Institut Mines-Télécom

Representations of signals

Roland Badeau
`roland.badeau@telecom-paris.fr`

Contents

Roland Badeau   roland.badeau@telecom-paris.fr

Roland Badeau   **roland.badeau@telecom-paris.fr**

TELECOM
Paris

IP PARIS

List of Tables

Roland Badeau **roland.badeau@telecom-paris.fr**

List of Figures

Roland Badeau   roland.badeau@telecom-paris.fr

TELECOM
Paris

IP PARIS

Acronyms

AC *Aliasing Cancellation*

AR *Autoregressive*

BIBO *Bounded Input $\Rightarrow$ Bounded Ouput*

CQF *Conjugate Quadrature Filters*

dB *Decibel*

DFT *Discrete Fourier Transform*

DTFT *Discrete Time Fourier Transform*

FFT *Fast Fourier Transform*

FIR *Finite Impulse Response*

FR *Frequency Response*

I/O *Input/Output*

IIR *Infinite Impulse Response*

IR *Impulse Response*

OLA *OverLapp-Add*

PR *Perfect Reconstruction*

STFT *Short Time Fourier Transform*

ZT *Z-Transform*

Mathematical notation

$\mathbb{N}$ set of natural numbers

$\mathbb{Z}$ set of integers

$\mathbb{R}$ set of real numbers

$\mathbb{C}$ set of complex numbers

$Re(.)$ real part of a complex number

$Im(.)$ imaginary part of a complex number

$\arg(.)$ argument (phase) of a complex number

x (normal font, lower case) scalar

$\mathbf{x}$ (bold font, lower case) vector

$\mathbf{A}$ (bold font, upper case) matrix

$\text{diag}(\mathbf{x})$ diagonal matrix whose diagonal entries are those of vector $\mathbf{x}$

$\|.\|_1$ l_1-norm

$\|.\|_\infty$ supremum

$l^1(\mathbb{Z})$ set of sequences whose l_1-norm is finite

$l^\infty(\mathbb{Z})$ set of sequences whose l_∞-norm is finite

$.^\top$ transpose of a vector/matrix

$.^*$ conjugate of a number/vector/matrix

$.^H$ conjugate transpose of a vector/matrix

$H(e^{2\iota\pi\nu}) = \sum_{n\in\mathbb{Z}} h(n)e^{-2\iota\pi\nu n}$ discrete time Fourier transform

$H(z)$ Z-transform

$\mathbf{1}_I(x)$ indicator function of set I ($\mathbf{1}_I(x) = 1$ if $x \in I$, and $\mathbf{1}_I(x) = 0$ otherwise)

$*$ convolution product between two sequences of numbers

Foreword

This course introduces a set of digital signal processing tools that every student should know before learning more advanced techniques:

- Chapter 1 presents various reminders regarding discrete filtering, the Z-transform and recursive filters;

- Chapter 2 introduces several methods for synthesizing both recursive filters and linear phase FIR filters;

- Chapter 3 shows how the sampling rate of a signal can be efficiently converted to a different value;

- Chapter 4 introduces the most famous time-frequency representation: the short-time Fourier transform;

- finally, Chapter 5 introduces the general framework of M-channel filter banks.

These various notions will be used in the forthcoming courses in the TSIA study track. For instance:

- multirate processing and filter banks allow us to pre-process signals before applying machine learning methods, in a way that highlights their characteristics and compresses the information;

- time-frequency representations are essential tools for analyzing, synthesizing and modifying speech, audio, and multimedia signals;

- wavelets (which can be implemented as a binary tree-structured filter bank) find many applications, including denoising.

Roland Badeau roland.badeau@telecom-paris.fr

Contexte académique } sans modifications
Voir Page 79   8/79

TELECOM
Paris

IP PARIS

Chapter 1

Reminders

1   Discrete filtering

In this course, a *signal* is a sequence of real or complex numbers $x(n)$, defined for all times $n \in \mathbb{Z}$. A *filter* is a transform which turns a signal into another signal, and which is both linear and invariant to translations over time: if $x(n)$ and $y(n)$ respectively denote the *input* and *output* signals of the filter, then for any $m \in \mathbb{Z}$, the same filter transforms $x(n - m)$ into $y(n - m)$.

It can be proved that all filters are characterized by their *Impulse Response* (IR) $h(n)$, which is defined as the transform of the impulse signal $\delta_0(n)$ (centered at time $n = 0$, *i.e.* $\delta_0(n) = 1$ if $n = 0$, and $\delta_0(n) = 0$ otherwise). Then the general relationship between the input and output signals is a *convolution product*:

$$y(n) = (h * x)(n) = \sum_{m \in \mathbb{Z}} h(m)x(n - m). \tag{1.1}$$

Note that both h and x have to satisfy some conditions for the convolution product (1.1) to be mathematically well-defined. The most commonly used condition in signal processing is the *Bounded Input $\Rightarrow$ Bounded Ouput* (BIBO) stability condition that will be introduced in the next section.

When both h and x have well-defined Fourier transforms $H(e^{2\imath\pi\nu}) = \sum_{n \in \mathbb{Z}} h(n)e^{-2\imath\pi\nu n}$ and $X(e^{2\imath\pi\nu})$ (defined in the same way) for all *normalized frequencies* $\nu \in \mathbb{R}$, then the Fourier transform of the output signal y is

$$Y(e^{2\imath\pi\nu}) = H(e^{2\imath\pi\nu}) X(e^{2\imath\pi\nu}). \tag{1.2}$$

For this reason, function $H(e^{2\imath\pi\nu})$ is called the *Frequency Response* (FR) of the filter.

1.1   Causality and stability

Beyond its linearity and its invariance to translations over time, one may expect a filter to additionally fulfill one or both of the two following conditions: *causality* and *stability*.

1.1.1   Causality

The *causality* condition is compulsory when we need a filter to act in real-time. It is defined as follows:

Definition 1 (Causal filter). *A filter of impulse response $h(n)$ is causal if and only if, for any input signal $x(n)$, the output $y(n) = (h * x)(n)$ only depends on $x(k)$, $k \leq n$.*

Causality is characterized by the following necessary and sufficient condition:

Proposition 1 (Characterization of causality). *A filter of impulse response $h(n)$ is causal if and only if $h(m) = 0$ $\forall m < 0$.*

TELECOM
Paris

IP PARIS

Proof. We note that $y(n) = \sum_{m<0} h(m)x(n-m) + \sum_{m\geq 0} h(m)x(n-m)$. However, if $m < 0$, then $n - m > n$, thus the definition is equivalent to $h(m) = 0 \;\forall m < 0$. $\qquad\square$

At first, causality was introduced as a property of filters. However, seeing its characterization in Proposition 1, we may like to extend its definition to signals:

Definition 2 (Causal signal). *A signal $x(n)$ is causal if and only if $x(n) = 0 \;\forall n < 0$.*

Now we will investigate what happens when a causal signal is filtered by a causal filter:

Proposition 2 (Causal filtering of a causal signal). *If both the impulse response h and the input signal x are causal, then the output signal $y = h * x$ is also causal.*

Proof. Since h is causal, we have $y(n) = \sum_{m\geq 0} h(m)x(n-m)$. However $\forall n < 0$ and $m \geq 0$, $n - m < 0$, hence $x(n-m) = 0$. Therefore $\forall n < 0$, $y(n) = 0$. $\qquad\square$

1.1.2 Stability

The stability condition is *always* required whenever we want to numerically compute the output of a filter, as we will see below. It is defined as follows:

Definition 3 (Bounded Input $\Rightarrow$ Bounded Output stability). *A filter of impulse response $h(n)$ is stable if and only if, for any bounded input signal $x \in l^\infty(\mathbb{Z})$, the output signal $y = h * x$ is also bounded: $y \in l^\infty(\mathbb{Z})$.*

Stability is characterized by the following necessary and sufficient condition:

Proposition 3 (Characterization of stability). *A filter of impulse response $h(n)$ is stable if and only if $\sum_{m=-\infty}^{+\infty} |h(m)| < +\infty$ (i.e. $h \in l^1(\mathbb{Z})$). Consequently, the frequency response of a stable filter is always a continuous function.*

Proof. First, let us prove that if $h \in l^1(\mathbb{Z})$, then the filter is stable. Indeed, suppose that $h \in l^1(\mathbb{Z})$ and $x \in l^\infty(\mathbb{Z})$. Then $|y(n)| \leq \sum_{m\in\mathbb{Z}} |h(m)||x(n-m)| \leq \sum_{m\in\mathbb{Z}} |h(m)|\|x\|_\infty = \|h\|_1\|x\|_\infty$. Therefore $\|y\|_\infty \leq \|h\|_1\|x\|_\infty < +\infty$.

Reciprocally, let us prove that if the filter is stable, then $h \in l^1(\mathbb{Z})$. Indeed, suppose that the filter is stable. Then let $x(n)e^{-i\arg h(-n)}$. This input signal is bounded: $x \in l^\infty(\mathbb{Z})$. Since the filter is stable, the output signal $y = h * x$ is also bounded. In particular, $|y(0)| < +\infty$. However, $y(0) = \sum_{m\in\mathbb{Z}} h(m)x(-m) = \sum_{m\in\mathbb{Z}} \left(|h(m)|e^{i\arg h(m)}\right) e^{-i\arg h(m)} = \sum_{m\in\mathbb{Z}} |h(m)| = \|h\|_1$. Therefore $h \in l^1(\mathbb{Z})$. $\qquad\square$

Note that the stability property is compulsory to make the numerical implementation of any filter possible. Indeed, assume for instance that $x_q = x+e$ is a quantified version of the input signal x, where e denotes the rounding error. Then, by applying filter h to x_q, we get the output $y_q = y + h * e$. If $h \in l^1(\mathbb{Z})$, then $\|h * e\|_\infty \leq \|h\|_1\|e\|_\infty$, so the resulting rounding error is bounded. However, if $h \notin l^1(\mathbb{Z})$, then $h * e$ may be unbounded: the resulting error can even dominate the desired output y.

1.2 Ideal filters

An *ideal* filter selects a given *frequency band* called *pass-band* of the input signal, and rejects all other frequencies, which form the *stop-band*.

1.2.1 Ideal low-pass filter

The *pass-band* of the ideal *low-pass* filter contains all frequencies lower (in absolute value) than a given threshold, called the *cut-off* frequency $\nu_c \in]0, \frac{1}{2}[$. Its frequency response is thus defined as

$$H_{LP}(e^{2i\pi\nu}) = \left\{ \begin{array}{ll} 1 & \text{if } |\nu| < \nu_c \\ 0 & \text{if } |\nu| > \nu_c \end{array} \right.$$

for all $\nu \in [-\frac{1}{2}, \frac{1}{2}]$, and it is represented in Figure 1.1a (note that $H_{LP}(e^{2i\pi\nu})$ is a 1-periodic function):

TELECOM
Paris

IP PARIS
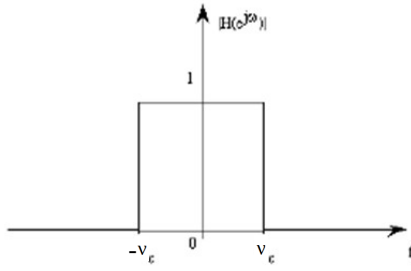
(a) Frequency response of the ideal low-pass filter
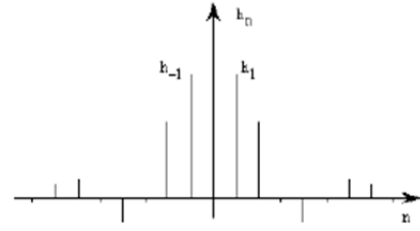
(b) Impulse response of the ideal low-pass filter

Figure 1.1: Ideal low-pass filter

Its impulse response can be calculated by applying the inverse *Discrete Time Fourier Transform* (DTFT) to H_{LP}:

$$h_{LP}(n) = \int_{\nu=-\frac{1}{2}}^{\frac{1}{2}} H_{LP}(e^{2\iota\pi\nu}) \, e^{+2\iota\pi\nu n} d\nu = \int_{\nu=-\nu_c}^{\nu_c} e^{+2\iota\pi\nu n} d\nu = 2\nu_c \, \text{sinc}(2\pi\nu_c n). \tag{1.3}$$

This impulse response is represented in Figure 1.1b. Equation (1.3) shows that the ideal low-pass filter is neither causal ($\exists n < 0$ such that $h_{LP}(n) \neq 0$) nor stable ($h_{LP} \notin l^1(\mathbb{Z})$). Note that calculating the closed-form expression of $h_{LP}(n)$ was actually not necessary to prove it: indeed, since H_{LP} is real and even, then so is h_{LP} (therefore h_{LP} cannot be causal), and since H_{LP} is discontinuous, then $h_{LP} \notin l^1(\mathbb{Z})$ (therefore h_{LP} is not stable).

1.2.2 Ideal band-pass filter

The pass-band of the ideal *band-pass* filter is characterized by its *center frequency* ν_0, and its *bandwidth* $2\nu_c$. Its frequency response is thus defined as

$$H_{BP}(e^{2\iota\pi\nu}) = \left\{ \begin{array}{ll} 1 & \text{if } \left|\,|\nu| - \nu_0\,\right| < \nu_c \\ 0 & \text{if } \left|\,|\nu| - \nu_0\,\right| > \nu_c \end{array} \right.$$

for all $\nu \in [-\frac{1}{2}, \frac{1}{2}]$, where ν_0 and ν_c are such that $\nu_0 - \nu_c > 0$ and $\nu_0 + \nu_c < \frac{1}{2}$.

We note that $H_{BP}(e^{2\iota\pi\nu}) = H_{LP}(e^{2\iota\pi(\nu-\nu_0)}) + H_{LP}(e^{2\iota\pi(\nu+\nu_0)})$. Therefore its impulse response is

$$h_{BP}(n) = h_{LP}(n) \, e^{2i\pi\nu_0 n} + h_{LP}(n) \, e^{-2i\pi\nu_0 n} = 4\nu_c \, \text{sinc}(2\pi\nu_c n) \, \cos(2\pi\nu_0 n).$$

Again, the ideal band-pass filter is neither causal nor stable, for the same reasons as the ideal low-pass filter.

1.3 Transient and steady states of a causal filter

The transient and steady states of a filter are defined when a causal filter is applied to a causal input signal. Let us introduce these notions on an example. Consider for instance the causal averaging filter of length N: $h(n) = \frac{1}{N} 1_{[0,N-1]}(n)$. Then suppose that the input signal $x(n)$ is the unit step signal: $x(n) = 1_{[0,+\infty[}(n)$.

Due to Proposition 2, we already know that $y = h * x$ is a causal signal. Besides, we could expect that averaging the unit step signal produces $y(n) = 1$ for $n \geq 0$. This is untrue because of the causality: actually, $y(n) = 1$ only for $n \geq N - 1$. In the interval $n \in [0, N - 2]$, $y(n)$ looks like a ramp, which smoothly transitions from $y(n) = 0$ to $y(n) = 1$. This transition is called the *transient state* of the filter; its duration is $N - 1$ samples. This transient state is followed, for $n \geq N - 1$, by the *steady state* of the filter, where $y(n)$ becomes constant in this particular case.

1.4 Phase and group delays

Let $x(n)$ be a narrowband signal: $x(n) = a(n) \, e^{i2\pi\nu_0 n}$ where $a(n)$ is the *temporal envelope* of the signal (the support of $A(e^{2i\pi\nu})$ is assumed to be limited to a neighborhood of the zero frequency), and ν_0 is called the *carrier frequency*. Equivalently, we have $X(e^{2i\pi\nu}) = A(e^{2i\pi(\nu-\nu_0)})$.

Roland Badeau   `roland.badeau@telecom-paris.fr`

We consider the polar factorization of the filter frequency response: $H(e^{i2\pi\nu}) = H_R(\nu) e^{i\phi(\nu)}$ where the real amplitude $H_R(\nu) \in \mathbb{R}$ may be negative at some frequencies (so as to guarantee its continuity w.r.t. ν), and $\phi(\nu)$ is the phase of the filter. The phase and group delays of the filter are then defined as follows:

$$\begin{cases} \tau_p(\nu_0) & = & -\frac{1}{2\pi}\frac{\phi(\nu_0)}{\nu_0}; \\ \tau_g(\nu_0) & = & -\frac{1}{2\pi}\frac{d\phi}{d\nu}(\nu_0). \end{cases}$$

In order to explain these definitions, let us approximate the filter frequency response $H(e^{i2\pi\nu})$ in the neighborhood of ν_0, by means of a first order Taylor expansion of the phase, and a zero order Taylor expansion of the amplitude:

$$\begin{array}{rcl} H(e^{i2\pi\nu}) & \simeq & H_R(\nu_0) e^{i(\phi(\nu_0)+(\nu-\nu_0)\frac{d\phi}{d\nu}(\nu_0)} \\ & \simeq & H_R(\nu_0) e^{-i2\pi(\nu_0\tau_p(\nu_0)+(\nu-\nu_0)\tau_g(\nu_0))}. \end{array}$$

Now let us consider the signal y obtained by applying filter h to the narrowband signal x: $y = h * x$. We then have

$$\begin{array}{rcl} Y(e^{i2\pi\nu}) & = & H(e^{i2\pi\nu})X(e^{i2\pi\nu}) \\ & \simeq & H_R(\nu_0) e^{-i2\pi(\nu_0\tau_p(\nu_0)+(\nu-\nu_0)\tau_g(\nu_0))}A(e^{2i\pi(\nu-\nu_0)}), \end{array}$$

therefore

$$Y(e^{i2\pi(\nu+\nu_0)}) \simeq H_R(\nu_0) e^{-i2\pi(\nu_0\tau_p(\nu_0)+\nu\tau_g(\nu_0))}A(e^{2i\pi\nu}).$$

Reverting to the time domain, we get

$$y(n) e^{-i2\pi\nu_0 n} \simeq H_R(\nu_0) e^{-i2\pi\nu_0\tau_p(\nu_0)}a(n - \tau_g(\nu_0)),$$

therefore

$$y(n) \simeq H_R(\nu_0) a(n - \tau_g(\nu_0)) e^{i2\pi\nu_0(n-\tau_p(\nu_0))}.$$

This equation proves that $y(n)$ is a narrowband signal, of same carrier frequency ν_0 as $x(n)$, of temporal envelope $H_R(\nu_0) a(n - \tau_g(\nu_0))$, which is shifted by the group delay $\tau_g(\nu_0)$ w.r.t. $x(n)$, and of initial phase which is shifted by the phase delay $\tau_p(\nu_0)$ w.r.t. $x(n)$.

We notice that the temporal waveform of the input signal x is not distorted by the filter if and only the group and phase delays are equal: $\tau_g(\nu) = \tau_p(\nu) \, \forall \nu \in \mathbb{R}$. The necessary and sufficient condition for this property to hold is that h is a *linear phase filter*, i.e. $\phi(\nu) = -2\pi\alpha\nu$ with $\alpha \in \mathbb{R}$.

2 Z-transform

The *Z-Transform* (ZT) is a very powerful tool for characterizing the properties of discrete filters.

2.1 Definition

Remember the definition of the DTFT of a discrete time signal $x \in l^1(\mathbb{Z})$: $X(e^{2i\pi\nu}) = \sum\limits_{n=-\infty}^{+\infty} x(n) e^{2i\pi\nu n}$. By introducing the complex number $z = e^{i\pi\nu}$, we get $X(z) = \sum\limits_{n=-\infty}^{+\infty} x(n) z^{-n}$. If we now apply this expression to *any* complex number, we get the Z-transform of x:

Definition 4. *The Z-transform of a discrete time signal x is defined for $z \in \mathbb{C}$ as $X(z) = \sum\limits_{n=-\infty}^{+\infty} x(n)z^{-n}$. Its domain is defined as $\mathcal{D} = (z \in \mathbb{C}/ \sum_{n=-\infty}^{+\infty} |h(n)||z|^{-n} < +\infty)$.*

We define the Z-transform of an impulse response h in the same way. In this case, $H(z)$ is often referred to as the *transfer function* of the filter. Let us study the domain of convergence of $H(z)$. We have

$$H(z) = \sum_{n=-\infty}^{+\infty} h(n) z^{-n} = \sum_{n<0} h(n) z^{-n} + \sum_{n\geq0} h(n) z^{-n} = \sum_{n>0} h(-n) z^{n} + \sum_{n\geq0} h(n) \left(\frac{1}{z}\right)^{n}.$$

TELECOM
Paris

IP PARIS

In the last member of this equality, the first series is the power series of the sequence $h(-n)$ for $n > 0$: its domain is thus a disk in $\mathbb{C}$, of radius $R_2 > 0$. The second series is the power series of the sequence $h(n)$ for $n \geq 0$: its domain is also a disk in $\mathbb{C}$. However, this second power series is applied to $\frac{1}{z}$, so its domain as a function of z is actually the complement of a disk, of radius $R_1 > 0$. Finally, the domain $\mathcal{D}$ of $H(z)$ is the intersection of those two domains: $\mathcal{D}$ is thus a ring (see Figure 1.2), in the most general sense:

- if h is a finite impulse response, then $\mathcal{D} = \mathbb{C}$ (possibly without $z = 0$ because of the terms in $\frac{1}{z}$);

- if h is causal ($h(-n) = 0$ for $n > 0$), $\mathcal{D}$ is the complement of the disk of radius R_1;

- if h is anti-causal ($h(n) = 0$ for $n \geq 0$), $\mathcal{D}$ is the disk of radius R_2;

- if e.g. $h(n) = e^{n^2}$, $\mathcal{D}$ is empty.
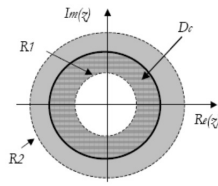
$$z = 1 \text{ at } \nu = 0$$
$$z = i \text{ at } \nu = 1/4$$

Figure 1.2: Z-transform

Finally, Proposition 3 shows that h is stable if and only if $\sum_{n=-\infty}^{+\infty} |h(n)| < +\infty$, which means that $\mathcal{D}$ contains the unit circle. In this case, the frequency response of the filter is well-defined. We note in Figure 1.2 that $\nu = 0$ corresponds to $z = 1$, $\nu = 1/4$ to $z = i$, and the *Nyquist frequency* $\nu = \frac{1}{2}$ to $z = -1$.

2.2 Basic properties

The ZT satisfies the following properties (their proof is left to the reader):

- Linearity: for any $a_1 \in \mathbb{C}$ and $a_2 \in \mathbb{C}$, the ZT of the sequence $a_1 h_1 + a_2 h_2$ is $a_1 H_1 + a_2 H_2$ and its domain $\mathcal{D}$ is such that $\mathcal{D} \supset \mathcal{D}_1 \cup \mathcal{D}_2$;

- Delay: two signals y and x are such that $y(n) = x(n - k)$ if and only if $Y(z) = z^{-k} X(z)$;

- Inversion of time direction: if $f(n) = h(-n) \; \forall n \in \mathbb{Z}$, then $F(z) = H(1/z)$ and $\mathcal{D}_F = 1/\mathcal{D}_H$;

- Convolution product: if $y = h * x$, then $Y(z) = H(z)X(z)$ and $\mathcal{D}_y \supset \mathcal{D}_h \cap \mathcal{D}_x$;

- Insertion of zeros: $Y(z) = X(z^L)$ if and only if $y(nL) = x(n) \; \forall n \in \mathbb{Z}$ and $y(m) = 0$ elsewhere;

- Inverse filter: the inverse filter h_i of a filter h is the filter such that $h * h_i = \delta_0$; it thus satisfies $H(z)H_i(z) = 1$ for $z \in \mathcal{D}_h \cap \mathcal{D}_{h_i}$.

2.3 Examples of Z-transforms

Here are a few simple examples of ZT:

- Identity filter: $h(n) = \delta_0(n) \Rightarrow H(z) = 1 \; \forall z \in \mathbb{C}$;

- Step filter: $h(n) = \mathbf{1}_{[0,+\infty[}(n) \Rightarrow H(z) = \frac{1}{1-z^{-1}} \; \forall |z| > 1$;

- Rectangular filter: $h(n) = \mathbf{1}_{[0...N-1]}(n) \Rightarrow H(z) = \frac{1-z^{-N}}{1-z^{-1}} \; \forall z \neq 0$.

The *Autoregressive* (AR) filter of order 1 (also referred as the *AR1 filter*), which will be introduced in the next sections, provides us with another fundamental example of ZT that will lead us to the notion of *recursive filters*.

2.3.1 Causal AR1 filter

Let $h(n) = \begin{pmatrix} a^n & \text{if } n \geq 0 \\ 0 & \text{if } n < 0 \end{pmatrix}$, where $a \in \mathbb{C}$. This impulse response is causal, so we already know that $\mathcal{D}$ is the complement of a disk. Then $H(z) = \sum_{n \in \mathbb{Z}} x(n)z^{-n} = \sum_{n \geq 0} a^n z^{-n} = \sum_{n \geq 0} (\frac{a}{z})^n$. This power series converges if and only if $|\frac{a}{z}| < 1$, therefore $\mathcal{D} = \{z \in \mathbb{C} / |z| > |a|\}$. Note that this filter is stable if and only if $\mathcal{D}$ contains the unit circle, which means $|a| < 1$. Then for all $z \in \mathcal{D}$, $H(z) = \frac{1}{1-\frac{a}{z}} = \frac{1}{1-az^{-1}}$.

