow channel continued from the open cartilaginous glottis, with a moderate constriction of the lower pharynx. Devoiced vowels exhibit a wide open glottis and a reduction of tongue articulation. Phonetic distinctions of voiced and voiceless consonants further involve fine temporal control over the larynx

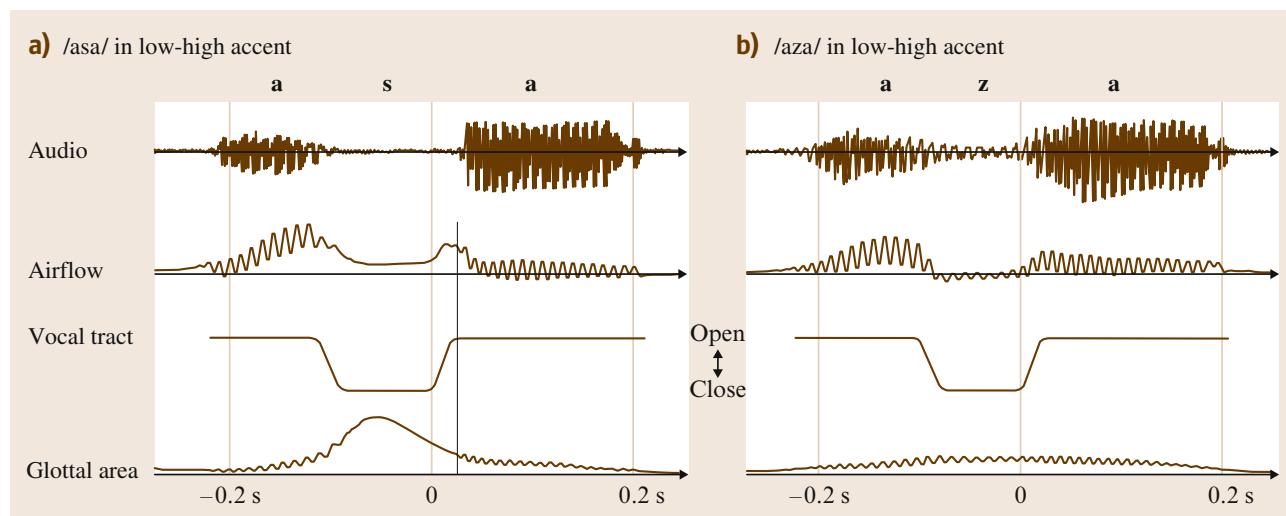


Fig. 2.16a,b Laryngeal articulatory patterns in producing VCV utterances with voiceless and voiced fricatives as in /asa/ and /aza/. From the top to bottom, speech signals, oral airflow, schematic patterns of vocal tract constriction, and glottal area variations are shown schematically. This figure is based on the author's recent experiment with anemometry with an open-type airflow transducer and photoglottography with an external lighting technique, conducted by Dr. Shinji Maeda (ENST) and the author

and supra-laryngeal articulators in language-specific ways.

In the production of voiced consonants, vocal fold vibration typically continues during the voiced segments. In voiced stops and fricatives, the closure or narrowing of the vocal tract results in decrease in glottal airflow and transglottal pressure difference. The glottal airflow during the stop closure is maintained during the closure due to the increases in vocal tract volume: the expansion of the oral cavity (jaw lowering and cheek wall expansion) and the expansion of the pharyngeal cavity (lateral wall expansion and larynx lowering). During the closure period, air pressure variations are radiated not only from the vocal tract wall but also from the anterior nares due to transvelar propagation of the intra-oral sound pressure into the nasal cavities.

In the production of voiceless consonants, vocal fold vibration is suppressed due to a rapid reduction of the transglottal pressure difference and abduction of the vocal folds. During stop closures, the intra-oral pressure builds up to reach the subglottal pressure, which enhances the rapid airflow after the release of the closure. Then, vocal fold vibration restarts with a delay to the release, which is observed as a long voice onset time (**VOT**) for voiceless stops. The process of suppressing vocal fold vibration is not merely a passive aerodynamic process on the vocal folds, but is assisted by a physiological process to control vocal fold stiffness. The cricothyroid muscle has been observed to increase its activity in producing voiceless consonants. This activity results in a high-falling F_0 pattern during the following vowel, contributing a phonetic attribute to voiceless consonants [2.33]. In glottal stops, vocal fold vibration stops due to forced adduction of the vocal folds with an effort closure of the supraglottic laryngeal cavity.

