Digital Signal Processing (Ohio State EE700)

By:
Phil Schniter

Digital Signal Processing (Ohio State EE700)

By:

Phil Schniter

C O N N E X I O N S

Table of Contents

Chapter 1

Multirate Signal Processing

1.1 Fundamentals of Multirate Signal Processing

1.1.1 Upsampling[1]

1.1.1.1 Upsampling

The operation of **upsampling** by factor $L \in \mathbb{N}$ describes the insertion of $L - 1$ zeros between every sample of the input signal. This is denoted by "$\uparrow (L)$" in block diagrams, as in Figure 1.1.
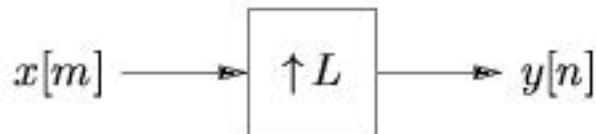
Figure 1.1

Formally, upsampling can be expressed in the time domain as

$$y[n] = \begin{cases} x\left[\frac{n}{L}\right] & \text{if } \frac{n}{L} \in \mathbb{Z} \\ 0 & \text{otherwise} \end{cases}$$

In the z-domain,

$$Y(z) = \sum_{n} y[n] z^{-n} = \sum_{n, \frac{n}{L} \in \mathbb{Z}} x\left[\frac{n}{L}\right] z^{-n} = \sum_{k} x[k] z^{(-k)L} = X\left(z^L\right)$$

and substituting $z = e^{j\omega}$ for the DTFT,

$$Y\left(e^{j\omega}\right) = X\left(e^{j\omega L}\right) \tag{1.1}$$

---

[1] This content is available online at $<$http://cnx.org/content/m10403/2.15/$>$.

As shown in Figure 1.2, upsampling compresses the DTFT by a factor of L along with the ω axis.
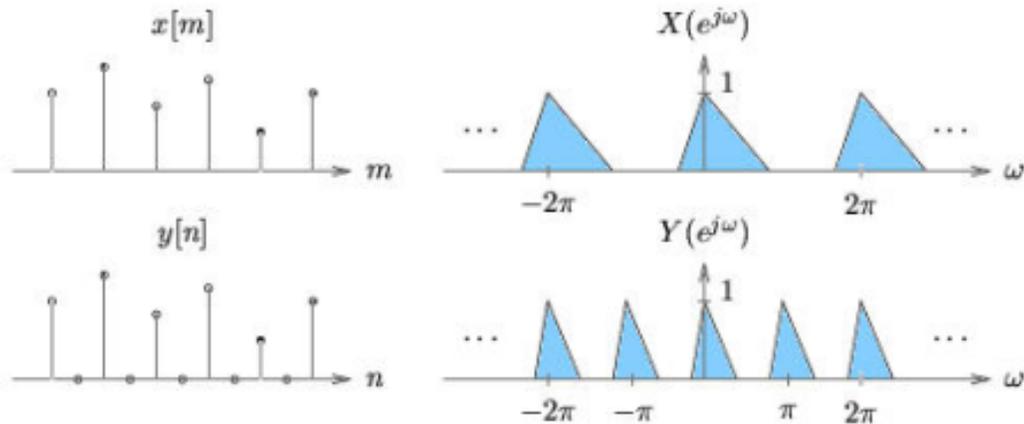
Figure 1.2

1.1.2 Downsampling[2]

The operation of **downsampling** by factor $M \in \mathbb{N}$ describes the process of keeping every M^{th} sample and discarding the rest. This is denoted by " $\downarrow (M)$" in block diagrams, as in Figure 1.3.
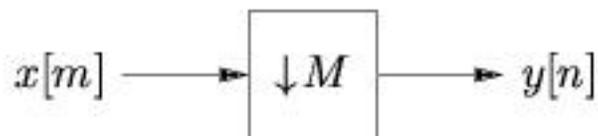
Figure 1.3

Formally, downsampling can be written as

$$y[n] = x[nM]$$

[2]This content is available online at $<$http://cnx.org/content/m10441/2.12/$>$.

In the z domain,

$$
\begin{aligned}
Y(z) &= \sum_n y[n] z^{-n} \\
&= \sum_n x[nM] z^{-n} \\
&= \sum_m x[m] \left(\frac{1}{M} \sum_{p=0}^{M-1} e^{j\frac{2\pi}{M}pm} \right) z^{\frac{-m}{M}}
\end{aligned}
\tag{1.2}
$$

where $\frac{1}{M} \sum_{p=0}^{M-1} e^{j\frac{2\pi}{M}pm} = \begin{cases} 1 & \text{if } m \text{ is a multiple of } M \\ 0 & \text{otherwise} \end{cases}$

$$
\begin{aligned}
Y(z) &= \frac{1}{M} \sum_{p=0}^{M-1} \sum_m x[m] \left(e^{-\left(j\frac{2\pi}{M}p\right)} z^{\frac{1}{M}} \right)^{-m} \\
&= \frac{1}{M} \sum_{p=0}^{M-1} X \left(e^{-\left(j\frac{2\pi}{M}p\right)} z^{\frac{1}{M}} \right)
\end{aligned}
\tag{1.3}
$$

Translating to the frequency domain,

$$
Y\left(e^{j\omega}\right) = \frac{1}{M} \sum_{p=0}^{M-1} X\left(e^{j\frac{\omega-2\pi p}{M}}\right)
\tag{1.4}
$$

As shown in Figure 1.4, downsampling expands each 2π -periodic repetition of $X\left(e^{j\omega}\right)$ by a factor of M along the ω axis, and reduces the gain by a factor of M. If $x[m]$ is not bandlimited to $\frac{\pi}{M}$, aliasing may result from spectral overlap.

NOTE: When performing a frequency-domain analysis of systems with up/downsamplers, it is strongly recommended to carry out the analysis in the z-domain until the last step, as done above. Working directly in the $e^{j\omega}$-domain can easily lead to errors.
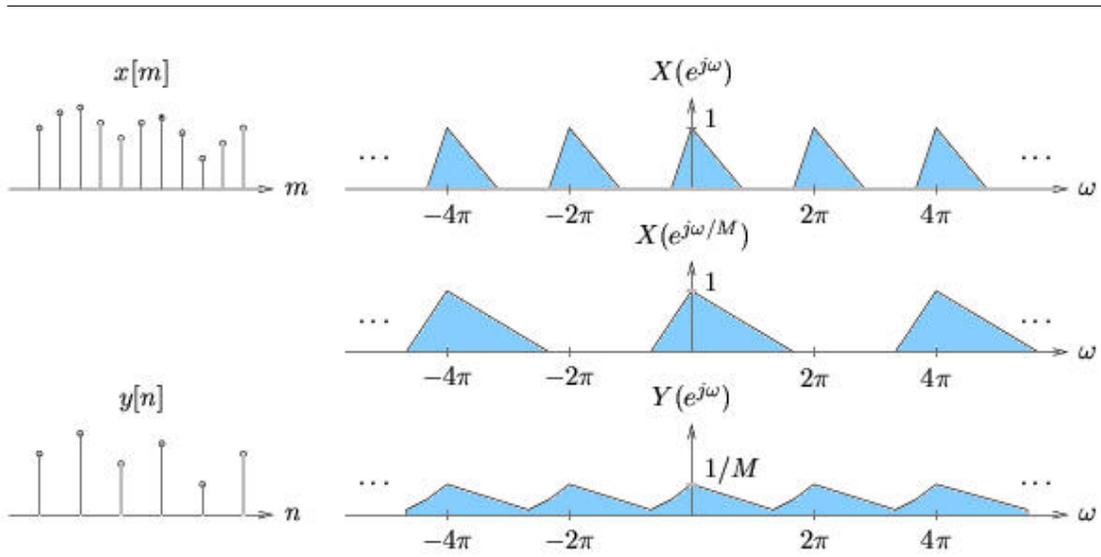
Figure 1.4

1.1.3 Interpolation[3]

1.1.3.1 Interpolation

Interpolation is the process of upsampling and filtering a signal to increase its effective sampling rate. To be more specific, say that $x[m]$ is an (unaliased) T-sampled version of $x_c(t)$ and $v[n]$ is an L-upsampled version version of $x[m]$. If we filter $v[n]$ with an ideal $\frac{\pi}{L}$-bandwidth lowpass filter (with DC gain L) to obtain $y[n]$, then $y[n]$ will be a $\frac{T}{L}$-sampled version of $x_c(t)$. This process is illustrated in Figure 1.5.
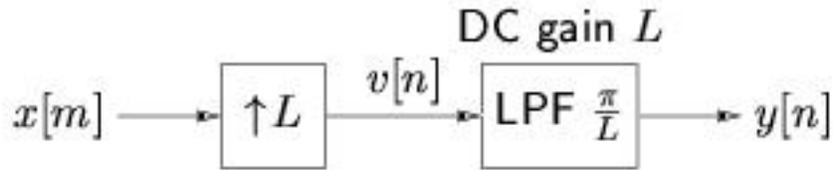
Figure 1.5

We justify our claims about interpolation using frequency-domain arguments. From the sampling theorem, we know that T- sampling $x_c(t)$ to create $x[n]$ yields

$$X\left(e^{j\omega}\right) = \frac{1}{T}\sum_k X_c\left(j\frac{\omega - 2\pi k}{T}\right) \tag{1.5}$$

After upsampling by factor L, (1.5) implies

$$V\left(e^{j\omega}\right) = \frac{1}{T}\sum_k X_c\left(j\frac{\omega L - 2\pi k}{T}\right) = \frac{1}{T}\sum_k X_c\left(j\frac{\omega - \frac{2\pi}{L}k}{\frac{T}{L}}\right)$$

Lowpass filtering with cutoff $\frac{\pi}{L}$ and gain L yields

$$Y\left(e^{j\omega}\right) = \frac{L}{T}\sum_{\frac{k}{L}\in\mathbb{Z}} X_c\left(j\frac{\omega - \frac{2\pi}{L}k}{\frac{T}{L}}\right) = \frac{L}{T}\sum_l X_c\left(j\frac{\omega - 2\pi l}{\frac{T}{L}}\right)$$

since the spectral copies with indices other than $k = lL$ (for $l \in \mathbb{Z}$) are removed. Clearly, this process yields a $\frac{T}{L}$-shaped version of $x_c(t)$. Figure 1.6 illustrates these frequency-domain arguments for $L = 2$.