2.3.2 Anti-causal AR1 filter

Let $h(n) = \begin{pmatrix} -a^n & \text{if } n < 0 \\ 0 & \text{if } n \geq 0 \end{pmatrix}$, where $a \in \mathbb{C}$. This impulse response is anti-causal, so we already know that $\mathcal{D}$ is a disk. Then $H(z) = \sum_{n \in \mathbb{Z}} x(n)z^{-n} = \sum_{n < 0} -a^n z^{-n} = -\sum_{n > 0} a^{-n} z^n = -\frac{z}{a} \sum_{n \geq 0} (\frac{z}{a})^n$. This power series converges if and only if $|\frac{z}{a}| < 1$, therefore $\mathcal{D} = \{z \in \mathbb{C} / |z| < |a|\}$. Note that this filter is stable if and only if $\mathcal{D}$ contains the unit circle, which means $|a| > 1$. Then for all $z \in \mathcal{D}$, $H(z) = -\frac{z}{a} \frac{1}{1-\frac{z}{a}} = \frac{1}{1-az^{-1}}$: the transfer function is the same as that of the causal AR1 filter. So the difference with the AR1 filter entirely lies in the domain of convergence.

2.3.3 Implementations of the AR1 filter

Since the causal and the anti-causal AR1 filters have the same transfer function, they will now be referred to as the two *implementations* of the same filter, that we will simply call the *AR1 filter*.

Indeed, in both cases we have $H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1-az^{-1}}$. Therefore $(1 - az^{-1})Y(z) = X(z)$, which in the time domain can be rewritten $y(n) - ay(n - 1) = x(n)$. This equation is called the *Input/Output* (I/O) relationship of the filter. It can be used to implement the filter in two different ways (note that in practice the filter cannot be implemented by means of a convolution product, because of its *Infinite Impulse Response* (IIR)):

- forward recursion: for increasing times n, $y(n) = ay(n - 1) + x(n)$. This is the causal implementation introduced in Section 2.3.1. Indeed, this implementation is numerically stable if and only if $|a| < 1$, since any error in the computation $y(n - 1)$ is reduced by a factor $|a|$ in $y(n)$.

- backward recursion: for decreasing times n, $y(n) = \frac{y(n+1)-x(n+1)}{a}$. This is the anti-causal implementation introduced in Section 2.3.2. Indeed, this implementation is numerically stable if and only if $|a| > 1$, since any error in the computation $y(n + 1)$ is reduced by a factor $\frac{1}{|a|}$ in $y(n)$.

These observations are summarized in Table 1.1.

I/O relationship	$y(n) - ay(n - 1) = x(n)$	
Transfer function	$H(z) = \frac{1}{1-az^{-1}}$	
Implementation	$y(n) = ay(n - 1) + x(n)$	$y(n) = \frac{y(n+1)-x(n+1)}{a}$
IR $(x(n)=\delta_0(n))$	$h(n) = a^n 1_{\{n \geq 0\}}$	$h(n) = -a^n 1_{\{n < 0\}}$
Domain $\mathcal{D}$	$\{z \in \mathbb{C} / |z| > |a|\}$	$\{z \in \mathbb{C} / |z| < |a|\}$
Properties	causal, stable if $|a|<1$	anti-causal, stable if $|a|>1$

Table 1.1: Autoregressive filter of order 1

The different filter implementations, that we have here studied in the particular case of the AR1 filter, will be extended to the general context of *recursive filters* in Section 3.

3 Recursive filters

Recursive filters generalize the notion of AR1 filters that was summarized in Section 2.3.3.

Roland Badeau   roland.badeau@telecom-paris.fr

TELECOM
Paris

IP PARIS

3.1 Definition

A recursive filter is defined by its I/O relationship:

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k), \tag{1.4}$$

which relates the samples of the output signal y to those of the input signal x. Note that the I/O relationship actually defines a filter:

- the output samples are linear functions of the input samples;

- delaying the input signal is equivalent to delaying the output signal by the same delay.

Therefore we know that there is an impulse response h such that $y = h * x$ (but in the general case, h cannot be expressed as a simple function of the coefficients a_k and b_k, as we will see later).

The I/O relationship admits several implementations. For instance, the causal implementation is

$$y(n) = \frac{1}{a_0} \left[\sum_{k=0}^{M} b_k x(n-k) - \sum_{k=1}^{N} a_k y(n-k) \right].$$

Contrary to the impulse response, the transfer function can be expressed as a simple function of the coefficients a_k and b_k, as shown in the following proposition:

Proposition 4. *The transfer function of a recursive filter defined by its I/O relationship* (1.4) *can be written as*

$$H(z) = \frac{\sum\limits_{k=0}^{M} b_k z^{-k}}{\sum\limits_{k=0}^{N} a_k z^{-k}} = \frac{b_0}{a_0} \frac{\prod\limits_{k=1}^{M} (1 - c_k z^{-1})}{\prod\limits_{k=1}^{N} (1 - d_k z^{-1})}, \tag{1.5}$$

where the coefficients c_k are referred to as the zeros *of the filter, and the coefficients d_k are referred to as its* poles.

Proof. Applying the Z-transform to the I/O relationship yields $\sum\limits_{k=0}^{N} a_k Y(z) z^{-k} = \sum\limits_{k=0}^{M} b_k X(z) z^{-k}$, or equivalently $\left(\sum\limits_{k=0}^{N} a_k z^{-k} \right) Y(z) = \left(\sum\limits_{k=0}^{M} b_k z^{-k} \right) X(z)$. Therefore $H(z) = \frac{Y(z)}{X(z)} = \frac{\sum\limits_{k=0}^{M} b_k z^{-k}}{\sum\limits_{k=0}^{N} a_k z^{-k}}$. Since the numerator and denominator are polynomials of z^{-1}, they can be factorized as products of monomials, therefore $H(z) == \frac{b_0}{a_0} \frac{\prod\limits_{k=1}^{M}(1-c_k z^{-1})}{\prod\limits_{k=1}^{N}(1-d_k z^{-1})}$. $\qquad\square$

Based on Proposition 4, we can particularize the general properties of the domain $\mathcal{D}$ of the transfer function, that we have already investigated in Section 2.1, to the particular case of recursive filters:

- $\mathcal{D}$ is a ring bounded by two poles, with no pole inside;

- the causal implementation is unique, and its domain $\mathcal{D}$ is the complement of the disk whose radius is equal to the maximum modulus of the poles;

- there is a stable implementation, which is unique, if and only if there is no pole on the unit circle; in this case its domain $\mathcal{D}$ is the largest ring containing the unit circle and no pole;

- there is a unique *stable and causal* implementation if and only if all poles are strictly inside the unit circle;

- if all coefficients a_k and b_k are real, then the poles and zeros are either real, or form conjugate pairs.

3.2 Examples of recursive filters

There are two particular classes of recursive filters which are widely employed in signal processing: the AR filters and the *Finite Impulse Response* (FIR) filters.

3.2.1 Autoregressive filters

AR filters are recursive filters such that $M = 0$ (so that they have no zero). In this case, we say that h is an AR filter of order N. The I/O relationship is simplified as

$$y(n) = \frac{b_0}{a_0} x(n) - \sum_{k=1}^{N} \frac{a_k}{a_0} y(n - k).$$

AR filters are usually employed in speech processing, because they correspond to a physical model of the vocal tract.

3.2.2 Finite impulse response filters

FIR filters are recursive filters such that $N = 0$ (so that they have no pole). In this case, we say that h is an FIR filter of length M. Their impulse response admits a simple expression as a function of the coefficients a_k and b_k:

$$h(n) = \left(\begin{array}{ll} \frac{b_n}{a_0} & \text{if } n = 0 \ldots M, \\ 0 & \text{otherwise.} \end{array} \right.$$

FIR filters are widely employed in signal processing because they are both causal and unconditionally stable. In addition, if the IR is symmetric or anti-symmetric, the phase is linear (this property will be proved in Chapter 2, Section 2).

3.2.3 Exercise: computation of the impulse response from the Z-transform

Let $H(z) = \frac{1}{1 - \frac{5}{2} z^{-1} + z^{-2}}$. Prove that this transfer function has three domains of convergence, and that for each of these domain, the impulse response is as follows:

- Domain $|z| < 1/2$ (anti-causal implementation): $h(n) = \left(\begin{array}{ll} \frac{2}{3} \left(\frac{1}{2} \right)^{n+1} - \frac{2}{3} 2^{n+1} & \text{if } n \leq -1 \\ 0 & \text{if } n > -1 \end{array} \right.$;

- Domain $1/2 < |z| < 2$ (stable implementation): $h(n) = -\frac{1}{3} \left(\frac{1}{2} \right)^{|n|}$;

- Domain $|z| > 2$ (causal implementation): $h(n) = \left(\begin{array}{ll} -\frac{2}{3} \left(\frac{1}{2} \right)^{n+1} + \frac{2}{3} 2^{n+1} & \text{if } n \geq 0 \\ 0 & \text{if } n < 0 \end{array} \right.$.

We remind you the methodology that will allow you to compute the IR from the ZT:

1. If necessary, extract from $H(z)$ a term in z^n, so that the numerator and denominator of $H(z)$ only contain negative powers of z plus a constant;

2. Factorize the numerator and denominator as products of monomials of z^{-1};

3. Compute the partial fraction decomposition of $H(z)$ as a function of z^{-1};

4. Expand this partial fraction decomposition in power series:

   - of z^{-1} if we compute the causal IR, or if we compute the stable IR and if the pole is inside the unit circle ;
   - of z if we compute the anti-causal IR, or if we compute the stable IR and if the pole is outside the unit circle.

5. Identify the coefficients of the resulting expression with those of $H(z) = \sum_{n \in \mathbb{Z}} h(n) z^{-n}$.

3.3 Geometric interpretation of the frequency response

The shape of the FR of a recursive filter can be easily deduced from the expression of its transfer function. For example, let us consider the causal AR2 filter:

$$H(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{1}{(1 - p z^{-1})(1 - p^* z^{-1})}, \tag{1.6}$$

where we have assumed that the coefficients $a_1 \in \mathbb{R}$ and $a_2 \in \mathbb{R}$ are such that the two poles form a conjugate pair.

Then let M be the point in the Z-plane of affix z, let P be the point of affix p, and let Q be the point of affix p^*, as represented in Figure 1.3a. Then equation (1.6) implies

$$|H(z)| = \frac{1}{PM \times QM} \tag{1.7}$$

and $\arg H(z) = 2 \arg(OM) - \arg(PM) - \arg(QM)$.

Applying equation (1.7) to $z = e^{2i\pi\nu}$ permits us to plot the magnitude of the frequency response $|H(e^{2i\pi\nu})|$, as in Figure 1.3b. Indeed, when ν goes from 0 to $+\frac{1}{2}$, M travels over the unit circle counterclockwise, from the point of affix $z = 1$ to the same point of affix $z = -1$. In particular, when M is closest to P, the distance PM is small, so $|H(z)|$ is high. Indeed, we observe a resonance in Figure 1.3b, at the resulting frequency $\nu = \frac{\arg p}{2\pi}$.
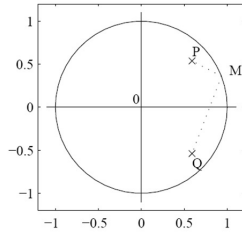
(a) Z-plane
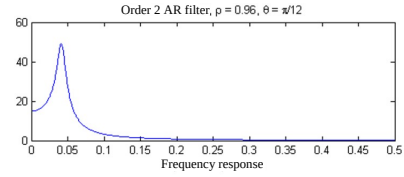
(b) Frequency response

Figure 1.3: Geometric interpretation of the FR

3.4 Spectral factorization

We have already seen in equation (1.5) that the transfer function of a recursive filter can be factorized as

$$H(z) = \frac{b_0}{a_0} \frac{\prod\limits_{k=1}^{M}(1 - c_k z^{-1})}{\prod\limits_{k=1}^{N}(1 - d_k z^{-1})}.$$

On the unit circle, *i.e.* when $z = e^{2i\pi\nu}$, the squared magnitude of the frequency response can thus be written as

$$|H(e^{2i\pi\nu})|^2 = H(z)H(1/z^*)^* = \left|\frac{b_0}{a_0}\right|^2 \frac{\prod\limits_{k=1}^{M}(1 - c_k z^{-1})(1 - c_k^* z)}{\prod\limits_{k=1}^{N}(1 - d_k z^{-1})(1 - d_k^* z)}.$$

We note that this expression would be unchanged (up to a multiplicative factor) if any zero c_k was replaced by $1/c_k^*$, or if any pole d_k was replaced by $1/d_k^*$, leading to a frequency response of same magnitude (up to a multiplicative factor). This remark is important because we already know that when all poles are strictly inside the unit circle, then the filter admits a causal and stable implementation. Therefore, given any recursive filter, we can move any

pole which would lie outside the unit circle into the unit circle, so as to get a causal and stable implementation, without altering the magnitude of the frequency response.

In the same way, it is also possible to move any zero which would lie outside the unit circle into the unit circle without altering the magnitude of the frequency response, so that the inverse of the resulting filter also admits a causal and stable implementation. When all the poles and zeros are strictly inside the unit circle, then we say that the filter admits a *minimal phase* implementation. In the following sections, we will further develop this notion of minimal phase filters.

3.4.1 All-pass filters

First, we need to introduce a particular class of recursive filters: the *all-pass* filters.

Definition 5 (All-pass filters). *An all-pass filter is a recursive filter whose transfer function can be written in the form $H(z) = \prod\limits_{k=1}^{N} \frac{z^{-1}-c_k^*}{1-c_k z^{-1}}$ where $|c_k| < 1 \ \forall k \in \{1 \dots N\}$ for $N \in \mathbb{N}^*$.*

If $|z| = 1$, we note that $H(z) = \prod\limits_{k=1}^{N} z^{-1} \frac{1-c_k^* z}{1-c_k z^{-1}}$, therefore $|H(z)| = \prod\limits_{k=1}^{N} |z^{-1}| \left| \frac{1-c_k^* z}{1-c_k z^{-1}} \right| = 1$. Therefore, if x and y respectively denote the input and output signals, we have $|Y(e^{2i\pi\nu})| = |X(e^{2i\pi\nu})| \ \forall \nu \in \mathbb{R}$: all the frequencies of the input signal pass through the filter. This is why it is called an *all-pass* filter. This filter also satisfies other interesting properties:

Proposition 5. *The filter $H(z)$ introduced in Definition 5 admits a causal and stable implementation. Moreover, if $y = h * x$, then:*

- $\sum\limits_{n=-\infty}^{+\infty} |x(n)|^2 = \sum\limits_{n=-\infty}^{+\infty} |y(n)|^2$;

- $\forall N \in \mathbb{Z}, \ \sum\limits_{n=-\infty}^{N} |x(n)|^2 \geq \sum\limits_{n=-\infty}^{N} |y(n)|^2$.

Proof. Since all the poles are inside the unit circle, the filter admits a causal and stable implementation.

Moreover, let $y = h * x$. Then $|Y(e^{2i\pi\nu})| = |H(e^{2i\pi\nu})||X(e^{2i\pi\nu})| = |X(e^{2i\pi\nu})|$, which proves that $\int_0^1 |Y(e^{2i\pi\nu})|^2 d\nu = \int_0^1 |X(e^{2i\pi\nu})|^2 d\nu$. Parseval's equality permits us to conclude that $\sum\limits_{n=-\infty}^{+\infty} |x(n)|^2 = \sum\limits_{n=-\infty}^{+\infty} |y(n)|^2$. Now let $x_N(n) = x(n) \times \mathbf{1}_{]-\infty,N]}(n)$ and $y_N = h * x_N$. We have $\sum\limits_{n=-\infty}^{N} |x(n)|^2 = \sum\limits_{n=-\infty}^{+\infty} |x_N(n)|^2 = \sum\limits_{n=-\infty}^{+\infty} |y_N(n)|^2 = \sum\limits_{n=-\infty}^{N} |y_N(n)|^2 + \sum\limits_{n=N+1}^{+\infty} |y_N(n)|^2$. However, since the filter is causal and $x_N(n) = x(n) \ \forall n \in]-\infty, N]$, then $y_N(n) = y(n) \ \forall n \in]-\infty, N]$. Therefore we get $\sum\limits_{n=-\infty}^{N} |x(n)|^2 = \sum\limits_{n=-\infty}^{N} |y(n)|^2 + \sum\limits_{n=N+1}^{+\infty} |y_N(n)|^2$. Since $\sum\limits_{n=N+1}^{+\infty} |y_N(n)|^2 \geq 0$, we conclude that $\sum\limits_{n=-\infty}^{N} |x(n)|^2 \geq \sum\limits_{n=-\infty}^{N} |y(n)|^2$. □

3.4.2 Minimal phase filters

We can know introduce the definition of a minimal phase filter:

Definition 6 (Minimal phase filter). *A minimal phase filter is a recursive filter whose poles and zeros are all strictly inside the unit circle.*

The proof of the following proposition was sketched at the beginning of Section 3.4:

Proposition 6. *Any causal and stable recursive filter is the product of an all-pass filter and a minimal phase filter.*

An example of factorization as described in Proposition 6 is illustrated in Figure 1.4.
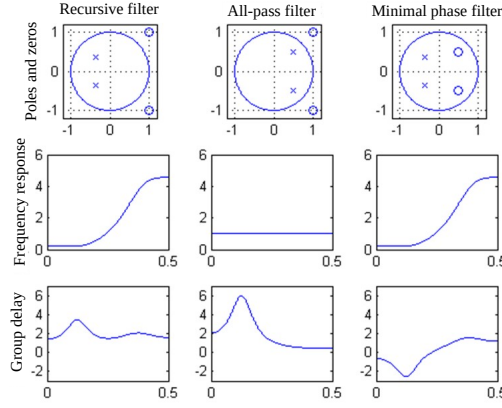
TELECOM
Paris

IP PARIS

Figure 1.4: Spectral factorization

Minimal phase filters satisfy several interesting properties:

Proposition 7. *Let g be a causal and stable recursive filter, and let g_m be the unique minimal phase filter having the same frequency response magnitude as g (i.e. $\forall \nu \in \mathbb{R}$, $|G(e^{2i\pi\nu})| = |G_m(e^{2i\pi\nu})|$). Let x and y respectively denote the input and output signals of filter g (i.e. $y = g * x$). In the same way, let $y_m = g_m * x$. Then:*

- *The inverse filter of g_m admits a causal and stable implementation;*

- $\forall N \in \mathbb{Z}$, $\sum\limits_{n=-\infty}^{N} |y_m(n)|^2 \geq \sum\limits_{n=-\infty}^{N} |y(n)|^2$;

- *at all frequencies $\nu_0 \in \mathbb{R}$, g_m has a lower group delay $\tau_{g_m}(\nu_0)$ than g.*

Proof. Since all the zeros of g_m are inside the unit circle, so are all the poles of g_m^{-1}. Therefore g_m^{-1} admits a causal and stable implementation.

Moreover, let h be the all-pass filter such that $g = h * g_m$, whose existence was proved in Proposition 6. Then $y = g * x = (h * g_m) * x = h * (g_m * x) = h * y_m$. Therefore Proposition 5 proves that $\sum\limits_{n=-\infty}^{N} |y_m(n)|^2 \geq \sum\limits_{n=-\infty}^{N} |y(n)|^2$.

Regarding the last assertion of the proposition, let us consider one zero c_1 of the transfer function $G(z)$. Let this zero be located inside the unit circle (*i.e.* $|c_1| < 1$), and let us see how the group delay is affected. First, the polar decomposition of c_1 yields

$$c_1 = |c_1| e^{i\theta_{c_1}} \text{ where } \theta_{c_1} = \arg(c_1).$$

Noting that the zero c_1 appears in the factor $1 - c_1 z^{-1}$ of the transfer function, the phase of this term when $z = e^{i2\pi\nu}$ is such that

$$
\begin{aligned}
\phi_{c_1}(\nu) &= \arg\left(1 - c_1 e^{-i2\pi\nu}\right) \\
&= \arg\left(1 - |c_1| e^{i\theta_{c_1}} e^{-i2\pi\nu}\right) \\
&= \arg\left(1 - |c_1| e^{-i(2\pi\nu - \theta_{c_1})}\right) \\
&= \arg\left((1 - |c_1|\cos(2\pi\nu - \theta_{c_1})) + i\left(|c_1|\sin(2\pi\nu - \theta_{c_1})\right)\right) \\
&= \arg\left(\left(\tfrac{1}{|c_1|} - \cos(2\pi\nu - \theta_{c_1})\right) + i\left(\sin(2\pi\nu - \theta_{c_1})\right)\right).
\end{aligned}
$$

Besides, we note that for any complex function of the form $\psi(\nu) = \alpha(\nu) + i\beta(\nu)$ with $\alpha(\nu) \in \mathbb{R}$ and $\beta(\nu) \in \mathbb{R}$, we have $\dfrac{d \arg \psi}{d\nu} = \dfrac{\frac{d\alpha}{d\nu}\beta(\nu) - \alpha(\nu)\frac{d\beta}{d\nu}}{\alpha^2(\nu) + \beta^2(\nu)}$.

Roland Badeau   `roland.badeau@telecom-paris.fr`

Therefore the contribution of $\phi_{c_1}(\nu)$ to the group delay is:

$$-\frac{1}{2\pi}\frac{d\phi_{c_1}(\nu)}{d\nu} = \frac{\sin^2(2\pi\nu - \theta_{c_1}) + \cos^2(2\pi\nu - \theta_{c_1}) - \frac{1}{|c_1|}\cos(2\pi\nu - \theta_{c_1})}{\sin^2(2\pi\nu - \theta_{c_1}) + \cos^2(2\pi\nu - \theta_{c_1}) + \frac{1}{|c_1|^2} - \frac{2}{|c_1|}\cos(2\pi\nu - \theta_{c_1})}$$

$$= \frac{|c_1| - \cos(2\pi\nu - \theta_{c_1})}{|c_1| + \frac{1}{|c_1|} - 2\cos(2\pi\nu - \theta_{c_1})}.$$

The denominator and θ_{c_1} are invariant to reflecting the zero c_1 outside of the unit circle, *i.e.*, replacing c_1 with $\frac{1}{c_1^*}$. However, by reflecting c_1 outside of the unit circle, we increase the magnitude of $|c_1|$ in the numerator. Thus, having c_1 inside the unit circle minimizes the group delay induced by the factor $1 - c_1 z^{-1}$. We can extend this result to the general case of more than one zero, since the phases of all multiplicative factors of the form $1 - c_k z^{-1}$ are additive, *i.e.*, for a transfer function with N zeros, *i.e.*

$$\arg\left(\prod_{k=1}^{N}\left(1 - c_k z^{-1}\right)\right) = \sum_{k=1}^{N}\arg\left(1 - c_k z^{-1}\right).$$

Consequently, a minimum phase filter with all zeros inside the unit circle minimizes the group delay, since the group delay of each individual zero is minimized. $\square$

The second property shown in Proposition 7 is illustrated on an example in Figure 1.5:

- the top figure displays the impulse response of filter g;

- the middle figure displays the impulse response of the minimum phase filter g_m;

- when $x = \delta_0$ (thus $y = g$ and $y_m = g_m$), the bottom figure compares the accumulated energy of the output signal $\sum_{n=-\infty}^{N}|y_m(n)|^2$ (red curve) w.r.t. that of the input signal $\sum_{n=-\infty}^{N}|y(n)|^2$ (blue curve);
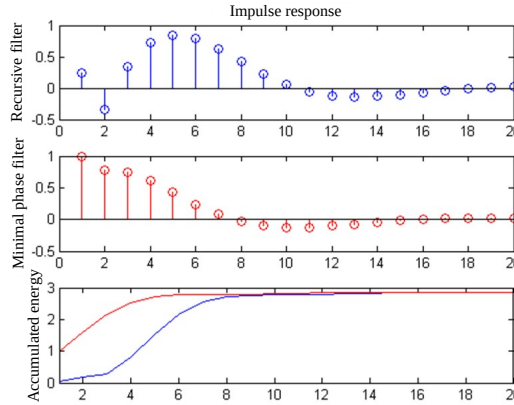

Figure 1.5: Properties of minimal phase filters

The third property shown in Proposition 7 is illustrated on an example in the bottom row of Figure 1.4.

Chapter 2

Synthesis of digital filters

This chapter is mostly extracted from a course handout by Bertrand David, translated in English.

The choice of using a recursive or a FIR filter usually depends on the context. The main advantage of FIR filters is the possibility of synthesizing them with an exactly linear phase in frequency, which allows the conservation of waveforms. However, in the case of selective filters, they often require a computational complexity (*i.e.* a number of additions and multiplications) much greater than that of a recursive filter to achieve the same function.

Another advantage of FIR filters is their smaller sensitivity to rounding errors, due to their non-recursive nature. These errors therefore do not propagate. A recursive filter is also more sensitive to the finite precision with which its parameters are represented, which can, in the case of poles near the unit circle, cause instabilities.