Figure 2.16 illustrates the time course of the processes during vowel-consonant-vowel (**VCV**) utterances with a voiceless fricative in comparison to the case with a voiced fricative. The voiceless segment initiates with glottal abduction and alveolar constriction, and vocal fold vibration gradually fades out during the phase of glottal opening. After reaching the maximum glottal abduction, the glottis enters the adduction phase, followed by the release of the alveolar constriction. Then, the glottis becomes narrower and vocal fold vibration restarts. There is the time lag between the release of the constriction and full adduction of the glottis, which results in the peak flow seen in Fig. 2.16a, presumably accompanied by aspiration sound at the glottis.

2.3.4 Articulators' Mobility and Coarticulation

The mobility of speech articulators varies across organs and contributes certain phonetic characteristics to speech sounds. Rapid movements are essential to a sequence from one distinctive feature to another, as observed in the syllable /sa/ from a narrow constriction to the vocalic opening, while gradual movements are found to produce nasals and certain labial sounds. These variations are principally due to the nature of articulators with respect to their mobility. The articulatory mechanism involves a complex system that is built up by organs with different motor characteristics. Their variation in temporal mobility may be explained by a few biological factors. The first is the phylogenetic origin of the organs: the tongue muscles share their origin with the fast motor systems such as the eyeball or finger, while other muscles such as in the lips or velum originate from the slow motor system similar to the musculature of the alimentary tract. The second is the innervation density to each muscle: the faster organs are innervated by thicker nerve bundles, and vice versa, which derives from an adaptation of the biological system to required functions. In fact, the human hypoglossal nerve that supplies the tongue muscles is much thicker than that of other members of the primate family. The third is the composition of muscle fiber types in the musculature, which varies from organ to organ. The muscles in the larynx have a high concentration of the ultrafast fibers (type 2B), while the muscle to elevate the velum predominantly contains the slow fibers (type 1). In accordance with these biological views, the rate of the articulators movement indexed by the maximum number of syllables per second follows the order of the tongue apex, body, and lips: the tongue moves at a maximum rate of 8.2 syllables per second at the apex, and 7.1 syllables per second with the back of the tongue, while the lips and facial structures move at a maximum rate of 2.5–3 syllables per second [2.34]. More recent measurements indicate that the lips are slower than the tongue apex but faster than the tongue dorsum. The velocities during speech tasks reach 166 mm/sec for the lower lip, 196 mm/sec for the tongue tip, and 129 mm/sec for the tongue dorsum [2.35]. The discrepancy between these two reports regarding the mobility of the lips may be explained by the types of movements measured: opening–closure movement by the jaw–lower lip complex is faster than the movement of the lips themselves, such as protrusion and spreading.

It is often noted that speech is characterized by asynchrony among articulatory movements, and the degree

of asynchrony varies with the feature to be realized. Each articulator does not necessarily strictly keep pace with other articulators in a syllable sequence. The physiological basis of this asynchrony may be explained by the mobility of the articulatory organs and motor precision required for the target of articulation. The slower articulators such as the lips and velum tend to exhibit marked coarticulation in production of labial and nasal sounds. In stop–vowel–nasal sequences (such as /tan/), the velopharyngeal port is tightly closed at the stop onset and the velum begins to lower before the nasal consonant. Thus, the vowel before the nasal consonant is partly nasalized. When the vowel /u/ is preceded by /s/, the lips start to protrude during the consonant prior to the rounded vowel.

