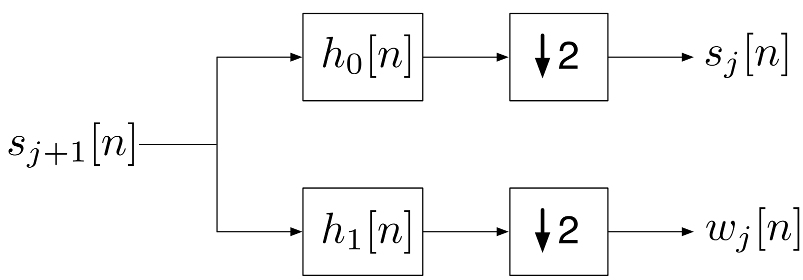
Haar wavelet filterbanks

Recall that we can compute the scaling coefficients $s_{j,n}$ and wavelet coefficients $w_{j,n}$ at scale j from the scaling coefficients $s_{j+1,n}$ at scale $j+1$:

$$\begin{aligned} s_{j,n} &= \frac{1}{\sqrt{2}}\, s_{j+1,2n} + \frac{1}{\sqrt{2}}\, s_{j+1,2n+1} \\ w_{j,n} &= \frac{1}{\sqrt{2}}\, s_{j+1,2n} - \frac{1}{\sqrt{2}}\, s_{j+1,2n+1}. \end{aligned}$$

If we think of the scaling/wavelet coefficients at scale j as a discrete time sequence, so $s_j[n] := s_{j,n}$ and $w_j[n] := w_{j,n}$, then the expressions above suggest that the scaling coefficients at scale j can be broken down scaling and wavelet coefficients at scale $j-1$ using **filters** and **downsampling** arranged in the following architecture:

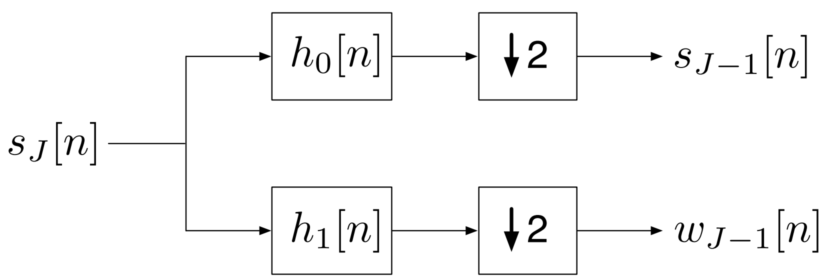


where the $\downarrow 2$ block means "downsample by 2" and the impulse responses for the filters are:

$$h_0[n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 0, -1 \\ 0 & \text{otherwise} \end{cases} \qquad h_1[n] = \begin{cases} -\frac{1}{\sqrt{2}} & n = -1 \\ \frac{1}{\sqrt{2}} & n = 0 \\ 0 & \text{otherwise.} \end{cases}$$

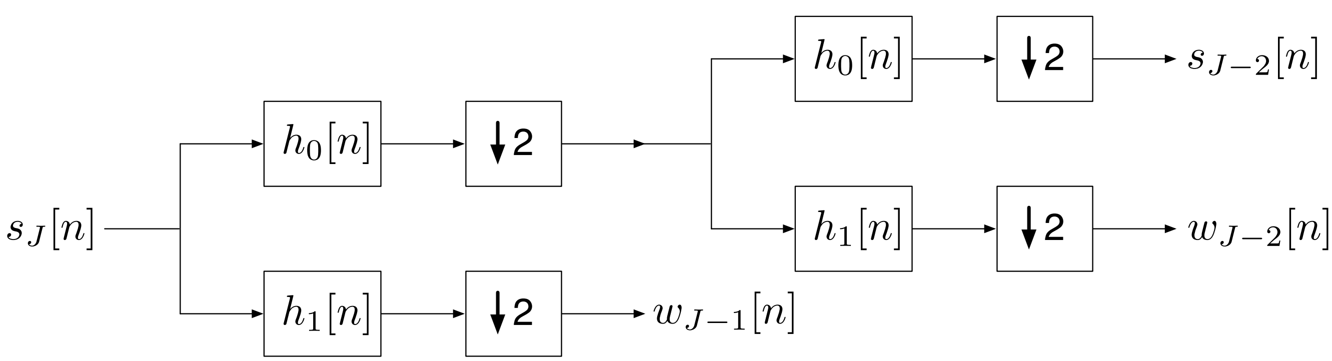
Of course, we can continue on and break up the $\{s_{j,n}\}$ into scaling and wavelet coefficients at the next coarsest scale. This gives rise to a **filter bank** structure that we can associate with each of the ways we can write the approximation at scale J, $\hat{x}_J(t) = \boldsymbol{P}_{\mathcal{V}_J}[x(t)]$.

$$\begin{aligned}\hat{x}_J(t) &= \boldsymbol{P}_{\mathcal{V}_{J-1}}[x(t)] + \boldsymbol{P}_{\mathcal{W}_{J-1}}[x(t)] \\ &= \sum_{n=-\infty}^{\infty} s_{J-1,n}\phi_{J-1,n}(t) + \sum_{n=-\infty}^{\infty} w_{J-1,n}\psi_{J-1,n}(t)\end{aligned}$$



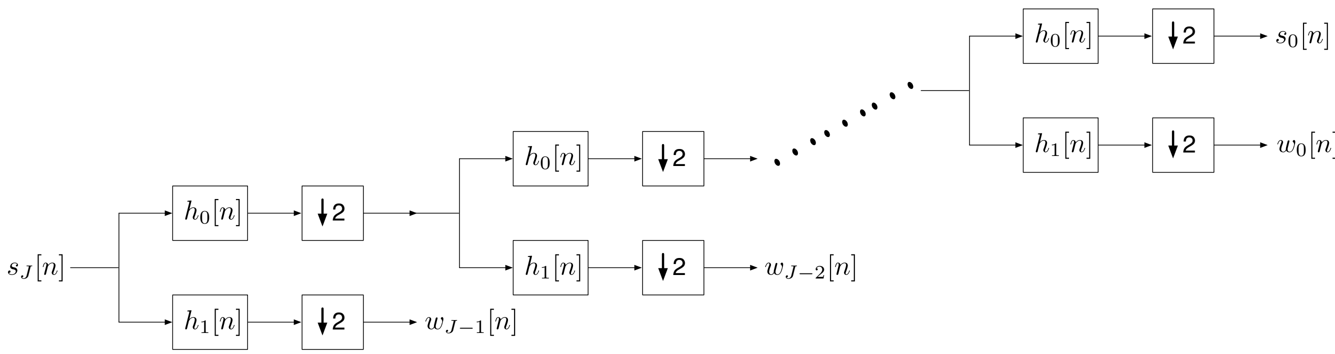
Iterating this process on $s_{J-1}[n]$ we obtain

$$\begin{aligned}\hat{x}_J(t) &= \boldsymbol{P}_{\mathcal{V}_{J-2}}[x(t)] + \boldsymbol{P}_{\mathcal{W}_{J-2}}[x(t)] + \boldsymbol{P}_{\mathcal{W}_{J-1}}[x(t)] \\ &= \sum_{n=-\infty}^{\infty} s_{J-2,n}\phi_{J-2,n}(t) + \sum_{n=-\infty}^{\infty} w_{J-2,n}\psi_{J-2,n}(t) + \sum_{n=-\infty}^{\infty} w_{J-1,n}\psi_{J-1,n}(t)\end{aligned}$$