1 Specification of a digital filter

A common way to specify a filter is to give one template, *i.e.* to specify the maximum allowable limits for the modulus of its frequency response. We consider for example a low-pass filter, which will be easy to generalize to other types of filters. These limits are then given in terms of maximum ripple δ_1 in the passband and δ_2 in the stopband, and of the width of the transition band. δ_1 and δ_2 are frequently specified in dB. The transition band, which is all the more narrow that the selectivity of the filter is high, lies between a lower bound ν_c and an upper bound ν_a. A template of this type is provided in Figure 2.1. Note that this template is normalized to 1 in the passband, so a static gain value may be given as a supplement.
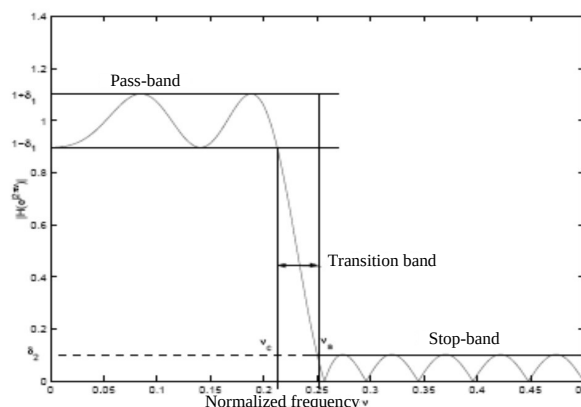

Figure 2.1: Template of a normalized digital low-pass filter, with 10 dB of attenuation in the stopband ($\delta_2 = 0.1$), and 0.9 dB of ripple in the passband ($\delta_1 = 0.1$).

Roland Badeau   roland.badeau@telecom-paris.fr

2 Linear phase FIR filters

The impulse response of such a filter is composed of a finite number N of non-zero coefficients. We consider the case of **real and causal** filters and these coefficients are then:

$$h(n) \in \mathbb{R}, \quad n = 0...N - 1.$$

We will have $h(n) = 0$ for all $n \notin [0, \ N - 1]$.

The frequency response of a linear phase filter is written

$$H(e^{i2\pi\nu}) = e^{i2\pi(\beta - \alpha\nu)} H_R(\nu), \quad \alpha \in \mathbb{R}, \quad \beta \in [0\,0.5]$$

where H_R is a real function of the continuous real variable ν. This is why we can restrict the variation interval of β to $[0\,0.5]$, without altering the generality of the expression.

2.1 Interest

These filters have the advantage of a phase delay constant with the frequency. If the input signal has its frequency support in the passband, the waveform of the output signal is similar to that of the input signal. This result is analogous to that encountered in wave physics when the wave propagation occurs without dispersion (the phase velocity is independent from the frequency): the components travel without getting out of phase with one another.

Indeed, let us assume that $X(e^{i2\pi\nu})$ and $H(e^{i2\pi\nu})$ are summable in the interval $[-0.5\ 0.5]$. In this interval, the support of X reduces to a compact set $B \in [-0.5\ 0.5]$ which satisfies

$$|H(e^{i2\pi\nu})| = 1, \quad \nu \in B.$$

Let $H(e^{i2\pi\nu}) = c$ where c is a complex number of modulus 1. In this way, we can define the output sequence of the filter in the form:

$$y(n) = ce^{i2\pi\beta} \int_B X(e^{i2\pi\nu}) e^{i2\pi\nu(n-\alpha)} d\nu. \tag{2.1}$$

We define the continuous time signal $x_a(t)$ such that $x(n)$ is obtained by sampling x_a at sample rate 1. That is to say, by using the interpolation formula:

$$x_a(t) = \sum_{\mathbb{Z}} x(n) \operatorname{sinc}(t - n).$$

The existence of x_a is assured because its Fourier transform is $X_a(\nu) = X(e^{i2\pi\nu}), \quad \nu \in [-0.5\ 0.5]$ and $X_a(\nu) = 0$ elsewhere. We then deduce from equation (2.1) the relationship:

$$y(n) = ce^{i2\pi\beta} x_a(n - \alpha)$$

2.2 Impulse response symmetries

To find the shape of the impulse response of this type of filter, we will use two properties: the Hermitian symmetry of the frequency response $H(e^{i2\pi\nu})$ and its 1-periodicity.

The 1-periodicity of $H(e^{i2\pi\nu})$ leads to half-integer values of α:

$$\alpha = p/2, \quad p \in \mathbb{Z}$$

and to the 2-periodicity of the real function $H_R(\nu)$.

Proof. The equality $H(e^{i2\pi\nu}) = H(e^{i2\pi(\nu+1)})$ leads to

$$H_R(\nu) = e^{-i2\pi\alpha} H_R(\nu + 1),$$

which leads to half-integer values of α, taking into account the property $H_R \in \mathbb{R}$. We then have $H_R(\nu + 1) = (-1)^p H_R(\nu)$ *i.e.* $H_R(\nu)$ is 2-periodic. □

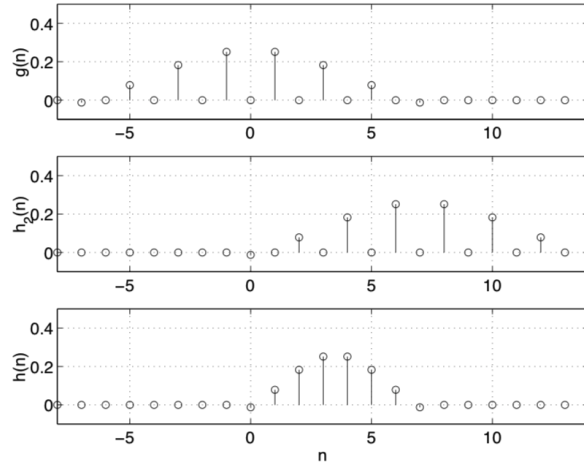**Contexte académique } sans modifications**

TELECOM
Paris

IP PARIS

Figure 2.2: Impulse response of an FIR filter for $N = 8$, $h(n) = h(N - 1 - n)$: the sequence $h(n)$ is obtained from $g(n)$ by a shift of $N - 1$ samples and a decimation by factor 2

$H_R \in L^1_{\mathcal{P}}(2)$ allows us to search the sequence obtained by inverse Fourier transform of the 1-periodic function $H_R(2\nu)$, that is:

$$g(n) = \int_{[0,1]} H_R(2\nu)e^{i2\pi\nu n}d\nu.$$

The **Hermitian symmetry** of the frequency response leads to possible values of β such as:

$$d \triangleq e^{i2\pi\beta} = 1 \text{ or } i,$$

which leads to H_R being even or odd, and therefore the Fourier transform of $g(n)$, $G(e^{i2\pi\nu})$ taken as a function of the variable ν, being also even or odd:

$$G(e^{i2\pi\nu}) = \pm G(e^{-i2\pi\nu})$$

In the whole domain of convergence of $H(z)$, that is to say the whole complex plane without the origin point, we can then write

$$H(z^2) = d\, z^{-p}\, G(z).$$

This last relation can be read simply. $H(z^2)$ is the Z-transform of the sequence $h_2(n)$, obtained from $h(n)$ by interpolation by a factor 2; $h_2(n)$ is obtained from $g(n)$ by an offset of p samples and a multiplication by d. We can thus build 4 types of filters $h(n)$ from 4 possible choices for the form of $g(n)$, depending on the value of d and the parity of N.

The choice of the value d implies a type of symmetry for G and therefore for $g(n)$. For example the choice $d = 1$ leads to a real and even function $G(e^{i2\pi\nu})$. $g(n)$ will be real and even and therefore $h(N - 1 - k) = h(k)$. We can then distinguish between the two cases of N being even or odd. In all cases, the corresponding length of $g(n)$ is $2N - 1$, which is odd. In order to obtain a causal sequence, we must shift $g(n)$ by $p = N - 1$ samples to the right, which leads to $\alpha = (N - 1)/2$.

The four types of linear phase FIR filters therefore correspond to the following cases:

- $d = 1$, N even or odd, $h(n) = h(N - 1 - n)$;

- $d = i$, N even or odd, $h(n) = -h(N - 1 - n)$.

An example corresponding to the type $d = 1$ and N even is shown in Figure 2.2.

Each type of linear phase FIR filter, which the traditional signal processing nomenclature refers to as type I (symmetric, N odd), type II (symmetric, N even), type III (antisymmetric, N odd) and type IV (antisymmetric, N

even), exhibits characteristic properties which make it suitable for specific uses, or on the contrary which limit its use.

Types I and II can fit the usual filters including a passband and a stopband (low-pass, high-pass, band-pass). However, the type II frequency response has a zero at $\nu = 0.5$, so it will not be able to fit a high-pass filter.

The frequency response of type III and IV filters takes the particular shape:

$$H(e^{i2\pi\nu}) = ie^{-i2\pi\frac{N-1}{2}}H_R(\nu).$$

They are mainly used to make *Hilbert transforms* and *differentiators*.
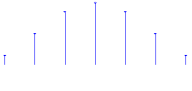
Table 2.1 summarizes the various properties stated above.

Type	Example of $h(n)$	Properties	Functions
Type I N odd symmetric		-	Low-pass High-pass Band-pass
Type II N even symmetric		$H(-1) = 0$	Low-pass, Band-pass
Type III N odd antisym.		$H(1) =$ $H(-1) = 0$	Differentiator, Hilbert Transform, Band-pass
Type IV N even antisym.		$H(1) = 0$	Differentiator, Hilbert Transform, High-pass

Table 2.1: The 4 types of FIR filters

2.3   Position of the zeros

Let z_0 be a zero of $H(z)$. First, since $h(n)$ is a real impulse response, z_0^* is also a zero of $H(z)$. Second, assuming that $z_0 \neq 0$, since $h(n)$ is either symmetric or antisymmetric, we have $H(\frac{1}{z_0}) = \sum_{n=0}^{N-1} h(n)(\frac{1}{z_0})^{-n} = \sum_{m=0}^{N-1} h(N-1-m)(\frac{1}{z_0})^{-N+1+m} = \pm z_0^{N-1} \sum_{m=0}^{N-1} h(m)z_0^{-m} = \pm z_0^{N-1}H(z_0) = 0$ with $m = N-1-n$.

Therefore, if z_0 be a zero of $H(z)$, then z_0^*, $\frac{1}{z_0}$ and $\frac{1}{z_0^*}$ are also zeros of $H(z)$. This property is illustrated in Figure 2.3, where we distinguish the cases $(|z_0| \neq 1, z_0 \notin \mathbb{R})$ (Figure 2.3a), $(|z_0| = 1, z_0 \notin \mathbb{R})$ (Figure 2.3b), $(|z_0| \neq 1, z_0 \in \mathbb{R})$ (Figure 2.3c), and $(|z_0| = 1, z_0 \in \mathbb{R})$ (Figure 2.3d).

2.4   Specialized filters

2.4.1   Hilbert transform

These filters perform an approximation of the Hilbert transform, which gives access to the analytic signal. The analytic signal is the complex signal associated with a real signal $x(n)$. Its Fourier transform is simply obtained from $X(e^{2\pi\nu})$ by canceling the "negative frequencies" part of the spectrum, *i.e.* by multiplying $X(e^{2\pi\nu})$ by the window $2 \times \mathbf{1}_{[0.5]}(\nu)$, periodized at rate 1. The type III and IV filters are suitable for this use because the frequency response contains the phase factor $e^{i\pi/2}$ ($d = i$): filtering a real sequence $x(n)$ provides the sequence $x_h(n)$ such that the analytic signal associated with $x(n)$ is simply:

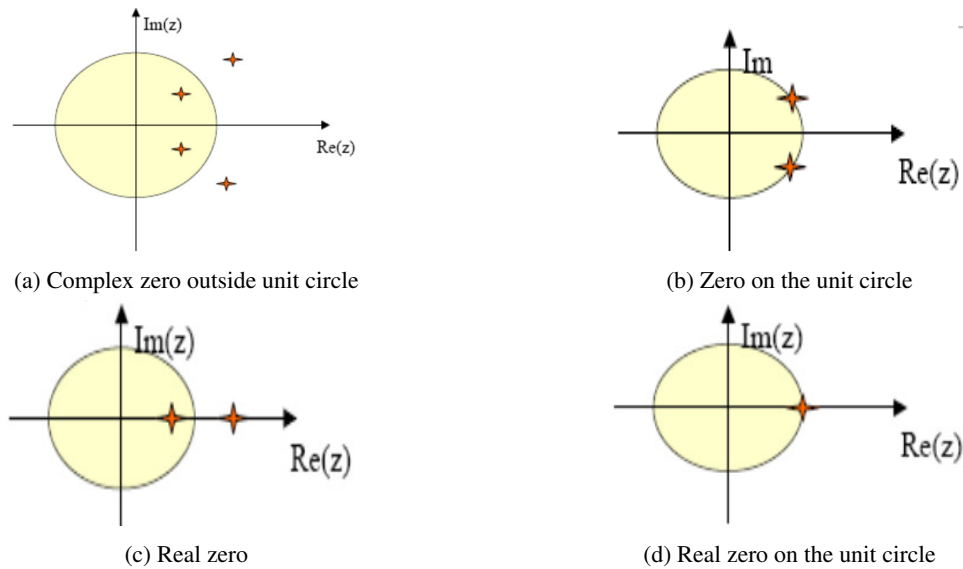$$x_a(n) = x(n) + jx_h(n), \quad x_h(n) = \{h * x\}(n).$$

TELECOM
Paris

IP PARIS

(a) Complex zero outside unit circle

(b) Zero on the unit circle

(c) Real zero

(d) Real zero on the unit circle

Figure 2.3: Position of the zeros

This result can be translated in the frequency domain as

$$H_a(e^{2\pi v}) = 1 + jH(e^{2\pi v})$$

where H_a is the frequency response of the filter that outputs x_a to the input x; with $H(e^{2\pi v}) = -i$ for $v \in [0\ 0.5]$ and $H(e^{2\pi v}) = i$ for $v \in [-0.5\ 0]$.

In practice. The MATLAB code below performs the Hilbert transform as described above.

```
Ne = 150; % number of samples
N = 61 ; % filter length
n = 0:Ne-1; % discrete time vector
nu0 = 0.05; % fundamental frequency of the input vector
x = cos(2*pi*nu0*n) ; % input vector
h = firpm(N-1,2*[.01 .25 .3 .5],[1 1 0 0],[1 10],'h'); % filter synthesis
% by Remez's exchange algorithm
xd = [zeros(1,(N-1)/2) x(1:end-(N-1)/2)] ; % delayed version of x
xh = filter(h,1,x); % Hilbert filtering
xa = xd+i*xh; % analytic signal
```

Figure 2.4 shows the obtained result and the frequency response of the Hilbert filter (here we have chosen a band-pass filter to show that the Hilbert transform can be performed in a selected frequency band). The filtering is operated on the sinusoidal input signal shifted by $(N - 1)/2$ samples to compensate for the filter delay, which makes the oddness of N essential.

In other respects, the filter h being of length N and causal, the "steady state" of the filter is only reached at $y(N - 1)$ (to be convinced of this, one can represent the achieved convolution). The obtained result is $y(n) \approx \sin(2\pi v_0 n)$. The imperfections are related to the transient of the filter ($N - 1$ points) and to that of the frequency response (ripples in the passband in particular). These imperfections are hardly visible on the graph presented here due to the order of the filter, which is sufficient to ensure a low ripple rate in the passband.
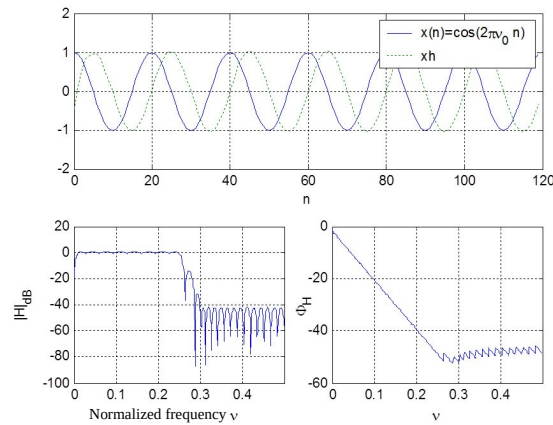
Figure 2.4: Hilbert filtering of a sinusoidal sequence $x(n)$ of frequency $\nu_0 = 0.05$

2.4.2 Differentiators

Due to the factor $d = i$, the type III and IV filters can also be synthesized to approximate the frequency response $H(e^{i2\pi\nu}) = i2\pi\nu$, that is to say the temporal derivation operation. Figure 2.5 shows an example of such a filter, in the band $\nu \in [0.1\ 0.3]$, with the command:

```
N = 21; % filter length
h = firpm(N-1,[0 0.45]*2,[0 2*pi*.45],'d');
```
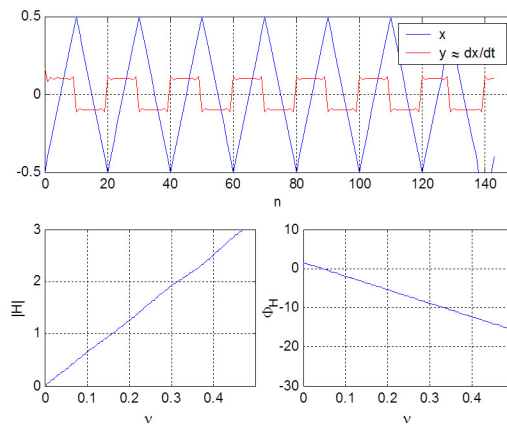

Figure 2.5: Differentiator filter of order 20 applied to a triangle sequence

In practice. To correct the delay related to filtering, we are led to use an odd filter length N. The order of the filter is even and it is therefore a type III filter. These filters feature a zero in the frequency response at $\nu = 0.5$. This is why the highest frequency of the passband is fixed at $0.45 < 0.5$. We also notice the Gibbs phenomenon caused by transitions of the square sequence.

3   FIR synthesis methods

3.1   Window method

We consider an ideal frequency response, such as that given as $H_i(e^{i2\pi\nu}) = 1$ for $\nu \in [-\nu_c + \nu_c]$ and 0 elsewhere (modulo 1). The impulse and frequency responses of this ideal low-pass filter are represented in Figure 2.6.
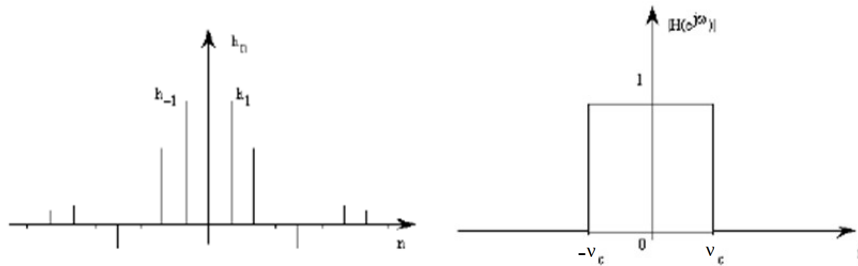

Figure 2.6: Impulse and frequency responses of the ideal low-pass filter

If a closed-form expression of the impulse response is known, we construct a finite impulse response by multiplication of this response by a finite window. We thus obtain an FIR filter that we can make causal by a temporal shift of the obtained response.

In the example cited above the closed-form expression is simply written (by inverse Fourier transform of H_i):

$$h_i(n) = 2\nu_c \, \text{sinc}(2\nu_c n).$$

We can then construct a finite impulse response by means of a window $w(n)$ with finite temporal support:

$$h(n) = w(n)h_i(n).$$

The spectral result obtained in this way is a convolution of the Fourier transform of the ideal response and the Fourier transform of the window, that is:

$$H(e^{i2\pi\nu}) = \int_{-0.5}^{+0.5} W(e^{i2\pi\gamma})H_i(e^{i2\pi(\nu-\gamma)})d\gamma.$$
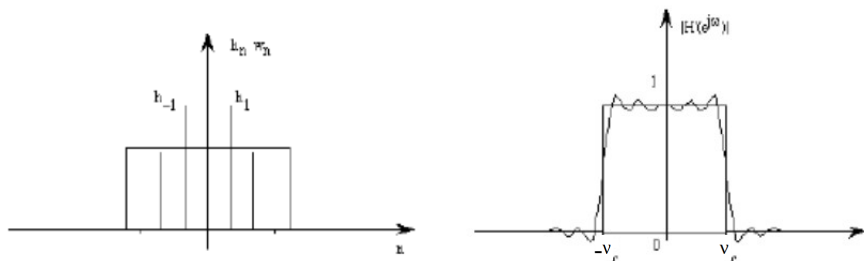
This result is illustrated in Figures 2.7 and 2.8.

Figure 2.7: Impulse and frequency responses of the FIR filter synthesized by the window method

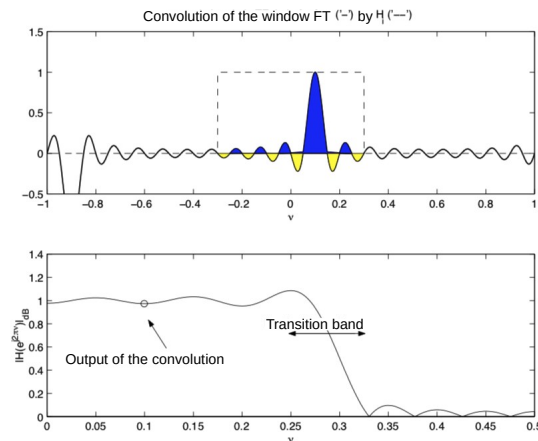**Roland Badeau**   roland.badeau@telecom-paris.fr

Figure 2.8: Synthesis of an FIR filter by the window method: appearance of a transition band and of ripple in the passband and stopband

Two effects appear:

- ripples in the passband and in the stopband, whose level is directly related to that of the sidelobes of $W(e^{i2\pi\nu})$. This also means that the stopband attenuation is also related to the level of these sidelobes;

- the obtained filter is no longer infinitely selective. A *transition band* appears between the passband and the stopband. The width of this transition band and therefore the filter selectivity are related to the width of the main lobe of $W(e^{i2\pi\nu})$.

The selection of the window's shape is based on a trade-off between the level of the sidelobes and the width of the main lobe. Usual windows include:

- the rectangular window (see Figure 2.9a) has width $1/M$, second lobe at -13 dB, and decrease of -6 dB/octave;

- the Bartlett window (see Figure 2.9b) has width $2/M$, second lobe at -26 dB, and decrease of -12 dB/octave;

- the Hann window (see Figure 2.9c) has width $2/M$, second lobe at -31 dB, and decrease of -18 dB/octave;

- the Hamming window (see Figure 2.9d) has width $2/M$, second lobe at -41 dB, and decrease of -6 dB/octave;

- the Blackman window (see Figure 2.9e) has width $3/M$, second lobe at -57 dB, and decrease of -18 dB/octave.
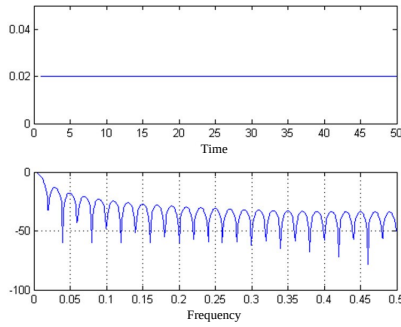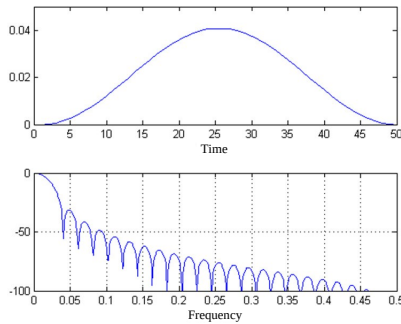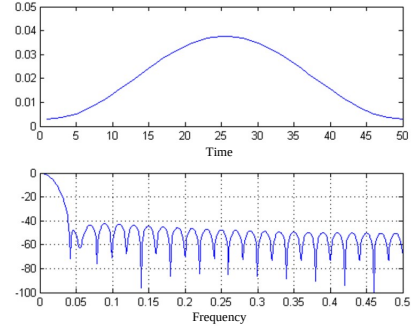
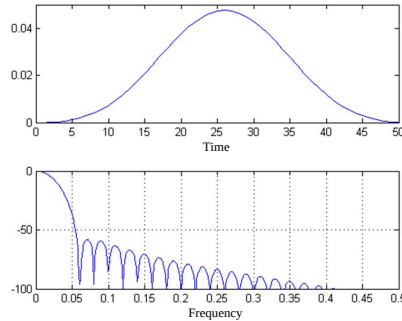(a) Rectangular window:
$$w(k) = \mathbf{1}_{[-(M-1)...M-1]}(k).$$
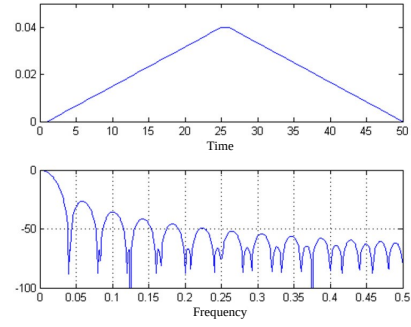
(b) Bartlett window:
$$w(k) = \left(1 - \frac{|k|}{M}\right) \mathbf{1}_{[-(M-1)...M-1]}(k).$$

(c) Hann window:
$$w(k) = \left(0.5 + 0.5 \cos(\frac{\pi k}{M})\right) \mathbf{1}_{[-(M-1)...M-1]}(k).$$

(d) Hamming window:
$$w(k) = \left(0.54 + 0.46 \cos(\frac{\pi k}{M-1})\right) \mathbf{1}_{[-(M-1)...M-1]}(k).$$

(e) Blackman window:
$$w(k) = \left(0.4266 + 0.4965 \cos(\frac{\pi k}{M-1}) + 0.076 \cos(\frac{2\pi k}{M-1})\right) \mathbf{1}_{[-(M-1)...M-1]}(k).$$

Figure 2.9: Common window shapes

In practice, it is difficult to optimize the method — in the sense of respecting a given template while minimizing the order of the filter — using common windows such as the rectangular, Bartlett, Hann,... windows. Indeed, we cannot adjust the trade-off between the main lobe width and the sidelobe level.