The articulators' mobility also contributes some variability to speech movements. The faster articulators such as parts of the tongue show various patterns from target undershooting to overshooting. In articulation of close–open–close vowel sequences such as /iai/, tongue movements naturally show undershooting for the open vowel. In contrast, when the alveolar voiceless stop /t/ is placed in the open vowel context as in /ata/, the tongue blade sometimes shows an extreme overshoot with a wide contact on the hard palate because such articulatory variations do not significantly affect the output sounds. On the contrary, in alveolar and postalveolar fricatives such as /s/ and /sh/, tongue movements also show a dependence on articulatory precision because the position of the tongue blade must be controlled precisely to realize the narrow passage for generating friction

sounds. The lateral /l/ is similar to the stops with respect to the palatal contact, while the rhotic /r/ with no contact to the palate can show a greater extent of articulatory variations from retroflex to bunched types depending on the preceding sounds.

2.3.5 Instruments for Observing Articulatory Dynamics

X-ray and palatography have been used as common tools for articulatory observation. Custom instruments are also developed to monitor articulatory movements, such as the X-ray microbeam system and magnetic sensor system. The various types of newer medical imaging techniques are being used to visualize the movements of articulatory system using sonography and nuclear magnetic resonance. These instruments are generally large scale, although relatively compact instruments are becoming available (e.g., magnetic probing system or portable ultrasound scanner).

Palatography

The palatograph is a compact device to record temporal changes in the contact pattern of the tongue on the palate. There are traditional static and modern dynamic types. The dynamic type is called electropalatography, or dynamic palatography, which employs an individually customized palatal plate to be placed on the upper jaw. As shown in Fig. 2.17a, this system employs a palatal plate with many surface electrodes to monitor electrical contacts on the tongue's surface.

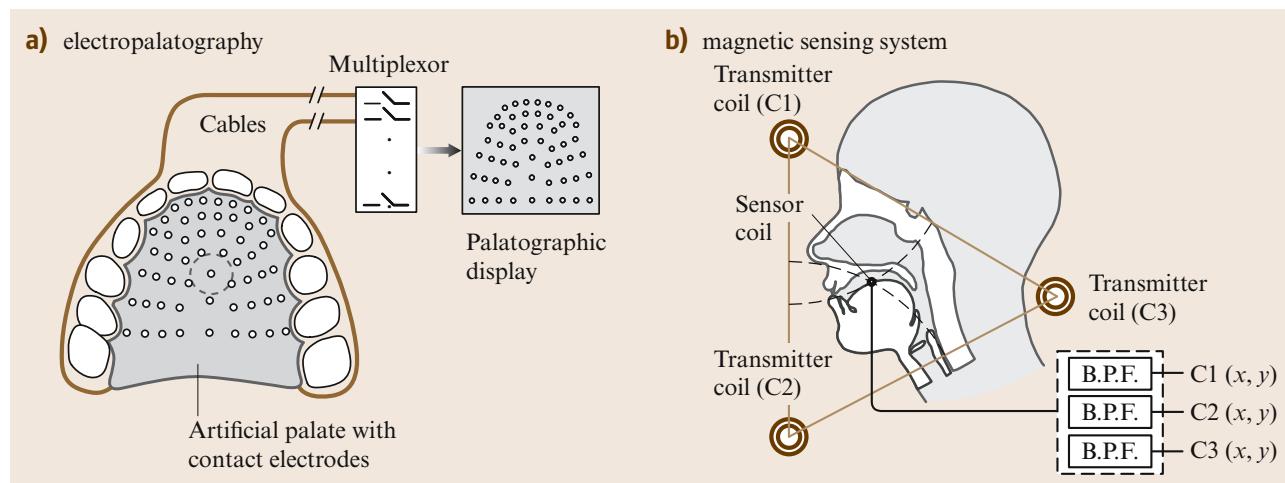


Fig. 2.17a,b Electropalatography and magnetic sensing system. **(a)** Electropalatography displays tongue–palate contact patterns by detecting weak electrical current caused by the contact between the electrodes on the artificial palate and the tongue tissue. **(b)** Magnetic sensing system is based on detection of alternate magnetic fields with different frequencies using miniature sensor coils