We can continue this process to obtain

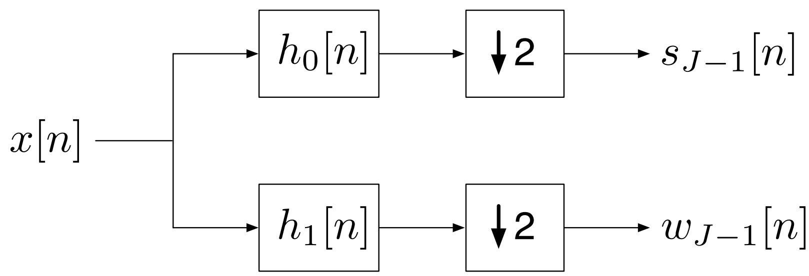
$$\begin{aligned}\hat{x}_J(t) &= \boldsymbol{P}_{\mathcal{V}_0}[x(t)] + \boldsymbol{P}_{\mathcal{W}_0}[x(t)] + \cdots + \boldsymbol{P}_{\mathcal{W}_{J-1}}[x(t)] \\ &= \sum_{n=-\infty}^{\infty} s_{0,n}\phi_{0,n}(t) + \sum_{j=0}^{J-1}\sum_{n=-\infty}^{\infty} w_{j,n}\psi_{j,n}(t)\end{aligned}$$



This provides an extraordinarily efficient way to compute the full set of scaling and wavelet coefficients given an initial approximation at scale J.

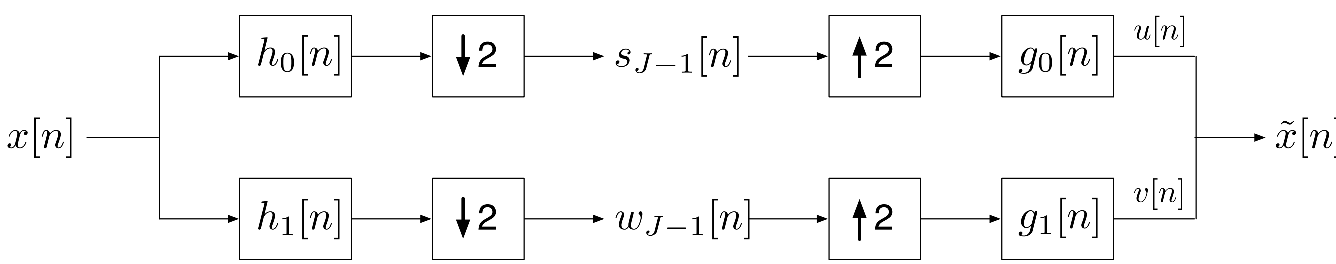
The discrete Haar transform

The connection to filter banks above gives us a natural way to define a wavelet transform for discrete-time signals. Basically, we just treat $x[n]$ like it was a sequence of scaling coefficients at fine scale, then apply as many levels of the filter bank as we like. So the following structure:



takes $x[n]$ and transforms it into two sequences, $s_{J-1}[n]$ and $w_{J-1}[n]$, each of which have **half the rate** of the input.

How do we **invert** this particular transform? With another filter filter bank. Consider the following structure:



If we take

$$g_0[n] = h_0[-n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 0, 1 \\ 0 & \text{otherwise} \end{cases}$$

$$g_1[n] = h_1[-n] = \begin{cases} \frac{1}{\sqrt{2}} & n = 0 \\ -\frac{1}{\sqrt{2}} & n = 1 \\ 0 & \text{otherwise} \end{cases},$$

then we will have $\tilde{x}[n] = x[n]$. To see this, recall that

$$s_{J-1}[n] = \frac{1}{\sqrt{2}}(x[2n] + x[2n+1]),$$

and so

$$u[n] = \begin{cases} \frac{1}{2}\left(x[n] + x[n+1]\right) & n \text{ even} \\ \frac{1}{2}\left(x[n-1] + x[n]\right) & n \text{ odd} \end{cases},$$

that is, the values in $u[n]$ appear in pairs,

$$u[0] = u[1] = \frac{1}{2}(x[0] + x[1]), \quad u[2] = u[3] = \frac{1}{2}(x[2] + x[3]), \text{ etc.}$$

Similarly, since

$$w_{J-1}[n] = \frac{1}{\sqrt{2}}\left(x[2n] - x[2n+1]\right),$$

we have

$$v[n] = \begin{cases} \frac{1}{2}\left(x[n] - x[n+1]\right) & n \text{ even} \\ \frac{1}{2}\left(-x[n-1] + x[n]\right) & n \text{ odd} \end{cases},$$

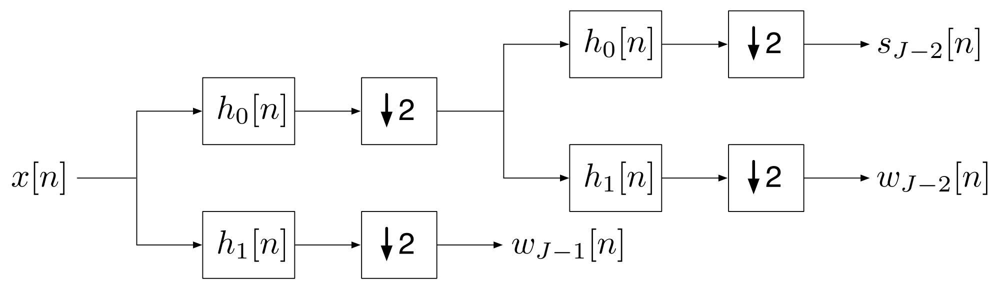
that is, the values in $v[n]$ appear in pairs of $\pm$ terms,

$$v[0] = -v[1] = \frac{1}{2}(x[0] - x[1]), \quad v[2] = -v[3] = \frac{1}{2}(x[2] - x[3]), \text{ etc.}$$

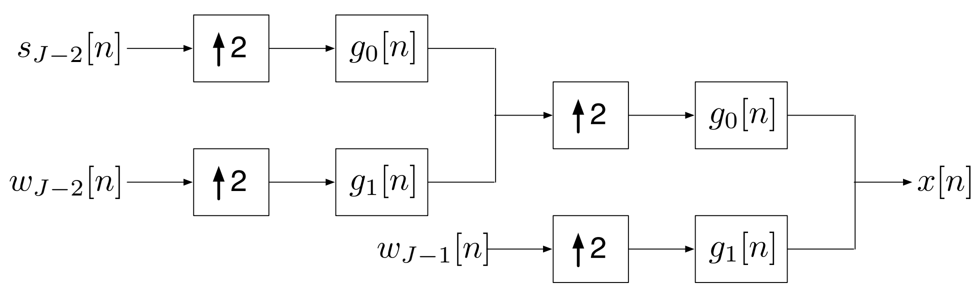
Now it is easy to see that

$$\tilde{x}[n] = u[n] + v[n] = x[n] \quad \text{for all } n \in \mathbb{Z}.$$

We can repeat this to as many levels as we desire. For example, the following structure