Kaiser looked into this problem in the 1970s. The result is the family of Kaiser windows, parameterized by a coefficient β. This parameter adjusts the attenuation level (δ_2) in the stopband. We can then adjust the transition

band by varying the length of the filter. Its expression is given, for a length $N = 2M + 1$ by:

$$w(n) = \frac{I_0(\beta \sqrt{1 - n^2/M^2})}{I_0(\beta)} \tag{2.2}$$

where I_0 denotes the modified Bessel function of the first kind. The result is a type I FIR filter. We also note that for the window method, we obtain a comparable ripple level in the two bands, *i.e.* $\delta_1 \approx \delta_2$.

3.2 Window optimization under constraint

There are several ways of optimizing the window method, depending on the applied constraint. The example of *prolate spheroid* sequences given below makes it possible to understand how to formulate such a problem and introduces another type of optimal filter: the eigenfilters [Vai93].

3.2.1 Prolate sequences

Let $h(n)$ be a real, causal, low-pass FIR filter of length N. We want to optimize this filter in the sense of maximizing $\int_{-\nu_c}^{\nu_c} |H(e^{i2\pi\nu})|^2 \, d\nu$, that is to say the area of the passband, under unitary constraint, namely:

$$\int_{-0.5}^{0.5} H(e^{i2\pi\nu})|^2 \, d\nu = \sum_{0}^{N-1} h(n)^2 = 1. \tag{2.3}$$

We use the following vector notation:

$$\mathbf{h} = [h(0) \, h(1) \, \ldots \, h(N-1)]^\top \, , \; \mathbf{e}(z) = [1 \, z^{-1} \, \ldots \, z^{-N+1}]^\top.$$

The problem can then be written as the maximization of the quadratic form

$$Q = \mathbf{h}^H \mathbf{R} \mathbf{h} \, , \; \mathbf{R} = \int_{-\nu_c}^{\nu_c} \mathbf{e}(e^{i2\pi\nu}) \mathbf{e}^H(e^{i2\pi\nu}) \, d\nu$$

(where ^{H} denotes the conjugate transpose) subject to

$$\mathbf{h}^H \mathbf{h} = 1.$$

A quick calculation gives the general term of matrix $\mathbf{R}$:

$$r_{mn} = 2\nu_c \, \text{sinc}(2\nu_c(m - n)).$$

Matrix $\mathbf{R}$ is real, symmetric (thus Hermitian), Q is therefore real. According to Rayleigh's principle, the maximum of Q is reached when $\mathbf{h}$ is the unit eigenvector associated with the greatest eigenvalue of $\mathbf{R}$, denoted λ_M. A synthesis example of such a filter is given in Figure 2.10.

3.2.2 Optimal eigenfilters

These filters are obtained by a method similar to the previous one, but extended to jointly minimize the error in the passband E_p and in the stopband E_s. These errors are defined as follows: In the stopband, it is simply written:

$$E_s = 2 \int_{\nu_s}^{0.5} H(e^{i2\pi\nu})|^2 \, d\nu = \mathbf{h}^H \mathbf{S} \mathbf{h}$$

in the same way as in the previous paragraph. In the passband, we define the deviation to the static value $H(1) = \mathbf{h}^H \mathbf{1}$ as

$$\epsilon_p(\nu) = \mathbf{h}^H [\mathbf{1} - \mathbf{e}(e^{i2\pi\nu})].$$
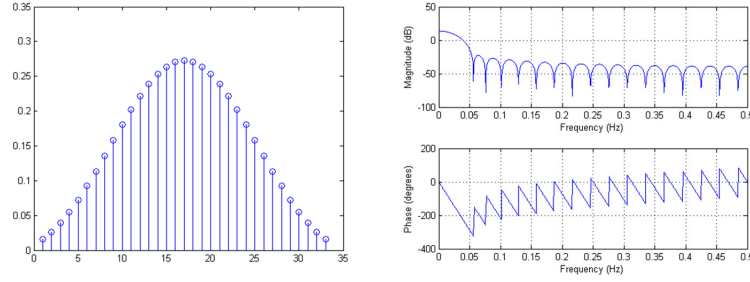
Figure 2.10: Impulse response and frequency response of a prolate spheroid filter with $N = 33$, $\nu_c = 0.05$

E_p is the average of this deviation taken as a quadratic value, that is:

$$E_p = \int_{-\nu_c}^{\nu_c} |H(1) - H(e^{i2\pi\nu})|^2 \, d\nu = \mathbf{h}^H \mathbf{P} \mathbf{h}$$

where $P = \int_{-\nu_c}^{\nu_c} [(\mathbf{1} - \mathbf{e}(e^{i2\pi\nu})][\mathbf{1} - \mathbf{e}(e^{i2\pi\nu})]^H \, d\nu$.

We then introduce a weight factor distributed between the passband and the stopband in the form

$$E = \alpha E_p + (1 - \alpha) E_s$$

where $\alpha \in [0\,1]$. We thus get the quadratic form to be minimized, which is:

$$Q = \mathbf{h}^H \mathbf{R} \mathbf{h}$$

with $\mathbf{R} = \alpha \mathbf{P} + (1 - \alpha) \mathbf{Q}$. We check that $\mathbf{R}$ is real and symmetric. We then find vector $\mathbf{h}$ as the unit eigenvector associated with the minimum eigenvalue λ_0. In this case, as in the previous one, we can verify that Q is positive and thus that $\lambda_0 > 0$.

3.3   Iterative methods

These methods consist in minimizing the Chebyshev error criterion defined below. This error reflects the deviation between the desired response and the obtained response, weighted by higher values in regions where the desired minimization is stronger. The minimization is obtained iteratively by Remez's exchange algorithm [RMP75] or by linear programming techniques [RG75].

3.3.1   Problem formulation

We want to optimize an FIR filter of length N whose frequency response is given by $H(e^{i2\pi\nu}) = e^{-i2\pi\nu(\frac{N-1}{2})} H_R(\nu)$. The optimization criterion is based on the calculation of the error

$$E(\nu) = W(\nu)|D(\nu) - H_R(\nu)| \tag{2.4}$$

which reflects the deviation between the zero phase part of the obtained response and the ideal function (D for "desired") $D(\nu)$ that is aimed at. The function $W(\nu) \geq 0$ weights this error according to the frequency (for instance, we can set $W(\nu) = 1/\delta_1$ in the passband, $W(\nu) = 1/\delta_2$ in the stop-band, and $W(\nu) = 0$ in the transition band).

The problem is formulated as follows: the filter that we are looking for is the one whose frequency response minimizes the infinite norm (the maximum error) of the weighted Chebyshev error on a frequency band B, namely:

$$H(e^{i2\pi\nu}) = \arg\min_H \|E(\nu)\|_\infty \tag{2.5}$$

where $\|E(\nu)\|_\infty = \max_B E(\nu)$.

3.3.2 Remez's exchange algorithm

First of all, we note that the four FIR filter types can be parametrized in a similar way: $H_R(\nu) = P(\nu)Q(\nu)$, where the particular forms of functions $P(\nu)$ and $Q(\nu)$ are specified in Table 2.2.

$H_R(\nu)$	$P(\nu)$	$Q(\nu)$
$\sum\limits_{n=0}^{\frac{N-1}{2}} a_n \cos(2\pi\nu n)$	$\sum\limits_{n=0}^{\frac{N-1}{2}} a_n \cos(2\pi\nu n)$	1
$\sum\limits_{n=1}^{\frac{N}{2}} b_n \cos\left(2\pi\nu\left(n - \frac{1}{2}\right)\right)$	$\sum\limits_{n=0}^{\frac{N}{2}-1} b_n' \cos(2\pi\nu n)$	$\cos(\pi\nu)$
$\sum\limits_{n=1}^{\frac{N-1}{2}} c_n \sin(2\pi\nu n)$	$\sum\limits_{n=0}^{\frac{N-3}{2}} c_n' \cos(2\pi\nu n)$	$\sin(2\pi\nu)$
$\sum\limits_{n=1}^{\frac{N}{2}} d_n \sin\left(2\pi\nu\left(n - \frac{1}{2}\right)\right)$	$\sum\limits_{n=0}^{\frac{N}{2}-1} d_n' \cos(2\pi\nu n)$	$\sin(\pi\nu)$

Table 2.2: Factorization of $H_R(\nu)$

We note that function $Q(\nu)$ is specific to the filter type, whereas function $P(\nu) = \sum\limits_{n=0}^{M} a_n \cos(2\pi\nu n)$ is a trigonometric polynomial whose expression is the same one for the four filter types (with different values of a_n and M). With the factorization $H_R(\nu) = P(\nu)Q(\nu)$, we get:

$$
\begin{aligned}
E(\nu) &= W(\nu)(D(\nu) - P(\nu)\,Q(\nu)) \\
&= W'(\nu)(D'(\nu) - P(\nu))
\end{aligned}
$$

with $W'(\nu) = W(\nu)Q(\nu)$ and $D'(\nu) = \frac{D(\nu)}{Q(\nu)}$.

The *alternation theorem* shows that the optimization problem $H(e^{i2\pi\nu}) = \min\limits_{H}\|E(\nu)\|_\infty$ on a closed set of frequencies B admits a unique solution, which is such that there exist $M + 2$ frequencies $\nu_0 \ldots \nu_{M+1}$ in B which satisfy $E(\nu_k) = \pm(-1)^k \delta$ where $\delta = \|E(\nu)\|_\infty$.

The Remez's exchange algorithm empirically looks for this solution. It is summarized in Algorithm 1. There is no proof of convergence, but usually it converges in a few iterations.

Algorithm 1 Remez algorithm

Initialization: the $M + 2$ alternations are set uniformly in B
Iteration

Direct resolution of the linear system $\sum\limits_{n=0}^{M} a_n \cos(2\pi\nu_k n) + \frac{(-1)^k \delta}{W'(\nu_k)} = D'(\nu_k)$ (or solution by Lagrangian interpolation)

Search of the extrema of this polynomial
Choice of the new values of the ν_k

3.3.3 Example of a low-pass filter

The example taken here is that of a low-pass filter of length 21, of cutoff frequency $\nu_C = .16$, which delimits the passband, and frequency $\nu_A = .24$, which marks the beginning of the stopband. The transition band therefore corresponds to the band lying between ν_C and ν_A. The frequency band on which the minimization operates is $B = [0\ \nu_C] \bigcup [\nu_A\ 0.5]$. $H_R(\nu)$ and $D(\nu)$ are represented (before optimization) in Figure 2.11. We thus define

$$
D(\nu) = \begin{cases} 1, & \text{for } \nu \in [0\ \nu_C]; \\ 0, & \text{for } \nu \in [\nu_A\ 0.5]. \end{cases}
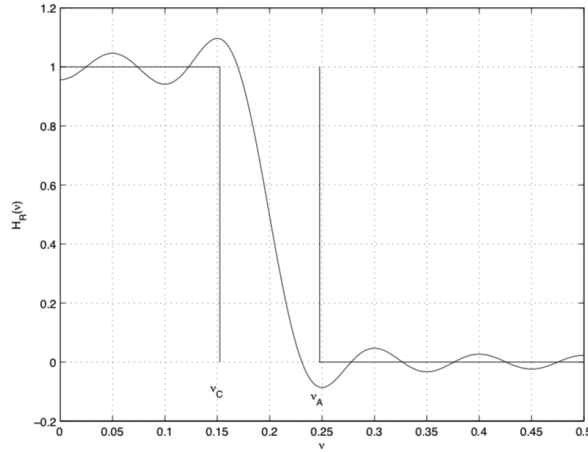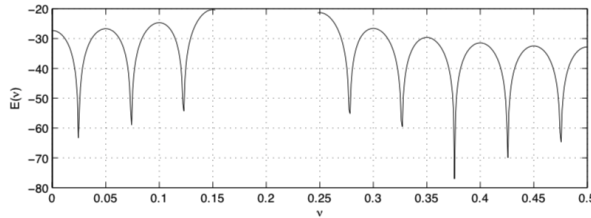$$

Figure 2.11: Obtained response and ideal response of a low-pass filter

Figure 2.12: Error in dB in domain B

We obtain the error represented in dB in Figure 2.12. We consider that if a ripple of about 10% in the band can be tolerated in some cases, however, a -20 dB cut-off is often insufficient in the stopband. This remark leads us to take a higher weight in the stopband, and for example to define function W as

$$W(\nu) = \begin{cases} 1, & \text{for } \nu \in [0 \; \nu_C]; \\ 100, & \text{for } \nu \in [\nu_A \; 0.5]. \end{cases}$$

After optimization, the obtained filter is represented in Figure 2.13.

4   Recursive filters and bilinear transform

4.1   Generalities

The transfer function of a recursive filter is written in the form

$$H(z) = K \frac{N(z)}{D(z)} \tag{2.6}$$

where the numerator $N(z)$ and the denominator $D(z)$ are usually written as polynomials of z^{-1} and K is a real positive constant. Consider the filter $N(z) = \prod_k (1 - z_k z^{-1})$, with $z_k = e^{i2\pi\nu_k}$. This filter has a zero on the unit circle, its frequency response presents cancellations at normalized frequency values $\nu = \nu_k$. This property leads us to build a stopband by setting the ν_k in this band. An example is given by the filter whose frequency response is plotted in Figure 2.14.
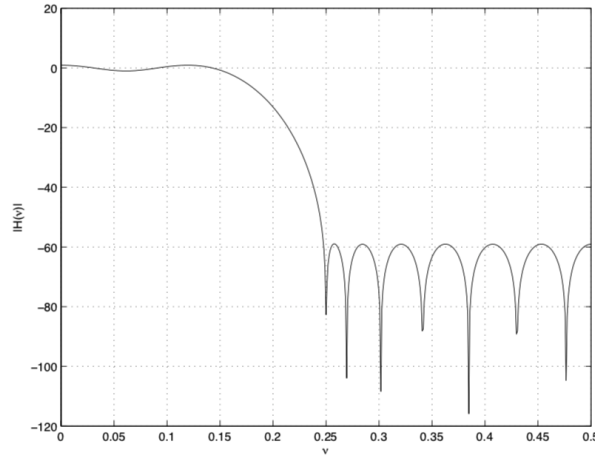
Figure 2.13: Filter obtained after optimization with weights 1 and 100 respectively in the passband and in the stopband
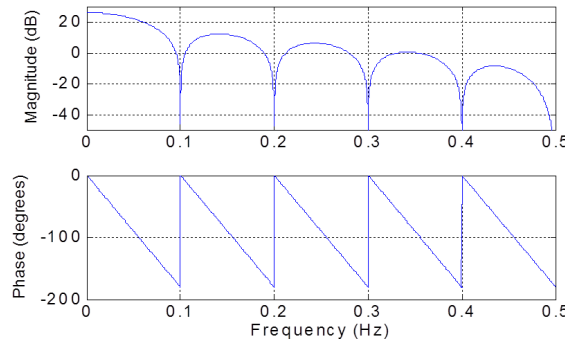

Figure 2.14: Frequency response of the filter $N(z)$ with zeros at the normalized frequencies 0.1, 0.2, ..., 0.5 and their conjugates

We can see that this filter is low-pass but it is not flat in its passband. To overcome this behavior we introduce a pole of $D(z)$ such that in the passband $|D(e^{i2\pi\nu})| \approx |N(e^{i2\pi\nu})|$. We take for instance

$$D(z) = (1 - pz^{-1})(1 - p^*z^{-1})$$

with $p = \rho e^{i2\pi\nu_1}, \rho < 1$ (causal filter). After normalization by the factor $K = \prod_i |1 - p_i|^2 / \prod_k |1 - z_k|^2$, we get the filter $H(z) = KN(z)/D(z)$ whose frequency response is plotted in Figure 2.15. We notice the flattening of the passband and the non-linearity of the phase introduced in the neighborhood of $\nu = 0.1$, which is the frequency of the pole.

Generally, we will thus find the poles in the vicinity of the transition bands of the filters and the cutoff will be even more selective (with a steep slope) that the poles are close to the unit circle.

An example of low-pass recursive filter is illustrated in Figure 2.16.

4.2 Bilinear transform

The most popular technique for synthesizing recursive digital filters consists in converting transfer functions of continuous time systems into Z transfer functions of discrete filters. To do this, we turn the Laplace transform
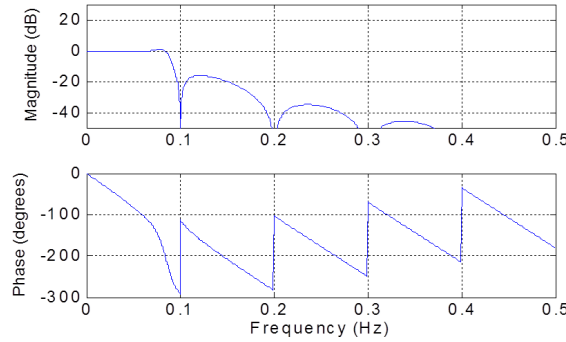
Figure 2.15: Frequency response of the filter $H(z)$ with zeros at the normalized frequencies 0.1, 0.2, ..., 0.5 and their conjugates, and two conjugated poles ($p = 0.95e^{i2\pi\,0.085}$ and p^*)

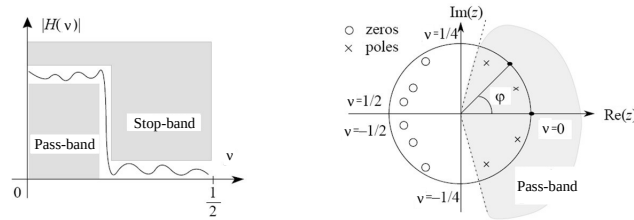
Figure 2.16: Position of the poles and zeros of a low-pass recursive filter

$H_a(p)$ into the Z-transform $H(z)$ by setting:

$$p = k\frac{1 - z^{-1}}{1 + z^{-1}}, \tag{2.7}$$

where k is a positive real number.

This operation transforms the left half-plane of the Laplace transform into the unit disk of the Z plane (see Figure 2.17). The stability of the filters is therefore preserved. The imaginary line is transformed into the unit circle. Let $z = e^{i2\pi\nu}$, then we find using (2.7):

$$p = i\Omega = ik\tan(\pi\nu). \tag{2.8}$$

In this way, all the "analog" frequencies are put in correspondence with the normalized frequencies in the interval $[-0.5\,0.5]$. The value k gives additional freedom to adjust the parameters of the transfer function in p.

4.2.1 Equivalence with the trapezoid method

This equivalence is based on the approximation of the following integral by the trapezoid method:

$$y(nT_e) = \int_{(n-1)T_e}^{nT_e} y'(t)\,dt + y((n-1)T_e)$$

where $y(t)$ is a continuous time function which is at least C^1, and T_e is the sampling period. We deduce

$$y(nT_e) = y((n-1)T_e) + T_e/2[y'(nT_e) + y'((n-1)T_e)]. \tag{2.9}$$

We now want to represent the system whose evolution equation is written
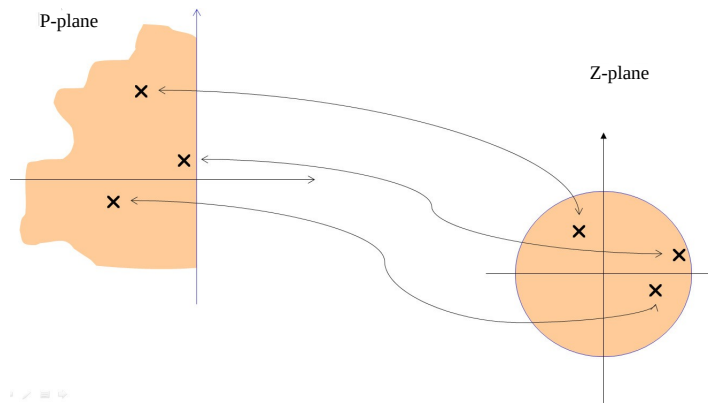
$$x(t) = y(t) + \tau y'(t), \tag{2.10}$$

Figure 2.17: Bilinear transform

and whose transfer function is therefore

$$H_a(p) = \frac{1}{1 + \tau p}. \tag{2.11}$$

We then write the evolution equation at times nT_e and $(n-1)T_e$, and by using equation (2.9), we deduce by applying the Z-transform:

$$\frac{Y(z)}{X(z)} = \frac{1}{1 + \tau \dfrac{2}{T_e} \dfrac{1 - z^{-1}}{1 + z^{-1}}}.$$

The transition from $H_a(p)$ to the Z transfer function of the discrete system is thus performed with the change of variable

$$p = \frac{2}{T_e} \frac{1 - z^{-1}}{1 + z^{-1}}.$$

4.2.2 Example of the first order low-pass filter

We are interested here in digitally simulating the case of an analog low-pass filter of RC type. This filter has the transfer function $H(p)$ written in (2.11) (with $\tau = RC$). Its cut-off frequency is given by $f_c = \frac{1}{2\pi\tau}$. Suppose for example that $f_c = 1000$ Hz. The sampled system operates at a 8000 Hz sampling rate. These conditions fix the *normalized cutoff frequency* of the frequency response of the digital filter: $\nu_c = 1/8$. We then get the value $\Omega_c = k \tan \pi \nu_c$ by using (2.8). One simple way to choose k is then to take $\Omega_c = 1$. We obtain the correspondence between transfer functions described in Figure 2.18.

We note that the choice of k has no influence, since the figures $H_a(i\Omega)$ and $\nu = \frac{1}{\pi} \frac{\arctan \Omega}{k}$ transform both in a similar way, by expansion or contraction of the x-axis, when k varies. Moreover, we notice that the nonlinear part of the arctan curve distorts the frequency response of the sampled filter in the high frequency part. However the cutoff to -3dB is respected. The transfer function of the obtained discrete filter is

$$H(z) = \frac{1}{k + 1} \frac{1 + z^{-1}}{1 + \frac{1-k}{1+k} z^{-1}}.$$

This transformation is used to calculate the digital filters equivalent to common analog filters (e.g. Butterworth, Chebyshev) whose expression of $H_a(i\Omega)$ is known (*cf.* for example [OS75]).
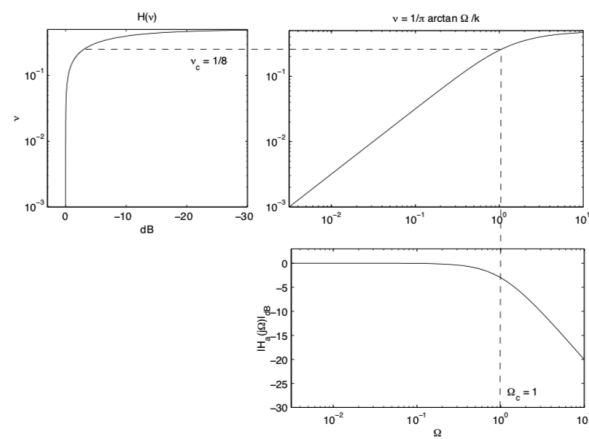
Figure 2.18: Bilinear transform of the first-order analog low-pass filter

Chapter 3

Conversion of sampling rate

Changing the sampling rate has many applications in signal processing. We can cite the improvement of the signal to quantization noise ratio for digital signal acquisition (delta-sigma converters), digital-to-analog conversion by oversampling (audio CD oversampling), switching between different normalized frequencies (32kHz, 44.1Hz and 48kHz from DAT), etc.

1 Upsampling (or oversampling)

Let us consider an analog signal $x_a(t)$, with finite spectral support. This signal is sampled at a sample period T, such that the assumption of the Shannon sampling theorem is fulfilled: $\text{supp}(X_a(f)) \subset \left[-\frac{1}{2T}, \frac{1}{2T} \right]$. The discrete time signal obtained in this way will be denoted $x(n) = x_a(nT)$. Then the Nyquist reconstruction formula yields

$$x_a(t) = \sum_{m \in \mathbb{Z}} x(m) \operatorname{sinc}\left(\pi \left(\frac{t}{T} - m \right) \right).$$

Now, let us consider a smaller sampling period $\frac{T}{L}$ where $L \in \mathbb{N}^*$, so that the discrete time signal $y(n) = x_a\left(n\frac{T}{L} \right)$ corresponds to an interpolation of $x(n)$ by a factor L. The Nyquist reconstruction formula now yields

$$y(n) = \sum_{m \in \mathbb{Z}} x(m) \operatorname{sinc}\left(\pi \left(\frac{n}{L} - m \right) \right). \tag{3.1}$$

This formula is similar to, but is not exactly, a convolution product. In order to rewrite it with a convolution product, we first need to introduce the following signal:

$$\begin{cases} \forall n \in L\mathbb{Z}, w(n) = x\left(\frac{n}{L} \right) \\ \forall n \notin L\mathbb{Z}, w(n) = 0 \end{cases}.$$

A quick calculation shows that the Z-transform of $w(n)$ is $W(z) = X(z^L)$. Indeed, we have $W(z) = \sum_{n \in \mathbb{Z}} w(n) z^{-n}$. With $n = mL$, we get $W(z) = \sum_{m \in \mathbb{Z}} w(mL) z^{-mL} = \sum_{m \in \mathbb{Z}} x(m)(z^L)^{-m} = X(z^L)$. In the frequency domain, this equation yields $W(e^{i2\pi\nu}) = X(e^{i2\pi L\nu})$.