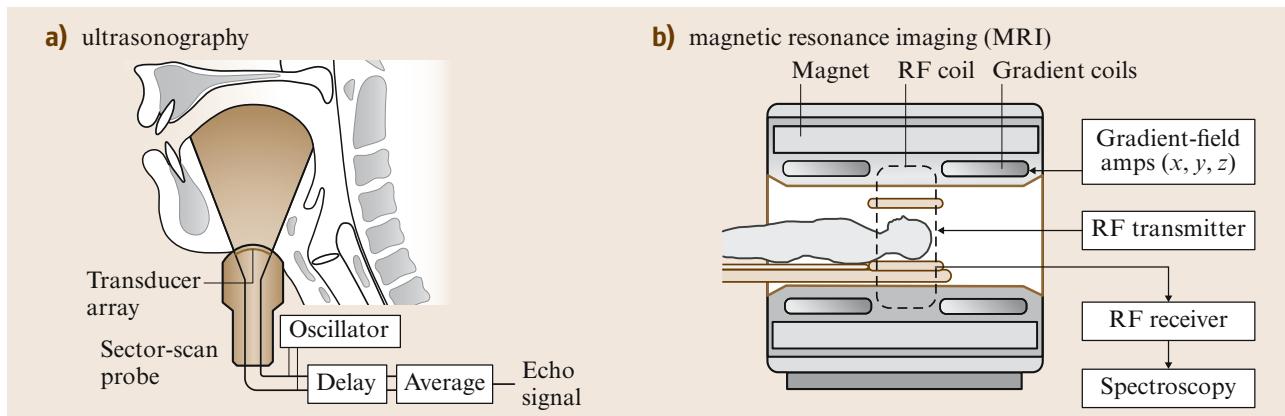


Fig. 2.18a,b Medical imaging techniques. **(a)** Ultrasound scanner uses an array of transmitters and receivers to detect echo signals from regions where the ultrasound signals reflect strongly such as at the tissue-air boundaries on the tongue surface. **(b)** Magnetic resonance imaging (MRI) generates strong static magnetic field, controlled gradient fields in the three directions, and radio-frequency (RF) pulses. Hydrogen atoms respond to the RF pulses to generate echo signals, which are detected with a receiver coil for spectral analysis

Marker Tracking System

A few custom devices have been developed to record movements of markers attached on the articulatory organs. X-ray microbeam and magnetic sensing systems belong to this category. Both can measure 10 markers simultaneously. The X-ray microbeam system uses a computer-controlled narrow beam of high-energy X-rays to track small metal pellets attached on the articulatory organs. This system allows automatic accurate measurements of pellets with a minimum X-ray dosage.

The magnetic sensing system (magnetometer, or magnetic articulograph) is designed to perform the same function as the microbeam system without X-rays. The system uses a set of transmitter coils that generate alternate magnetic fields and miniature sensor coils attached to the articulatory organs, as shown in Fig. 2.17b. The positions of the receiver coils are computed from the filtered signals from the coils.

Medical Imaging Techniques

X-ray cinematography and X-ray video fluorography have been used for recording articulatory movements in two-dimensional projection images. The X-ray images show clear outlines of rigid structures, while they pro-

vide less-obvious boundaries for soft tissue. The outline of the tongue is enhanced by the application of liquid contrast media on the surface. Metal markers are often used to track the movements of flesh points on the soft-tissue articulators.

Ultrasonography is a diagnostic technique to obtain cross-sectional images of soft-tissues in real time. Ultrasound scanners consist of a sound probe (phased-array piezo transducer and receiver) and image processor, as illustrated in Fig. 2.18a. The probe is attached to the skin below the tongue to image the tongue surface in the sagittal or coronal plane.

Magnetic resonance imaging (MRI), shown in Fig. 2.18b, is a developing medical technique that excels at soft-tissue imaging of the living body. Its principle relies on excitation and relaxation of the hydrogen nuclei in water in a strong homogeneous magnetic field in response to radio-frequency (RF) pulses applied with variable gradient magnetic fields that determine the slice position. MRI is essentially a method for recording static images, while motion imaging setups with stroboscopic or real-time techniques have been applied to the visualization of articulatory movements or vocal tract deformation three-dimensionally [2.36].

