Lecture 2

Introductory signal processing ideas

In this lecture we'll look at some basic properties of finite energy digital signals, in other words sequences $x = \{x_n\}_n$ in ℓ^2; if needed, we could restrict attention to sequences in which only finitely many of the x_n are non-zero, but that's usually not necessary from a strictly mathematical point of view. From a practical point of view, the values of the x_n would almost certainly be real, but it's actually more illuminating to allow complex-valued sequences, and so that's what we shall do. To appreciate fully the mathematical and signal processing ideas, however, you need to keep track of how and where this space ℓ^2 is being used. On some occasions it is a space of finite energy signals, while on others it is a space of *coefficients* of elements of an inner product space with respect to some orthonormal family. This interplay between mathematics and signal processing ideas will be a recurring theme in this and many subsequent lectures!

(2.1) Examples, operators on signals. 1. The simplest such signal is the *unit impulse*

$$\delta = (\ldots, 0, 1, 0, \ldots) = \{\delta_n\}, \qquad \delta_n = \begin{cases} 1, & n = 0, \\ 0, & n \neq 0, \end{cases}$$

i.e., $\delta = \varepsilon^{(0)}$ to use the notation of the previous lecture. Similarly, each of the $\varepsilon^{(n)}$ is a finite energy digital signal in which only one entry is non-zero.

2. Each of $\varphi^{(n)}$ and $\widetilde{\varphi}^{(n)}$ is a finite energy signal such that $E(\varphi^{(n)}) = E(\widetilde{\varphi}^{(n)}) = 1$.

3. There are also *oscillatory* signals: fix a real number ξ_0 and define e_{ξ_0} by

$$(e_{\xi_0})_n = e^{2\pi i n\xi_0}.$$

For no value of ξ_0 does e_{ξ_0} have finite energy, however. Now fix integers k, N, $0 \le k < N$, and define e_w by

$$(e_w)_n = w^n, \qquad w = e^{2\pi i k/N}.$$

Then w is an N^{th} root of unity since $w^N = e^{2\pi i k} = 1$ for each choice of k. Again the digital signal e_w cannot have finite energy, but it has the important property that it is N*-periodic.*

4. By the Integral test in calculus, the sequence $\{f(n)\}_n$ of all integer samples of a function f will be a finite energy digital signal whenever $f(x)$ is monotonic-decreasing as $|x| \to \infty$ and $f = f(x)$ has finite energy; in particular, if $f(x) = 1/(1+|x|)$, then the sequence

$$\left\{ \frac{1}{1+|n|} \right\}_n$$

will have finite energy.

Various linear operators on ℓ^2, will be needed. Engineers make frequent use of the *Delay* operator

$$S : \{x_n\}_n \longrightarrow \{x_{n-1}\}_n\,;$$

mathematicians would call this a *translation* operator on the additive group $\mathbb{Z}$ of integers! Since

$$E(Sa) \;=\; \sum_n |a_{n-1}|^2 \;=\; \sum_n |a_n|^2 \;=\; S(a),$$

S is energy-preserving; notice also that the adjoint, S^*, of S is the advancing operator

$$S^* : \{x_n\}_n \longrightarrow \{x_{n+1}\}_n,$$

translating the sequence in the opposite direction to S. Consequently,

$$\varepsilon^{(n)} \;=\; S^n(\varepsilon^{(0)}) \;=\; S^n(\delta), \qquad \varepsilon^{(-n)} \;=\; (S^*)^n(\varepsilon^{(0)}) \;=\; (S^*)^n\,\delta, \qquad n \geq 0.$$

The most important operator of all, however, is convolution, or *filtering* as engineers call it. Recall first that the convolution $h * x$ of two sequences is defined by

$$(h * x)_n \;=\; \sum_m h_m\, x_{n-m} \;=\; \sum_m h_{n-m}\, x_m\,;$$

in operator terms

$$h * x \;=\; \sum_m h_m\, S^m(x).$$

To check that convolution is well-defined on finite energy signals, observe that by the triangle inequality,

$$\begin{aligned} E^{1/2}(h * x) \;&=\; E^{1/2}\Big(\sum_m h_m\, S^m x\Big) \\ &\leq\; \sum_m \Big\{|h_m|\, E^{1/2}(S^m x)\Big\} \;=\; \Big\{\sum_m |h_m|\Big\} E^{1/2}(x) \end{aligned}$$

since $E(Sx) \;=\; E(x)$. Consequently, the convolution $h * x$ of a finite energy signal x will itself have finite energy provided $\sum_n |h_n| < \infty$, *i.e.*, when the coefficients of h are absolutely convergent; in particular, the convolution $x \longrightarrow h * x$ will map finite energy signals to finite energy ones if only finitely many $h_n \neq 0$.