---

[3]This content is available online at $<$http://cnx.org/content/m10444/2.14/$>$.
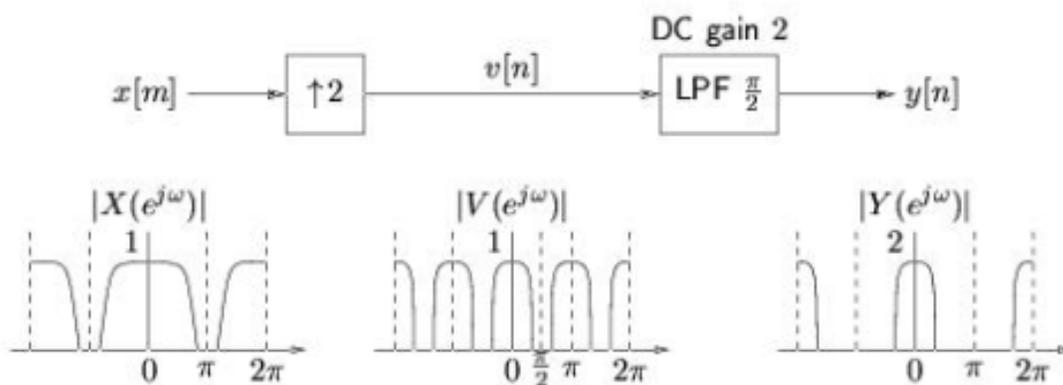
Figure 1.6

1.1.4 Application of Interpolation - Oversampling in CD Players[4]

1.1.4.1 Application of Interpolation- Oversampling in CD Players

The digital audio signal on a CD is a 44.1kHz sampled representation of a continuous signal with bandwidth 20kHz. With a standard ZOH-DAC, the analog reconstruction filter would have passband edge at 20kHz and stopband edge at 24.1kHz. (See Figure 1.7) With such a narrow transition band, this would be a difficult (and expensive) filter to build.
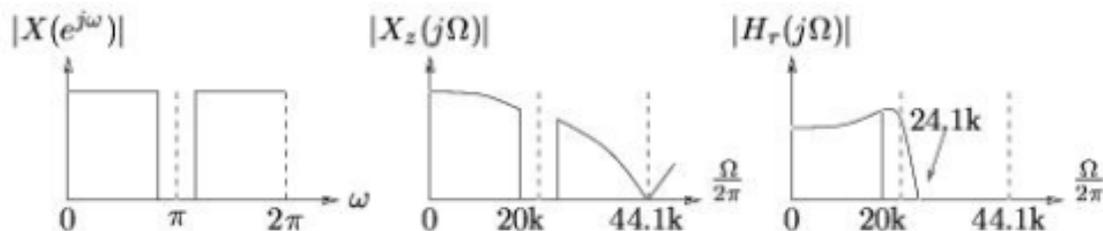
Figure 1.7

If digital interpolation is used prior to reconstruction, the effective sampling rate can be increased and the reconstruction filter's transition band can be made much wider, resulting in a much simpler (and cheaper) analog filter. Figure 1.8 illustrates the case of interpolation by 4. The reconstruction filter has passband edge at 20kHz and stopband edge at 156.4kHz, resulting in a much wider transition band and therefore an easier filter design.

---

[4]This content is available online at <http://cnx.org/content/m11006/2.3/>.
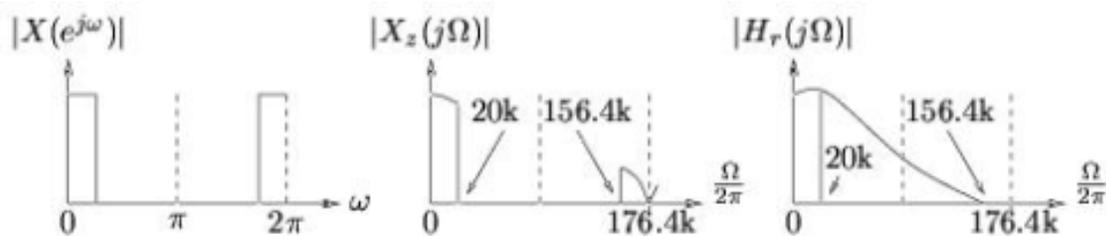
Figure 1.8

1.1.5 Decimation[5]

Decimation is the process of filtering and downsampling a signal to decrease its effective sampling rate, as illustrated in Figure 1.9. The filtering is employed to prevent aliasing that might otherwise result from downsampling.
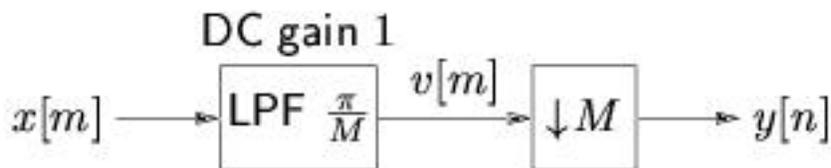
Figure 1.9

To be more specific, say that

$$x_c(t) = x_l(t) + x_b(t)$$

where $x_l(t)$ is a lowpass component bandlimited to $\frac{1}{2MT}$ Hz and $x_b(t)$ is a bandpass component with energy between $\frac{1}{2MT}$ and $\frac{1}{2T}$ Hz. If sampling $x_c(t)$ with interval T yields an unaliased discrete representation $x[m]$, then decimating $x[m]$ by a factor M will yield $y[n]$, an unaliased MT-sampled representation of lowpass component $x_l(t)$.

We offer the following justification of the previously described decimation procedure. From the sampling theorem, we have

$$X\left(e^{j\omega}\right) = \frac{1}{T}\sum_k X_l\left(j\frac{\omega - 2\pi k}{T}\right) + \frac{1}{T}\sum_k X_b\left(j\frac{\omega - 2\pi k}{T}\right)$$

The bandpass component $X_b(j\Omega)$ is the removed by $\frac{\pi}{M}$-lowpass filtering, giving

$$V\left(e^{j\omega}\right) = \frac{1}{T}\sum_k X_l\left(j\frac{\omega - 2\pi k}{T}\right)$$

---

[5]This content is available online at <http://cnx.org/content/m10445/2.11/>.

Finally, downsampling yields

$$
\begin{aligned}
Y\left(e^{j\omega}\right) &= \frac{1}{MT} \sum_{p=0}^{M-1} \sum_{k} X_l \left(j\frac{\frac{\omega-2\pi p}{M}-2\pi k}{T}\right) \\
&= \frac{1}{MT} \sum_{p=0}^{M-1} \sum_{k} X_l \left(j\frac{\omega-(2\pi)(kM+p)}{MT}\right) \\
&= \frac{1}{MT} \sum_{l} X_l \left(j\frac{\omega-2\pi l}{MT}\right)
\end{aligned}
\tag{1.6}
$$

which is clearly a MT-sampled version of $x_l(t)$. A frequency-domain illustration for $M=2$ appears in Figure 1.10.
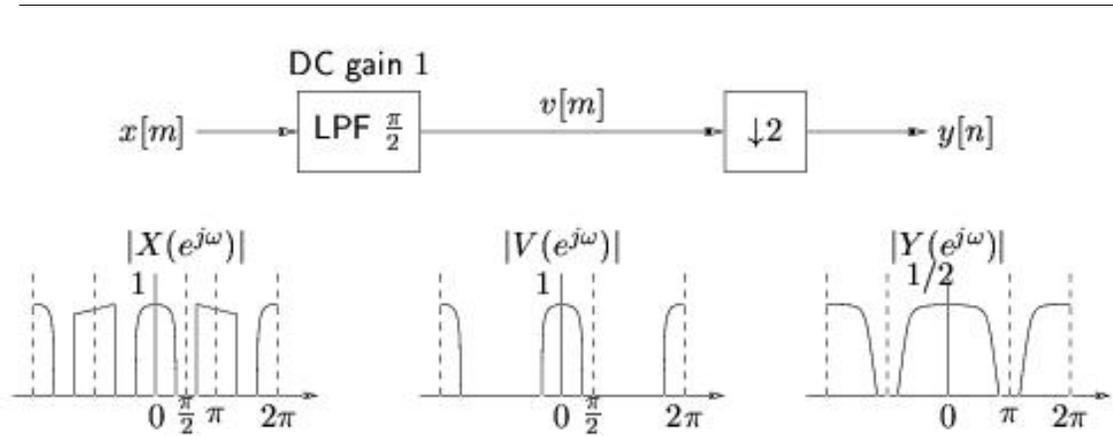
Figure 1.10

1.1.6 Resampling with Rational Factor[6]

Interpolation by L and decimation by M can be combined to change the effective sampling rate of a signal by the rational factor $\frac{L}{M}$. This process is called **resampling** or **sample-rate conversion**. Rather than cascading an anti-imaging filter for interpolation with an anti-aliasing filter for decimation, we implement one filter with the minimum of the two cutoffs $\left[\frac{\pi}{L}, \frac{\pi}{M}\right]$ and the multiplication of the two DC gains (L and 1), as illustrated in Figure 1.11.