2.4 Summary

This chapter described the structures of the human speech organs and physiological mechanisms for producing speech sounds. Physiological processes during

speech are multidimensional in nature as described in this chapter. Discoveries of their component mechanisms have been dependent on technical developments

for visualizing the human body and analyses of biological signals, and this is still true today. For example, the hypopharyngeal cavities have long been known to exist, but their acoustic role was underestimated until recent MRI observations. The topics in this chapter were chosen with the author's hope to provide a guideline for the sophistication of speech technologies by reflecting the

real and detailed processes of human speech production. Expectations from these lines of studies include speech analysis by recovering control parameters of articulatory models from speech sounds, speech synthesis with full handling of voice quality and individual vocal characteristics, and true speech recognition through biologic, acoustic, and phonetic characterizations of input sounds.

References

- 2.1 M.H. Draper, P. Ladefoged, D. Whittenridge: Respiratory muscles in speech, *J. Speech Hearing Res.* **2**, 16–27 (1959)
- 2.2 T.J. Hixon, M. Goldman, J. Mead: Kinematics of the chest wall during speech production: volume displacements of the rib cage, abdomen, and lung, *J. Speech Hearing Res.* **16**, 78–115 (1973)
- 2.3 G. Weismar: Speech production. In: *Handbook of Speech-Language Pathology and Audiology*, ed. by N.J. Lass, L.V. McReynolds, D.E. Yoder (Decker, Toronto 1988) pp. 215–252
- 2.4 J. Kahane: A morphological study of the human pre-pubertal and pubertal larynx, *Am. J. Anat.* **151**, 11–20 (1979)
- 2.5 M. Hirano, Y. Kakita: Cover-body theory of vocal cord vibration. In: *Speech Science*, ed. by R. Daniloff (College Hill, San Diego 1985) pp. 1–46
- 2.6 E.B. Holmberg: Glottal airflow and transglottal air pressure measurements for male and female speakers in soft, normal, and loud voice, *J. Acoust. Soc. Am.* **84**, 511–529 (1988)
- 2.7 M.R. Rothenberg: Acoustic interaction between the glottal source and the vocal tract. In: *Vocal Fold Physiology*, ed. by K.N. Stevens, M. Hirano (Univ. Tokyo Press, Tokyo 1981) pp. 305–328
- 2.8 G. Fant, J. Liljencrants, Q. Lin: A four-parameter model of glottal flow, *Speech Transmission Laboratory – Quarterly Progress and Status Report (STL-QPSR)* **4**, 1–13 (1985)
- 2.9 B.R. Fink, R.J. Demarest: *Laryngeal Biomechanics* (Harvard Univ. Press, Cambridge 1978)
- 2.10 J.E. Atkinson: Correlation analysis of the physiological features controlling fundamental frequency, *J. Acoust. Soc. Am.* **63**, 211–222 (1978)
- 2.11 K. Honda, H. Hirai, S. Masaki, Y. Shimada: Role of vertical larynx movement and cervical lordosis in F0 control, *Language Speech* **42**, 401–411 (1999)
- 2.12 H. Takemoto: Morphological analysis of the human tongue musculature for three-dimensional modeling, *J. Speech Lang. Hearing Res.* **44**, 95–107 (2001)
- 2.13 S. Takano, K. Honda: An MRI analysis of the extrinsic tongue muscles during vowel production, *Speech Commun.* **49**, 49–58 (2007)
- 2.14 S. Maeda: Compensatory articulation during speech: evidence from the analysis and synthesis of vocal-tract shapes using an articulatory model. In: *Speech Production and Speech Modeling*, ed. by W.J. Hardcastle, A. Marchal (Kluwer Academic, Dordrecht 1990) pp. 131–149
- 2.15 K. Honda, T. Kurita, Y. Kakita, S. Maeda: Physiology of the lips and modeling of lip gestures, *J. Phonetics* **23**, 243–254 (1995)
- 2.16 K. Ishizaka, M. Matsudaira, T. Kaneko: Input acoustic-impedance measurement of the subglottal system, *J. Acoust. Soc. Am.* **60**, 190–197 (1976)
- 2.17 U.G. Goldstein: An articulatory model for the vocal tracts of growing children. Ph.D. Thesis (Massachusetts Institute of Technology, Cambridge 1980)
- 2.18 W.T. Fitch, J. Giedd: Morphology and development of the human vocal tract: A study using magnetic resonance imaging, *J. Acoust. Soc. Am.* **106**, 1511–1522 (1999)
- 2.19 J. Dang, K. Honda, H. Suzuki: Morphological and acoustic analysis of the nasal and paranasal cavities, *J. Acoust. Soc. Am.* **96**, 2088–2100 (1994)
- 2.20 O. Fujimura, J. Lindqvist: Sweep-tone measurements of the vocal tract characteristics, *J. Acoust. Soc. Am.* **49**, 541–557 (1971)
- 2.21 S. Maeda: The role of the sinus cavities in the production of nasal vowels, *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP'82)*, Vol. 2 (1982) pp. 911–914, Paris
- 2.22 T. Chiba, M. Kajiyama: *The Vowel – Its Nature and Structure* (Tokyo-Kaiseikan, Tokyo 1942)
- 2.23 G. Fant: *Acoustic Theory of Speech Production* (Mouton, The Hague 1960)
- 2.24 T. Baer, P. Alfonso, K. Honda: Electromyography of the tongue muscle during vowels in /pVp/ environment, *Ann. Bull. RILP* **22**, 7–20 (1988)
- 2.25 K. Honda: Organization of tongue articulation for vowels, *J. Phonetics* **24**, 39–52 (1996)
- 2.26 I. Lehiste, G.E. Peterson: Some basic considerations in the analysis of intonation, *J. Acoust. Soc. Am.* **33**, 419–425 (1961)
- 2.27 K. Honda: Relationship between pitch control and vowel articulation. In: *Vocal Fold Physiology*, ed. by D.M. Bless, J.H. Abbs (College-Hill, San Diego 1983) pp. 286–297
- 2.28 J. Sundberg: Articulatory interpretation of the singing formant, *J. Acoust. Soc. Am.* **55**, 838–844 (1974)