Now, if the convolution operator $x \longrightarrow h * x$ bounded on ℓ^2 it will have an adjoint. But what form will this adjoint take? Well, given finite energy sequences $x = \{x_n\}_n$ and $y = \{y_n\}_n$,

$$(h * x,\, y) \;=\; \sum_n \Big(\sum_m h_{n-m}\, x_m\Big)\, \overline{y_n}$$

$$= \sum_m \Big(\sum_n h_{n-m}\, \overline{y_n}\Big)\, x_m \;=\; (x,\, h^* * y)$$

where the last convolution is defined by

$$h^* * y \;=\; \sum_n \overline{h_{n-m}}\, y_n, \qquad i.e., \quad h^* \;=\; \{\, \overline{h_{-n}}\,\}_n.$$

(2.2) Discrete-Time Fourier Transform. Given a sequence $a = \{a_n\}_n$, its *Discrete-Time Fourier Transform* $(DTFT)$, $a \to \widehat{a}$, is defined by

$$\widehat{a}(\xi) \;=\; \sum_n a_n\, e^{-2\pi i n \xi}\,;$$

in signal analysis one usually writes $A(\xi)$ instead of $\widehat{a}$ and we shall often follow this convention. The sum makes good sense if only finitely many $a_n \neq 0$. In this case $\widehat{a}(\xi)$ can be interpreted as the inner product

$$\widehat{a}(\xi) \;=\; (a,\, e_\xi) \;=\; A(\xi)$$

of a with the oscillatory signal e_ξ defined in the previous section except that we have to be careful because e_ξ does not have finite energy; in addition, when only finitely many $a_n \neq 0$, it is clear that $\widehat{a}(\xi)$ is a period 1 trigonometric polynomial, hence certainly a continuous function. Later we shall see that it also makes sense when $\sum_n |a_n| < \infty$. It is often useful to think of these period 1 functions $\widehat{a}(\xi)$, $A(\xi)$ as functions on $[-\frac{1}{2},\, \frac{1}{2}]$. Since $\{\, e^{2\pi i n\xi} \,:\, -\infty < n < \infty \,\}$ is orthonormal in $L^2[-\frac{1}{2},\, \frac{1}{2}]$, it follows that

$$E(A) \;=\; \int_{-1/2}^{1/2} \big|A(\xi)\big|^2\, d\xi$$

$$= \int_{-1/2}^{1/2} \Big|\sum_n a_n\, e^{-2\pi i n\xi}\Big|^2 d\xi \;=\; \sum_n |a_n|^2 \;=\; E(a).$$

Consequently, $a \to A(\xi)$ is energy-preserving as a mapping from ℓ^2 into $L^2[-\frac{1}{2},\, \frac{1}{2}]$. One property of the $(DTFT)$ is that

$$\widehat{Sa}(\xi) \;=\; \sum_n a_{n-1}\, e^{-2\pi i n\xi} \;=\; e^{-2\pi i \xi}\, A(\xi),$$

i.e., the Discrete time Fourier transform *maps delay into modulation* which is a signal-processing way of talking about pointwise multiplication by the oscillatory function $e^{-2\pi i\xi}$. But a more crucial property is the following.

Theorem. *The* $(DTFT)$ *maps convolution into pointwise multiplication, more precisely,*

$$\widehat{h*x}(\xi) \;=\; \widehat{h}(\xi)\,\widehat{x}(\xi) \;=\; H(\xi)\,X(\xi)$$

for all real ξ.

Proof: by definition

$$\begin{aligned}\widehat{h*x}(\xi) \;&=\; \sum_n \Bigl(\sum_m \; h_{n-m}\,x_m\Bigr)\,e^{-2\pi in\xi}\\ &=\; \sum_n \Bigl(\sum_m \; h_{n-m}\,x_m\Bigr)\,e^{-2\pi im\xi}\,e^{-2\pi i(n-m)\xi},\end{aligned}$$

which after simplification becomes

$$\widehat{h*x}(\xi) \;=\; \Bigl(\sum_m \; h_{n-m}\,e^{-2\pi i(n-m)\xi}\Bigr)\sum_m \; x_m\,e^{-2\pi im\xi} \;=\; H(\xi)\,X(\xi),$$

completing the proof. $\square$

Corollary. *The Frequency Response function of the adjoint* $h^* \;=\; \{\overline{h_{-n}}\}_n$ *is given by*

$$\widehat{h^*}(\xi) \;=\; \overline{H(\xi)}, \qquad H(\xi) \;=\; \sum_n \; h_n\,e^{2\pi in\xi}.$$

Proof: by definition,

$$\widehat{h^*}(\xi) \;=\; \sum_n \;\overline{h_{-n}}\,e^{-2\pi in\xi} \;=\; \overline{\Bigl\{\sum_n \; h_{-n}\,e^{2\pi in\xi}\Bigr\}} \;=\; \overline{\Bigl\{\sum_n \; h_n\,e^{-2\pi in\xi}\Bigr\}} \;=\; \overline{H(\xi)},$$

completing the proof. $\square$

The adjoint of the $(DTFT)$ defines a linear mapping from $L^2[-\frac{1}{2},\,\frac{1}{2}]$ into ℓ^2. As we shall see in the next lecture this is (almost) the same as the Fourier coefficient mapping. By Parseval's theorem applied to the $(DTFT)$,

$$\begin{aligned}(x,\,y) \;&=\; \int_{-1/2}^{1/2} \; X(\xi)\,\overline{Y(\xi)}\,d\xi \;=\; \int_{-1/2}^{1/2} \Bigl(\sum_n \; x_n\,e^{-2\pi in\xi}\Bigr)\,\overline{Y(\xi)}\,d\xi\\ &=\; \sum_n x_n \Bigl(\int_{-1/2}^{1/2} \; \overline{Y(\xi)}\,e^{-2\pi in\xi}\,d\xi\Bigr) \;=\; \sum_n \; x_n\,\overline{y_n}\end{aligned}$$

for all finite energy sequences x, y. From this it follows that

$$y_n = \int_{-1/2}^{1/2} Y(\xi)\, e^{2\pi i n\xi}\, d\xi,$$

thereby recovering y from its Frequency Response function $Y(\xi)$. In fact, if we get 'clever' here and write

$$(\dagger\dagger) \qquad y = \sum_n \Big(\int_{-1/2}^{1/2} Y(\xi)\, e^{2\pi i n\xi}\, d\xi\Big)\, \varepsilon^{(n)},$$

we can actually regard this as yet another way of *representing* a signal, this time in terms of its Frequency Response function. When a signal is studied using Fourier analysis techniques, $(\dagger\dagger)$ is the most important way of representing a signal.