---

[6] This content is available online at <http://cnx.org/content/m10448/2.11/>.
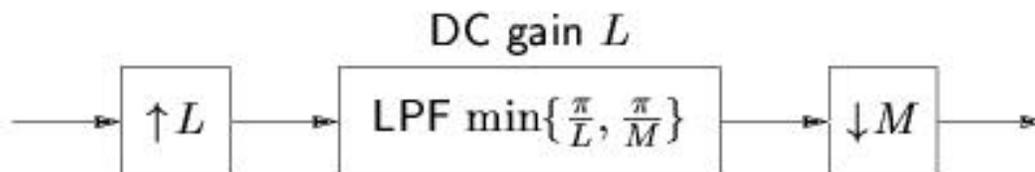
Figure 1.11

1.1.7 Digital Filter Design for Interpolation and Decimation[7]

First we treat filter design for interpolation. Consider an input signal $x[n]$ that is ω_0-bandlimited in the DTFT domain. If we upsample by factor L to get $v[m]$, the desired portion of $V\left(e^{j\omega}\right)$ is the spectrum in $\left[\frac{-\pi}{L}, \frac{\pi}{L}\right)$, while the undesired portion is the remainder of $[-\pi, \pi)$. Noting from Figure 1.12 that $V\left(e^{j\omega}\right)$ has zero energy in the regions

$$\left[\frac{2k\pi + \omega_0}{L}, \frac{2(k+1)\pi - \omega_0}{L}\right), k \in \mathbb{Z} \tag{1.7}$$

the anti-imaging filter can be designed with transition bands in these regions (rather than passbands or stopbands). For a given number of taps, the additional degrees of freedom offered by these transition bands allows for better responses in the passbands and stopbands. The resulting filter design specifications are shown in the bottom subplot below (Figure 1.12).

---

[7]This content is available online at <http://cnx.org/content/m10870/2.6/>.
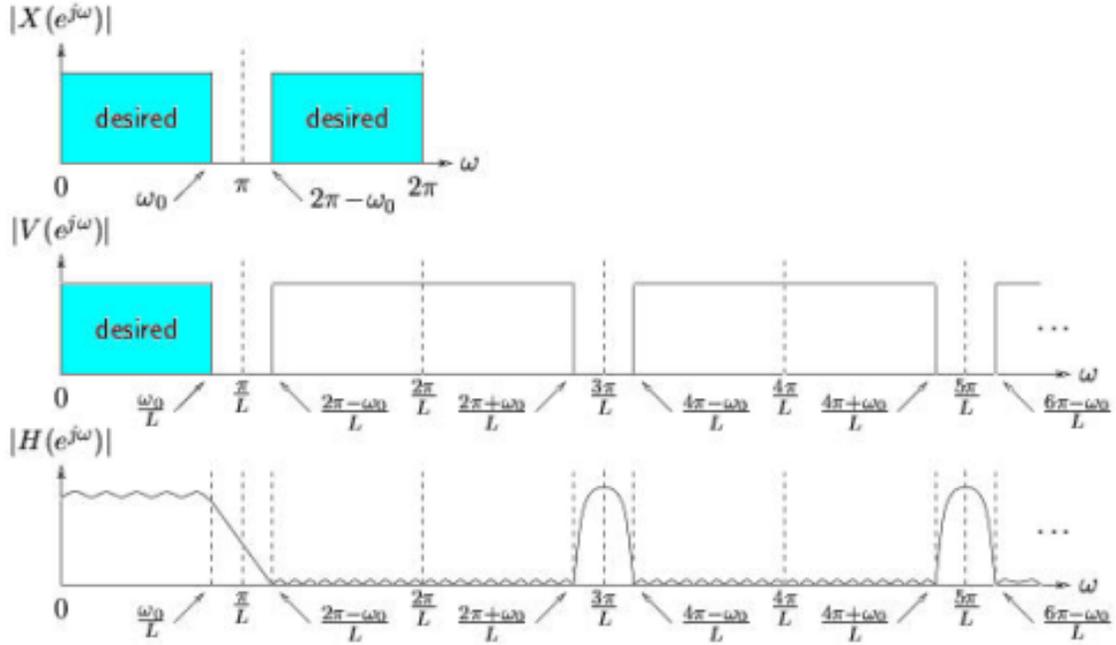
Figure 1.12

Next we treat filter design for decimation. Say that the **desired** spectral component of the input signal is bandlimited to $\frac{\omega_0}{M} < \frac{\pi}{M}$ and we have decided to downsample by M. The goal is to minimally distort the input spectrum over $\left[\frac{-\omega_0}{M}, \frac{\omega_0}{M}\right)$, *i.e.*, the post-decimation spectrum over $[-\omega_0, \omega_0)$. Thus, we must not allow any aliased signals to enter $[-\omega_0, \omega_0)$. To allow for extra degrees of freedom in the filter design, we **do** allow aliasing to enter the post-decimation spectrum outside of $[-\omega_0, \omega_0)$ within $[-\pi, \pi)$. Since the input spectral regions which alias outside of $[-\omega_0, \omega_0)$ are given by

$$\left[\frac{2k\pi + \omega_0}{L}, \frac{2(k+1)\pi - \omega_0}{L}\right), k \in \mathbb{Z} \tag{1.8}$$

(as shown in Figure 1.13), we can treat these regions as transition bands in the filter design. The resulting filter design specifications are illustrated in the middle subplot (Figure 1.13).
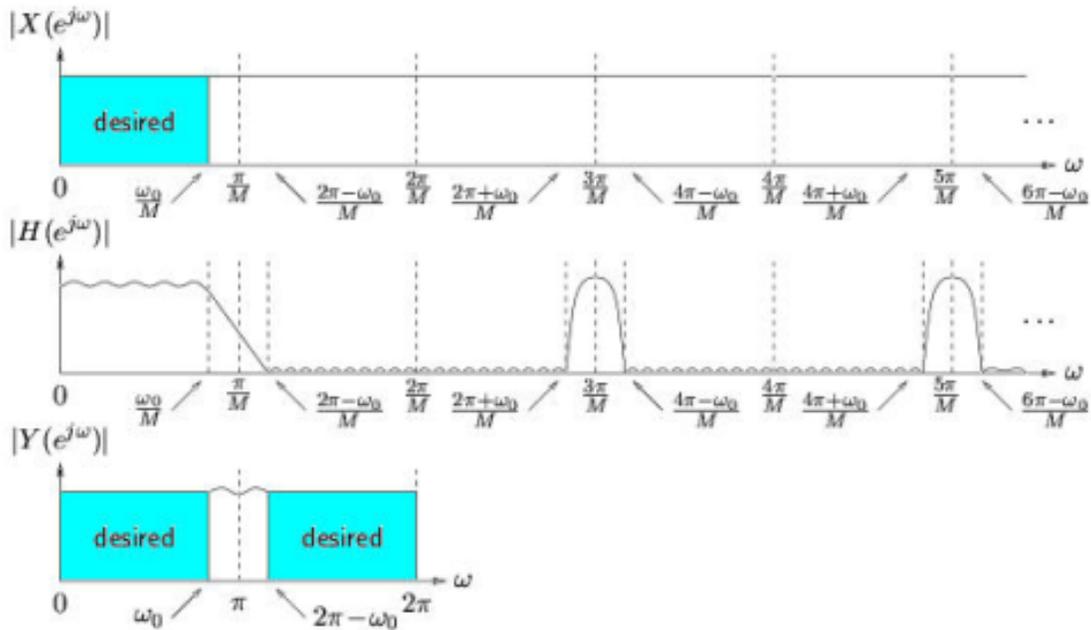
Figure 1.13

1.1.8 Noble Identities[8]

1.1.8.1

The Noble identities (illustrated in Figure 1.14 and Figure 1.15) describe when it is possible to reverse the order of upsampling/downsampling and filtering. We prove the Noble identities showing the equivalence of each pair of block diagrams.

The Noble identity for interpolation can be depicted as in Figure 1.14:

Figure 1.14

For the left side of the diagram, we have

$$Y(z) = H(z^L) V_1(z)$$

---

[8]This content is available online at $<$http://cnx.org/content/m10432/2.12/$>$.

where $V_1(z) = X(z^L)$

$$Y(z) = H(z^L) X(z^L)$$

while for the right side,

$$Y(z) = V_2(z^L)$$

where $V_2(z) = H(z) X(z)$

$$Y(z) = H(z^L) X(z^L)$$

Thus we have established the Noble identity for interpolation.