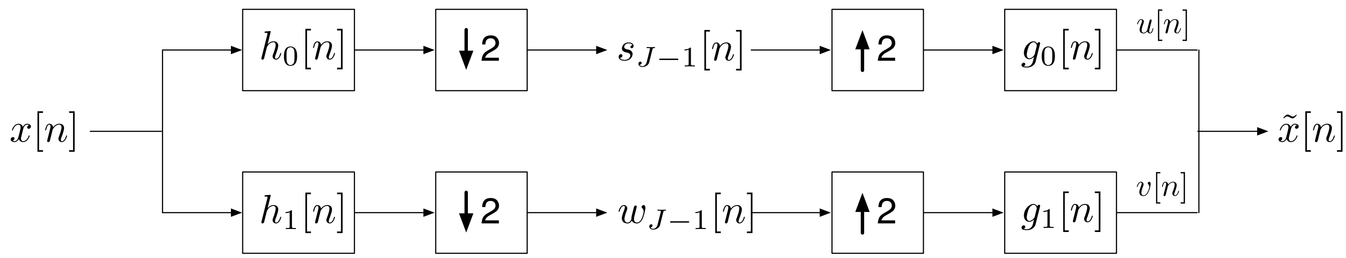
takes $x[n]$ and transforms it into three sequences, one of which is at half the rate of $x[n]$, and the other two are at a quarter of the rate. To invert it, we simply apply the inverse filter bank twice:



It should be clear how to extend this to an arbitrary number of levels.

Filterbanks: Beyond Haar

The filterbank architecture that defines the discrete Haar transform suggests a natural way in which to generalize our discussion of Haar wavelets. Specifically, recall the architecture:



You might wonder if there are other choices for the filters $h_0[n]$, $h_1[n]$, $g_0[n]$, and $g_1[n]$ that would correspond to alternative wavelet decompositions. This is indeed the case, but of course this will only be true when the filters satisfy certain restrictions.

To get some intuition for this, first we will consider a simpler question: when can we show that the above architecture gives rise to a **perfect reconstruction filterbank**, by which we simply mean a filterbank satisfying $\tilde{x}[n] = x[n]$?

An answer to this question requires careful thought about what happens when we **downsample** and **upsample** a discrete-time sequence. You can read the technical details provided later in these notes to see all the nitty-gritty details. It's not to hard if you remember the ***z*-transform**, but is mostly just a lot of algebra that doesn't tend to give you a whole lot of insight, so we can just jump to the first main conclusion.

In order to obtain a filterbank that produces a (possibly delayed) perfect reconstruction of the input $x[n]$, the filters used must satisfy two main properties, known as the **perfect reconstruction conditions**:[1]

$$G_0(e^{j\omega})H_0(e^{-j\omega}) + G_1(e^{j\omega})H_1(e^{-j\omega}) = 0 \qquad \text{(Alias cancellation)}$$

and

$$G_0(e^{j\omega})H_0(e^{j\omega}) + G_1(e^{j\omega})H_1(e^{j\omega}) = 2e^{-j\omega m}. \qquad \text{(No distortion)}$$

The "no distortion" condition is perhaps the more intuitive of the two. Imagine that h_0 was a perfect low-pass filter and h_1 a perfect high-pass filter. In this case, the downsampling step induces no aliasing. The no distortion condition simply implies that g_0 and g_1 have to undo the effects of h_0 and h_1 so that we can get back our original signal.

The "alias cancellation" condition simply accounts for the fact that in real-world settings (e.g., finite-length filters), the downsampling step may induce aliasing, and thus the filters must be well designed to perfectly compensate for this effect (g_0 needs to somehow filter out the aliasing that results from the output of h_0 not being perfectly bandlimited, and similarly for g_1).

Of course, the next natural question is: how can we construct filters that satisfy this property? There is, of course, a huge variety of choices, but it turns out that the conditions above impose a lot of structure. For example, if we fix g_0 and g_1, then there are natural choices for how to design h_0 and h_1 to ensure that we satisfy the conditions above (again, see the technical details if you are curious).

[1] In the technical details these are stated in terms of the z-transform, but here I will just use the DTFT to keep things simple.

All of the discussion above tells us only what restrictions are imposed by the desire to obtain a perfect reconstruction. Nothing we have said so far tells us anything about whether the representation we are computing corresponds to an *orthonormal* basis. If we additionally wish to ask that the filterbank correspond to computing the coefficients for some kind of orthonormal basis, this imposes even more structure. As shown in the technical details, if we want orthogonality, this constrains our design even more. In fact, if we want an *orthogonal filterbank*, then once we fix a single one of the filters (say, g_0), the rest are determined.

Moreover, not any filter g_0 will do. As shown in the technical details, g_0 must satisfy a particular condition for everything to still work out. Specifically, we need:

$$P(e^{j\omega}) = G_0(e^{j\omega})G_0(e^{-j\omega}) = |G_0(e^{j\omega})|^2$$

to satisfy

$$P(e^{j\omega}) + P(-e^{j\omega}) = 2. \qquad \text{(No distortion (v2))}$$

Thus, constructing an orthonormal filterbank boils down to designing a filter g_0 satisfying this condition. There is still a bit of freedom in how to best do this, and we will have more to say on this later, but before we do so, we will take a step back to relate all of this discussion back to the concepts we first introduced when discussing wavelets. We will see that (ignoring some additional technical constraints on g_0) any orthogonal filterbank has a natural correspondence to a continuous-time orthonormal wavelet basis.

Continuous-time orthonormal wavelet bases

We began our discussion of wavelets by considering the Haar wavelet basis for decomposing continuous-time signals $x(t) \in L_2(\mathbb{R})$, giving us a decomposition of the form

$$x(t) = \sum_{n=-\infty}^{\infty} s_{0,n}\phi_{0,n}(t) + \sum_{j=0}^{\infty}\sum_{n=-\infty}^{\infty} w_{j,n}\psi_{j,n}(t).$$

Recall that the (orthonormal) basis functions are **scaled and shifted** versions of the two template functions $\phi_0(t)$ and $\psi_0(t)$. Moreover, these two functions were linear combinations of shifts of a *contracted* version of $\phi_0(t)$:

$$\phi_0(t) = \phi_0(2t) + \phi_0(2t-1), \qquad \psi_0(t) = \phi_0(2t) - \phi_0(2t-1).$$

This gave us the very nice interpretation of the wavelet coefficients $w_{j,n}$ capturing the **differences** between piecewise-constant approximations of $x(t)$ at different dyadic scales:

$$x(t) = \underbrace{\underbrace{\underbrace{\boldsymbol{P}_{\mathcal{V}_0}[x(t)] + \boldsymbol{P}_{\mathcal{W}_0}[x(t)]}_{=\boldsymbol{P}_{\mathcal{V}_1}[x(t)]} + \boldsymbol{P}_{\mathcal{W}_1}[x(t)]}_{=\boldsymbol{P}_{\mathcal{V}_2}[x(t)]} + \boldsymbol{P}_{\mathcal{W}_2}[x(t)]}_{=\boldsymbol{P}_{\mathcal{V}_3}[x(t)]} + \cdots .$$

Along with this interpretation, we also developed an efficient filter-bank implementation for computing this decomposition from some initial approximation $\hat{x}_J(t) = \boldsymbol{P}_{\mathcal{V}_J}[x(t)]$.