(2.3) z-transform. The (forward) z-transform, $X(z)$, of a signal $x = \{x_n\}$ is defined by

$$X(z) = \sum_n x_n\, z^{-n}, \qquad z \in \mathbb{C}.$$

As $z = e^{2\pi i\xi}$ on the unit circle in the complex plane,

$$X(e^{2\pi i\xi}) = \sum_n x_n\, e^{-2\pi i n\xi} = \widehat{x}(\xi),$$

in other words, the z-transform of x can be regarded as the extension of the Discrete Fourier transform of x from the unit circle to the whole complex plane $\mathbb{C}$. Perhaps not surprisingly, engineers make a lot of use of complex analysis (poles, zeros, Residue theorem *etc.*) in signal analysis, as does Daubechies at a crucial stage in her construction of wavelets!

Basic to all of sub-band coding are two sample rate changes. These have a particularly informative description as operators on ℓ^2 as well as in terms of Discrete Time Fourier transforms.

(2.4) Sampling rate changes I. *Down-sampling* or *sub-sampling* a signal $x = \{x_n\}_n$ produces a new signal

$$(\downarrow 2)x = \{x_{2n}\}_n,$$

by discarding all *odd-indexed* terms from x and re-indexing; clearly,

$$E((\downarrow 2)x) = \sum_n |x_{2n}|^2 \le \sum_n |x_n|^2 = E(x),$$

so down-sampling is bounded on ℓ^2. Notice that $(\downarrow 2)x = (\downarrow 2)y$ irrespective of the values of their odd-indexed terms $x_{2n+1},\ y_{2n+1}$; thus different sequences may coincide after downsampling. On the other hand, in Fourier terms,

$$\tfrac{1}{2}\Big\{\widehat{x}(\tfrac{1}{2}\xi) + \widehat{x}(\tfrac{1}{2}\xi + \tfrac{1}{2})\Big\} \;=\; \tfrac{1}{2}\Big\{\sum_n\ x_n\, e^{-\pi i n\xi} + \sum_n\ x_n\, e^{-\pi i n(\xi+1)}\Big\}$$

$$=\; \tfrac{1}{2}\Big\{\sum_n\ x_n\, e^{-\pi i n\xi}(1 + (-1)^n)\Big\} \;=\; \sum_n\ x_{2n}\, e^{2\pi i n\xi}.$$

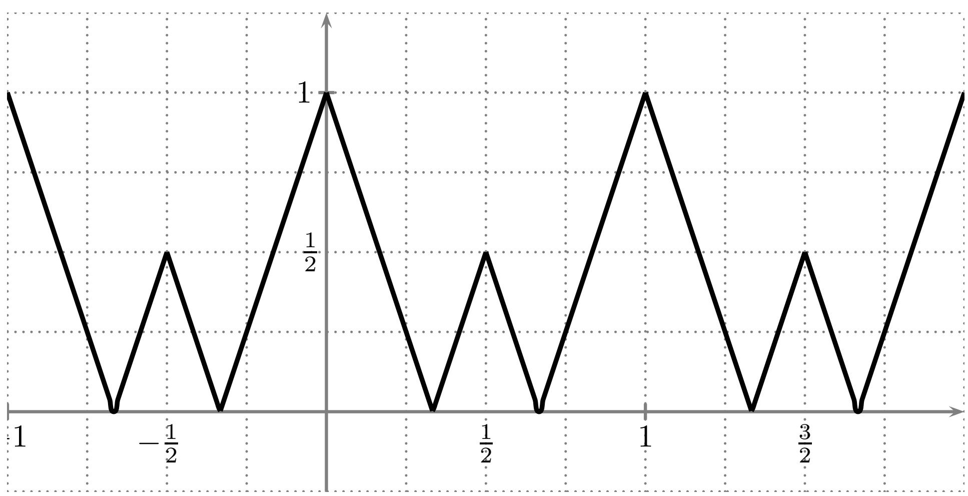
Consequently,

$$\widehat{(\downarrow 2)x}(\xi) \;=\; \tfrac{1}{2}\Big\{\widehat{x}(\tfrac{1}{2}\xi) + \widehat{x}(\tfrac{1}{2}\xi + \tfrac{1}{2})\Big\} \;=\; \tfrac{1}{2}\Big\{X(\tfrac{1}{2}\xi) + X(\tfrac{1}{2}\xi + \tfrac{1}{2})\Big\}.$$