The Noble identity for decimation can be depicted as in Figure 1.15:

Figure 1.15

For the left side of the preceding diagram, we have

$$V_1(z) = \frac{1}{M} \sum_{k=0}^{M-1} X\left(e^{(-j)\frac{2\pi}{M}k} z^{\frac{1}{M}}\right)$$

$$
\begin{aligned}
Y(z) &= H(z) V_1(z) \\
&= H(z)\left(\frac{1}{M} \sum_{k=0}^{M-1} X\left(e^{(-j)\frac{2\pi}{M}k} z^{\frac{1}{M}}\right)\right)
\end{aligned}
\tag{1.9}
$$

while for the right side,

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} V_z\left(e^{(-j)\frac{2\pi}{M}k} z^{\frac{1}{M}}\right) \tag{1.10}$$

where $V_2(z) = X(z) H(z^M)$

$$
\begin{aligned}
Y(z) &= \frac{1}{M} \sum_{k=0}^{M-1} X\left(e^{(-j)\frac{2\pi}{M}k} z^{\frac{1}{M}}\right) H\left(e^{(-j)\frac{2\pi}{M}kM} z^{\frac{M}{M}}\right) \\
&= H(z) \frac{1}{M} \sum_{k=0}^{M-1} X\left(e^{(-j)\frac{2\pi}{M}k} z^{\frac{1}{M}}\right)
\end{aligned}
\tag{1.11}
$$

Thus we have established the Noble identity for decimation. Note that the impulse response of $H(z^L)$ is the L-upsampled impulse response of $H(z)$.

1.1.9 Polyphase Interpolation[9]

1.1.9.1 Polyphase Interpolation Filter

Recall the standard interpolation procedure illustrated in Figure 1.16.
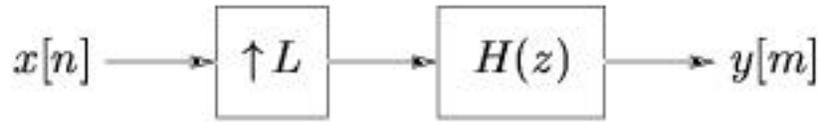
Figure 1.16

Note that this procedure is computationally inefficient because the lowpass filter operates on a sequence that is mostly composed of zeros. Through the use of the Noble identities, it is possible to rearrange the preceding block diagram so that operations on zero-valued samples are avoided.

In order to apply the Noble identity for interpolation, we must transform $H(z)$ into its upsampled polyphase components $H_p(z^L)$, $p = \{0, \ldots, L-1\}$.

$$
\begin{aligned}
H(z) &= \sum_{nn} h[n] z^{-n} \\
&= \sum_{kk} \sum_{p=0}^{L-1} h[kL+p] z^{-(kL+p)}
\end{aligned}
\tag{1.12}
$$

via $k := \lfloor \frac{n}{L} \rfloor$, $p := n \bmod L$

$$
H(z) = \sum_{p=0}^{L-1} \sum_{kk} h_p[k] z^{-(kL)} z^{-p}
\tag{1.13}
$$

via $h_p[k] := h[kL+p]$

$$
H(z) = \sum_{p=0}^{L-1} H_p(z^L) z^{-p}
\tag{1.14}
$$

Above, $\lfloor \cdot \rfloor$ denotes the floor operator and $\cdot \bmod M$ the modulo-M operator. Note that the pth polyphase filter $h_p[k]$ is constructed by downsampling the "master filter" $h[n]$ at offset p. Using the unsampled polyphase components, the Figure 1.16 diagram can be redrawn as in Figure 1.17.
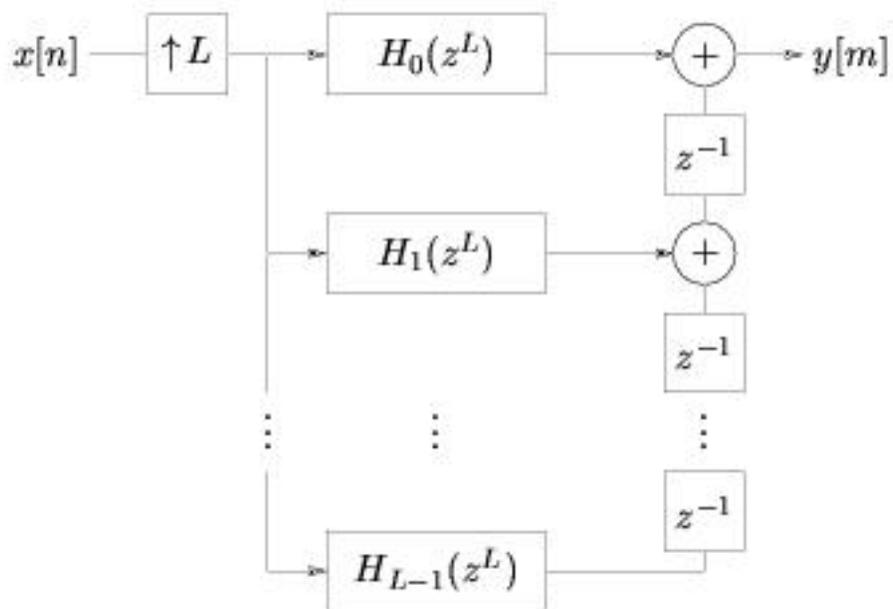
Figure 1.17

Applying the Noble identity for interpolation to Figure 1.18 yields Figure 1.17. The ladder of upsamplers and delays on the right below (Figure 1.17) accomplishes a form of parallel-to-serial conversion.
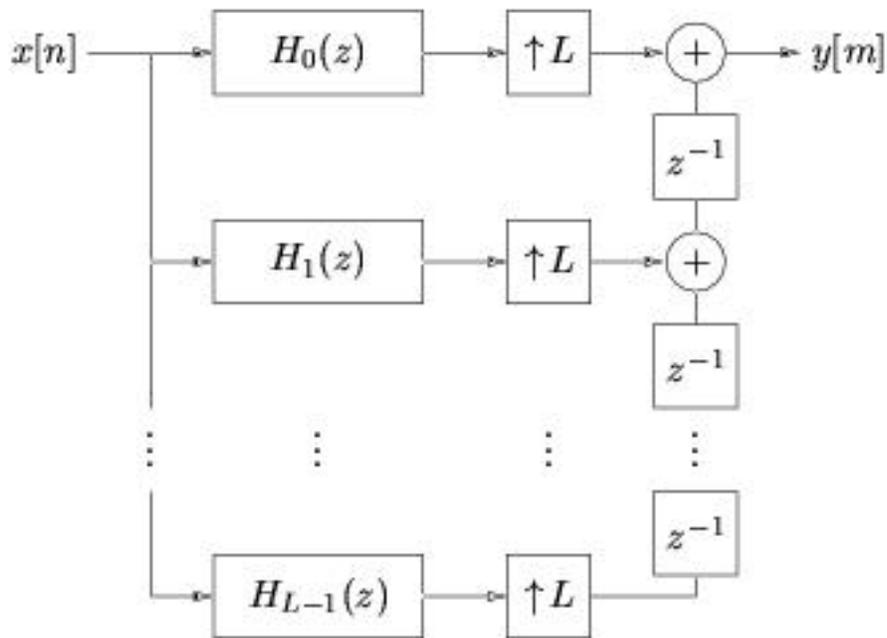
Figure 1.18

1.1.10 Polyphase Decimation Filter[10]

1.1.10.1 Polyphase Decimation

Recall the standard decimation method in Figure 1.19.
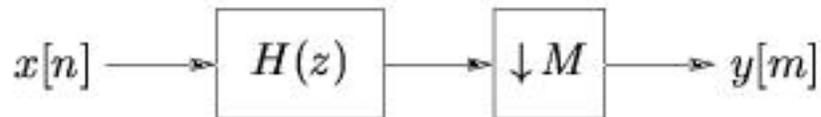
Figure 1.19

Note that this procedure is computationally inefficient because it discards the majority of the computed filter outputs. Through the use of the Noble identities, it is possible to rearrange Figure 1.19 so that filter outputs are not discarded.

---

[10]This content is available online at $<$http://cnx.org/content/m10433/2.12/$>$.

In order to apply the Noble identity for decimation, we must transform $H(z)$ into its upsampled polyphase components $H_p(z^M)$, $p = \{0, \ldots, M-1\}$, defined previously in the context of polyphase interpolation (Section 1.1.9).