Now that we have seen how to generalize this filterbank structure, it is natural to ask whether these new filterbanks have a similar correspondence to other types of approximation spaces $\mathcal{V}_j$ built using scaling functions $\phi_0(t)$ other than just piecewise-constant functions. Indeed we can, and it leads to a very rich family of **orthonormal wavelet bases**.

As in the Haar case, we will see that essentially all of the properties of any orthonormal wavelet basis will follow from properties of the scaling function $\phi_0(t)$. Before discussing these more general wavelet bases, we will first review some of the key properties of $\phi_0(t)$ that allowed us to interpret the Haar wavelet transform as providing a multiscale approximation.

Multiscale approximation: Scaling spaces

For a given $\phi_0(t)$, the first approximation space $\mathcal{V}_0$ is set of signals we can build up from different linear combinations[2] of the integer shifts of $\phi_0(t)$:

$$\mathcal{V}_0 = \overline{\text{Span}}(\{\phi_0(t-n)\}_{n\in\mathbb{Z}}).$$

The first thing we want is for $\{\phi_0(t-n)\}_{n\in\mathbb{Z}}$ to be an orthobasis, so we ask that

(P1)
$$\langle \phi_0(t-k), \phi_0(t-n)\rangle = \begin{cases} 1, & k = n, \\ 0, & k \neq n. \end{cases}$$

[2]Technically, this is the set of signals we can approximate arbitrarily well from different linear combinations — this is the closure of the span, which we will denote by $\overline{\text{Span}}$.

Now set

$$\phi_{j,n}(t) = 2^{j/2}\phi_0(2^j t - n),$$

so the function $\phi_0(2^j t - n)$ is formed by **contracting** $\phi_0(t)$ by a factor of 2^j, then shifting the result on a grid with spacing 2^{-j}. For a fixed scale j, define

$$\mathcal{V}_j = \overline{\mathrm{Span}}(\{\phi_{j,n}(t)\}_{n\in\mathbb{Z}}).$$

Following the Haar case, there are two more key properties we ask of this sequence of approximation spaces; we would like these spaces to be nested,

$$\textbf{(P2)} \qquad \mathcal{V}_j \subset \mathcal{V}_{j+1}, \quad \text{so } x(t) \in \mathcal{V}_j \Rightarrow x(t) \in \mathcal{V}_{j+1},$$

and we also want these approximation spaces to cover all of $L_2(\mathbb{R})$ in their limit:

$$\textbf{(P3)} \qquad \lim_{j\to\infty} \mathcal{V}_j = L_2(\mathbb{R}), \text{ so } \lim_{j\to\infty} \boldsymbol{P}_{\mathcal{V}_j}[x(t)] = x(t) \text{ for all } x(t) \in L_2(\mathbb{R}).$$

Now the question is: What properties does $\phi_0(t)$ have to have to ensure **(P1)**–**(P3)** hold? While the answer is not straightforward, this question was answered completely in the late 1980s/early 1990s. The conditions on $\phi_0(t)$ are actually most easily expressed in terms of the inter-scale relationships between the $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ and $\{\phi_{j+1,n}\}_{n\in\mathbb{Z}}$, which you may recall is exactly what gave rise to the filterbank structure for computing the Haar wavelet transform.

Specifically, given a $\phi_0(t)$, define the sequence of numbers $g_0[n]$

$$g_0[n] = \langle \phi_0(t), \sqrt{2}\phi_0(2t - n)\rangle. \tag{1}$$

It turns out that whether properties **(P1)**–**(P3)** hold depends entirely on properties of this sequence of numbers. Let $G_0(e^{j\omega})$ be the discrete-time Fourier transform of $g_0[n]$. Then we have following major result:

> If $g_0[n]$ obeys the following three properties, then the approximation spaces $\{\mathcal{V}_j\}_{j\geq 0}$ obey properties **(P1)**–**(P3)**:
>
> **(G1)** $$|G_0(e^{j\omega})|^2 + |G_0(e^{j(\omega+\pi)})|^2 = 2, \quad \text{for all} \quad -\pi \leq \omega \leq \pi$$
>
> **(G2)** $$G_0(e^{j0}) = \sum_n g_0[n] = \sqrt{2},$$
>
> **(G3)** $$|G_0(e^{j\omega})| > 0 \quad \text{for all} \quad |\omega| \leq \frac{\pi}{2}.$$

The proof of this result is long and complicated.[3] Note, however, that Condition **(G1)** is somewhat familiar. Specifically, **(G1)** is simply another way of writing v2 of the "No distortion" filterbank condition. The remaining conditions are more technical requirements that allow us to construct $\phi_0(t)$ from knowledge of $g_0[n]$ (see the additional technical details at end of notes.)

[3]There are a few good references here. I will recommend Chapter 7 of A Wavelet Tour of Signal Processing, by S. Mallat, and Daubechies' book *Ten Lectures on Wavelets*.

Multiscale approximation: Wavelet spaces

The complementary wavelet spaces and wavelet basis functions can also be generated from the coefficient sequence $g_0[n]$. This is detailed as our second major result:

Suppose $\phi_0(t)$ with corresponding $g_0[n]$ obeys **(G1)**–**(G3)**. Set[a]

$$g_1[n] = (-1)^{1-n} g_0[1-n],$$

and

$$\psi_0(t) = \sum_{n=-\infty}^{\infty} g_1[n]\sqrt{2}\,\phi_0(2t-n).$$

Then, along with integer shifts of the scaling function $\phi_{0,n}(t) = \phi_0(t-n)$, the set of all dyadic shifts and contractions of $\psi_0(t)$,

$$\psi_{j,n}(t) = 2^{j/2}\psi_0(2^j t - n), \quad n \in \mathbb{Z}, \quad j = 0, 1, 2, \ldots,$$

form an orthobasis for $L_2(\mathbb{R})$. That is,

$$x(t) = \sum_{n=-\infty}^{\infty} \langle \boldsymbol{x}, \boldsymbol{\phi}_{0,n}\rangle \phi_{0,n}(t) + \sum_{j=0}^{\infty}\sum_{n=-\infty}^{\infty} \langle \boldsymbol{x}, \boldsymbol{\psi}_{j,n}\rangle \psi_{j,n}(t)$$

for all $x(t) \in L_2(\mathbb{R})$.

[a] Note that the choice of $g_1[n]$ here is precisely the "alternating flip" construction we described in the context of filterbanks.

As with the Haar case, the wavelet coefficients at scale j represent the difference between the approximation of a signal in $\mathcal{V}_j$ and the approximation in $\mathcal{V}_{j+1}$. That is, if we set

$$\mathcal{W}_j = \overline{\text{Span}}\left(\{\psi_{j,n}(t)\}_{n\in\mathbb{Z}}\right)$$

then

1. For fixed j, $\langle \psi_{j,n}, \psi_{j,\ell}\rangle = 0$ for $n \neq \ell$. That is, the $\{\psi_{j,n}(t)\}_{n\in\mathbb{Z}}$ are orthobasis for $\mathcal{W}_j$.

2. $\mathcal{W}_j \perp \mathcal{V}_{j'}$ for all $j' \leq j$. Notice that since $\mathcal{W}_j \subset \mathcal{V}_{j+1}$, it follows that the sequence of spaces $\mathcal{V}_0, \mathcal{W}_0, \mathcal{W}_1, \ldots$ are all mutually orthogonal.