This last function has period 2, so down-sampling increases the period: it doubles it. As an illustration, consider the signal x whose Frequency Response function is the period 1 function

$$X(\xi) \;=\; |1 - 3|\xi||,\ \ 0 \le |\xi| \le \tfrac{1}{2}, \qquad X(\xi + 1) \;=\; X(\xi);$$

its graph is

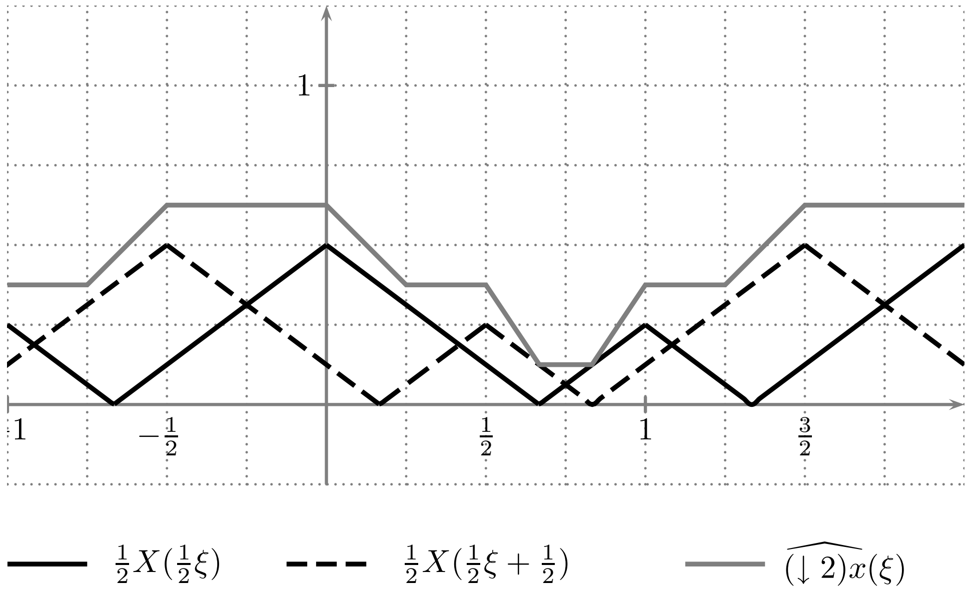


The graph of $X(\xi)$

Now the corresponding functions $X(\frac{1}{2}\xi)$ and $X(\frac{1}{2}\xi + \frac{1}{2})$ are period 2 functions, in fact they can be regarded as *stretched* versions of $X(\xi)$; in addition, the graph of $X(\frac{1}{2}\xi + \frac{1}{2})$ is just the graph of $X(\frac{1}{2}\xi)$ moved to the left by 1. Plotting the graphs of all three of

$$\tfrac{1}{2}X(\tfrac{1}{2}\xi), \qquad \tfrac{1}{2}X(\tfrac{1}{2}\xi + \tfrac{1}{2}), \qquad \widehat{(\downarrow 2)x}(\xi) \;=\; \tfrac{1}{2}X(\tfrac{1}{2}\xi) + \tfrac{1}{2}X(\tfrac{1}{2}\xi + \tfrac{1}{2})$$

on the same axes we thus obtain



The figure makes clear that the graph of $X(\frac{1}{2}\xi+\frac{1}{2})$ is simply the graph of $X(\frac{1}{2}\xi)$ shifted in frequency; signal processing language calls $X(\frac{1}{2}\xi+\frac{1}{2})$ an *alias* of $X(\frac{1}{2}\xi)$.

(2.4) Sampling rate changes II. *Upsampling* is the converse of down-sampling. Given a signal $y = \{y_n\}_n$, upsampling produces a new signal

$$(\uparrow 2)y \;=\; \{v_n\}, \qquad \begin{cases} v_{2n} \;=\; y_n, \\ v_{2n+1} \;=\; 0 \end{cases}$$

by inserting zeros between consecutive terms of y and relabelling. In mathematical terms, upsampling is the adjoint of downsampling: indeed, on ℓ^2,

$$((\downarrow 2)x,\, y) \;=\; \sum_n x_{2n}\,\overline{y_n} \;=\; (x,\, (\uparrow 2)y), \qquad x,\, y \,\in\, \ell^2;$$

in particular, up-sampling is bounded on ℓ^2. More is true, in fact. Since

$$E((\uparrow 2)y) \;=\; \sum_n \Big\{|v_{2n}|^2 + |v_{2n+1}|^2\Big\} \;=\; \sum_n |y_n|^2 \;=\; E(y),$$

it is clearly energy-preserving. On the other hand, in terms of the $(DTFS)$,

$$\widehat{(\uparrow 2)y}(\xi) \;=\; \sum_n y_n\, e^{-2\pi i 2n\xi} \;=\; Y(2\xi),$$

which is now a period $\frac{1}{2}$ function, so upsampling *decreases periodicity*; more precisely: it *halves it*.

Finally, we come to the notion that dominates the theory of wavelets whatever point of view is adopted.

(2.5) Filtering. The term 'filter' suggests 'removal' or 'selection', and this is precisely what a filter does to a signal as will be made precise shortly: think of passing water through a filter to purify it or to make a cup of coffee! We adopt a theoretical definition. A (discrete) *filter* is a *linear, time-invariant operator* acting on ℓ^2; in other words, it is a linear operator $\mathcal{H}$ mapping finite energy signals to finite energy signals and satisfying $\mathcal{H}(Sx) = S(\mathcal{H}(x))$ with respect to the delay operator S. Any such operator is given by convolution:

$$\mathcal{H}:\ x \;=\; \{x_n\}_n \;\longrightarrow\; h * x \;=\; \Big\{\sum_\ell h_\ell\, x_{n-\ell}\Big\}_n$$

with a given sequence $\{h_n\}_n$, the sequence of *filter coefficients*. Another way of writing this is

$$\mathcal{H} \;=\; \sum_\ell h_\ell\, S^\ell, \qquad \mathcal{H}(x) \;=\; \sum_\ell h_\ell\, S^\ell(x),$$

a form which has the advantage of making the time-invariance of $\mathcal{H}$ very clear since

$$S(\mathcal{H}x) \;=\; S\Big(\sum_\ell h_\ell\, S^\ell x\Big) \;=\; \sum_\ell h_\ell\, S^{\ell+1} x \;=\; \Big(\sum_\ell h_\ell\, S^\ell\Big) Sx \;=\; \mathcal{H}(Sx).$$