$$
\begin{aligned}
H(z) &= \sum_{nn} h[n] z^{-n} \\
&= \sum_{kk} \sum_{p=0}^{M-1} h[kM+p] z^{(-(kM))-p}
\end{aligned}
\tag{1.15}
$$

via $k := \lfloor \frac{n}{M} \rfloor$, $p := n \bmod M$

$$
H(z) = \sum_{p=0}^{M-1} \sum_{k} h_p[k] z^{-(kM)} z^{-p}
\tag{1.16}
$$

via $h_p[k] := h[kM+p]$

$$
H(z) = \sum_{p=0}^{M-1} H_p(z^M) z^{-p}
\tag{1.17}
$$

Using these unsampled polyphase components, the preceding block diagram can be redrawn as Figure 1.20.
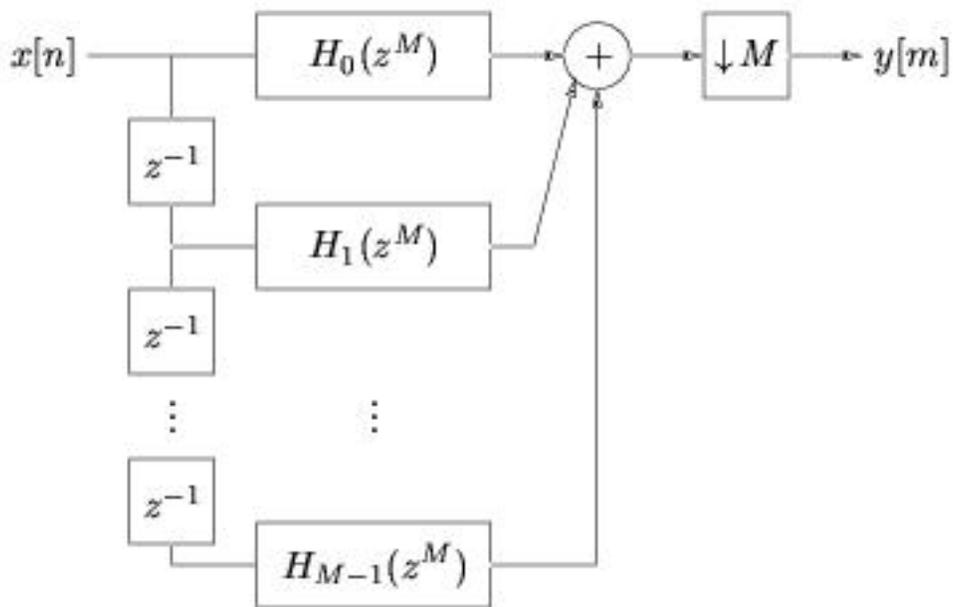
Figure 1.20

Applying the Noble identity for decimation to Figure 1.20 yields Figure 1.21. The ladder of delays and downsamplers on the left below accomplishes a form of serial-to-parallel conversion.
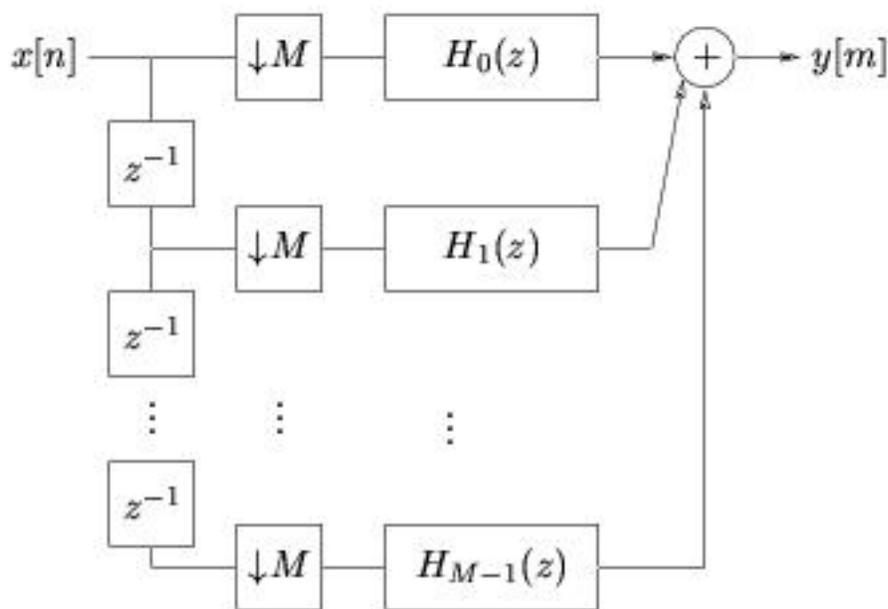
Figure 1.21

1.1.11 Computational Savings of Polyphase Interpolation/Decimation[11]

1.1.11.1 Computational Savings of Polyphase Interpolation/Decimation

Assume that we design FIR LPF $H(z)$ with N taps, requiring N multiplies per output. For standard decimation by factor M, we have N multiplies per intermediate sample and M intermediate samples per output, giving NM multiplies per output.

For polyphase decimation, we have $\frac{N}{M}$ multiplies per branch and M branches, giving a total of N multiplies per output. The assumption of $\frac{N}{M}$ multiplies per branch follows from the fact that $h[n]$ is downsampled by M to create each polyphase filter. Thus, we conclude that the standard implementation requires M times as many operations as its polyphase counterpart. (For decimation, we count multiples per output, rather than per input, to avoid confusion, since only every M^{th} input produces an output.)

From this result, it appears that the number of multiplications required by polyphase decimation is independent of the decimation rate M. However, it should be remembered that the length N of the $\frac{\pi}{M}$-lowpass FIR filter $H(z)$ will typically be proportional to M. This is suggested, *e.g.*, by the Kaiser FIR-length approximation formula

$$N \simeq \frac{-10\log(\delta_p \delta_s) - 13}{2.324\Delta(\omega)}$$

where $\Delta(\omega)$ in the transition bandwidth in radians, and δ_p and δ_s are the passband and stopband ripple levels. Recall that, to preserve a fixed signal bandwidth, the transition bandwidth $\Delta(\omega)$ will be linearly proportional

---

[11]This content is available online at <http://cnx.org/content/m11008/2.2/>.

to the cutoff $\frac{\pi}{M}$, so that N will be linearly proportional to M. In summary, polyphase decimation by factor M requires N multiplies per output, where N is the filter length, and where N is linearly proportional to M.

Using similar arguments for polyphase interpolation, we could find essentially the same result. Polyphase interpolation by factor L requires N multiplies per input, where N is the filter length, and where N is linearly proportional to the interpolation factor L. (For interpolation we count multiplies per input, rather than per output, to avoid confusion, since M outputs are generated in parallel.)

1.1.12 A Group-Delay Interpretation of Polyphase Filters[12]

Previously, polyphase interpolation (Section 1.1.9) and decimation (Section 1.1.10) were derived from the Noble identities and motivated for reasons of computational efficiency. Here we present a different interpretation of the (ideal) polyphase filter.

Assume that $H(z)$ is an ideal lowpass filter with gain L, cutoff $\frac{\pi}{L}$, and constant group delay of d:

$$H\left(e^{j\omega}\right) = \begin{cases} Le^{-(jd\omega)} & \text{if } \omega \in \left[-\frac{\pi}{L}, \frac{\pi}{L}\right) \\ 0 & \text{if } \omega \in \left[-\pi, -\frac{\pi}{L}\right) \\ & \left[\frac{\pi}{L}, \pi\right) \end{cases}$$

Recall that the polyphase filters are defined as

$$h_p[k] = h[kL + p] \quad, \quad p \in \{0, \ldots, L-1\}$$

In other words, $h_p[k]$ is an advanced (by p samples) and downsampled (by factor L) version of $h[n]$ (see Figure 1.22).
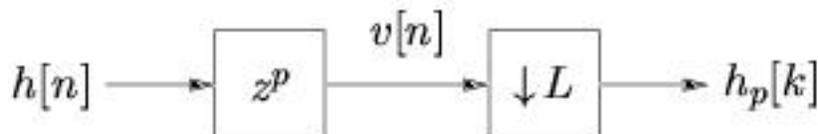
Figure 1.22

The DTFT of the pth polyphase filter impulse response is then

$$H_p(z) = \frac{1}{L} \sum_{l=0}^{L-1} V\left(e^{(-j)\frac{2\pi}{L}l}z^{\frac{1}{L}}\right) \tag{1.18}$$

where $V(z) = H(z)z^p$

$$H_p(z) = \frac{1}{L} \sum_{l=0}^{L-1} e^{(-j)\frac{2\pi}{L}lp}z^{\frac{p}{L}}H\left(e^{(-j)\frac{2\pi}{L}l}z^{\frac{1}{L}}\right) \tag{1.19}$$

---

[12]This content is available online at <http://cnx.org/content/m10438/2.12/>.

$$
\begin{aligned}
H_p\left(e^{j\omega}\right) &= \frac{1}{L}\sum_{l=0}^{L-1} e^{j\frac{\omega-2\pi l}{L}p} H\left(e^{j\frac{\omega-2\pi l}{L}}\right) \\
&= \frac{1}{L}\left(e^{j\frac{\omega}{L}p} H\left(e^{j\frac{\omega}{L}}\right)\right) \quad,\quad |\omega|\le\pi \\
&= e^{(-j)\frac{d-p}{L}\omega} \quad,\quad |\omega|\le\pi
\end{aligned}
\tag{1.20}
$$

Thus, the ideal p^{th} polyphase filter has a constant magnitude response of one and a constant group delay of $\frac{d-p}{L}$ samples. The implication is that if the input to the pth polyphase filter is the unaliased T-sampled representation $x[n] = x_c(nT)$, then the output of the filter would be the unaliased T-sampled representation $y_p[n] = x_c\left(\left(n - \frac{d-p}{L}\right)T\right)$ (see Figure 1.23).
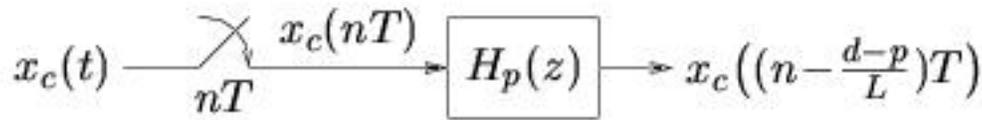
Figure 1.23