Then, with the change of variable $m = \frac{l}{M}$ in equation (3.1), we get

$$y(n) = \sum_{l \in \mathbb{Z}} w(l) \operatorname{sinc}\left(\pi \left(\frac{n-l}{L} \right) \right) = (h * w)(n), \tag{3.2}$$

where $h(n) = \operatorname{sinc}\left(\frac{\pi n}{L} \right) \forall n \in \mathbb{Z}$. It can be easily proved that the following frequency response corresponds to the impulse response $h(n)$:

$$\begin{cases} H(e^{i2\pi\nu}) = L & \forall |\nu| < \frac{1}{2L} \\ H(e^{i2\pi\nu}) = 0 & \forall \frac{1}{2L} < |\nu| < \frac{1}{2} \end{cases}.$$

Roland Badeau `roland.badeau@telecom-paris.fr`

TELECOM
Paris

IP PARIS

Indeed, if H is defined in this way, then we can calculate the impulse response by applying the inverse DTFT:

$h(n) = \int_{\nu=-\frac{1}{2}}^{\frac{1}{2}} H(e^{i2\pi\nu}) \, e^{+i2\pi\nu n} d\nu = L \int_{\nu=-\frac{1}{2L}}^{\frac{1}{2L}} e^{+i2\pi\nu n} d\nu = \text{sinc}\left(\frac{\pi n}{L}\right).$

Then equation (3.2) implies $\begin{cases} Y(e^{i2\pi\nu}) = LX(e^{i2\pi L\nu}) & \forall |\nu| < \frac{1}{2L} \\ Y(e^{i2\pi\nu}) = 0 & \forall \frac{1}{2L} < |\nu| < \frac{1}{2} \end{cases}$.

The spectral domain interpretation of upsampling is illustrated in Figure 3.1.
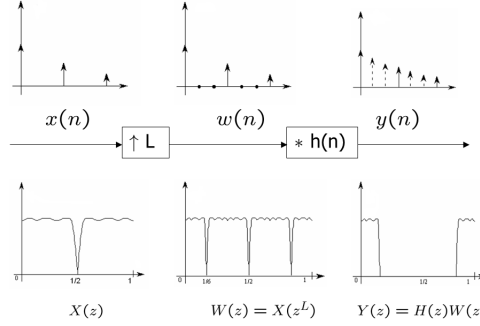
Figure 3.1: Upsampling (or oversampling)

Note that the operator which computes w from x is denoted $\uparrow L$, and it is called *insertion*, which is short for "insertion of zeros". Finally, the block diagram of upsampling is depicted in Figure 3.2.
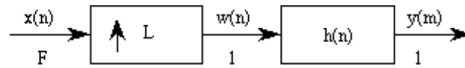

Figure 3.2: Upsampling diagram

2   Downsampling (or subsampling)

Downsampling is the reverse operation of upsampling. By inverting the order of three columns in Figure 3.1, we get the signals and spectra represented in Figure 3.3:
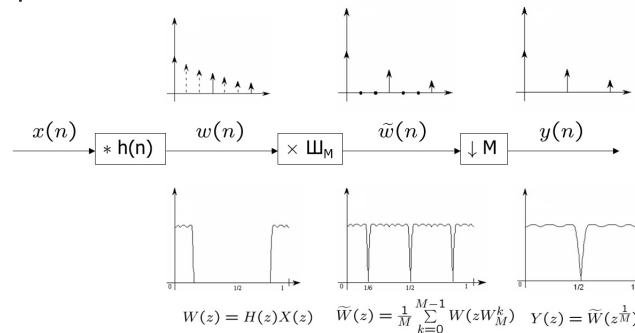

Figure 3.3: Downsampling (or subsampling)

Note that in Figure 3.3, the names of the signals have been changed compared to Figure 3.1, just for convenience (letter y usually denotes the output of a signal processing system). Also, the upsampling factor L has been

replaced by the downsampling factor $M \in \mathbb{N}^*$.

In order to get $\tilde{w}$ from w, and y from $\tilde{w}$, we introduced two new operators, denoted $\times \text{Ш}_M$ and $\downarrow M$, respectively. The latter is called *decimation* (by a factor M).

More precisely, we have $\tilde{w}(n) = w(n)\,\text{Ш}_M(n)$ with $\text{Ш}_M(n) = \frac{1}{M}\sum_{k=0}^{M-1} e^{2i\pi\frac{kn}{M}}$. Therefore $\tilde{W}(z) = \sum_{n\in\mathbb{Z}} \tilde{w}(n)z^{-n} = \sum_{n\in\mathbb{Z}} w(n)\left(\frac{1}{M}\sum_{k=0}^{M-1} e^{2i\pi\frac{kn}{M}}\right)z^{-n} = \frac{1}{M}\sum_{k=0}^{M-1}\sum_{n\in\mathbb{Z}} w(n)\left(z\,e^{-2i\pi\frac{k}{M}}\right)^{-n} = \frac{1}{M}\sum_{k=0}^{M-1} W\left(z\,e^{-2i\pi\frac{k}{M}}\right)$.

In other respects, we already know that $\tilde{W}(z) = Y(z^M)$. Then, we have $Y(z) = \tilde{W}(z^{\frac{1}{M}})$. Since $Y(e^{i2\pi\nu}) = \tilde{W}(e^{i2\pi\frac{\nu}{M}})$ and $\tilde{W}(e^{i2\pi\nu}) = \frac{1}{M}\sum_{k=0}^{M-1} W\left(e^{i2\pi\left(\nu-\frac{k}{M}\right)}\right)$, we thus get

$$Y(e^{i2\pi\nu}) = \frac{1}{M}\sum_{k=0}^{M-1} W\left(e^{i2\pi\left(\frac{\nu-k}{M}\right)}\right). \tag{3.3}$$

Also note that the spectrum of $w(n)$ in Figure 3.3 is assumed to have limited support. This is necessary to avoid spectral aliasing at the output of the downsampling chain. In order to enforce this property, we just assume that $w(n)$ is obtained from the input signal $x(n)$ by applying an ideal low-pass filter of cut-off frequency $\frac{1}{2M}$:

$$\begin{cases} H(e^{i2\pi\nu}) = 1 & \forall |\nu| < \frac{1}{2M} \\ H(e^{i2\pi\nu}) = 0 & \forall \frac{1}{2M} < |\nu| < \frac{1}{2} \end{cases}.$$

Finally, equation (3.3) yields $Y(e^{i2\pi\nu}) = \frac{1}{M}X(e^{i2\pi\frac{\nu}{M}})\ \forall |\nu| < \frac{1}{2}$.

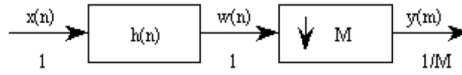The block diagram of downsampling is depicted in Figure 3.4.

Figure 3.4: Downsampling diagram

3 Resampling

Resampling by a factor $\frac{L}{M}$ is performed by performing first upsampling by factor L, then downsampling by factor M. The order of the two operations is important:

- if we downsample first, then the lowest sampling rate involved in the conversion is $\frac{F_s}{M}$;

- if we upsample first, then the lowest sampling rate involved in the conversion is $\min(F_s, \frac{L}{M}F_s)$;

Therefore starting with upsampling is the guarantee to lose as less information as possible during the conversion.

In other respects, performing upsampling first permits us to merge the two low-pass filters involved in upsampling and down sampling. The frequency response of the resulting low-pass filter is

$$\begin{cases} H(e^{i2\pi\nu}) = L & \forall |\nu| < \min\left(\frac{1}{2L}, \frac{1}{2M}\right) \\ H(e^{i2\pi\nu}) = 0 & \forall \min\left(\frac{1}{2L}, \frac{1}{2M}\right) < |\nu| < \frac{1}{2} \end{cases}.$$

Finally, we get $Y(e^{i2\pi\nu}) = \frac{L}{M}X(e^{i2\pi\frac{L}{M}\nu})\ \forall |\nu| < \min\left(\frac{1}{2}, \frac{M}{2L}\right)$ (and 0 elsewhere).

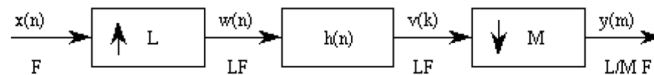The block diagram of resampling is depicted in Figure 3.5.

Figure 3.5: Resampling diagram

4 Noble identities

The noble identities described in this section aim to permute the decimation, insertion, and filtering operations, so as to produce *efficient* implementations of upsampling, downsampling, and resampling (in terms of computational complexity).

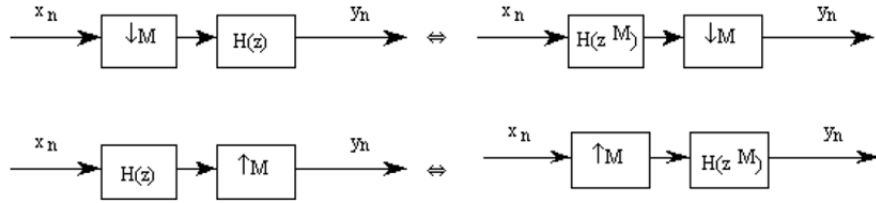The first two noble identities are described in Figure 3.6:

Figure 3.6: Permutation of filtering / decimation or insertion

The equivalence in the top sub-figure of Figure 3.6 is proved as follows:

- in the left block diagram, we get $Y(z) = H(z)\frac{1}{M} \sum_{k=0}^{M-1} X\left(z^{\frac{1}{M}} e^{-2i\pi \frac{k}{M}}\right)$;

- in the right block diagram, we get $Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} H\left(\left(z^{\frac{1}{M}} e^{-2i\pi \frac{k}{M}}\right)^M\right) X\left(z^{\frac{1}{M}} e^{-2i\pi \frac{k}{M}}\right)$.

So the two Z-transforms $Y(z)$ are equal.

In the same way, the equivalence in the bottom sub-figure of Figure 3.6 is proved as follows:

- in the left block diagram, we get $Y(z) = (H X)(z^M)$;

- in the right block diagram, we get $Y(z) = H(z^M)X(z^M)$.

So again the two Z-transforms $Y(z)$ are equal.

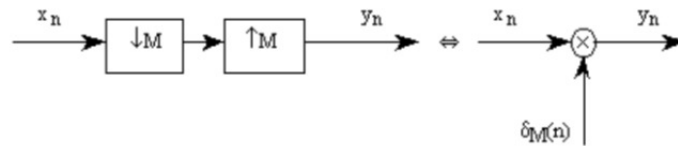Another noble identity is described in Figure 3.7:

Figure 3.7: Simplification insertion / decimation

The equivalence in Figure 3.7 is proved as follows:

- in the left block diagram, we get $Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} X\left((z^M)^{\frac{1}{M}} e^{-2i\pi \frac{k}{M}}\right)$;

- in the right block diagram, we get $Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} X\left(z\, e^{-2i\pi \frac{k}{M}}\right)$.

So the two Z-transforms $Y(z)$ are equal.

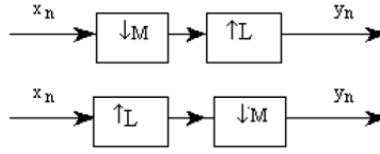Finally, the last noble identity is described in Figure 3.8:

Roland Badeau   roland.badeau@telecom-paris.fr

Figure 3.8: Permutation of insertion / decimation?

Be careful, the equivalence in Figure 3.8 holds if and only if M and N are coprime integers. Indeed,

- in the top block diagram, we get $Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} X\left((z^L)^{\frac{1}{M}} e^{-2i\pi \frac{k}{M}}\right)$;

- in the bottom block diagram, we get $Y(z) = X(z)\frac{1}{M} \sum_{k=0}^{M-1} X\left(\left(z^{\frac{1}{M}} e^{-2i\pi \frac{k}{M}}\right)^L\right)$.

So the two Z-transforms $Y(z)$ are equal if and only if the two sets $S_1 = \left\{e^{-2i\pi \frac{k}{M}}\right\}_{k=0}^{M-1}$ and $S_L = \left\{e^{-2i\pi \frac{kL}{M}}\right\}_{k=0}^{M-1}$ are equal. First, we note that we trivially have the inclusion $S_L \subset S_1$, so we just have to check whether $S_1 \subset S_L$.

The Bézout's identity shows that if d is the greatest common divisor of M and L, then there exist integers u and v such that $uM + vL = d$. Consequently, $e^{-2i\pi \frac{d}{M}} = e^{-2i\pi \frac{uM+vL}{M}} = e^{-2i\pi \frac{vL}{M}}$.

So if $d = 1$, $e^{-2i\pi \frac{1}{M}} \in S_L$, therefore we do have $S_1 \subset S_L$. However, if $d > 1$, let $M' = M/d$: it is clear that all elements in S_L are powers of $e^{-2i\pi \frac{1}{M'}}$, which is not the case of $e^{-2i\pi \frac{1}{M}}$, therefore $S_1 \not\subset S_L$.

5 Polyphase components

The upsampling, downsampling and resampling operations involve a low-pass filter $h(n)$. In order to implement them efficiently, we first need to introduce the *polyphase components* of $h(n)$.

5.1 General definition

The polyphase components of an impulse response $h(n)$ are simply obtained by decimating time-shifted versions of this impulse response, as illustrated in Figure 3.9:
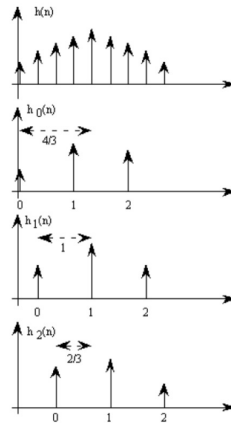

Figure 3.9: Polyphase components

In Figure 3.9, we consider polyphase components at order 3: three time-shifted versions of $h(n)$ are decimated by a factor 3. Note that if h is an ideal low-pass filter, then h_0, h_1, h_2 are all-pass filters, which differ by their phase.

5.2 Type I polyphase components

The relationships between the impulse responses of h and its polyphase components as described in Figure 3.9 can be written in the Z-transform domain as: $H(z) = \sum\limits_{m=0}^{M-1} E_m(z^M)z^{-m}$. This equation can be implemented as in the block-diagram represented in Figure 3.10, for the particular case $M = 2$.
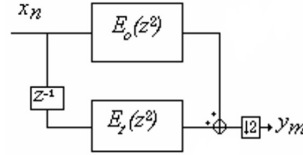

Figure 3.10: Type I polyphase components

5.3 Type II polyphase components

In a similar way, the relationships between the impulse responses of h and its polyphase components as described in Figure 3.9 can equivalently be written in the Z-transform domain as: $H(z) = \sum\limits_{l=0}^{L-1} R_l(z^L)z^{-(L-1-l)}$. This equation can be implemented as in the block-diagram represented in Figure 3.11, for the particular case $L = 2$.
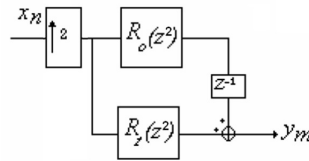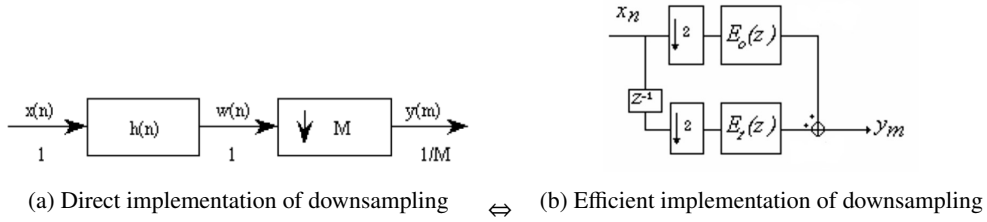

Figure 3.11: Type II polyphase components

6 Efficient structures

6.1 Efficient structure for downsampling

Type I polyphase components can be used to efficiently implement downsampling. Indeed, by substituting Figure 3.10 into Figure 3.12a with $M = 2$, and by using the noble identity in Figure 3.6, we get the block-diagram in Figure 3.12b.

However, in Figure 3.12a, filtering is performed with a filter of length N at a sampling rate of F_s, whereas in Figure 3.12b, filtering can be performed in parallel with polyphase filters of length $\simeq \frac{N}{M}$ at a sampling rate of $\frac{F_s}{M}$. So the computational complexity (in terms of additions and multiplications) is reduced by a factor M^2.

Roland Badeau   roland.badeau@telecom-paris.fr

Contexte académique } sans modifications
Voir Page 79   43/79

TELECOM
Paris

IP PARIS

(a) Direct implementation of downsampling $\Leftrightarrow$ (b) Efficient implementation of downsampling

Figure 3.12: Efficient structure for downsampling ($M = 2$)

6.2 Efficient structure for upsampling

In the same way, Type II polyphase components can be used to efficiently implement upsampling. Indeed, by substituting Figure 3.11 into Figure 3.13a with $L = 2$, and by using the noble identity in Figure 3.6, we get the block-diagram in Figure 3.13b.

However, in Figure 3.13a, filtering is performed with a filter of length N at a sampling rate of F_s, whereas in Figure 3.13b, filtering can be performed in parallel with polyphase filters of length $\simeq \frac{N}{L}$ at a sampling rate of $\frac{F_s}{L}$. So the computational complexity is reduced by a factor L^2.

(a) Direct implementation of upsampling $\Leftrightarrow$ (b) Efficient implementation of upsampling
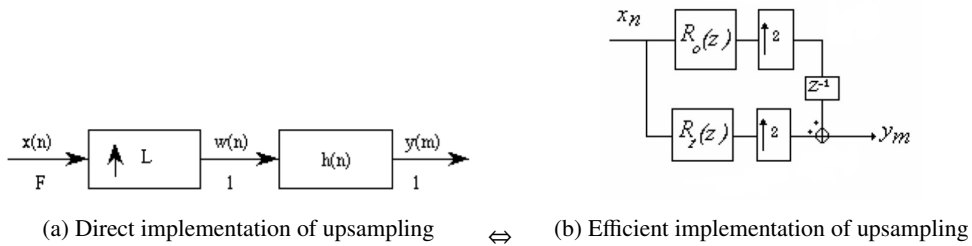
Figure 3.13: Efficient structure for upsampling ($L = 2$)

Chapter 4

Short time Fourier transform

This chapter is partly extracted from a course handout by Bertrand David, translated in English.

The *Short Time Fourier Transform* (STFT) belongs to the family of time-frequency analysis methods. It is defined as a sliding Fourier transform. The STFT is widely used in the analysis of speech and music signals. It is related to the *spectrogram*, which represents the modulus of the STFT in decibels with a color code. This representation is sometimes called *sonagram* by reference to a registered trademark (sonagraph from Kay electronics). The speed of the DFT computation algorithm, called *Fast Fourier Transform* (FFT), favored the reputation of this tool. The waveform and spectrogram of a sample speech signal are represented in Figure 4.1.
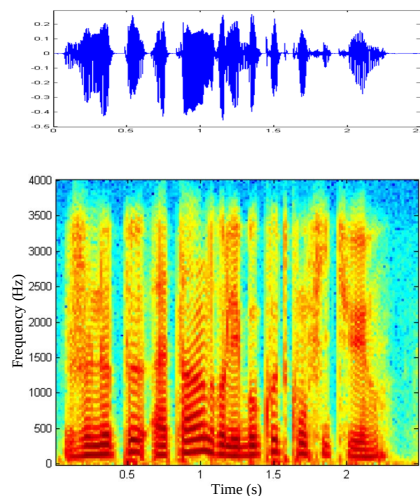

Figure 4.1: Waveform and spectrogram of a speech signal

1 Definition and computation

The block diagram of the STFT is represented in figure 4.2, as it is numerically computed. The principle is that of a sliding Fourier transform, performed on overlapping frames of the signal. The spectrum is calculated on a signal windowed by an analysis window $w_a(n)$ (most often of finished length, real and symmetric) at analysis times $t_a(u) = u\,M$, $u \in \mathbb{Z}$. These analysis times are thus regularly spaced by M samples. The STFT of $x(n)$ is then
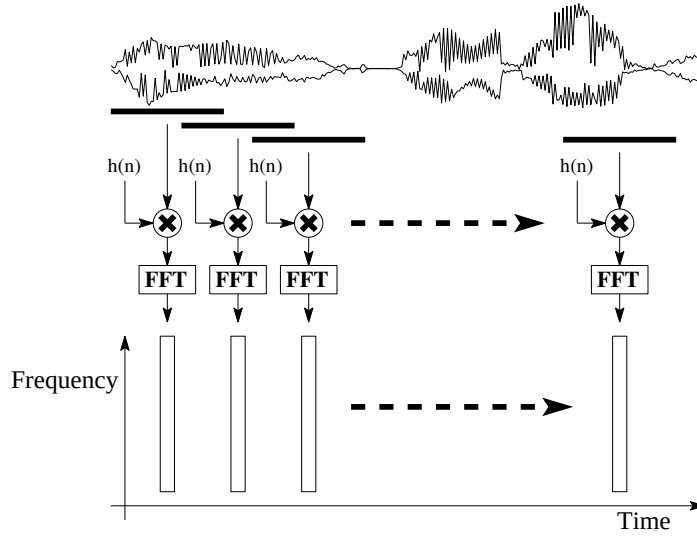
Roland Badeau   roland.badeau@telecom-paris.fr

TELECOM
Paris

IP PARIS

Figure 4.2: Diagram of the short time Fourier transform

defined as

$$\tilde{X}(t_a, \nu) \overset{\Delta}{=} \sum_{n \in \mathbb{Z}} x(n + t_a)w_a(n)e^{-j2\pi\nu n}. \tag{4.1}$$

We preferred here a notation function of ν while a notation function of $e^{i2\pi\nu}$ would have been more consistent with the interpretation in terms of sliding Fourier transform, but this choice simplifies the expressions.

1.1 Interpretation

A quick calculation shows that, by defining $h_k(n) = w_a(-n)e^{i2\pi\nu_k n}$, the expression (4.1) can be written in the form of a convolution product:

$$\tilde{X}(t_a, \nu_k) = [x * h_k](t_a). \tag{4.2}$$

If $w_a(n)$ is a real and pair window of finite length, its FT $W_a(e^{i2\pi\nu})$ is real and even. The FT of h_k is then simply $H_k(e^{j2\pi\nu}) = W_a(e^{j2\pi(\nu-\nu_k)})$. A typical result is given in figure 4.3 for $\nu_k = 0.3$. This example shows that $\tilde{X}(t_a, \nu_k)$ performs a band-pass FIR filtering around frequency ν_k. The characteristics of the filter are linked to that of the chosen analysis window. This interpretation is at the origin of the qualification of *band-pass convention* given to expression (4.1). There is another convention, called *low-pass*, often used for its ease of handling calculations. We will, however, stick to the band-pass convention because it corresponds to the practical realization.

1.2 Discrete version of the STFT

In practice, the Fourier transform is evaluated using the DFT. This is equivalent to setting $\nu_k = k/N$ in the expression (4.1) of $\tilde{X}(t_a, \nu_k)$. N is the order of the DFT. We thus obtain a *discrete* version of the STFT, i.e. sampled in frequency:

$$\tilde{X}(t_a, \nu_k) = \sum_{n=0}^{N-1} x(n + t_a)w_a(n)e^{-j2\pi\nu_k n}. \tag{4.3}$$

In order to avoid time aliasing, the length of the analysis windows M has to be less than or equal to N.

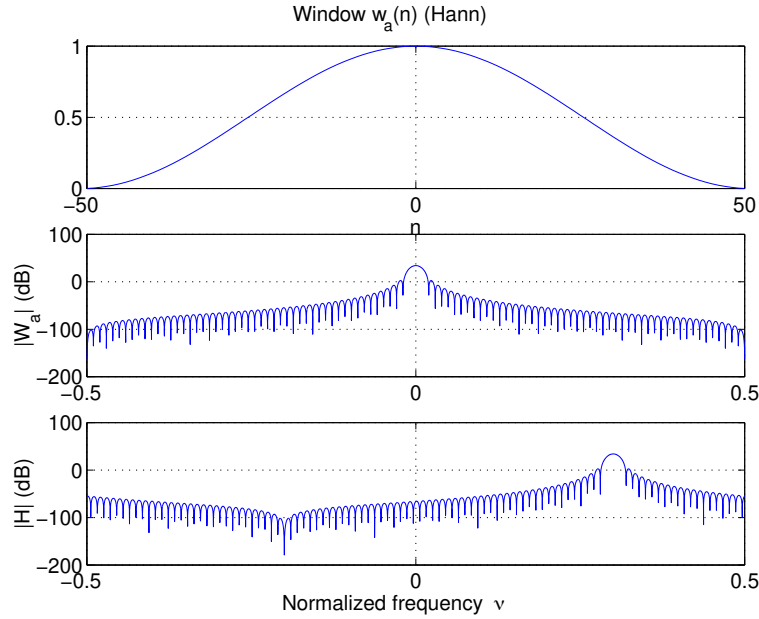**Roland Badeau** `roland.badeau@telecom-paris.fr`

Figure 4.3: Band-pass filtering equivalent to an STFT channel

1.3 Equivalent analysis filter bank

When we make $k = 0, \dots, N-1$ vary, we see that we obtain N signals $x_k(t_a) = \tilde{X}(t_a, \nu_k)$, *i.e.* a signal per *channel* of STFT. Each of these signals is the result of band-pass filtering with a FIR filter centered at ν_k such as previously described. The STFT is therefore equivalent to a *filter bank*. The filters are obtained by shifting in frequency a prototype low-pass filter, which is none other than the Fourier transform of the analysis window. For instance, this filter will be more selective in the case of a rectangular window than in the case of a Hamming window of the same length (narrower main lobe in the rectangular case) but the stopband will be less rejected (larger side lobes). For a same window shape, a shorter length (L lower) leads to a filter with broadened frequency support.