3. $\mathcal{V}_{j+1} = \mathcal{V}_j \oplus \mathcal{W}_j$. That is, every $v(t) \in \mathcal{V}_{j+1}$ can be written as

$$v(t) = \boldsymbol{P}_{\mathcal{V}_j}[v(t)] + \boldsymbol{P}_{\mathcal{W}_j}[v(t)].$$

   As the previous property states, these two components are orthogonal to one another.

In summary, this means we can break $L_2(\mathbb{R})$ into orthogonal parts,

$$L_2(\mathbb{R}) = \mathcal{V}_0 \oplus \mathcal{W}_0 \oplus \mathcal{W}_1 \oplus \mathcal{W}_2 \oplus \cdots$$

and we have an orthobases for each of these.

Vanishing moments and support size

In addition to forming an orthobasis with a certain multiscale form, there are other desirable properties that wavelet systems often have.

Vanishing moments. We say that $\psi_0(t)$ has p vanishing moments if

$$\int_{-\infty}^{\infty} t^q \psi_0(t)\,\mathrm{d}t = 0, \quad \text{for } q = 0, 1, \ldots, p-1.$$

This means that $\psi_0(t)$ is **orthogonal** to all **polynomials** of degree $p-1$ or smaller. Since shifting a polynomial just gives you another polynomial of the same order, $\psi_0(t-n)$ is also orthogonal to these polynomials. This means that polynomials that have degree at most $p-1$ are completely contained in the scaling space $\mathcal{V}_0$ — all of the wavelet coefficients of a polynomial are zero.

Compact support. The support of $\psi_0(t)$ is the size of the interval on which it is non-zero. If $\psi_0(t)$ is supported on $[0, L]$, then $\psi_{0,n}(t) = \psi_0(t-n)$ is supported on $[n, n+L]$, and

$$w_{0,n} = \langle \boldsymbol{x}, \boldsymbol{\psi}_{0,n} \rangle = \int_n^{n+L} x(t)\psi_{0,n}(t)\,\mathrm{d}t.$$

This means that $w_{0,n}$ only depends on what $x(t)$ is doing on $[n, n+L]$ — the wavelet coefficients are recording **local** information about the behavior of $x(t)$.

These two properties make wavelets very good for representing signals which are smooth except at a few singularities.

Daubechies Wavelets

In the late 1980s, Ingrid Daubechies presented a systematic framework for designing wavelets with vanishing moments and compact support. For any integer p, there is a method for solving for the $g_0[n]$ that corresponds to a wavelet with p vanishing moments and has support size $2p-1$.

Here are the filter coefficients for $p = 2, \ldots, 10$. ($p = 1$ gives you Haar wavelets.):

	n	$h_p[n]$
$p=2$	0	0.482962913145
	1	0.836516303738
	2	0.224143868042
	3	−0.129409522551
$p=3$	0	0.332670552950
	1	0.806891509311
	2	0.459877502118
	3	−0.135011020010
	4	−0.085441273882
	5	0.035226291882
$p=4$	0	0.230377813309
	1	0.714846570553
	2	0.630880767930
	3	−0.027983769417
	4	−0.187034811719
	5	0.030841381836
	6	0.032883011667
	7	−0.010597401785
$p=5$	0	0.160102397974
	1	0.603829269797
	2	0.724308528438
	3	0.138428145901
	4	−0.242294887066
	5	−0.032244869585
	6	0.077571493840
	7	−0.006241490213
	8	−0.012580751999
	9	0.003335725285
$p=6$	0	0.111540743350
	1	0.494623890398
	2	0.751133908021
	3	0.315250351709
	4	−0.226264693965
	5	−0.129766867567
	6	0.097501605587
	7	0.027522865530
	8	−0.031582039317
	9	0.000553842201
	10	0.004777257511
	11	−0.001077301085
$p=7$	0	0.077852054085
	1	0.396539319482
	2	0.729132090846
	3	0.469782287405
	4	−0.143906003929
	5	−0.224036184994
	6	0.071309219267
	7	0.080612609151
	8	−0.038029936935
	9	−0.016574541631
	10	0.012550998556
	11	0.000429577973
	12	−0.001801640704
	13	0.000353713800
$p=8$	0	0.054415842243
	1	0.312871590914
	2	0.675630736297
	3	0.585354683654
	4	−0.015829105256
	5	−0.284015542962
	6	0.000472484574
	7	0.128747426620
	8	−0.017369301002
	9	−0.04408825393
	10	0.013981027917
	11	0.008746094047
	12	−0.004870352993
	13	−0.000391740373
	14	0.000675449406
	15	−0.000117476784
$p=9$	0	0.038077947364
	1	0.243834674613
	2	0.604823123690
	3	0.657288078051
	4	0.133197385825
	5	−0.293273783279
	6	−0.096840783223
	7	0.148540749338
	8	0.030725681479
	9	−0.067632829061
	10	0.000250947115
	11	0.022361662124
	12	−0.004723204758
	13	−0.004281503682
	14	0.001847646883
	15	0.000230385764
	16	−0.000251963189
	17	0.000039347320
$p=10$	0	0.026670057901
	1	0.188176800078
	2	0.527201188932
	3	0.688459039454
	4	0.281172343661
	5	−0.249846424327
	6	−0.195946274377
	7	0.127369340336
	8	0.093057364604
	9	−0.071394147166
	10	−0.029457536822
	11	0.033212674059
	12	0.003606553567
	13	−0.010733175483
	14	0.001395351747
	15	0.001992405295
	16	−0.000685856695
	17	−0.000116466855
	18	0.000093588670
	19	−0.000013264203

From Mallat, *A Wavelet Tour of Signal Processing*

Here are pictures of some of the scaling functions ($N = 2p$ in the captions below):