Figure 1.24 shows the role of polyphase interpolation filters assume zero-delay ($d = 0$) processing. Essentially, the pth filter interpolates the waveform $\frac{p}{L}$-way between consecutive input samples. The L polyphase outputs are then interleaved to create the output stream. Assuming that $x_c(t)$ is bandlimited to $\frac{1}{2T}$Hz, perfect polyphase filtering yields a perfectly interpolated output. In practice, we use the casual FIR approximations of the polyphase filters $h_p[k]$ (which which correspond to some casual FIR approximation of the master filter $h[n]$).
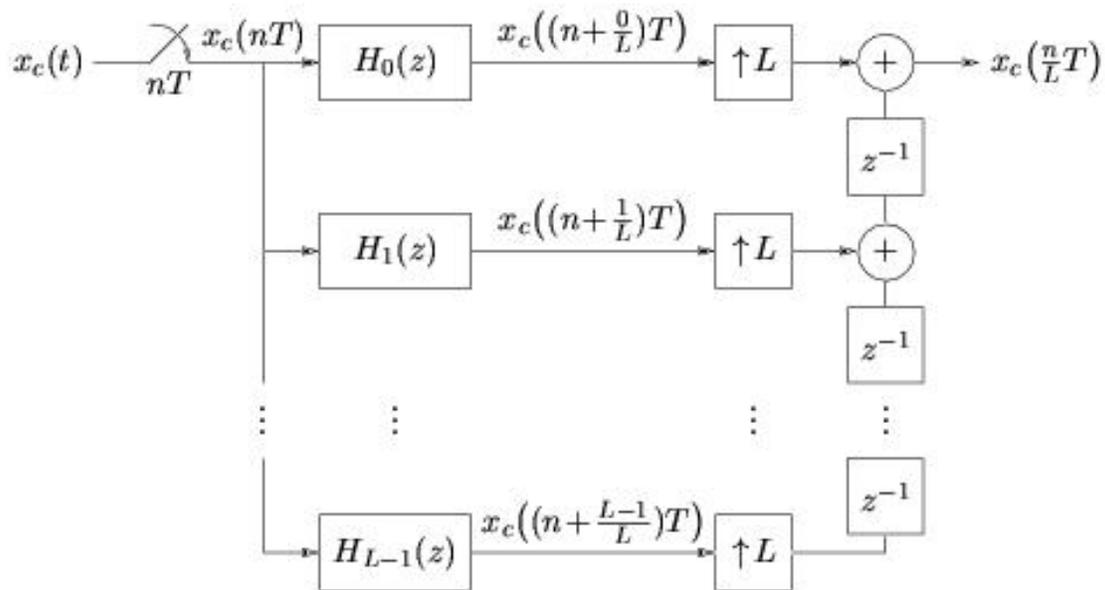
Figure 1.24

1.1.13 Polyphase Resampling with a Rational Factor[13]

1.1.13.1 Polyphase Resampling with a Rational Factor

Recall that resampling by rational rate $\frac{L}{M}$ can be accomplished through the following three-stage process(see Figure 1.25).
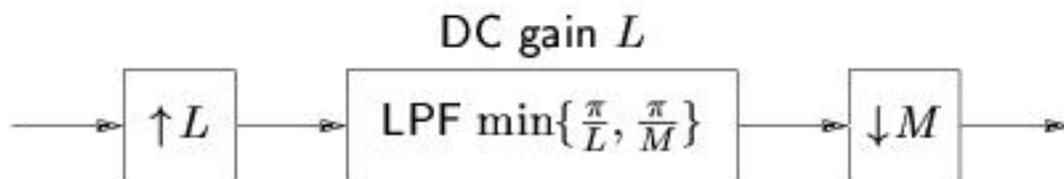
Figure 1.25

If we implemented the upsampler/LPF pair with a polyphase filterbank, we would still waste computations due to eventual downsampling by M. Alternatively, if we implemented the LPF/downsampler pair

---
[13]This content is available online at $<$http://cnx.org/content/m10443/2.9/$>$.

with a polyphase filterbank, we would waste computations by feeding it the (mostly-zeros) upsampler output. Thus, we need to examine this problem in more detail.

Assume for the moment that we implemented the upsampler/LPF pair with a polyphase filterbank, giving the architecture in Figure 1.26.
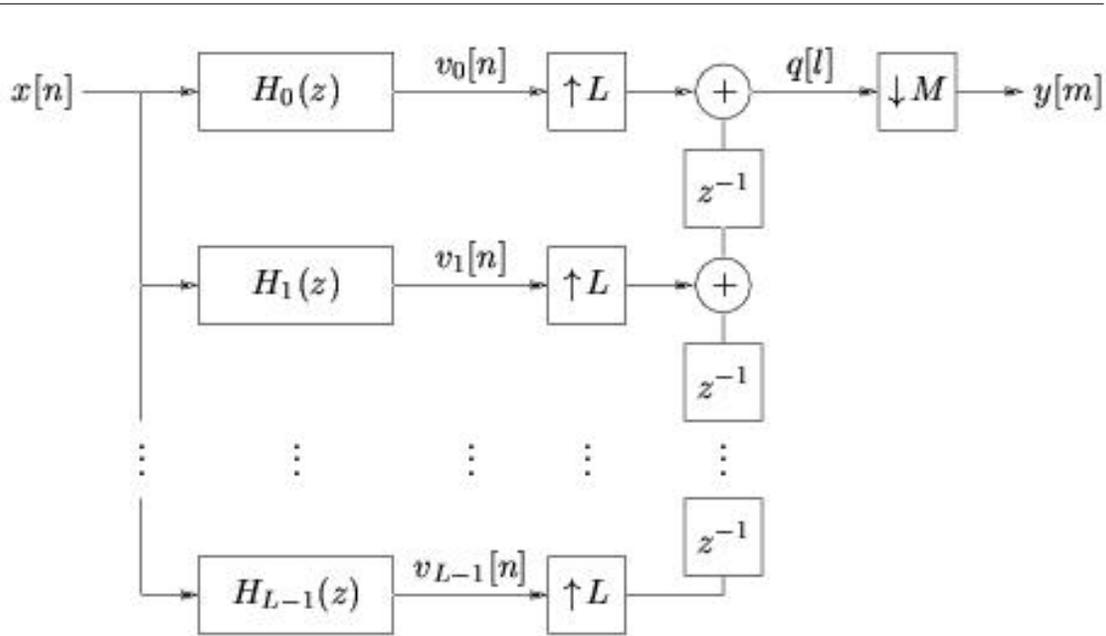
Figure 1.26

Keeping the "parallel-to-serial" interpretation of the upsampler/delay ladder in mind, the input sequence to the decimator $q[l]$ has the form as in Figure 1.27

$$
\ldots, \quad v_0[0], v_1[0], v_2[0], \ldots, v_{L-1}[0],
$$
$$
v_0[1], v_1[1], v_2[1], \ldots, v_{L-1}[1],
$$
$$
v_0[2], v_1[2], v_2[2], \ldots, v_{L-1}[2], \ldots
$$

Figure 1.27

leading to the observation that

$$q[l] = v_{<l>_L}\left[\lfloor\frac{l}{L}\rfloor\right]$$

$$
\begin{aligned}
y[m] &= q[mM] \\
&= v_{<mM>_L}\left[\lfloor\frac{mM}{L}\rfloor\right] \\
&= \sum_k h_{<mM>_L}[k]\, x\left[\lfloor\frac{mM}{L}\rfloor - k\right]
\end{aligned}
\tag{1.21}
$$

Thus, to calculate the resampled output at output index m, we should calculate only the output of branch number $mM mod L$ at input index $\lfloor\frac{mM}{L}\rfloor$. No other branch outputs are calculated, so that no computations are wasted. The resulting structure is depicted in Figure 1.28.
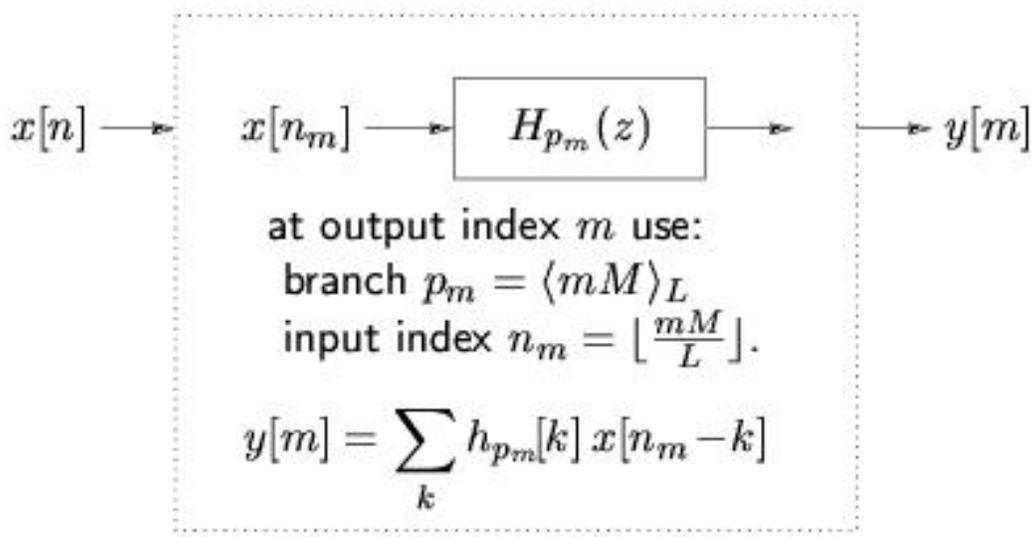
Figure 1.28

An equally-efficient structure could be obtained if we implemented the LPF/downsampler using the M-branch polyphase decimator which was fed with the proper sequence of input samples. However, this structure is not as elegant: rather than computing the output of **one** particular polyphase branch per output sample, we would need to add **all** branch outputs, but where each branch output was calculated using a particular **subset** of polyphase taps.