When only finitely many of the $h_m \neq 0$ it is usual to say that $\mathcal{H}$ is an FIR (Finite Impulse Response) filter, by contrast with the case of an IIR (Infinite Impulse Response) filter where infinitely many of the $h_m \neq 0$. A filter is said to be *causal* when $h_\ell = 0,\ \ell < 0$, the point being that in terms of convolution

$$(\mathcal{H}x)_n \;=\; \sum_{\ell \geq 0} h_\ell\, x_{n-\ell}$$

so $(\mathcal{H}x)_n$ depends only on x_n and *earlier* terms $x_m,\ m < n$, in the signal when $\mathcal{H}$ is causal. Most of the filters $\mathcal{H}$ we shall study are FIR filters, so there is no question that $\mathcal{H}$ is well-defined.

But how does a filter actually 'filter' a signal? Well, the $(DTFT)$ maps convolution to pointwise multiplication, so

$$y \;=\; \mathcal{H}(x) \;=\; h * x \quad\Longrightarrow\quad Y(z) \;=\; H(z)\,X(z), \quad Y(\xi) \;=\; H(\xi)\,X(\xi)$$

where, crucially,

$$H(\xi) \;=\; \sum_{\ell} h_\ell \, e^{-2\pi i \ell \xi}.$$

This function $H(\xi)$ is called the *Frequency Response function* of $\mathcal{H}$. Now we can see how a filter works, for representation (††) ensures that

$$\mathcal{H}(x) \;=\; \sum_n \left(\int_{-1/2}^{1/2} H(\xi)\, X(\xi)\, e^{2\pi i n \xi}\, d\xi \right) \varepsilon^{(n)};$$

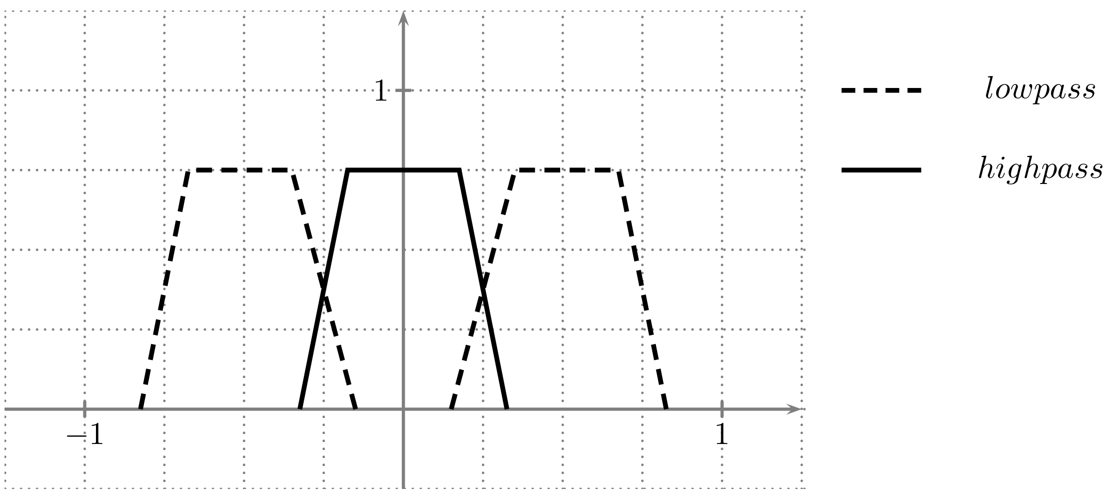
in other words, the action of $\mathcal{H}$ is to *select or reject frequencies* in a signal. Filters come in many 'flavors' depending on how these frequencies are selected or rejected:

(a) *ideal*: $H(\xi)$ takes only the values 0, 1;

(b) *low-pass*: $|\xi| > a \implies H(\xi) = 0$ for some $0 < a < \frac{1}{2}$;

(c) *high-pass*: $|\xi| < a \implies H(\xi) = 0$ for some $0 < a < \frac{1}{2}$;

Typical, but unrealistic, examples are



Thus a low-pass filter keeps only 'low' frequencies in some band about the origin, while a high-pass filter keeps only 'high' frequencies in some band not including the origin. No FIR filters can be low or high pass in this sense, however. Indeed, if $\mathcal{H}$ is, say, a causal FIR filter with filter coefficients $h_0, h_1, \ldots, h_{L-1}$ for some L, then

$$H(z) \;=\; h_0 + \frac{h_1}{z} + \ldots + \frac{h_{L-1}}{z^{L-1}} \;=\; \frac{P(z)}{z^{L-1}}$$

for some polynomial P. So $H(\xi)$ has at most finitely many zeros, hence cannot vanish on any interval. Nonetheless, FIR filters can capture the basic features of low and high-pass filters if 'they try hard'.

Examples. Let's look at a couple of examples to make this clearer.