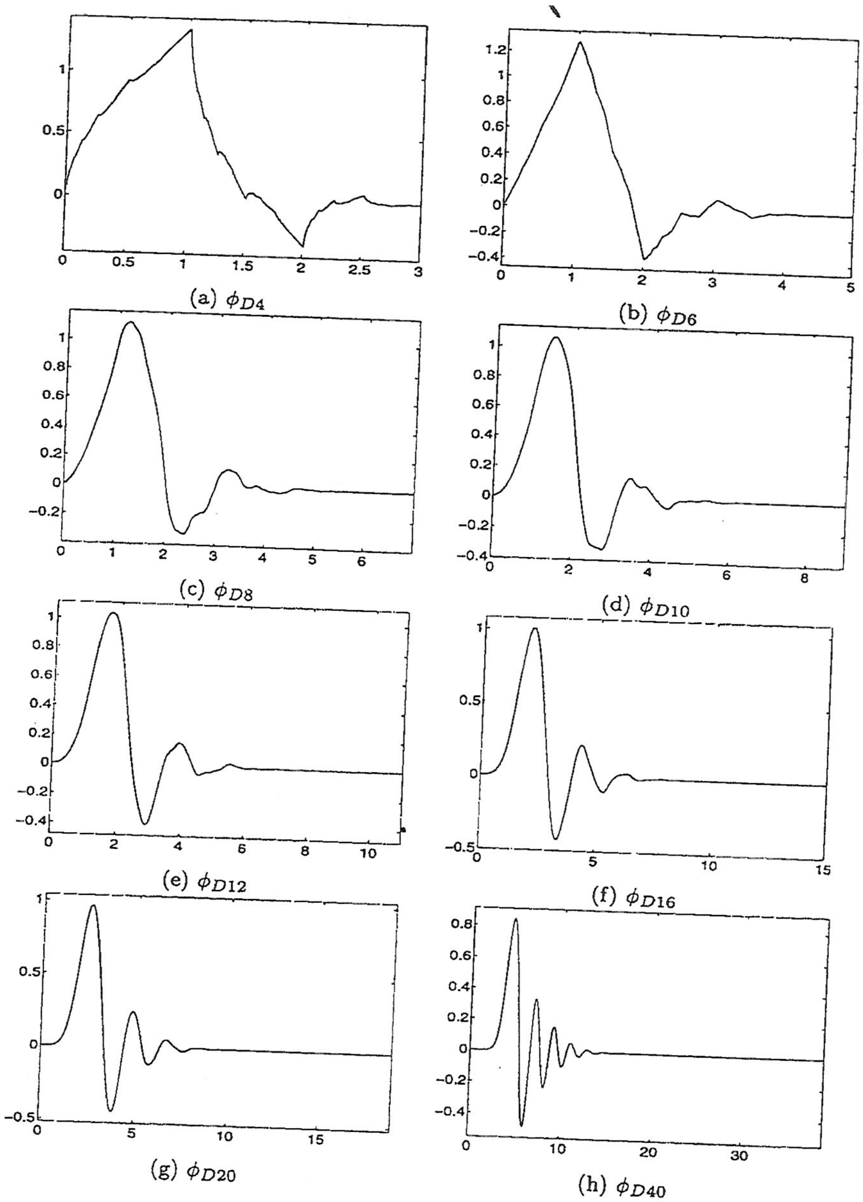


Figure 6.1. Daubechies Scaling Functions, $N = 4, 6, 8, \ldots, 40$

From Burrus et al, *Introduction to Wavelets ...*

Here are pictures of some of the wavelet functions ($N = 2p$ in the captions below):

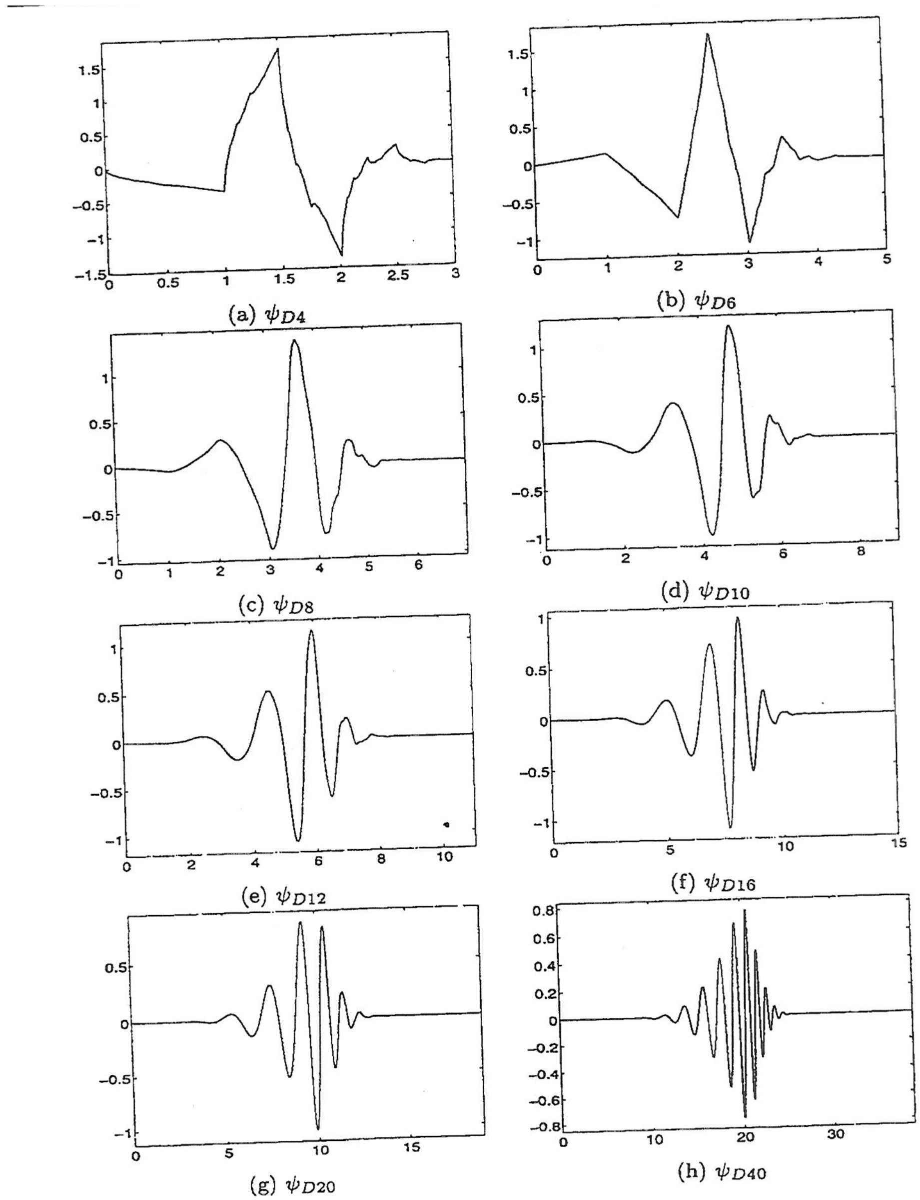


Figure 6.2. Daubechies Wavelets, $N = 4, 6, 8, \ldots, 40$

From Burrus et al, *Introduction to Wavelets ...*

Technical Details: Perfect reconstruction and orthogonal filterbanks

The z-transform

Recall that for a discrete-time signal $x[n]$, the z-transform is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n},$$

where z is a complex number.[4]

We can think of the z-transform as a generalization of the DTFT. Specifically, by evaluating the z-transform $X(z)$ at $z = e^{j\omega}$ we obtain the DTFT of $x[n]$, i.e. $X(e^{j\omega})$.

The z-transform also generalizes the familiar property of the DTFT that convolution in time is equivalent to multiplication in frequency. Specifically, if $y[n]$ denotes the convolution of $x[n]$ with $h[n]$, then we have

$$Y(z) = X(z)H(z).$$

Downsampling

Now consider the process of taking a signal $x[n]$ and **downsampling** it by a factor of 2. Specifically, let

$$y[n] = x[2n].$$

[4]For any given z, this sum may or may not converge, and so we also associate with $X(z)$ a *region of convergence* (which will depend on $x[n]$) that tells us the set of possible z for which the sum converges. Fortunately, we will not need to worry much about this.

What is the relationship between $X(z)$ and $Y(z)$?