1.1.14 Computational Savings of Polyphase Resampling[14]

1.1.14.1 Computational Savings of Polyphase Resampling

Recall the standard (non-polyphase) resampler in Figure 1.29.

[14]This content is available online at <http://cnx.org/content/m11009/2.1/>.
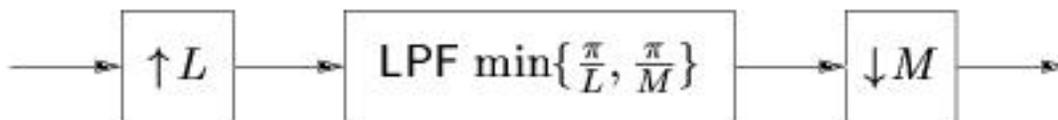
Figure 1.29

For simplicity, assume that $L > M$. Since the length of an FIR filter is inversely proportional to the transition bandwidth (recalling Kaiser's formula), and the transition bandwidth is directionally proportional to the cutoff frequency, we model the lowpass filter length as $N = \alpha L$, where α is a constant that determines the filter's (and thus the resampler's) performance (independent of L and M). To compute one output point, we require M filter outputs, each requiring $N = \alpha L$ multiplies, giving a total of αLM multiplies per output.

In the polyphase implementation, calculation of one output point requires the computation of only one polyphase filter output. With $N = \alpha L$ master filter taps and L branches, the polyphase filter length is α, so that only α multiplies are required per output. Thus, the polyphase implementation saves a factor of LM multiplies over the standard implementation!

1.1.15 CD to DAT Conversion[15]

1.1.15.1 Example: CD to DAT rate conversion

Digital audio signals stored on compact digital discs (CDs) are sampled at 44.1kHz, while those stored on digital audio tapes (DATs) are sampled at 48kHz. Conversion from CD to DAT requires an exchange rate of

$$Q = \frac{48000}{44100} = \frac{160}{147}$$

Assuming that the audio signal is bandlimited to 20kHz, we design our master lowpass filter with transition bands

$$\left[\left(2k + \frac{20}{22.05}\right)\frac{\pi}{160}, \left(2\left(k + 1\right) - \frac{20}{22.05}\right)\frac{\pi}{160}\right) \quad , \quad k \in \mathbb{Z}$$

Keeping the passband and stopband ripple levels below -96dB requires a filter with a length $N \simeq 10000$, implying that the standard polyphase resampler will require about $NM = 1.5$million multiplication per output, or 70billion multiplications per second! If the equivalent polyphase implementation is used instead, we require only $\frac{N}{L} \simeq 62$ multiplies per output, or 3million multiplications per second.

1.1.16 Polyphase Resampling with Arbitrary (Non-Rational or Time-Varying) Rate[16]

Though we have derived a computationally efficient polyphase resampler for rational factors $Q = \frac{L}{M}$, the structure will not be practical to implement for large L, such as might occur when the desired resampling factor Q is not well approximated by a ratio of two small integers. Furthermore, we may encounter applications in which Q is chosen on-the-fly, so that the number L of polyphase branches cannot be chosen a priori. Fortunately, a slight modification of our exisiting structure will allow us to handle both of these cases.

---

[15]This content is available online at <http://cnx.org/content/m11010/2.2/>.
[16]This content is available online at <http://cnx.org/content/m10456/2.9/>.

Say that our goal is to produce the $\frac{Q}{T}$-rate samples $x_c\left(m\frac{Q}{T}\right)$ given the $\frac{1}{T}$-rate samples $x_c\left(nT\right)$, where we assume that $x_c\left(t\right)$ is bandlimited to $\frac{1}{2T}$ and Q can be any positive real number. Consider, for a moment, the outputs of polyphase filters in an ideal zero-delay L-branch polyphase interpolation bank (as in Figure 1.30).
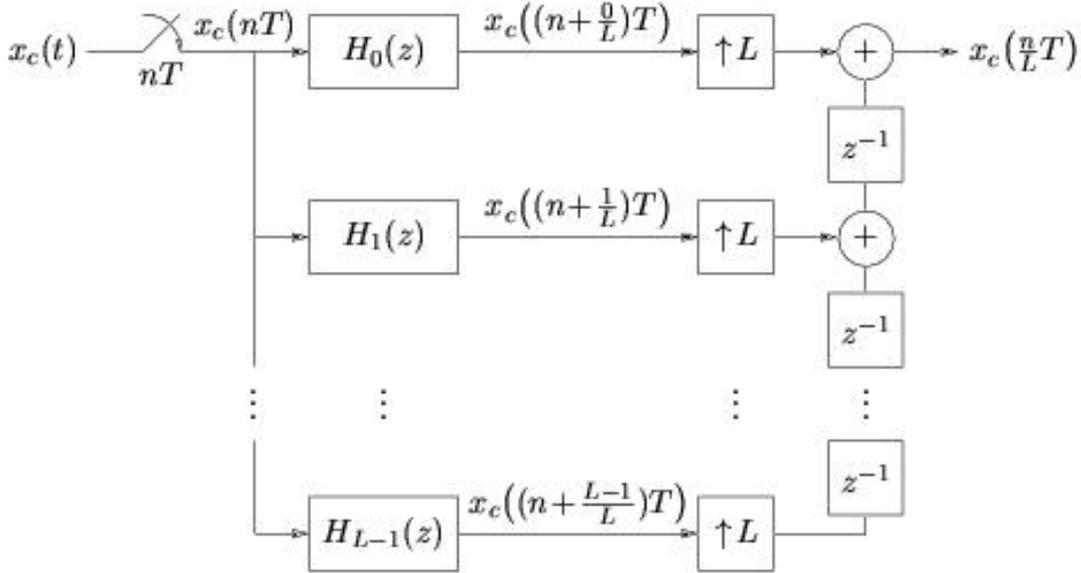
Figure 1.30

We know that, at time index n, the pth and $(p+1)^{th}$ filter outputs equal

$$x_c\left(\left(n+\frac{p}{L}\right)T\right) \quad x_c\left(\left(n+\frac{p+1}{L}\right)T\right)$$

respectively. Because the highest frequency in $x_c\left(t\right)$ is limited to $\frac{1}{2T}$, the waveform cannot not change abruptly, and therefore cannot change significantly over a very small time interval. In fact, when L is large, the waveform is nearly linear in the time interval between $t=\left(n+\frac{p}{L}\right)T$ and $t=\left(n+\frac{p+1}{L}\right)T$, so that, for any $\alpha\in[0,1)$,

$$x_c\left(\left(n+\frac{p+\alpha}{L}\right)T\right)=x_c\left(1\left(n+\frac{p}{L}\right)T+\alpha\left(n+\frac{p+1}{L}\right)T\right)$$

$$x_c\left(\left(n+\frac{p+\alpha}{L}\right)T\right)\simeq 1x_c\left(\left(n+\frac{p}{L}\right)T\right)+\alpha x_c\left(\left(n+\frac{p+1}{L}\right)T\right)$$

This suggests that we can closely approximate $x_c\left(t\right)$ at any $t\in\mathbb{R}$ by linearly interpolating adjacent-branch outputs of a polyphase filterbank with a large enough L. The details are worked out below.

Assume an ideal L-branch polyphase filterbank with d-delay master filter and T-sampled input, giving access to $x_c\left(\left(n+\frac{p-d}{L}\right)T\right)$ for $n\in\mathbb{Z}$ and $p\in\{0,\ldots,L-1\}$. By linearly interpolating branch outputs p and $p+1$ at time n, we are able to closely approximate $x_c\left(\left(n+\frac{p-d+\alpha}{L}\right)T\right)$ for any $\alpha\in[0,1)$. We would like to approximate $y\left[m\right]=x_c\left(m\frac{T}{Q}-d\frac{T}{L}\right)$ in this manner - note the inclusion of the master filter delay. So,

for a particular m, Q, d, and L, we would like to find $n \in \mathbb{Z}$, $p \in \{0, \ldots, L-1\}$, and $\alpha \in [0, 1)$ such that

$$\left(n + \frac{p - d + \alpha}{L}\right) T = m\frac{T}{Q} - d\frac{T}{L} \tag{1.22}$$

$$
\begin{aligned}
nL + p + \alpha &= \tfrac{mL}{Q} \\
&= \tfrac{m}{Q}L \\
&= \left(\lfloor \tfrac{m}{Q} \rfloor + \tfrac{m}{Q}mod1\right) L \\
&= \lfloor \tfrac{m}{Q} \rfloor L + \lfloor \tfrac{m}{Q}mod1L \rfloor + \tfrac{m}{Q}mod1Lmod1 \\
&= \lfloor \tfrac{m}{Q} \rfloor L + \lfloor \tfrac{m}{Q}mod1L \rfloor + \tfrac{mL}{Q}mod1
\end{aligned}
\tag{1.23}
$$

where $\lfloor \frac{m}{Q} \rfloor L \in \mathbb{Z}$, $\lfloor \frac{m}{Q}mod1L \rfloor \in \{0, \ldots, L-1\}$, $\frac{mL}{Q}mod1 \in [0, 1)$.