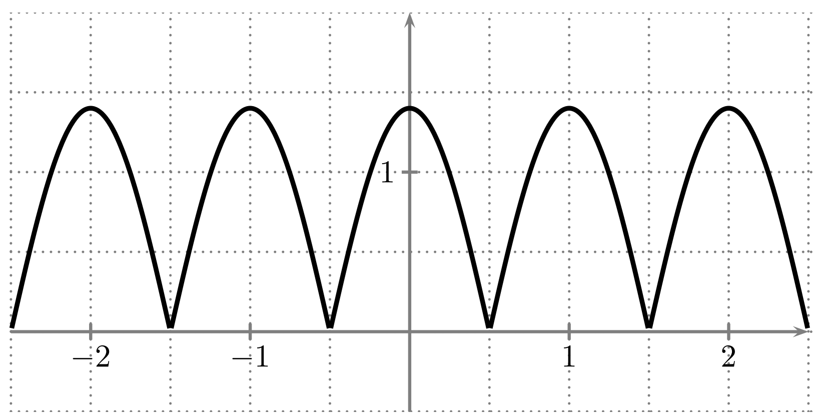
1. The Haar filters: $h = \{h_n\}_n$ *and* $\widetilde{h} = \{\widetilde{h}_n\}_n$ *are causal FIR filters defined by*

$$h_n = \begin{cases} \frac{1}{\sqrt{2}}\,, & n = 0, \\ \frac{1}{\sqrt{2}}\,, & n = 1, \\ 0, & n \neq 0,\, 1, \end{cases} \qquad \widetilde{h}_n = \begin{cases} \frac{1}{\sqrt{2}}\,, & n = 0, \\ -\frac{1}{\sqrt{2}}\,, & n = 1, \\ 0, & n \neq 0,\, 1. \end{cases}$$

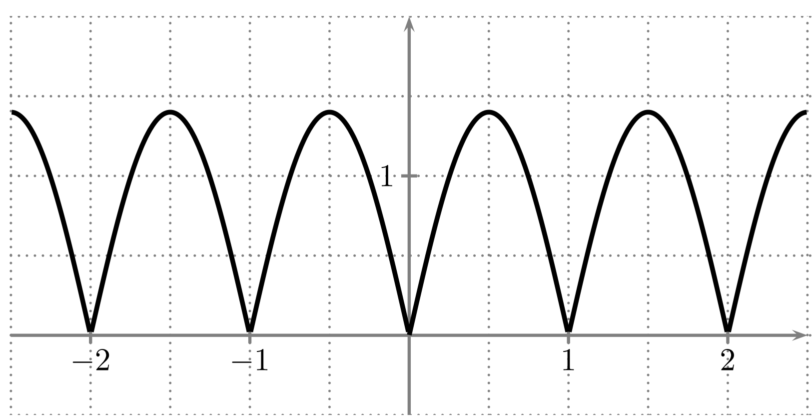
(I realize I'm using the same notation for different things - you have to work out the precise meaning from context!! Shortly everything will settle down and h will mean just one thing.) Simple calculations show that their respective Frequency Response functions $H(\xi)$ and $\widetilde{H}$ are given by

$$H(\xi) = \sqrt{2}\, e^{-\pi i \xi} \cos \pi\xi, \qquad \widetilde{H}(\xi) = \sqrt{2}\, i\, e^{-\pi i \xi} \sin \pi\xi.$$

As often happens with attempts to graph a Fourier transform, however, the presence of complex values forces us to graph *absolute values* of the particular Fourier transform. With that in mind we obtain



for the graph of $|H(\xi)|$ and



for the graph of $|\widetilde{H}(\xi)|$. Thus $|H(\xi)| > 0$ in a neighborhood of $\xi = 0$, while $H(\pm\frac{1}{2}) = 0$ This is about the best a polynomial can do in behaving like a low-pass filter; the only question, and an absolutely key one for the Daubechies filters, is how many zeros does $H(\xi)$ have at $\xi = \pm\frac{1}{2}$? Does it have zeros up to order 1, or 2, or some large finite integer value? In the Haar case, the **order of the zero of $H(\xi)$ at $\xi = \pm\frac{1}{2}$ is one**.

On the other hand, $|\widetilde{H}(\xi)| > 0$ in a neighborhood of $\xi = \pm\frac{1}{2}$, while $\widetilde{H}(0) = 0$. Again, this is the best a polynomial can do in behaving like a high-pass filter, the only question being the order of the zero at $\xi = 0$. In the Haar case, the **order of the zero of $\widetilde{H}(\xi)$ at $\xi = 0$ is one**. At the risk of flippancy, we shall say that a general FIR filter is *trying hard* to be a low or high-pass filter when its Frequency Response function has these properties rather than the more restrictive ones required in the earlier definition.

2. Daubechies *db*2-filter: even at this early stage it's impossible to resist introducing the famous Daubechies-*db*2 filter coefficients - we do resist for the moment saying where they come from, however! There are 4 coefficients and they are defined by

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

with corresponding Frequency response function

$$H(\xi) = \sqrt{2}e^{-3\pi i\xi}\,(\cos\pi\xi)^2\,(\cos\pi\xi - i\sqrt{3}\sin\pi\xi).$$

(To check this expression for $H(\xi)$ it's probably easiest to start with the given expression and then use trig identities to write $H(\xi)$ as a finite sum $\sum_{n=0}^{3} h_n e^{-2\pi in\xi}$.) Clearly *db*2 tries hard to be a low-pass filter. Notice that the presence of the $(\cos\pi\xi)^2$-term ensures that $H(\xi)$ **has a zero of order 2 at $\xi = \pm\frac{1}{2}$**. This is the reason why it is often referred to as the *db*2-filter; the Haar could well be called the *db*1-filter. Others refer to the *db*2-filter as the $D4$-filter because it has 4 coefficients. We have followed Matlab in the choice of notation because you will be making good use of its Wavelet toolbox!