The STFT can thus be interpreted as an analysis filter bank, as shown in Figure 4.4, where $t_a(u) = Mu$ and
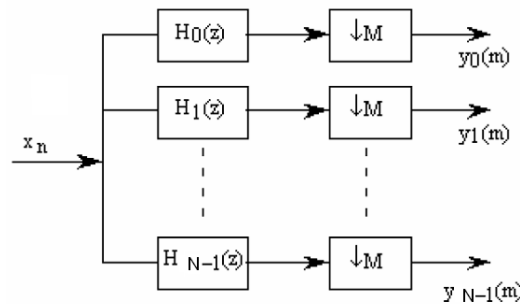
$$y_k(u) = \widetilde{X}(Mu, \nu_k). \tag{4.4}$$

Figure 4.4: Equivalent analysis diagram

Roland Badeau   roland.badeau@telecom-paris.fr

2 Signal reconstruction

In this section, we show that the signal $x(n)$ can be reconstructed from its STFT under mild conditions.

2.1 Perfect reconstruction condition

The reverse operation of the analysis is carried out for the synthesis: from the stream of discrete spectra $\tilde{X}(t_a(u), \nu_k)$ we compute an inverse DFT and we reconstruct the signal by *OverLapp-Add* (OLA). Figure 4.5 illustrates this reconstruction.
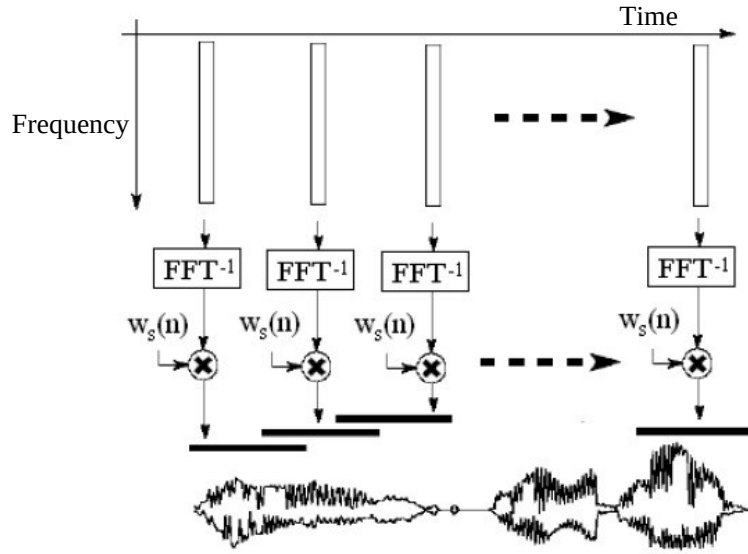

Figure 4.5: Reconstruction diagram

The corresponding mathematical operation is defined as

$$y(n) = \sum_u w_s(n - t_s(u)) y_w(n - t_s(u), t_s(u)). \tag{4.5}$$

The times $t_s = t_s(u)$, $u \in \mathbb{Z}$ denote the temporal synthesis marks, w_s denotes the synthesis window with support in $[0, N - 1]$, and

$$y_w(n, t_s(u)) = \frac{1}{N} \sum_{k=0}^{N-1} Y(t_s(u), \nu_k) e^{j2\pi\nu_k n}. \tag{4.6}$$

When $t_a(u) = t_s(u)$, $\forall u \in \mathbb{Z}$, and when $Y = \tilde{X}$, we can get a perfect reconstruction condition, which guarantees that $x(n) = y(n)$. Indeed, by substituting the expression (4.1) into (4.5), we show that $x(n) = y(n)$ is obtained by means of the sufficient condition:

$$\sum_u w_a(n - t_a(u)) w_s(n - t_a(u)) = 1. \tag{4.7}$$

Proof. By substituting equation (4.6), $t_s(u) = t_a(u)$, and $Y = \tilde{X}$ into (4.5), we get

$$y(n) = \sum_u w_s(n - t_a(u)) \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(t_a(u), \nu_k) e^{j2\pi\nu_k(n - t_a(u))}.$$

Then by substituting (4.3) (where we replace index n by m) into this last equation, we get

$$y(n) = \sum_{u \in \mathbb{Z}} w_s(n - t_a(u)) \sum_{m=0}^{N-1} w_a(m)\, x(m + t_a(u)) \frac{1}{N} \sum_{k=0}^{N-1} e^{j2\pi \nu_k(n-m-t_a(u))}.$$

Noticing that $\frac{1}{N} \sum_{k=0}^{N-1} e^{j2\pi \nu_k(n-m-t_a(u))} = \mathbf{1}_{n-m-t_a(u) \in N\mathbb{Z}}$, we get

$$y(n) = \sum_{u} w_s(n - t_a(u)) \sum_{m=0}^{N-1} w_a(m)\, x(m + t_a(u)) \mathbf{1}_{n-m-t_a(u) \in N\mathbb{Z}}.$$

However, considering that $m \in \{0 \dots M-1\}$ and $N \geq M$, $\mathbf{1}_{n-m-t_a(u) \in N\mathbb{Z}}$ can be non-zero if and only if $m = n - t_a(u)$. Therefore we get the simplification

$$y(n) = \sum_{u} w_s(n - t_a(u)) w_a(n - t_a(u))\, x(n),$$

which proves that $y(n) = x(n) \; \forall n \in \mathbb{Z}$ if and only equation (4.7) holds. $\square$

2.2 Equivalent synthesis filter bank

The reconstruction formula (4.5)-(4.6) can also be expressed as

$$y(n) = \sum_{k=0}^{N-1} [f_k * \widetilde{y}_k](n) \tag{4.8}$$

with

$$f_k(n) = \frac{1}{N} w_s(n)\, e^{i2\pi \nu_k n}, \tag{4.9}$$

or equivalently $F_k(z) = \frac{1}{N} W_s(z W_N^k) = F_0(z W_N^k)$ with $W_N^k = e^{-i2\pi \frac{k}{N}}$, and the signal $\widetilde{y}_k$ is defined as $\widetilde{y}_k(mM) = y_k(m)$, and $\widetilde{y}_k(n) = 0$ everywhere else. Therefore the reconstruction diagram is also equivalent to a synthesis fiter bank, as shown in Figure 4.6, where $\hat{x} = y$.
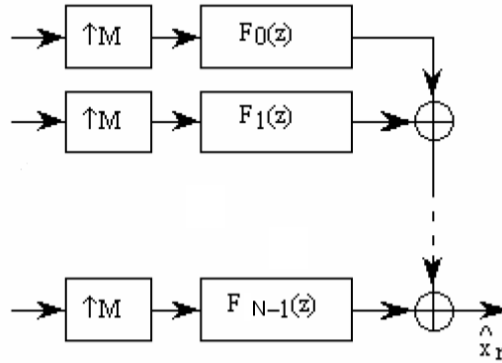

Figure 4.6: Equivalent synthesis diagram

Proof. Substituting the definition of $\widetilde{y}_k$ into (4.8), we get

$$y(n) = \sum_{k=0}^{N-1} \sum_{l \in \mathbb{Z}} f_k(n-l)\widetilde{y}_k(l) = \sum_{k=0}^{N-1} \sum_{u \in \mathbb{Z}} f_k(n-uM)y_k(u)$$

with $l = uM$. Then substituting (4.4) into this last equation, we get

$$y(n) = \sum_{k=0}^{N-1} \sum_{l \in \mathbb{Z}} f_k(n - l)\widetilde{y}_k(l) = \sum_{k=0}^{N-1} \sum_{u \in \mathbb{Z}} f_k(n - uM)\widetilde{X}(uM, \nu_k).$$

Finally, substituting (4.9) into this last equation, we get

$$y(n) = \sum_{u \in \mathbb{Z}} w_s(n - uM)\frac{1}{N} \sum_{k=0}^{N-1} \widetilde{X}(uM, \nu_k)\, e^{i2\pi\nu_k(n-uM)}.$$

With $t_s(u) = uM$ and $Y = \widetilde{X}$, we thus retrieve (4.5) and (4.6). $\qquad\square$

3 Analysis / synthesis filter bank interpretation

Finally, the STFT followed by the signal reconstruction can be interpreted as an M-channel analysis / synthesis filterbank, as shown in Figure 4.7.
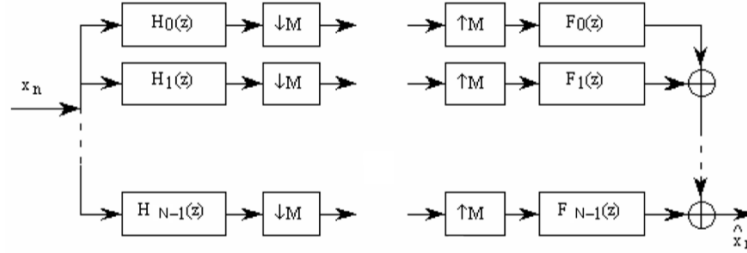

Figure 4.7: Analysis / synthesis diagram

It is a particular case of perfect reconstruction filter bank, as will be introduced in Chapter 5, with $H_k(z) = H_0(zW_N^k)$.

4 Conclusion

The STFT is a time/frequency analysis method which consists in computing Fourier transforms of successive windowed frames of the signal (with or without overlap). The frequency resolution varies as $1/N$, whereas the time resolution varies as N, so there is a trade-off to be found. The method can be implemented efficiently by means of the FFT algorithm. It can also be interpreted as a bank of band-pass filters centered at frequencies multiples of $1/N$.

We thus get a representation of the time evolution of each frequency component. The possibility of recovering the temporal signal is related to constraints on the analysis and synthesis windows, or equivalently to constraints on the analysis and synthesis filters in the filter bank interpretation.

Roland Badeau `roland.badeau@telecom-paris.fr`

Contexte académique } sans modifications
Voir Page 79   50/79

TELECOM
Paris

IP PARIS

Chapter 5

Filter banks

In Chapter 4, we have studied the filter bank interpretation of the short-time Fourier transform. This interpretation comes from the frequency approach of time-frequency analysis. We now introduce the general framework of filter banks [Vai93], which generalizes in various ways the short-time Fourier transform.

1 Sub-band processing

In many applications, it may be of interest to separate the input signal into several sub-band components. Indeed, this makes it possible to locate the frequency band(s) where the useful information can be found. One possible application is audio coding: in order to compress the information transmitted, we only take into account the frequency bands in which the energy is above a certain threshold. To do this, we need to use band-pass filters. However some precautions should be taken:

- the band-pass filters should be carefully designed, so that the amount of information to be processed does not increase dramatically;

- we must be able to reconstruct the original signal from the signals in each sub-band.

The problem of the growth of the quantity of information is solved enough simply. Consider that the signal is decomposed into M sub-bands of same spectral width, then it is possible to sub-sample the signal at the output of each filter by a factor M. The overall processing frequency is maintained, so that there is no increase in the amount of information (*cf.* Figure 5.1). However, sub-sampling creates an inevitable spectral aliasing with physically feasible filters. It then becomes difficult to retrieve the original signal from the sub-sampled signals in each sub-band. The elimination of aliasing effects is made possible by an appropriate synthesis of the filters, which will permit the perfect reconstruction of the input signal.

Figure 5.1: Sub-band processing

In order to make the study of filter banks easier, we first start with the particular case of 2-channel filter banks in Section 2.

Roland Badeau `roland.badeau@telecom-paris.fr`

2 Two-channel filter banks

The general form of a 2-channel filter bank is depicted in Figure 5.2:
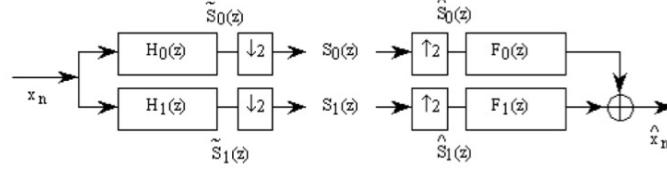
Figure 5.2: 2-channel filter bank

The left part of Figure 5.2, sometimes called *analysis filter bank*, involves the *analysis filters* H_0 and H_1, whereas the right part, sometimes called *synthesis filter bank*, involves the *synthesis filters* F_0 and F_1. The *input signal* is $x(n)$ and the *output signal* is $\widehat{x}(n)$. The two signals $s_0(n)$ and $s_1(n)$ are called *sub-band signals*.

2.1 Ideal 2-channel filter bank

The ideal filter bank is such that the sub-band signal $s_0(n)$ in Figure 5.2 contains exactly the low frequency part of the spectrum of the input signal $x(n)$, whereas the sub-band signal $s_1(n)$ contains its high frequency part; it also guarantees *Perfect Reconstruction* (PR) at the output: $\widehat{x}(n) = x(n)$.

To this purpose, we just assume that H_0 and F_0 are ideal low-pass half-band filters (i.e. with a cut-off frequency of $\frac{1}{4}$), and H_1 and F_1 are ideal high-pass half-band filters. Then in the top branch in Figure 5.2, $\widetilde{s_0}(n)$ is a low-pass half-band signal; after decimation by factor 2, $s_0(n)$ is a full-band signal whose spectrum is the low frequency part of $x(n)$, with a frequency scale stretched by a factor 2; after insertion of zeros, $\widehat{s_0}(n)$ is a full-band signal whose spectrum is the low frequency part of $x(n)$ periodized at period $\frac{1}{2}$; and after filtering by F_0, we retrieve $\widetilde{s_0}(n)$. In the bottom branch, $\widetilde{s_1}(n)$ is a high-pass half-band signal; after decimation by factor 2, $s_1(n)$ is a full-band signal whose spectrum is the high frequency part of $x(n)$ with a frequency scale stretched by a factor 2; after insertion of zeros, $\widehat{s_1}(n)$ is a full-band signal whose spectrum is the high frequency part of $x(n)$ periodized at period $\frac{1}{2}$; and after filtering by F_1, we retrieve $\widetilde{s_1}(n)$. Therefore at the output $\widehat{x}(n)$ is equal to $\widetilde{s_0}(n) + \widetilde{s_1}(n) = x(n)$: the ideal 2-channel filter bank indeed guarantees PR at the output.

2.2 Example with spectral aliasing

The ideal 2-channel filter bank introduced in Section 2.1 is not the only one which guarantees PR. Actually, the 2-channel filter bank depicted in Figure 5.3 is not frequency-selective and even generates a significant spectral aliasing in the sub-band signals $s_0(n)$ and $s_1(n)$. Nevertheless, it still guarantees PR at the output.
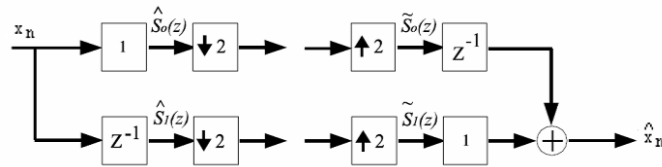

Figure 5.3: Example with aliasing

Indeed, in Figure 5.3 we have $H_0(z) = 1$, $F_0(z) = z^{-1}$, $H_1(z) = z^{-1}$ and $F_1(z) = 1$: all those filters are all-pass filters. Nevertheless, in the top branch the even samples of $x(n)$ are selected, and the odd samples are zeroed out; whereas in the bottom branch the odd samples of $x(n)$ are selected, and the even samples are zeroed out. Therefore

at the output the even and odd samples of $x(n)$ are interleaved, so that the input signal is reconstructed with a delay of one sample: $\widehat{x}(n) = x(n-1)$. In this case we still say that we have PR.

2.3 General case of 2-channel filter bank

In the general case, from Figure 5.2, we get the following relationships between the Z-transforms of the various signals:

$$
\begin{array}{rcl}
\widetilde{S}_0(z) & = & H_0(z)X(z), \\
\widetilde{S}_1(z) & = & H_1(z)X(z), \\
S_0(z) & = & \frac{1}{2}\left(\widetilde{S}_0(z^{\frac{1}{2}}) + \widetilde{S}_0(-z^{\frac{1}{2}})\right), \\
S_1(z) & = & \frac{1}{2}\left(\widetilde{S}_1(z^{\frac{1}{2}}) + \widetilde{S}_1(-z^{\frac{1}{2}})\right), \\
\widehat{S}_0(z) & = & S_0(z^2), \\
\widehat{S}_1(z) & = & S_1(z^2), \\
\widehat{X}(z) & = & F_0(z)\widehat{S}_0(z) + F_1(z)\widehat{S}_1(z).
\end{array}
$$

By means of successive substitutions, we get the following Input-Output (I/O) relationship:

$$\widehat{X}(z) = T(z)X(z) + A(z)X(-z) \tag{5.1}$$

where

$$T(z) = \frac{1}{2}(H_0(z)F_0(z) + H_1(z)F_1(z)), \tag{5.2}$$

$$A(z) = \frac{1}{2}(H_0(-z)F_0(z) + H_1(-z)F_1(z)). \tag{5.3}$$

The spurious term $X(-z)$ in (5.1) induces spectral aliasing in $\widehat{X}(z)$. In order to remove this term, the *Aliasing Cancellation* (AC) condition is simply $A(z) = 0$. Then we say that we get PR, i.e. the output signal $\hat{x}$ is equal to the input signal x up to a delay and a scale factor, iff $T(z) = cz^{-n_0}$ with $c \neq 0$ and $n_0 \in \mathbb{Z}$. By substituting equations (5.2) and (5.3) into the AC and PR conditions, we get the following linear system of equations:

$$
\left[\begin{array}{cc} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{array}\right]
\left[\begin{array}{c} F_0(z) \\ F_1(z) \end{array}\right] =
\left[\begin{array}{c} 2cz^{-n_0} \\ 0 \end{array}\right]
$$

whose solution is

$$
\left\{
\begin{array}{l}
F_0(z) = \frac{2cz^{-n_0}}{D(z)} H_1(-z) \\
F_1(z) = -\frac{2cz^{-n_0}}{D(z)} H_0(-z)
\end{array}
\right.
$$

with $D(z) = H_0(z)H_1(-z) - H_0(-z)H_1(z)$.

If in addition we want both filters H_k and F_k to have FIR, the two conditions become:

$$
\text{AC} \quad : \quad
\left\{
\begin{array}{l}
F_0(z) = H_1(-z), \\
F_1(z) = -H_0(-z),
\end{array}
\right. \tag{5.4}
$$

$$
\text{PR} \quad : \quad T(z) = \frac{1}{2}D(z) = cz^{-n_0}. \tag{5.5}
$$

3 Half-band filters

We are now interested in synthesizing filters H_k and F_k which best approximate the ideal filter bank introduced in Section 2.1. To this purpose, we first have to introduce *half-band filters*.

The ideal low-pass half-band filter is defined by its frequency response $G_R(\nu) = \left\{ \begin{array}{ll} 1 & \text{for } 0 \leq |\nu| < 0.25 \\ 0 & \text{for } 0.25 \leq |\nu| < 0.5 \end{array} \right.$
$\forall \nu \in [0, 0.5[$. This ideal frequency response satisfies the following *half-band condition*:

$$\forall \nu \in \mathbb{R}, \; G_R(\nu) + G_R\left(\nu + \frac{1}{2}\right) = 2c \tag{5.6}$$

Contexte académique } sans modifications

TELECOM
Paris

IP PARIS

with $c > 0$. Unfortunately, the ideal low-pass half-band filter is not stable, since its frequency response is not continuous. We now want to synthesize a type I causal FIR (thus stable) filter $g(n)$ which approximates the ideal low-pass half-band filter, while exactly satisfying the half-band condition in equation (5.6).

Since $g(n)$ is causal, we can write

$$G(e^{2\iota\pi\nu}) = G_R(\nu)\, e^{-2\iota\pi\nu(N-1)}, \tag{5.7}$$

where $G_R(\nu) \in \mathbb{R}$ and $N > 1$, so that the length of $g(n)$ is $2N - 1$.

In the time domain, equation (5.6) is equivalent to $(1 + (-1)^n)g_R(n) = 2c\,\delta_0(n)$, i.e. $g_R(n) = c\delta_0(n)$ when n is even. Since all even samples of $g_R(n)$ are zero (except when $n = 0$), N is necessarily even. Moreover, since $g(n)$ is translated from $g_R(n)$ by $N - 1$ samples, all odd samples of $g(n)$ are zero, except that of index $N - 1$, which is equal to c. Therefore there is a causal FIR filter $v(n)$ such that

$$G(z) = c(V(z^2) + z^{-(N-1)}). \tag{5.8}$$

By construction, $v(n)$ has length N which is even, therefore it is a type II FIR filter, such that

$$V(e^{2\iota\pi\nu}) = V_R(\nu)\, e^{-2\iota\pi\nu\frac{N-1}{2}}. \tag{5.9}$$

By substituting equations (5.7) and (5.9) into (5.8) with $z = e^{2\iota\pi\nu}$, we get the relationship

$$\frac{1}{2c}G_R(\nu) - \frac{1}{2} = \frac{1}{2}V_R(2\nu)$$

(*cf.* left part of Figure 5.4). Therefore $V(e^{2\iota\pi\nu})$ is nearly all-pass, but cuts frequency $1/2$ (*cf.* right part of Figure 5.4).

$$\frac{1}{2c}G_R(\nu) - \frac{1}{2} = \frac{1}{2}V_R(2\nu) \qquad\qquad V_R(\nu)$$
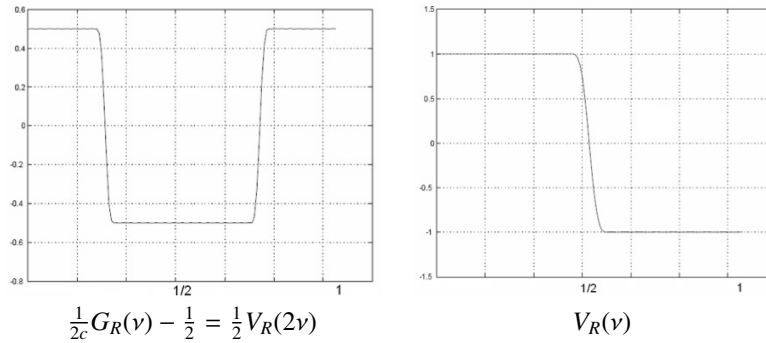
Figure 5.4: Synthesis of $V(z)$ by the Remez method

Finally, filter $v(n)$ can be synthesized by means of the Remez method.

4 Conjugate Quadrature Filters

We now introduce a particular class of 2-channel filter banks, called *Conjugate Quadrature Filters* (CQF) filter banks, which approximate the ideal 2-channel filter bank introduced in Section 2.1, by exploiting the properties of half-band filters described in Section 3.

4.1 CQF filter banks

We first consider the general framework introduced in Figure 5.2 and Section 2.3, especially the I/O relationship (5.1) with the definitions (5.2) and (5.3), and we focus on FIR solutions which satisfy the AC condition (5.4) and the PR condition (5.5).

Roland Badeau   roland.badeau@telecom-paris.fr

We now introduce a new constraint called CQF: we assume that N is even (as in Section 3), and that

$$H_1(z) = -z^{-(N-1)}\tilde{H}_0(-z), \tag{5.10}$$

where filter $\tilde{h}_k$ is obtained by reversing time and conjugating the samples of h_k: $\tilde{H}_k(z) = H_k^*(\frac{1}{z})$ (analysis filters are then called *conjugate quadrature filters*). Then by substituting equation (5.10) into (5.4), we get:

$$\forall k \in \{0, 1\}, \ F_k(z) = z^{-(N-1)}\widetilde{H}_k(z), \tag{5.11}$$

which means that f_k is a FIR filter obtained by reversing time and conjugating the samples of h_k, and finally applying a translation by $N - 1$ samples in order to restore causality. Analysis and synthesis filters are then called *paraconjugate filters*.

Finally, substituting (5.11) into equation (5.2) leads to the following expression of the transfer function:

$$T(z) = \frac{z^{-(N-1)}}{2}\left(\tilde{H}_0(z)H_0(z) + \tilde{H}_0(-z)H_0(-z)\right). \tag{5.12}$$

If we now introduce the *symmetric power* constraint

$$\widetilde{H}_0(z)H_0(z) + \widetilde{H}_0(-z)H_0(-z) = 2c \tag{5.13}$$

into equation (5.12), we retrieve the PR condition (5.5) with $n_0 = N - 1$, i.e. $T(z) = c\,z^{-(N-1)}$.

4.2 Synthesis of a perfect reconstruction CQF filter bank

The problem of synthesizing a PR-CQF filter bank thus reduces to synthesizing a filter h_0 which satisfies the symmetric power constraint (5.13).

To this purpose, the first step consists in synthesizing a symmetric low-pass half-band filter $g(n)$ of length $2N - 1$ as described in Section 3. We then proceed in two steps: the raising of the half-band filter (Section 4.2.1) and the factorization of the half-band filter (Section 4.2.2).

4.2.1 Raising of the half-band filter

The frequency response $G_R(\nu)$ obtained by using the Remez method is depicted in Figure 5.5.
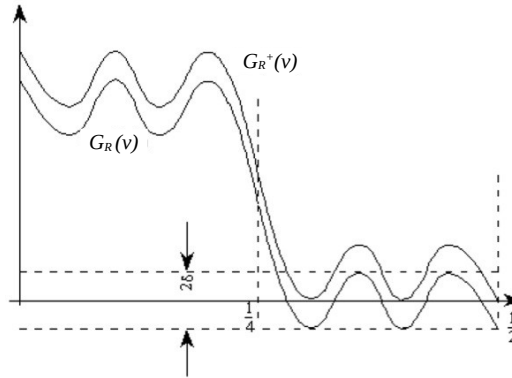
Figure 5.5: Raising of the half-band filter

The alternance theorem shows that this frequency response has ripples of constant magnitude (that we will denote ε from now on) in the stop-band, as can be noticed in Figure 5.5.