Observe that

$$\begin{aligned}
Y(z) &= \sum_{n=-\infty}^{\infty} x[2n]z^{-n} \\
&= \sum_{\ell \text{ even}} x[\ell]z^{-\ell/2} \\
&= \frac{1}{2}\sum_{\ell=-\infty}^{\infty} x[\ell]z^{-\ell/2} + \frac{1}{2}\sum_{\ell=-\infty}^{\infty} x[\ell](-1)^{-\ell}z^{-\ell/2} \\
&= \frac{1}{2}\sum_{\ell=-\infty}^{\infty} x[\ell](z^{1/2})^{-\ell} + \frac{1}{2}\sum_{\ell=-\infty}^{\infty} x[\ell](-z^{1/2})^{-\ell} \\
&= \frac{1}{2}\left[X(z^{1/2}) + X(-z^{1/2})\right].
\end{aligned}$$

This may seem a bit difficult to interpret, but things are a bit clearer when we look at the DTFT. Specifically, if we let $z = e^{j\omega}$, then

$$X(z^{1/2}) = X(e^{j\omega/2}) \qquad\qquad X(-z^{1/2}) = X(-e^{j\omega/2}) = X(e^{j\omega/2+\pi}).$$

Note that $X(e^{j\omega/2})$ is simply a dilated version of $X(e^{j\omega})$. The $X(e^{j\omega/2+\pi})$ term corresponds to a dilation of $X(e^{j\omega})$ shifted by π. This corresponds to exactly what one would have obtained if $x[n]$ corresponded to samples of a continuous-time signal which we then sampled at half of the original sampling rate – the spectrum is dilated (because of the lower sampling rate) but there is also potential *aliasing*, which is accounted for by the $X(e^{j\omega/2+\pi})$ term.

Upsampling

Now we turn to the problem of taking a signal $x[n]$ and **upsampling** it by a factor of 2. By this we mean generating a signal

$$y[n] = \begin{cases} x[n/2] & n \text{ even} \\ 0 & n \text{ odd} \end{cases}.$$

We again ask the question: what is the relationship between $X(z)$ and $Y(z)$?

The answer is straightforward:

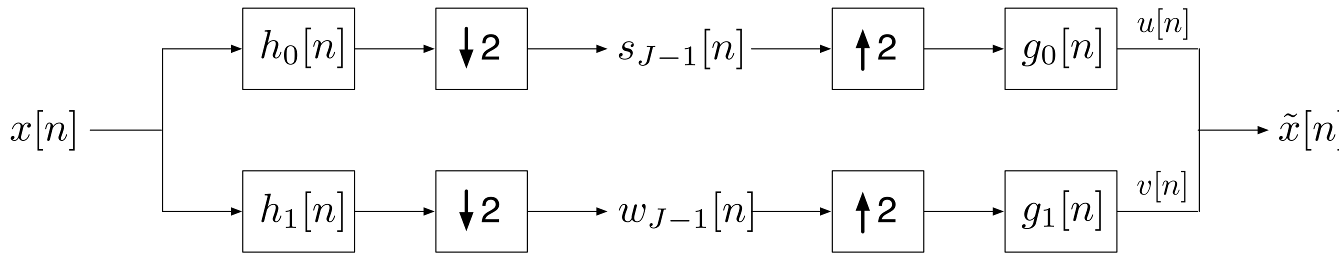
$$\begin{aligned} Y(z) &= \sum_{n=-\infty}^{\infty} y[n]z^{-n} \\ &= \sum_{n \text{ even}} x[n/2]z^{-n} \\ &= \sum_{\ell=-\infty}^{\infty} x[\ell]z^{2\ell} \\ &= \sum_{\ell=-\infty}^{\infty} x[\ell](z^2)^{\ell} \\ &= X(z^2) \end{aligned}$$

Note that this implies a *compression* of the DTFT:

$$X(e^{j\omega}) \to X(e^{j2\omega}).$$

Perfect reconstruction conditions

We are now in a position to derive conditions on the filters in a filterbank (in terms of their z-transforms) that will ensure that we perfectly reconstruct the input. Recall the architecture:



We want to ensure that $\tilde{x}[n] = x[n]$. If we use causal filters, this is not quite possible and we instead relax our notion of perfect reconstruction to instead require $\tilde{x}[n] = x[n-m]$ for some delay m.

Towards this end, note that we can write

$$S_{J-1}(z) = \frac{1}{2}\left[H_0(z^{1/2})X(z^{1/2}) + H_0(-z^{1/2})X(-z^{1/2})\right]$$

and thus

$$U(z) = \frac{1}{2}G_0(z)\left[H_0(z)X(z) + H_0(-z)X(-z)\right].$$

Similarly, we have

$$V(z) = \frac{1}{2}G_1(z)\left[H_1(z)X(z) + H_1(-z)X(-z)\right].$$

Combining these and rearranging we have

$$\begin{aligned}\widetilde{X}(z) = &\frac{1}{2}\left[G_0(z)H_0(z) + G_1(z)H_1(z)\right]X(z)\\ &+ \frac{1}{2}\left[G_0(z)H_0(-z) + G_1(z)H_1(-z)\right]X(-z).\end{aligned}$$

We want to have $\widetilde{X}(z) = z^{-m}X(z)$. The way to make this occur is straightforward: the filters must satisfy the following **perfect reconstruction conditions**:

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0 \qquad \text{(Alias cancellation)}$$

and

$$G_0(z)H_0(z) + G_1(z)H_1(z) = 2z^{-m}. \qquad \text{(No distortion)}$$

How can we design filters that will satisfy these conditions? Suppose for the moment that the filters $G_0(z)$ and $G_1(z)$ are given – what is a natural choice for $H_0(z)$ and $H_1(z)$?

$$H_0(z) = G_1(-z) \qquad \text{and} \qquad H_1(z) = -G_0(-z).$$

With these choices we immediately have that

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = G_0(z)G_1(z) - G_1(z)G_0(z) = 0,$$

and thus the alias cancellation condition is satisfied.

What do these filters look like? Consider the FIR filters with z-transforms

$$\begin{aligned} G_0(z) &= \alpha_0 + \alpha_1 z^{-1} + \alpha_2 z^{-2} \\ G_1(z) &= \beta_0 + \beta_1 z^{-1} + \beta_2 z^{-2} + \beta_3 z^{-3}. \end{aligned}$$

In this case,

$$\begin{aligned} H_0(z) &= \beta_0 - \beta_1 z^{-1} + \beta_2 z^{-2} - \beta_3 z^{-3} \\ H_1(z) &= -\alpha_0 + \alpha_1 z^{-1} - \alpha_2 z^{-2}. \end{aligned}$$

The question then becomes, how can we design $G_0(z)$ and $G_1(z)$ to ensure that they satisfy the no distortion condition? With the choices for $H_0(z)$ and $H_1(z)$ given above, we can write this condition as

$$T(z) = G_0(z)G_1(-z) - G_1(z)G_0(-z) = 2z^{-m}.$$

There are an endless possible variety of choices at this point. We will discuss the most common approach next time, but for now we simply note that if we fix $G_0(z)$, then our design problem reduces to constructing a $G_1(z)$ such that $T(z) = 2z^{-m}$, at which point the rest of the filterbank $H_0(z)$ and $H_1(z)$ are determined.

Orthogonal filterbanks

Recall that a perfect reconstruction filterbank can be designed by first constructing $G_0(z)$ and $G_1(z)$ satisfying

$$G_0(z)G_1(-z) - G_1(z)G_0(-z) = 2z^{-m}. \qquad \text{(No distortion)}$$

Once we have such a $G_0(z)$ and $G_1(z)$, we can then define the filters

$$H_0(z) = G_1(-z) \qquad \text{and} \qquad H_1(z) = -G_0(-z),$$

and automatically form a perfect reconstruction filterbank.