(1.8) Filtering and up/down sampling. Finally, let's put these crucial operations of filtering and up/down sampling together. Given an FIR filter $\mathcal{H}$, consider the operators

$$\downarrow 2 \circ \mathcal{H}^* : x \longrightarrow \Big\{\sum_\ell x_\ell\,\overline{h_{\ell-2n}}\Big\}_n, \qquad \mathcal{H} \circ \uparrow 2 : x \longrightarrow \Big\{\sum_\ell x_\ell\,h_{n-2\ell}\Big\}_n.$$

What's probably far from clear is why we use $\downarrow 2 \circ \mathcal{H}^*$ instead of $\downarrow 2 \circ \mathcal{H}$. That won't really emerge until lecture 5, but be content with noting that $\mathcal{H} \circ \uparrow 2$ is the adjoint of $\downarrow 2 \circ \mathcal{H}^*$ since

$$\begin{aligned}((\downarrow 2 \circ \mathcal{H}^*)x,\, y) &= \sum_n \Big\{\sum_m \overline{h_{\ell-2n}}\,x_\ell\Big\}\overline{y_n} \\ &= \sum_\ell x_\ell\,\overline{\Big\{\sum_n h_{\ell-2n}\,y_n\Big\}} = (x,\, (\mathcal{H} \circ \uparrow 2)y\,),\end{aligned}$$

showing indeed that

$$\mathcal{H} \circ \uparrow 2 \;=\; (\downarrow 2 \circ \mathcal{H}^*)^*.$$

(That helps!!) To illustrate the ideas, let's apply these operators in the case of the Haar filters

$$h_n \;=\; \begin{cases} \frac{1}{\sqrt{2}}\,, & n \,=\, 0, \\ \frac{1}{\sqrt{2}}\,, & n \,=\, 1, \\ 0, & n \,\neq\, 0,\, 1, \end{cases} \qquad \widetilde{h}_n \;=\; \begin{cases} \frac{1}{\sqrt{2}}\,, & n \,=\, 0, \\ -\frac{1}{\sqrt{2}}\,, & n \,=\, 1, \\ 0, & n \,\neq\, 0,\, 1. \end{cases}$$

So fix sequences $a \,=\, \{a_n\}_n,\ c \,=\, \{c_n\}_n$ and recall the signals $\{\varphi^{(n)}\}_n,\ \{\widetilde{\varphi}^{(n)}\}_n$,

$$\varphi^{(n)} \;=\; \frac{1}{\sqrt{2}}\big(\varepsilon^{(2n)} + \varepsilon^{(2n+1)}\big), \qquad \widetilde{\varphi}^{(n)} \;=\; \frac{1}{\sqrt{2}}\big(\varepsilon^{(2n)} - \varepsilon^{(2n+1)}\big),$$

defined in lecture 1. Then the filter coefficients are real, allowing us to omit complex conjugates and compute:

1. $\downarrow 2 \circ \mathcal{H}^*$: (averaging operator)

$$(\downarrow 2 \circ \mathcal{H}^*)a \;=\; \sum_n \frac{1}{\sqrt{2}}\big(a_{2n} + a_{2n+1}\big)\,\varepsilon^{(n)} \;=\; \big\{\,(a,\ \varphi^{(n)})\,\big\}_n;$$

2. $\downarrow 2 \circ \widetilde{\mathcal{H}}^*$: (difference operator)

$$(\downarrow 2 \circ \widetilde{H}^*)a \;=\; \sum_n \frac{1}{\sqrt{2}}\big(a_{2n} - a_{2n+1}\big)\,\varepsilon^{(n)} \;=\; \big\{\,(a,\ \widetilde{\varphi}^{(n)})\,\big\}_n;$$

3. $\mathcal{H} \circ \uparrow 2$: (spreading operator)

$$\sum_\ell c_\ell\, h_{n-2\ell} \;=\; \begin{cases} \frac{1}{\sqrt{2}}\, c_n, & n \,=\, 2m, \\ \frac{1}{\sqrt{2}}\, c_m, & n \,=\, 2m+1, \end{cases} \qquad (\mathcal{H} \circ \uparrow 2)c \;=\; \sum_n c_n\, \varphi^{(n)}\,;$$

4. $\widetilde{\mathcal{H}} \circ \uparrow 2$: (spread and oscillate)

$$\sum_\ell c_\ell\, \widetilde{h}_{n-2\ell} \;=\; \begin{cases} \frac{1}{\sqrt{2}}\, c_m, & n \,=\, 2m, \\ -\frac{1}{\sqrt{2}}\, c_m, & n \,=\, 2m+1, \end{cases} \qquad (\widetilde{\mathcal{H}} \circ \uparrow 2)c \;=\; \sum_n c_n\, \widetilde{\varphi}^{(n)}\,;$$