Therefore function $G_R^+(\nu) = G_R(\nu) + \varepsilon$ (which is named "$G_{0+}(e^{j2\pi f})$" in Figure 5.5) is nonnegative. In the time domaine, we get $g^+(n) = g(n) + \varepsilon\delta_0(n - (N - 1))$, which is still a symmetric low-pass half-band filter.

4.2.2 Factorization of the half-band filter

Since filter g^+ has length $2N - 1$, its transfer function $G^+(z)$ has $2N - 2$ roots in the complex plane, that can be grouped by pairs of same argument (*cf.* Figure 5.6).
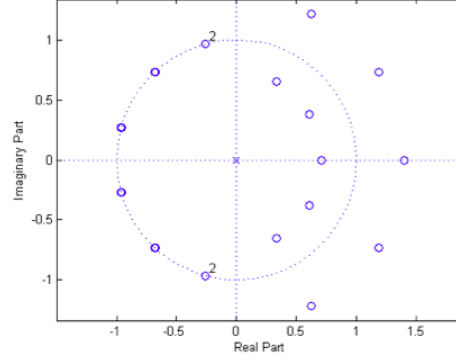
Figure 5.6: Factorization of the half-band filter

In Figure 5.6, also note that function $G^+(z)$ has double roots located on the unit circle in the left half plane, which correspond to the double zeros in the stop-band of function $G_R^+(\nu)$ in Figure 5.5.

A transfer function $H_0(z)$ can then be obtained by keeping only the roots that are *inside* the unit circle in the right half plane, and by making the double roots *simple* in the left half plane, so that $h_0(n)$ is a minimal phase filter of length N (the total number of remaining roots is $N - 1$). Consequently, $\widetilde{H}_0(z)$ is the transfer function obtained by keeping only the roots that are *outside* the unit circle in the right half plane, and by also making the double roots *simple* in the left half plane. In this way, we have the relationship

$$G_R^+(\nu) = \widetilde{H}_0(e^{2\iota\pi\nu})H_0(e^{2\iota\pi\nu}). \tag{5.14}$$

Note that both $H_0(e^{2\iota\pi\nu})$ and $\widetilde{H}_0(e^{2\iota\pi\nu})$ are low-pass half-band filters, but their phase is no longer linear, contrary to that of $G_R(\nu)$. Consequently, $F_0(e^{2\iota\pi\nu})$ is also a low-pass half-band filter, and $H_1(e^{2\iota\pi\nu})$ and $F_1(e^{2\iota\pi\nu})$ are high-pass half-band filters. Therefore the CQF filter bank constructed in this way approximates the ideal 2-channel filter bank introduced in Section 2.1. Moreover, it guarantees perfect reconstruction, since by substituting equation (5.14) into the half-band condition (5.6), we retrieve the symmetric power constraint (5.13).

5 Bi-orthogonal filters

Bi-orthogonal filter banks from a generalization of CQF filter banks, where the analysis and synthesis filters are no longer paraconjugate. This allows more flexibility in the design of the analysis and synthesis filters.

Again, we consider the general framework introduced in Figure 5.2 and Section 2.3, especially the I/O relationship (5.1) with the definitions (5.2) and (5.3), and we focus on FIR solutions which satisfy the AC condition (5.4) and the PR condition (5.5). Now, let $G(z) = H_0(z) F_0(z)$. Then, substituting equations (5.2) and (5.4) into (5.5) leads to

$$G(z) - G(-z) = 2cz^{-n_0}, \tag{5.15}$$

which guarantees the PR property. The synthesis of a bi-orthogonal filter bank then proceeds in three steps:

- Synthesis of a low-pass, half-band, type-I causal FIR filter $G(z)$ of length $2N - 1$ with N even as described in Section 3. Then the half-band condition (5.6) implies equation (5.15), which guarantees PR;

- Raising of the half-band filter as described in Section 4.2.1;

Contexte académique } sans modifications

TELECOM
Paris

IP PARIS

- Factorization of $G(z)$ as a product $H_0(z)F_0(z)$ by selecting the roots of $G(z)$ in various ways (not only the particular way described in Section 4.2.2).

This flexible design allows to synthesize analysis filters H_0 and H_1 with various spectral properties.

6   Application: transmultiplexer

The design of a 2-channel PR filter bank can be directly applied to the design of the transmultiplexer represented in Figure 5.7. The purpose of the transmultiplexer is to transmit several signals (we consider here two input signals s_0 and s_1) in a single communication channel. It thus involves an *encoder* (left part of Figure 5.7) which encodes the two input signals into a single transmitted signal, and a *decoder* (right part of Figure 5.7) which decodes the transmitted signal in order to retrieve the two input signals at the output.

It can be noticed that the diagram in Figure 5.7 is the same as that of the 2-channel filter bank in Figure 5.2, except that the left and right parts have been permuted. Indeed, the PR property of the 2-channel filter bank implies that the synthesis filter bank (right part of Figure 5.2) represents a linear transform which is the exact inverse of the linear transform represented by the analysis filter bank (left part of Figure 5.2), combined downstream with a filter of transfer function $T(z) = c\,z^{-(N-1)}$. Therefore the decoder in Figure 5.7, combined upstream with a filter of transfer function $1/T(z)$, also corresponds to a linear transform which is the exact inverse of the linear transform represented by the encoder. This proves that the transmultiplixer, designed by using the same filters as those of a 2-channel PR filter bank, also guarantees perfect reconstruction at the output, provided that $X(z) = \widehat{X}(z)/T(z)$. In practice, because N is even and because the decoder is only composed of filters and of decimations by a factor of 2, perfect reconstruction is also obtained (up to a delay and a multiplicative factor) if we just set $X(z) = z^{-1}\widehat{X}(z)$.
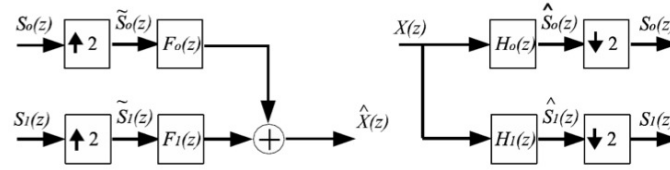
Figure 5.7: Application: transmultiplexer

In practice, the transmission of the signal in the channel is subject to convolutive effects. The problem posed to preserve the PR property of the transmultiplexer is then called *channel equalization*: it consists in estimating and inverting the impulse response of the channel at the decoder.

7   Filter banks: from 2 channels to M channels

We are now interested in designing filter banks with an arbitrary number M of channels. The first approach presented in Section 7.1 is based on the 2-channel filter banks that we have introduced in Section 2.

7.1   Pyramidal structure

Suppose that the number of channels M is a power of 2. Then it is possible to chain several 2-channel analysis filter banks according to a pyramidal structure, in order to produce an M-channel analysis filter bank. An example with $M = 4$ and a pyramid with two levels is represented in Figure 5.8. In the same way, the M-channel synthesis filter bank can be obtained by chaining several 2-channel synthesis filter banks according to the reverse pyramidal structure. If the 2-channel filter bank is designed to guarantee PR, then the M-channel filter bank obtained in this way also guarantees PR. If the 2-channel filter bank involves half-band filters as in Section 4, then the pyramidal structure decomposes the input signal into M uniform sub-bands (of spectral width $1/M$).
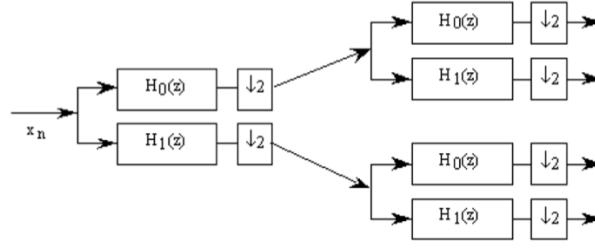
Figure 5.8: Pyramidal structure

7.2 M-channel filter banks

We now address the general theory of M-channel filter banks, with an arbitrary number $M \geq 2$ of channels. The general form of an M-channel filter bank is depicted in Figure 5.9.
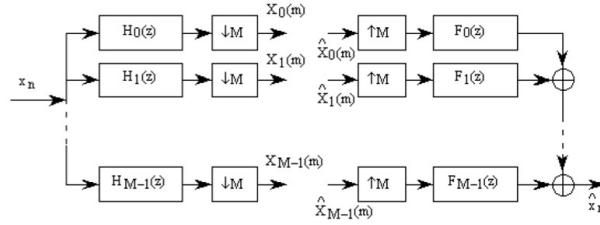
Figure 5.9: General implementation of an M-channel filter bank

From Figure 5.2, we get the following relationships between the Z-transforms of the various signals:

$$X_k(z) = \frac{1}{M} \sum_{l=0}^{M-1} H_k(z^{\frac{1}{M}} W_M^l) X(z^{\frac{1}{M}} W_M^l), \tag{5.16}$$

$$\widehat{X}(z) = \sum_{k=0}^{M-1} F_k(z) X_k(z^M). \tag{5.17}$$

where $W_M^l = e^{-2\iota\pi \frac{l}{M}}$. By substituting equation (5.16) into (5.17), we get the following I/O relationship:

$$\widehat{X}(z) = T(z)X(z) + \sum_{l=1}^{M-1} A_l(z) X(z W_M^l) \tag{5.18}$$

where

$$T(z) = \frac{1}{M} \sum_{k=0}^{M-1} H_k(z) F_k(z), \tag{5.19}$$

$$A_l(z) = \frac{1}{M} \sum_{k=0}^{M-1} H_k(z W_M^l) F_k(z). \tag{5.20}$$

The spurious terms $X(z W_M^l)$ in (5.18) induce spectral aliasing in $\widehat{X}(z)$. In order to remove these terms, the AC condition is simply $\forall l \in \{0 \ldots M-1\}$, $A_l(z) = 0$. Then we get PR iff $T(z) = cz^{-n_0}$ with $c \neq 0$ and $n_0 \in \mathbb{Z}$.

By substituting equations (5.19) and (5.20) into the AC and PR conditions, we get the following linear system of equations:

$$\underbrace{\begin{bmatrix} H_0(z) & \dots & H_{M-1}(z) \\ H_0(zW_M^1) & \dots & H_{M-1}(zW_M^1) \\ \vdots & \dots & \vdots \\ H_0(zW_M^{M-1}) & \dots & H_{M-1}(zW_M^{M-1}) \end{bmatrix}}_{\mathbf{H}_M^\top(z)} \underbrace{\begin{bmatrix} F_0(z) \\ F_1(z) \\ \vdots \\ F_{M-1}(z) \end{bmatrix}}_{\mathbf{f}(z)} = \underbrace{\begin{bmatrix} Mcz^{-n_0} \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{t}(z)}$$

whose solution is $\mathbf{f}(z) = \left(\mathbf{H}_M^\top(z)\right)^{-1} \mathbf{t}(z)$.

If in addition we want both filters H_k and F_k to have FIR, the two conditions become:

$$\text{AC} \quad : \quad \mathbf{f}(z) = \text{Adj}(\mathbf{H}_M(z))\,\mathbf{e}_1, \tag{5.21}$$

$$\text{PR} \quad : \quad T(z) = \tfrac{1}{M}\det(\mathbf{H}_M(z)) = cz^{-n_0}. \tag{5.22}$$

where $\mathbf{e}_1$ is the vector $[1, 0 \dots 0]^\top$, and $\text{Adj}(\mathbf{H}_M(z))$ denotes adjugate matrix, that is the transpose of the cofactor matrix, so that $(\mathbf{H}_M(z))^{-1} = \frac{\text{Adj}(\mathbf{H}_M(z))^\top}{\det(\mathbf{H}_M(z))}$.

7.3 Polyphase implementation

For every $k \in \{0 \dots M-1\}$, let us now consider the type I polyphase decomposition of the analysis filter H_k at order M:

$$H_k(z) = \sum_{l=0}^{M-1} E_{kl}(z^M)z^{-l}. \tag{5.23}$$

By concatenating equation (5.23) for all $k \in \{0 \dots M-1\}$, we get the relationship

$$\underbrace{\begin{bmatrix} H_0(z) \\ H_1(z) \\ \vdots \\ H_{M-1}(z) \end{bmatrix}}_{\mathbf{h}(z)} = \underbrace{\begin{bmatrix} E_{0,0}(z^M) & E_{0,1}(z^M) & \dots & E_{0,M-1}(z^M) \\ E_{1,0}(z^M) & E_{1,1}(z^M) & \dots & E_{1,M-1}(z^M) \\ \vdots & \vdots & \vdots & \vdots \\ E_{M-1,0}(z^M) & E_{M-1,1}(z^M) & \dots & E_{M-1,M-1}(z^M) \end{bmatrix}}_{\mathbf{E}(z^M)} \underbrace{\begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(M-1)} \end{bmatrix}}_{\mathbf{e}(z)}. \tag{5.24}$$

In the same way, for every $k \in \{0 \dots M-1\}$, let us consider the type II polyphase decomposition of the synthesis filter F_k at order M:

$$F_k(z) = \sum_{l=0}^{M-1} R_{lk}(z^M)z^{-(M-1-l)}. \tag{5.25}$$

By concatenating equation (5.25) for all $k \in \{0 \dots M-1\}$, we get the relationship

$$\underbrace{\begin{bmatrix} F_0(z) \\ F_1(z) \\ \vdots \\ F_{M-1}(z) \end{bmatrix}}_{\mathbf{f}(z)} = \underbrace{\begin{bmatrix} R_{0,0}(z^M) & R_{1,0}(z^M) & \dots & R_{M-1,0}(z^M) \\ R_{0,1}(z^M) & R_{1,1}(z^M) & \dots & R_{M-1,1}(z^M) \\ \vdots & \vdots & \vdots & \vdots \\ R_{0,M-1}(z^M) & R_{1,M-1}(z^M) & \dots & R_{M-1,M-1}(z^M) \end{bmatrix}}_{\mathbf{R}^\top(z^M)} \underbrace{\begin{bmatrix} z^{-(M-1)} \\ z^{-(M-2)} \\ \vdots \\ 1 \end{bmatrix}}_{\widetilde{\mathbf{e}}(z)}. \tag{5.26}$$

With matrices $\mathbf{E}(z^M)$ and $\mathbf{R}(z^M)$ defined in equations (5.24) and (5.26), the M-channel filter bank depicted in Figure 5.9 can also be equivalently represented as in Figure 5.10.
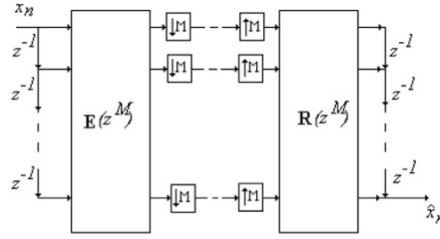
Figure 5.10: Polyphase implementation

By using the noble identities, the implementation in Figure 5.10 can be turned into the efficient implementation represented in Figure 5.11.
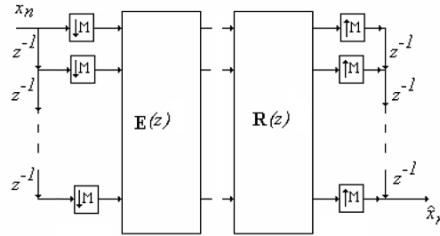
Figure 5.11: Efficient implementation

Then the transfer function as defined in equation (5.19) can be rewritten in the following form:

$$T(z) = \frac{1}{M}\mathbf{f}(z)^\top \mathbf{h}(z) = \frac{1}{M}\widetilde{\mathbf{e}}(z)^\top \underbrace{\mathbf{R}(z^M)\mathbf{E}(z^M)}_{\mathbf{P}(z^M)}\mathbf{e}(z). \tag{5.27}$$

where $\widetilde{\mathbf{e}}(z) = \left[z^{-(M-1)}, \dots, z^{-1}, 1\right]^\top$. With the new matrix $\mathbf{P}(z^M)$ introduced in equation (5.27), the diagram in Figure 5.11 can be simplified as in Figure 5.12.
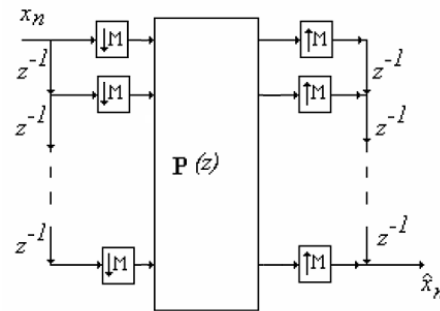
Figure 5.12: Equivalent implementation

Finally, the PR condition (5.22) is fulfilled if we choose the analysis and synthesis filters so that

$$\mathbf{P}(z) = c z^{-n'_0}\mathbf{I}_M \tag{5.28}$$

Roland Badeau `roland.badeau@telecom-paris.fr`

Contexte académique } sans modifications
Voir Page 79   60/79

TELECOM
Paris

IP PARIS

with $n'_0 \in \mathbb{Z}$. Indeed, in this case equation (5.27) implies $T(z) = \frac{1}{M}\mathbf{f}(z)^\top \mathbf{h}(z) = \frac{1}{M}\widetilde{\mathbf{e}}(z)^\top \mathbf{P}(z^M)\mathbf{e}(z) = cz^{-n_0}$ with $n_0 = Mn'_0 + M - 1$.

7.4 Paraunitary filter banks

In this section, we intend to generalize the notion of CQF filter banks introduced in Section 4 to the general case of M-channel filter banks.

We first introduce the filter matrix $\widetilde{\mathbf{E}}(z) = \mathbf{E}^H(z^{-1})$, which is called the *para-conjugate* of matrix $\mathbf{E}(z)$. Then we assume that matrix $\mathbf{E}(z)$ is *paraunitary*, i.e. $\mathbf{E}(z)\widetilde{\mathbf{E}}(z) = c\,\mathbf{I}_M$. Paraunitarity is a generalization of the notion of *unitary matrix* to filter matrices. Then we can define $\mathbf{R}(z) = z^{-\left(\frac{N}{M}-1\right)}\widetilde{\mathbf{E}}(z)$ where N is a multiple of M. We thus have $\mathbf{P}(z) = \mathbf{R}(z)\mathbf{E}(z) = c\,z^{-\left(\frac{N}{M}-1\right)}\mathbf{I}_M$, which corresponds to equation (5.28) with $n'_0 = \frac{N}{M} - 1$. We thus retrieve the PR condition $T(z) = c\,z^{-n_0}$ with $n_0 = N - 1$. Finally, equation (5.26) shows that $\mathbf{f}(z) = \mathbf{R}^\top(z^M)\widetilde{\mathbf{e}}(z) = z^{-(N-M)}\mathbf{E}^*(z^{-M})\widetilde{\mathbf{e}}(z)$, which proves that

$$\forall k \in \{0\ldots M-1\},\ F_k(z) = z^{-(N-1)}\widetilde{H}_k(z).$$

This is a generalization to M-channel filter banks of equation (5.11), that we obtained in the framework of CQF filter banks. Finally, equation (5.27) proves that $T(z) = \frac{1}{M}z^{-(N-1)}\sum_{k=0}^{M-1} H_k(z)\widetilde{H}_k(z)$, which generalizes equation (5.12).

7.4.1 Synthesis based on M-th band filters

The synthesis of the analysis filters follows the same principle as that of the CQF filter banks presented in Section 4.2: we first synthesize a low-pass, M-th band, linear phase FIR filter $G(z)$ (with a cut-off frequency of $\frac{1}{2M}$) by means of the Remez method. This filter satisfies the M-th band condition:

$$\forall \nu \in \mathbb{R},\ \frac{1}{M}\sum_{k=0}^{M-1} G_R\left(\nu - \frac{k}{M}\right) = c$$

with $c > 0$. Then this filter is raised so as to produce a non-negative frequency response $G_R^+(\nu)$. Finally, the transfer function is factorized as $G_R^+(z) = H_0(z)\widetilde{H}_0(z)$, so as to get a low-pass, M-th band, minimal phase filter $H_0(z)$ (whose phase is not linear). The other analysis filters for $k \in \{1\ldots M\}$ are obtained by factorizing $G_R^+(zW_M^k) = H_k(z)\widetilde{H}_k(z)$.

7.4.2 Other example: DFT filter bank

Another possible construction of paraunitary filter bank relies on the polyphase component matrix $\mathbf{E}(z) = \mathbf{W}^H$ where $\mathbf{W}$ is the $M \times M$ Discrete Fourier Transform (DFT) matrix. With $N = M$, we then get $\mathbf{R}(z) = \widetilde{\mathbf{E}}(z) = \mathbf{W}$. Therefore $\mathbf{P}(z) = M\mathbf{I}_M$. We thus get perfect reconstruction, with $T(z) = z^{-n_0}$ with $n_0 = M - 1$.

This particular *DFT filter bank* corresponds to the STFT with non-overlapping rectangular windows, as illustrated in Figure 5.13.
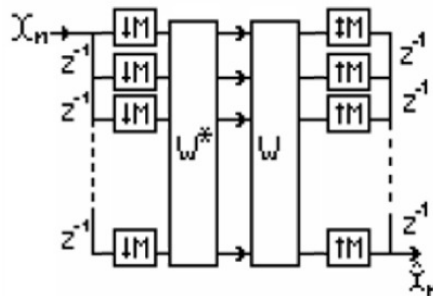
Figure 5.13: Rectangular window STFT

Next section introduces the general class of DFT filter banks, which in general are *not* paraunitary filter banks.

7.5 DFT filter bank

DFT filter banks involve a polyphase component matrix of the form $\mathbf{E}(z) = \mathbf{W}^H \text{diag}(E_0(z) \ldots E_{M-1}(z))$, as depicted in Figure 5.14.
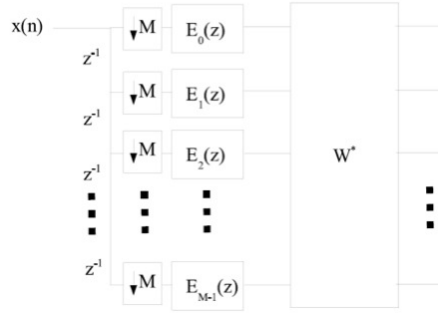
Figure 5.14: Diagram of DFT filter bank

Then equation (5.24) implies $H_0(z) = \sum_l E_l(z^M))z^{-l}$ (i.e. filters $E_l(z)$ are the type I polyphase components of filter $H_0(z)$ at order M), and $\forall k \in \{0 \ldots M - 1\}$, $H_k(z) = H_0(zW_N^k)$. The latter property, depicted in Figure 5.15, is the same one as that already encountered when studying the filter bank interpretation of the STFT. Indeed, DFT filter banks are formally equivalent to the *critically sampled* STFT (i.e. such that the number of sub-bands is equal to the decimation factor M). However, in this critically sampled case, the perfect reconstruction condition of the STFT only allows trivial solutions (with constant filters $E_l(z)$, as in Section 7.4.2). Thus in the general case, DFT filter banks cannot achieve perfect reconstruction. Note that the perfect reconstruction condition of the STFT admits non-trivial solutions when the number of sub-bands is higher than the decimation factor M, which goes beyond the framework of DFT filter banks.
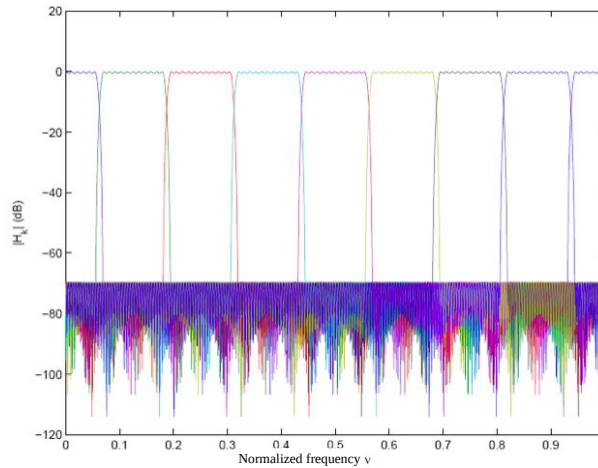
Figure 5.15: Frequency responses of DFT filter bank

Bibliography

[OS75]   A. V. Oppenheim and R. W. Schafer. *Digital Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1975.

[RG75]   L. R. Rabiner and B. Gold. *Theory and Application of Digital Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1975.

[RMP75] L.R. Rabiner, J.H. McClellan, and T.W. Parks. FIR digital filter design techniques using weighted Chebyshev approximation. *Proceedings of the IEEE*, 63:595–610, April 1975.

[Vai93] P.P. Vaidyanathan. *Multirate systems and filter banks*. Signal Processing Series. Prentice Hall, 1993.

Roland Badeau   roland.badeau@telecom-paris.fr

Tutorial on filter synthesis

Roland Badeau

1 Rejector filter

Let us consider the transfer function $H(z) = \frac{1-2\cos(\theta)z^{-1}+z^{-2}}{1-2\rho\cos(\theta)z^{-1}+\rho^2 z^{-2}}$, with $0 < \rho < 1$. Check that $H(z)$ can be factorized in the form $H(z) = \frac{(1-z_0 z^{-1})(1-z_0^* z^{-1})}{(1-\rho z_0 z^{-1})(1-\rho z_0^* z^{-1})}$, where z_0 is to be expressed as a function of θ. What is the normalized frequency rejected by this filter? What is the domain of convergence of its stable implementation? Is this implementation causal? Write the corresponding input/output relationship.