Thus our central challenge is to construct $G_0(z)$ and $G_1(z)$ satisfying the no distortion condition. Here we will discuss one possible solution which satisfies some particularly nice properties. Suppose that $G_0(z)$ is given. Then the **alternating flip** construction is to set

$$G_1(z) = -z^{-m}G_0(-z^{-1}),$$

where m is odd and will correspond to the total delay of the system. What does $G_1(z)$ look like in this case? Suppose that

$$G_0(z) = \alpha_0 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \alpha_3 z^{-3}.$$

Then for $m = 3$ we have

$$\begin{aligned} G_0(z^{-1}) &= \alpha_0 + \alpha_1 z + \alpha_2 z^2 + \alpha_3 z^3 \\ G_0(-z^{-1}) &= \alpha_0 - \alpha_1 z + \alpha_2 z^2 - \alpha_3 z^3 \\ -G_0(-z^{-1}) &= -\alpha_0 + \alpha_1 z - \alpha_2 z^2 + \alpha_3 z^3 \\ G_1(z) &= \alpha_3 - \alpha_2 z^{-1} + \alpha_1 z^{-2} - \alpha_0 z^{-3}. \end{aligned}$$

Note that with this construction (when m is odd), $G_1(-z) = z^{-m} G_0(z^{-1})$, and thus the no distortion condition reduces to

$$z^{-m}\left[G_0(z)G_0(z^{-1}) + G_0(-z)G_0(-z^{-1})\right] = 2z^{-m}.$$

Alternatively, if we set $P(z) = G_0(z)G_0(z^{-1})$, then this simply reduces to

$$P(z) + P(-z) = 2. \qquad \text{(No distortion (v2))}$$

Recalling that

$$P(z) = \sum_{n=-\infty}^{\infty} p[n] z^{-n},$$

we can see that the above condition on $P(z)$ reduces to

$$p[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0, n \text{ even} \\ \text{anything} & n \text{ odd.} \end{cases}$$

But what exactly is $p[n]$, and what does it tell us about $g_0[n]$? Using the fact that $P(z) = G_0(z)G_0(z^{-1})$, we have

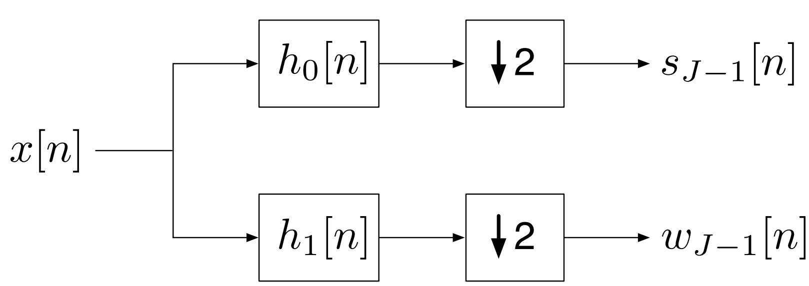
$$\begin{aligned}\sum_{n=-\infty}^{\infty} p[n]z^{-n} &= \left(\sum_{k=-\infty}^{\infty} g_0[k]z^{-k}\right)\left(\sum_{m=-\infty}^{\infty} g_0[m]z^{m}\right) \\ &= \sum_{k=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} g_0[k]g_0[m]z^{m-k} \\ &= \sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} g_0[k]g_0[k-n]z^{-n}.\end{aligned}$$

Thus, we can conclude that

$$p[n] = \sum_{k=-\infty}^{\infty} g_0[k]g_0[k-n],$$

i.e. $p[n]$ is just the autocorrelation function of $g_0[n]$, and the perfect reconstruction condition reduces to a simple constraint on this function.

To summarize, the procedure we have described to design a filterbank consists of designing $g_0[n]$ so that the autocorrelation function $p[n]$ is zero for all even n except $n = 0$, and then using the "alternating flip" construction of $g_1[n]$, which together dictate $h_0[n]$ and $h_1[n]$. In this context, the constraint on $p[n]$ has significant consequences. Specifically, for *any* filterbank designed in this way, the filterbank architecture



can be thought of as computing a representation of $x[n]$ in an orthogonal basis.

Specifically, note that here

$$s_{J-1}[n] = \sum_{k=-\infty}^{\infty} x[k]h_0[2n-k]$$
$$w_{J-1}[n] = \sum_{k=-\infty}^{\infty} x[k]h_1[2n-k].$$

If we define $\boldsymbol{u}_n = h_0[2n-k]$ and $\boldsymbol{v}_n = h_1[2n-k]$, then we can interpret this as

$$s_{J-1}[n] = \langle \boldsymbol{u}_n, \boldsymbol{x} \rangle \qquad \text{and} \qquad w_{J-1}[n] = \langle \boldsymbol{v}_n, \boldsymbol{x} \rangle.$$

The constraint on $p[n]$ turns out to be exactly what we need to ensure that $\boldsymbol{\mathcal{B}} = \{\boldsymbol{u}_n\}_n \cup \{\boldsymbol{v}_n\}$ are orthonormal. This together with the perfect reconstruction property of the filterbank implies that $\boldsymbol{\mathcal{B}}$ forms an orthonormal basis for ℓ_2.

Technical Details: Constructing $\phi_0(t)$

Note that with **(P2)** established, we know that $\phi_0(t) \in \mathcal{V}_1$. This gives us an additional interpretation of the $g_0[n]$; they tell us how to build up $\phi_0(t)$ out of shifts of the contracted version $\phi_0(2t)$:

$$\phi_0(t) = \sum_{n=-\infty}^{\infty} g_0[n]\,\sqrt{2}\phi_0(2t-n). \tag{2}$$

Given a particular $\phi_0(t)$, we can of course generate the $g_0[n]$ using (1) – but we can also go the other way. If we design a sequence $g_0[n]$ that obeys the three properties above, it specifies a unique scaling function $\phi_0(t)$. To get $\phi_0(t)$ from $g_0[n]$, we take the continuous-time Fourier transform of both sides of (2):

$$\begin{aligned}\Phi_0(j\Omega) &= \sum_{n=-\infty}^{\infty} g_0[n]\sqrt{2}\int_{-\infty}^{\infty}\phi_0(2t-n)e^{-j\Omega t}\ \mathrm{d}t\\ &= \sum_{n=-\infty}^{\infty} g_0[n]\,\frac{1}{\sqrt{2}}e^{j\Omega n/2}\Phi_0(j\Omega/2)\\ &= \frac{1}{\sqrt{2}}\overline{G(e^{j\Omega/2})}\Phi_0(j\Omega/2)\end{aligned}$$

We can again expand $\Phi_0(j\Omega/2) = \frac{1}{\sqrt{2}}\overline{G(e^{j\Omega/4})}\Phi_0(j\Omega/4)$, etc. Condition **(G3)** above means that the limit exists, and we have

$$\Phi_0(j\Omega) = \left(\prod_{p=1}^{\infty}\frac{\overline{G(e^{j2^{-p}\Omega})}}{\sqrt{2}}\right)\Phi_0(j0) = \prod_{p=1}^{\infty}\frac{\overline{G(e^{j2^{-p}\Omega})}}{\sqrt{2}},$$

since $\Phi_0(j0) = 1$ (this follows from integrating both sides of (2) and applying Condition **(G2)** above). Unfortunately, except in special cases it is hard to compute $\Phi_0(j\Omega)$ past the iterative expression above. This is why wavelets are usually specified in terms of their corresponding sequences $g_0[n]$.