What's the point? Combining 1. and 3., we see that

$$(\mathcal{H} \circ \uparrow 2) \circ (\downarrow 2 \circ \mathcal{H}^*)a \;=\; \sum_n \,(a,\, \varphi^{(n)})\, \varphi^{(n)}$$

which is just the orthonormal series expansion of a with respect to the family $\{\varphi^{(n)}\}_n$, while combining 2. and 4. we see that

$$(\widetilde{\mathcal{H}} \circ \uparrow 2) \circ (\downarrow 2 \circ \widetilde{\mathcal{H}}^*)a \;=\; \sum_n \,(a,\, \varphi^{(n)})\, \varphi^{(n)}$$

which is the orthonormal series expansion with respect to the second family $\{\widetilde{\varphi}^{(n)}\}_n$. But together we know that the two families form a complete orthonormal family in ℓ^2. Hence we have obtained a *splitting*:

$$a \;=\; \sum_n \,(a,\, \varphi^{(n)})\, \varphi^{(n)} + \sum_n \,(a,\, \widetilde{\varphi}^{(n)})\, \widetilde{\varphi}^{(n)}$$

of a into what will turn out to be its 'coarse' and 'fine' details. This is exactly what sub-band coding is all about as we shall see in lectures 4 and 5. The individual properties of these four operators associated with the Haar filters will be explored further in the problems for this section.

Summary. So why all this emphasis on down-sampling and up-sampling? Well, we are going to use two filters, one low-pass to single out low frequency terms and the other high-pass to single out the high frequencies in a signal: the Haar $\mathcal{H}$ and $\widetilde{\mathcal{H}}$, for instance. But a convolution $h * x$ *contains just as many terms* as the original sequence x, so if we use two of them we will end up with 'twice' as many terms as we had before filtering started. We may have separated out the frequencies bands in a signal, but doubling the number of terms is hardly impressive compression! Thus the role of down-sampling is to ensure that we still have the *same number of terms after* filtering with both low-pass and high-pass filters. As down-sampling introduces aliasing, however, we've then got to up-sample and filter again to eliminate the aliasing and hence reconstruct the original signal. For digital signals all this is made precise in the notion of *filter bank* introduced in lecture 5, which then carries over to produce the decomposition of analogue signals into coarse and fine details. As always, the big problem is how to design the appropriate filters. That's the miracle achieved by the Daubechies FIR filters!

Problems.

1. Let $X = X(\xi)$ be the Frequency Response function of a signal $x = \{x_n\}_n$. Use the orthonormality relations in section (1.4) to determine each x_n knowing $X(\xi)$.

2. Let $x = \{x_n\}_n$ be the sequence whose Frequency Response function is the period 1 extension of the function

$$X(\xi) \;=\; |1 - 3|\xi||, \qquad 0 \le |\xi| \le \tfrac{1}{2}.$$

By using your result in the previous problem find the sequence $\{x_n\}_n$ for which $X(\xi)$ is its Frequency Response function. (Hint: get rid of the outer absolute value by using the fact that X is even, *i.e.*, $X(-\xi) = X(\xi)$; then get rid of the inner absolute value by splitting the new integral into two parts.)

3. The z transforms of down-sampled and up-sampled signals can be computed. Give a signal $x = \{x_n\}_n$, set

$$v \;=\; (\downarrow 2)x, \qquad u \;=\; (\uparrow 2)v \;=\; (\uparrow 2)(\downarrow 2)x.$$

Show that

$$V(z) \;=\; \tfrac{1}{2}\Big\{X(z^{1/2}) + X(-z^{1/2})\Big\}, \qquad U(z) \;=\; V(z^2),$$

and then deduce that

$$U(z) \;=\; \tfrac{1}{2}\Big\{X(z) + X(-z)\Big\}.$$

4. Given a signal $x = \{x_n\}_n$, compute $(\uparrow 2)(\downarrow 2)x$ as a sequence and then compute the Frequency Response function of $(\uparrow 2)(\downarrow 2)x$.

5. Show that $(\downarrow 2)(\uparrow 2)x = x$, *i.e.*, up-sampling followed by down-sampling always preserves a signal.

6. As we shall see later, one crucial aspect of the Daubechies construction is that a companion filter having coefficients

$$\tilde{h}_\ell \;=\; (-1)^\ell h_{1-\ell}$$

is always associated with a filter $\{h_\ell\}_\ell$. Show that Frequency Response function $\widetilde{H}(\xi)$ of this associated filter is given by

$$\widetilde{H}(\xi) \;=\; -e^{-2\pi i\xi}\,\overline{H(\xi + \tfrac{1}{2})}.$$

7. The companion filter in the $db2$-case, for instance, is

$$\tilde{h}_{-2} \;=\; \frac{1-\sqrt{3}}{4\sqrt{2}}, \quad \tilde{h}_{-1} \;=\; -\frac{3-\sqrt{3}}{4\sqrt{2}}, \quad \tilde{h}_0 \;=\; \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad \tilde{h}_1 \;=\; -\frac{1+\sqrt{3}}{4\sqrt{2}}$$

(it's not causal I know, but that's not important!). Use problem 6 and the known expression for $H(\xi)$ to show that

$$\widetilde{H}(\xi) \;=\; \sqrt{2}\, e^{\pi i \xi}\, (\sin \pi\xi)^2 (\sqrt{3} \cos \pi\xi - i \sin \pi\xi)$$

(the $db2$-filter coefficients are *real* remember!). Deduce that $\{\, \tilde{h}_\ell \,\}_\ell$ is trying very hard to be a high pass filter.

8. Use some graphing facility to draw the graph of $|H(\xi)|$ and $|\widetilde{H}(\xi)|$ for the Daubechies $db2$-filter.

9. What condition on $H(\xi)$ ensures that the mapping $\mathcal{H} \circ \uparrow 2$ is energy-preserving on ℓ^2 for a given FIR filter $\mathcal{H}$? Deduce that both the Haar and Daubechies filters satisfy this condition.