2 Hilbert filter

Let $x_a(t)$ be a continuous time (analog) real signal. The *analytic* signal associated to $x_a(t)$ is the signal $z_a(t)$ whose the CTFT is expressed as $Z_a(f) = 2 U_a(f) X_a(f)$, where $U_a(f)$ is the unit step function, whose value is 1 for $f > 0$, and 0 for $f < 0$. For continuity reasons, we assume that $U_a(0) = \frac{1}{2}$. The filter of frequency response $2U_a(f)$ is referred to as the *analytic filter*.

1. Which property does function $X_a(f)$ satisfy? Deduce the expression of $\frac{1}{2}(Z_a(f) + Z_a^*(-f))$ as a function of $X_a(f)$, and prove that the real part of $z_a(t)$ is equal to $x_a(t)$. We can then write $z_a(t) = x_a(t) + i y_a(t)$, where the real signal $y_a(t)$ is defined as the imaginary part of $z_a(t)$.

2. Prove that $y_a(t)$ can be obtained from $x_a(t)$ by linear filtering of frequency response $H_a(f) = -i\,\text{sign}(f)$, where $\text{sign}(f) = 1$ for $f > 0$, $\text{sign}(f) = -1$ for $f < 0$, and $\text{sign}(0) = 0$. Filter $H_a(f)$ is referred to as the *Hilbert filter*, and $y_a(t)$ is called the *Hilbert transform* of $x_a(t)$.

Let us assume that the signal $x_a(t)$ satisfies the assumptions of the sampling theorem: there exists a frequency F_s such that the support of $X_a(f)$ is included in $]-\frac{F_s}{2}, \frac{F_s}{2}[$. We then consider the sampled signals $x(n) = x_a(nT_s)$ and $y(n) = y_a(nT_s)$, where $T_s = 1/F_s$. We remind the relationship between the DTFT $X(e^{2i\pi\nu})$ and the CTFT $X_a(f)$:

$$X(e^{2i\pi\nu}) = \frac{1}{T_s} \sum_{k\in\mathbb{Z}} X_a\left(\frac{\nu+k}{T_s}\right) \tag{1}$$

3. Simplify the expression (1) when $\nu \in\,]-\frac{1}{2}, \frac{1}{2}[$. Check that $Y(e^{2i\pi\nu})$ satisfies a similar expression. Deduce that the signal $y(n)$ can also be expressed as the output of the discrete filter of frequency response $H(e^{2i\pi\nu}) = -i\,\text{sign}(\nu)$ for $\nu \in\,]-\frac{1}{2}, \frac{1}{2}[$ (and $H(e^{2i\pi\nu}) = 0$ for $\nu = \pm\frac{1}{2}$), applied to the input signal $x(n)$.

Remark: the discrete filter $H(e^{2i\pi\nu})$ allows us to directly compute the samples $y(n)$ of the Hilbert transform from the samples $x(n)$, without having to perform a digital/analog conversion.

4. By applying the inverse DTFT, prove that the impulse response $h(n)$ satisfies $h(n) = \frac{2}{\pi n}$ if n is odd, and 0 if n is even.

Roland Badeau

Contexte académique } sans modifications
Voir Page 2   1/2

TELECOM
Paris

IP PARIS

5. Is this filter causal? Stable? Of finite (FIR) or infinite (IIR) impulse response?

6. For a discrete filter of impulse response $g(n)$ and of transfer function $G(z)$, what is the impulse response of the filter of transfer function $G(z^2)$? By using the fact that the even coefficients of $h(n)$ are zero, deduce that there exists a transfer function $G(z)$, such that $H(z) = z^{-1}G(z^2)$. What is the impulse response $g(n)$?

7. We want to approximate the ideal filter $G(z)$ by using the window method, in order to synthesize a linear phase FIR filter, of type 4 (even length N, antisymmetric impulse response $g(n)$). Quickly summarize the principle of the window method, its advantages and its drawbacks.

Contexte académique } sans modifications

Roland Badeau

Tutorial on the conversion of sampling frequency and STFT

Roland Badeau

TELECOM
ParisTech

This tutorial aims to carry out filtering in a multirate system (with an application to the conversion of sampling frequency), and to understand a perfect reconstruction filter bank for audio equalization.

1 Conversion of sampling frequency

We want to achieve the conversion of sampling frequency from $F_s = 48kHz$ to $F_s = 32kHz$.

1. Describe and draw the digital processing chain that will permit you to achieve such a conversion (in particular you need to specify the characteristics of the ideal filter $H(e^{2i\pi\nu})$ that you will have to use).

The purpose of the rest of this section is to get an efficient implementation of this conversion of sampling frequency and to compare its performance with that of a direct implementation.
The efficient implementation will be achieved by means of polyphase decompositions (of type I and type II). The use of noble identities and of the equivalence below (Figure 1) will permit you to apply the filtering operations to signals at a lower sampling frequency.

2. Check the equivalence between the two diagrams in Figure 1.
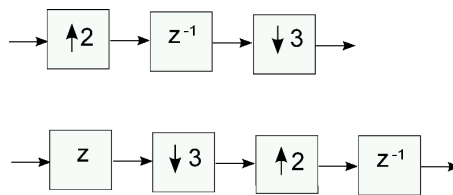
Figure 1: Equivalence

3. Find the efficient implementation of this conversion of sampling rate by using two successive polyphase decompositions and the equivalence in Figure 1.

2 STFT audio equalization

2.1 STFT analysis

The definition of the STFT that is referred to as "low-pass convention" is given in discrete time by :

$$W_x(\lambda, b) = \sum_{n \in \mathbb{Z}} x(n)w(n - b)e^{-j2\pi\lambda n}, \tag{1}$$

where $w(n)$ is the analysis window in discrete time, which is supposed summable, real and symmetric.

1. We choose $w(n)$ as a Hann (Hanning) window of length N_w. What is the width of the main lobe as a function of N_w ?

2. Note that the expression (1), taken at fixed λ, can be written as a convolution and deduce an interpretation of the STFT in terms of filtering. Explain the role of the corresponding filter (low-pass? band-pass? high-pass?). As a linear phase FIR filter, specify its type (type 1,2,3,4?) according to its length (even or odd).

3. Another definition of the STFT, referred to as "band-pass convention", is given by

$$\tilde{X}(\lambda, b) = \sum_{n \in \mathbb{Z}} x(n+b)w(n)e^{-j2\pi\lambda n}. \tag{2}$$

Explain this latter designation and express $\tilde{X}$ as a function of W_x.

2.2 Reconstruction

The reconstruction is achieved by an *overlap-add* operation, which will be written as:

$$y(n) = \sum_{u \in \mathbb{Z}} y_s(u, n - uR),$$

where $y_s(u, n) = \text{DFT}^{-1}[\tilde{X}(k, u)](n) \, w_s(n) = \frac{1}{M} \sum_{k=0}^{M-1} \tilde{X}(k, u)e^{j2\pi\frac{kn}{M}} \times w_s(n)$

4. Show that a sufficient condition for perfect reconstruction is $f(n) = 1 \; \forall n$ where $f(n) = \sum_{u \in \mathbb{Z}} w(n - uR)w_s(n - uR)$.

Contexte académique } sans modifications

Roland Badeau

TELECOM
ParisTech

Tutorial on filter banks

Roland Badeau

1 CQF filter bank

A two-channel filter bank is defined by the diagram in Figure 1.
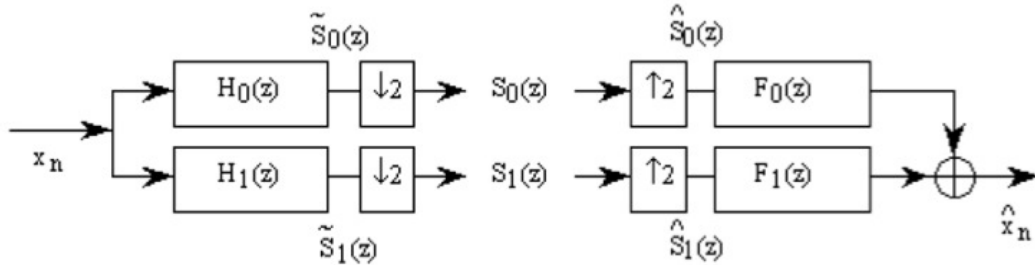
Figure 1: General diagram of a two-channel filter bank.

1. Express $\hat{X}(z)$ as a function of $X(z)$.

2. Deduce that the aliasing cancellation (AC) conditions of a 2-channel filter bank are $F_0(z) = H_1(-z)$ and $F_1(z) = -H_0(-z)$, and that its transfer function (TF) is $T(z) = \frac{1}{2}(H_0(z)F_0(z) + H_1(z)F_1(z))$.

3. Now we assume that H_0 and H_1 are *conjugate quadrature filters* (CQF): $H_1(z) = -z^{-(N-1)}\widetilde{H}_0(-z)$ where $\tilde{H}_0(z) = H_0^*(\frac{1}{z})$ and N is even. Prove that equations (AC) and (CQF) imply that $\forall k \in \{0, 1\}$, $F_k(z) = z^{-(N-1)}\widetilde{H}_k(z)$: we say that the analysis and synthesis filters are *paraconjugate* (PC).

4. Finally we assume that $H_0(z)$ is a *symmetric power* (SP) filter: $\widetilde{H}_0(z)H_0(z) + \widetilde{H}_0(-z)H_0(-z) = 2c$. Prove that equations (TF), (CQF), (PC) et (SP) imply that $T(z) = cz^{-(N-1)}$: the CQF filter bank guarantees perfect reconstruction.

2 Transmultiplexer

We implement the transmultiplexer represented in Figure 2 by means of the filters defined in Exercise 1.
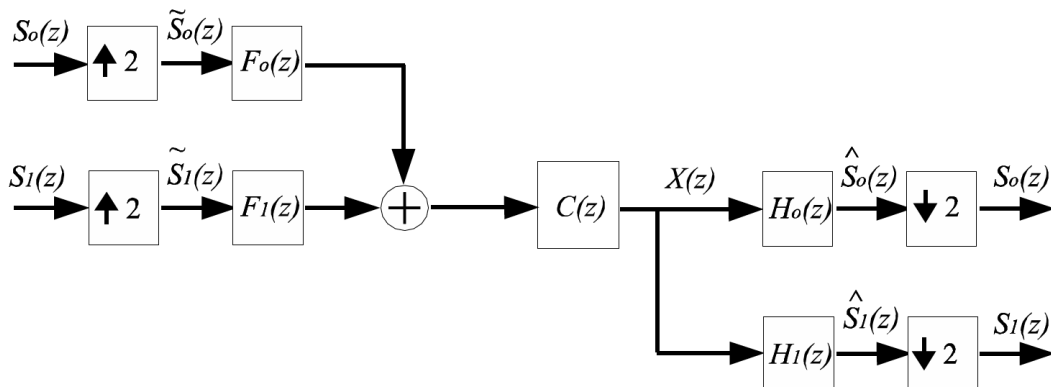
Figure 2: Transmultiplexer.

Roland Badeau

TELECOM
Paris

IP PARIS

1. Prove that the transmultiplexer guarantees perfect reconstruction at the output when $C(z) = z^{-1}$.

2. From now on, filter $C(z)$ will represent the transfer function of a transmission channel between the encoder and the decoder. It is uniformly equal to 1 if the channel is transparent, but in general, the transmission channel is imperfect, and its transfer function $C(z)$ is not constant. In order to simplify, let us assume that $C(z)$ is of the form $1 - \alpha z^{-1}$. In order to keep the perfect reconstruction property at the output of the transmultiplexer, we will have to introduce just after $C(z)$ a causal filter $D(z)$ such that $C(z)D(z) = dz^{-n_0}$, where n_0 is an odd number.

   (a) How to choose $D(z)$ if $\alpha = 0.9$?

   (b) What problem do we encounter if $\alpha = 1.2$? Propose an approximate solution.

3. If the channel transfer function $C(z)$ is unknown, propose a method for estimating it from the output signals, by choosing appropriate filter $D(z)$ and input signals s_0 and s_1.

Contexte académique } sans modifications

Roland Badeau

Practical work on the conversion of sampling rate and STFT

Roland Badeau

This practical work aims to carry out filtering in a multirate system (with an application to the conversion of sampling rate), and to understand and implement a perfect reconstruction filter bank for audio equalization. The files related to this practical work are available on the eCampus website of TSIA201.

1 Conversion of sampling rate

We want to achieve the conversion of sampling rate from $Fs = 48kHz$ to $Fs = 32kHz$.

1. Synthesize an impulse reponse $h(n)$ appropriate for this conversion of sampling rate. You can use the following implementations of the Remez method:

   - Matlab function `firpm`: `[h,delta] = firpm(odd order,2*[0,nu_c,nu_a,.5],[L L 0 0])`
   - Python function `remez`: `h = scipy.signal.remez(even length,[0,nu_c,nu_a,.5],[L 0])`

   Plot the frequency response $|H(e^{2i\pi\nu})|$ to ensure your synthesis is correct (we expect a difference of at least 50 dB between the pass-band and the stop-band).

2. Program the simplest digital processing chain that will permit you to achieve the conversion. You can use the Matlab function `filter` or the Python function `scipy.signal.lfilter`.

3. Program the efficient implementation of this conversion of sampling rate by using two successive polyphase decompositions.

4. Compare the results obtained with the two implementations and compare the running times of the two approaches (you can use the Matlab functions `tic` and `toc`, or the Python function `time.time()`).

2 STFT audio equalization

2.1 STFT analysis

In Matlab, you will use the program `stft.m`, which provides a framework for the computation of the short time Fourier transform (STFT). In Python, you can use the provided notebook template.
The definition of the STFT that is referred to as "low-pass convention" is given in discrete time by :

$$W_x(\lambda, b) = \sum_{n\in\mathbb{Z}} x(n)w(n-b)e^{-j2\pi\lambda n}, \tag{1}$$

where $w(n)$ is the analysis window in discrete time, which is supposed summable, real and symmetric.

1. We choose $w(n)$ as a Hann (Hanning) window of length N_w. Plot the DFT of w, i.e. $\hat{w}(k/M)$ where $M, M \geq N_w$ is the order of the DFT.

2. Another definition of the STFT, referred to as "band-pass convention", is given by

$$\tilde{X}(\lambda, b) = \sum_{n\in\mathbb{Z}} x(n+b)w(n)e^{-j2\pi\lambda n}. \tag{2}$$

   Which one of these two conventions correspond to the computation implemented in `stft.m` or in the notebook template?

From now on, the computation of the STFT will be carried out by means of fast Fourier transforms (FFT), by only considering causal windows of finite length, lower than the order M of the DFT. In order to simplify, the STFT will now be indexed by the index k of the frequency channel located around $\lambda = k/M$ and the temporal index u of the considered frame, i.e.

$$\tilde{X}(k, u) = \sum_{n \in \mathbb{Z}} x(n + uR)w(n)e^{-j2\pi \frac{kn}{M}}. \tag{3}$$

where R represents the temporal shift between successive analysis frames.

3. Based on the Matlab script `stft.m` or on the Python notebook template, for $M = 32$ and $R = 1$, compute the signal $x_k(u) = \tilde{X}(k, u)$, for $k = 3$. Is it real or complex ? Check the interpretation in terms of filtering by using the Matlab functions `filter` and `spectrogram` or the Python functions `scipy.signal.lfilter` and `scipy.signal.spectrogram`. Listen to Re(x_k).

2.2 Reconstruction

The reconstruction is achieved by an *overlap-add* operation, which will be written as:

$$y(n) = \sum_{u \in \mathbb{Z}} y_s(u, n - uR),$$

where $y_s(u, n) = \mathrm{DFT}^{-1}[\tilde{X}(k, u)](n) \, w_s(n) = \frac{1}{M} \sum_{k=0}^{M-1} \tilde{X}(k, u)e^{j2\pi \frac{kn}{M}} \times w_s(n)$

4. A sufficient condition for perfect reconstruction is $f(n) = 1 \; \forall n$ where $f(n) = \sum_{u \in \mathbb{Z}} w(n - uR)w_s(n - uR)$. Use the program `ola.m` or function `ola` in the notebook template to check this condition for the product window $w_p(n) = w(n)w_s(n) = h(n)^2$, where $h(n)$ is a Hann window correctly normalized, and for a 75% overlap.

5. In the program `stft.m` or in the notebook template, implement the resynthesis according to the overlap-add approach described in question (e), and check the validity of perfect reconstruction.

2.3 STFT audio equalizer

We exploit the previous results to use the STFT as an equalizer with $M/2 + 1$ real channels (the order M is supposed even). We will then get the synthesis frames as $y_s(u, n) = \mathrm{DFT}^{-1}[\tilde{Y}(k, u)](n) \, w_s(n)$, where $\tilde{Y}$ is obtained by assigning weights to every channel, i.e. $\tilde{Y}(k, u) = w_k \tilde{X}(k, u)$. The perfect reconstruction condition then ensures that $y(n) = x(n)$ when $w_k = 1 \; \forall k$.

6. Implement the equalizer and test it.

Roland Badeau

Contexte académique } sans modifications
Voir Page 3

2/3

TELECOM
ParisTech

Contexte académique } sans modifications

Roland Badeau

TELECOM
ParisTech

Examination of the teaching unit
Représentations des signaux - TSIA201

Roland Badeau, Marco Cagnazzo

Friday, November 6, 2020

Duration: 1:30
All documents are permitted. However electronic devices (including calculators) are forbidden.
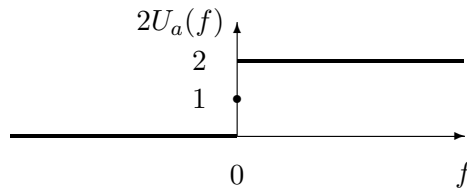◇◇◇

1 Rejector filter

Let us consider the transfer function $H(z) = \frac{1-2\cos(\theta)z^{-1}+z^{-2}}{1-2\rho\cos(\theta)z^{-1}+\rho^2 z^{-2}}$, with $0 < \rho < 1$. Check that $H(z)$ can be factorized in the form $H(z) = \frac{(1-z_0 z^{-1})(1-z_0^* z^{-1})}{(1-\rho z_0 z^{-1})(1-\rho z_0^* z^{-1})}$, where z_0 is to be expressed as a function of θ. What is the normalized frequency rejected by this filter? What is the domain of convergence of its stable implementation? Is this implementation causal? Write the corresponding input/output relationship.

---

2 Hilbert filter

Let $x_a(t)$ be a continuous time (analog) real signal. The *analytic* signal associated to $x_a(t)$ is the signal $z_a(t)$ whose the CTFT is expressed as $Z_a(f) = 2\,U_a(f)X_a(f)$, where $U_a(f)$ is the unit step function, whose value is 1 for $f > 0$, and 0 for $f < 0$. For continuity reasons, we assume that $U_a(0) = \frac{1}{2}$. The filter of frequency response $2U_a(f)$ is referred to as the *analytic filter*.

1. Which property does function $X_a(f)$ satisfy? Deduce the expression of $\frac{1}{2}(Z_a(f)+Z_a^*(-f))$ as a function of $X_a(f)$, and prove that the real part of $z_a(t)$ is equal to $x_a(t)$. We can then write $z_a(t) = x_a(t) + iy_a(t)$, where the real signal $y_a(t)$ is defined as the imaginary part of $z_a(t)$.

2. Prove that $y_a(t)$ can be obtained from $x_a(t)$ by linear filtering of frequency response $H_a(f) = -i\,\text{sign}(f)$, where $\text{sign}(f) = 1$ for $f > 0$, $\text{sign}(f) = -1$ for $f < 0$, and $\text{sign}(0) = 0$. Filter $H_a(f)$ is referred to as the *Hilbert filter*, and $y_a(t)$ is called the *Hilbert transform* of $x_a(t)$.

1

Let us assume that the signal $x_a(t)$ satisfies the assumptions of the sampling theorem: there exists a frequency F_s such that the support of $X_a(f)$ is included in $]-\frac{F_s}{2}, \frac{F_s}{2}[$. We then consider the sampled signals $x(n) = x_a(nT_s)$ and $y(n) = y_a(nT_s)$, where $T_s = 1/F_s$. We remind the relationship between the DTFT $X(e^{2i\pi\nu})$ and the CTFT $X_a(f)$:

$$X(e^{2i\pi\nu}) = \frac{1}{T_s} \sum_{k \in \mathbb{Z}} X_a\left(\frac{\nu + k}{T_s}\right) \tag{1}$$

3. Simplify the expression (1) when $\nu \in]-\frac{1}{2}, \frac{1}{2}[$. Check that $Y(e^{2i\pi\nu})$ satisfies a similar expression. Deduce that the signal $y(n)$ can also be expressed as the output of the discrete filter of frequency response $H(e^{2i\pi\nu}) = -i\,\text{sign}(\nu)$ for $\nu \in]-\frac{1}{2}, \frac{1}{2}[$ (and $H(e^{2i\pi\nu}) = 0$ for $\nu = \pm\frac{1}{2}$), applied to the input signal $x(n)$.

Remark: the discrete filter $H(e^{2i\pi\nu})$ allows us to directly compute the samples $y(n)$ of the Hilbert transform from the samples $x(n)$, without having to perform a digital/analog conversion.

4. By applying the inverse DTFT, prove that the impulse response $h(n)$ satisfies $h(n) = \frac{2}{\pi n}$ if n is odd, and 0 if n is even.

5. Is this filter causal? Stable? Of finite (FIR) or infinite (IIR) impulse response?

6. For a discrete filter of impulse response $g(n)$ and of transfer function $G(z)$, what is the impulse response of the filter of transfer function $G(z^2)$? By using the fact that the even coefficients of $h(n)$ are zero, deduce that there exists a transfer function $G(z)$, such that $H(z) = z^{-1}G(z^2)$. What is the impulse response $g(n)$?

7. We want to approximate the ideal filter $G(z)$ by using the window method, in order to synthesize a linear phase FIR filter, of type 4 (even length N, antisymmetric impulse response $g(n)$). Quickly summarize the principle of the window method, its advantages and its drawbacks.

8. Now, we want to prove that the following diagram provides an efficient implementation of the discrete Hilbert filter $H(z)$:
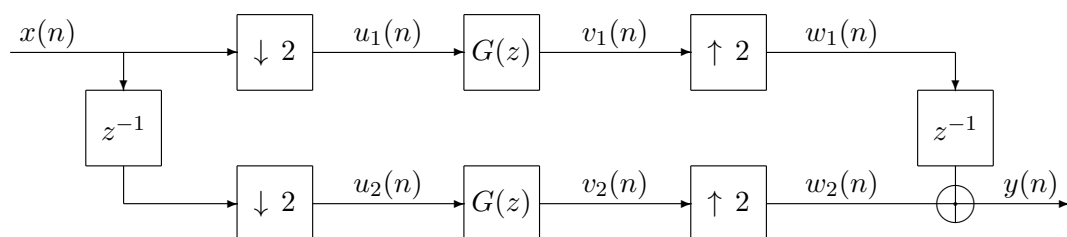
We remind that $U_1(z) = \frac{1}{2}(X(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}}))$. Express $U_2(z)$ as a function of $X(z)$, then $V_1(z)$ and $V_2(z)$ as a function of $U_1(z)$ and $U_2(z)$, then $W_1(z)$ and $W_2(z)$ as a function of $V_1(z)$ and $V_2(z)$, and finally $Y(z)$ as a function of $W_1(z)$ and $W_2(z)$. By substitution, retrieve the relationship $Y(z) = H(z)X(z)$.

3 MRA and Wavelet transform

1. Show that, for the Haar filter ($h_0[0] = h_0[1] = \frac{\sqrt{2}}{2}$, the other coefficients are zero), a suitably scaled indicator function of the interval $(0, 1)$ is the father wavelet for an MRA (hint: use the dilation equation). Propose a mother wavelet function and show that with such a choice of ψ the wavelet equation holds.

2. Let $\phi(t)$ be following function, see also Fig.3:

$$\phi(t) = \begin{cases} t & \text{if } t \in (0, 1) \\ 2 - t & \text{if } t \in (12) \\ 0 & \text{otherwise} \end{cases}$$

It can be shown that this ϕ is the father wavelet of an MRA. We say that $\phi \in V_0$ and that the integer-shifts of ϕ produce a basis of V_0. By using the dilation equation, show (by sketching the graphs of $c_k \phi(2t - k)$) that , with suitable normalization, the representation of ϕ on V_1 is given with coefficients c_k as follows:

$$c_k = \begin{cases} 1/2 & \text{if } k = 0 \\ 1 & \text{if } k = 1 \\ 1/2 & \text{if } k = 2 \\ 0 & \text{otherwise} \end{cases}$$

Find the coefficients d_k from the coefficients c_k (normalization can be ignored for simplicity); finally, find the corresponding mother wavelet ψ by using the wavelet equation.
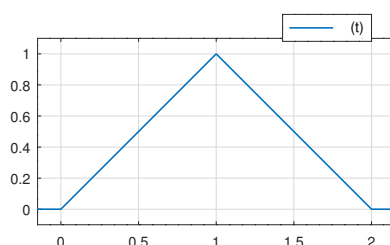

Figure 1: Father wavelet $\phi(t)$

3. We want to compute the projection of a signal $x : t \in \mathbb{R} \to \mathbb{R}$ onto a MRA, but we only have access to the samples $x[n] = x(t)|_{t=n \in \mathbb{Z}}$. Is it correct to use these samples as input of the analysis filterbank? Is there any implicit assumption in doing this, in particular on the father wavelet function ϕ?

———————————