- 2.29 J. Dang, K. Honda: Acoustic characteristics of the piriform fossa in models and humans, *J. Acoust. Soc. Am.* **101**, 456–465 (1996)
- 2.30 H. Takemoto, S. Adachi, T. Kitamura, P. Mokhtari, K. Honda: Acoustic roles of the laryngeal cavity in vocal tract resonance, *J. Acoust. Soc. Am.* **120**, 2228–2238 (2006)
- 2.31 T. Kitamura, H. Takemoto, S. Adachi, P. Mokhtari, K. Honda: Cyclicity of laryngeal cavity resonance due to vocal fold vibration, *J. Acoust. Soc. Am.* **120**, 2239–2249 (2006)
- 2.32 I.R. Titze, B.H. Story: Acoustic interactions of the voice source with the lower vocal tract, *J. Acoust. Soc. Am.* **101**, 2234–2243 (1997)
- 2.33 A. Lofqvist, N.S. McGarr, K. Honda: Laryngeal muscles and articulatory control, *J. Acoust. Soc. Am.* **76**, 951–954 (1984)
- 2.34 R.G. Daniloff: Normal articulation processes. In: *Normal Aspect of Speech, Hearing, and Language*, ed. by F.D. Minifie, T.J. Hixon, F. Williams (Prentice-Hall, Englewood Cliffs 1983) pp.169–209
- 2.35 D.P. Kuehn, K.L. Moll: A cineradiographic study of VC and CV articulatory velocities, *J. Phonetics* **4**, 303–320 (1976)
- 2.36 K. Honda, H. Takemoto, T. Kitamura, S. Fujita, S. Takano: Exploring human speech production mechanisms by MRI, *IEICE Info. Syst.* **E87-D**, 1050–1058 (2004)