Thus, we have found suitable n, p, and α. Making clear the dependence on output time index m, we write

$$n_m = \lfloor \frac{m}{Q} \rfloor$$

$$p_m = \frac{m}{Q}mod1L$$

$$\alpha_m = \frac{mL}{Q}mod1$$

and generate output $y[m] \simeq x_c\left(m\frac{T}{Q} - d\frac{T}{L}\right)$ via

$$y[m] = 1 \sum_{kk} h_{p_m}[k]\, x[n_m - k] + \alpha_m \sum_{kk} h_{p_m+1}[k]\, x[n_m - k]$$

The arbitrary rate polyphase resampling structure is summarized in Figure 1.31.
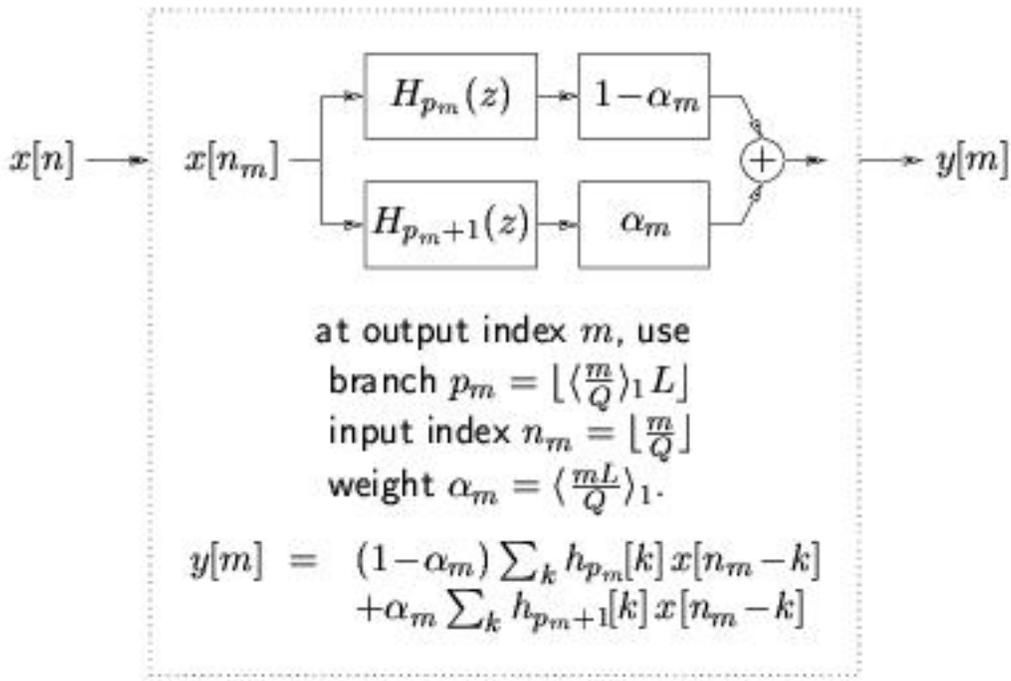
Figure 1.31

Note that our structure refers to polyphase filters $H_{p_m}(z)$ and $H_{p_m+1}(z)$ for $p_m \in \{0, \ldots, L-1\}$. This specifies the standard polyphase bank $\{H_0(z), \ldots, H_{L-1}(z)\}$ plus the additional filter $H_L(z)$. Ideally the pth filter has group delay $\frac{d-p}{L}$, so that $H_L(z)$ should advance the input one full sample relative to $H_0(z)$, i.e., $H_L(z) = zH_0(z)$. There are a number of ways to design/implement the additional filter.

1. Design a master filter of length $LN_p + 1$ (where N_p is the polyphase filter length), and then construct

$$h_p[k] = h[kL+p] \quad, \quad p \in \{0, \ldots, L\}$$

   Note that $h_L[k] = h_0[k+1]$ for $0 \le k \le N_p - 2$.
2. Set $H_L(z) = H_0(z)$ and advance the input stream to the last filter by one sample (relative to the other filters).

In certain applications the rate of resampling needs to be adjusted on-the-fly. The arbitrary rate resampler easily facilitates this requirement by replacing Q with Q_m in the definitions for n_m, p_m, and α_m.

Finally, it should be mentioned that a more sophisticated interpolation could be used, e.g., Lagrange interpolation involving more than two branch outputs. By making the interpolation more accurate, fewer polyphase filters would be necessary for the same overall performance, reducing the storage requirements for polyphase filter taps. On the other hand, combining the outputs of more branches requires more computations per output point. Essentially, the different schemes tradeoff between storage and computation.

1.1.17 Multi-stage Interpolation and Decimation[17]

1.1.17.1 Multistage Decimation

In the single-stage interpolation structure illustrated in Figure 1.32, the required impulse response of $H(z)$ can be very long for large L.

Figure 1.32

Consider, for example, the case where $L = 30$ and the input signal has a bandwidth of $\omega_0 = 0.9\pi$ radians. If we desire passband ripple $\delta_p = 0.002$ and stopband ripple $\delta_s = 0.001$, then Kaiser's formula approximates the required FIR filter length to be

$$N_h \simeq \frac{-10\log(\delta_P \delta_S) - 13}{2.3\Delta(\omega)} \simeq 900$$

choosing $\Delta(\omega) = \frac{2\pi - 2\omega_0}{L}$ as the width of the first transition band (*i.e.*, ignoring the other transition bands for this rough approximation). Thus, a polyphase implementation of this interpolation task would cost about 900 multiplies per input sample.

Consider now the two-stage implementation illustrated in Figure 1.33.
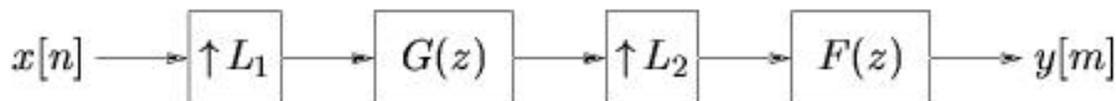
Figure 1.33

We claim that, when L is large and ω_0 is near Nyquist, the two-stage scheme can accomplish the same interpolation task with less computation.

Let's revisit the interpolation objective of our previous example. Assume that $L_1 = 2$ and $L_2 = 15$ so that $L_1 L_2 = L = 30$. We then desire a pair $\{F(z), G(z)\}$ which results in the same performance as $H(z)$. As a means of choosing these filters, we employ a Noble identity to reverse the order of filtering and upsampling (see Figure 1.34).

---

[17]This content is available online at <http://cnx.org/content/m10420/2.13/>.
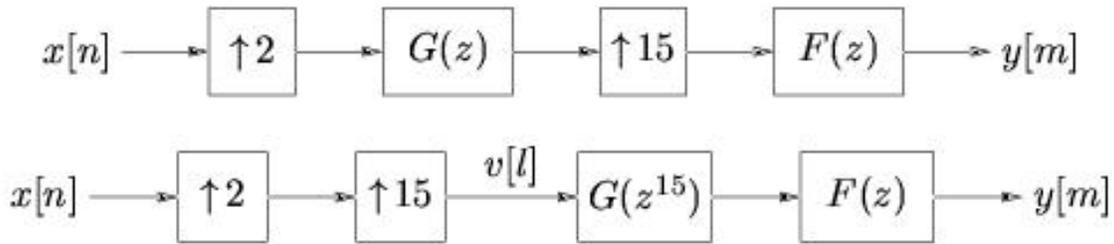
Figure 1.34

It is now clear that the composite filter $G\left(z^{15}\right)F\left(z\right)$ should be designed to meet the same specifications as $H\left(z\right)$. Thus we adopt the following strategy:

1. Design $G\left(z^{15}\right)$ to remove unwanted images, keeping in mind that the DTFT $G\left(e^{j15\omega}\right)$ is $\frac{2\pi}{15}$-periodic in ω.
2. Design $F\left(z\right)$ to remove the remaining images.

The first and second plots in Figure 1.35 illustrate example DTFTs for the desired signal $x\left[n\right]$ and its L-upsampled version $v\left[l\right]$, respectively. Our objective for interpolation, is to remove all but the shaded spectral image shown in the second plot. The third plot (Figure 1.35) shows that, due to symmetry requirements $G\left(z^{15}\right)$ will be able to remove only one image in the frequency range $\left[0, \frac{2\pi}{15}\right)$. Due to its periodicity, however, $G\left(z^{15}\right)$ also removes some of the other undesired images, namely those centered at $\frac{\pi}{15} + m\frac{2\pi}{15}$ for $m \in \mathbb{Z}$. $F\left(z\right)$ is then used to remove the remaining undesired images, namely those centered at $m\frac{2\pi}{15}$ for $m \in \mathbb{Z}$ such that m is not a multiple of 15. Since it is possible that the passband ripples of $F\left(z\right)$ and $G\left(z^{15}\right)$ could add constructively, we specify $\delta_p = 0.001$ for both $F\left(z\right)$ and $G\left(z\right)$, half the passband ripple specified for $H\left(z\right)$. Assuming that the transition bands in $F\left(z\right)$ have gain no greater than one, the stopband ripples will not be amplified and we can set $\delta_s = 0.001$ for both $F\left(z\right)$ and $G\left(z\right)$, the same specification as for $H\left(z\right)$.
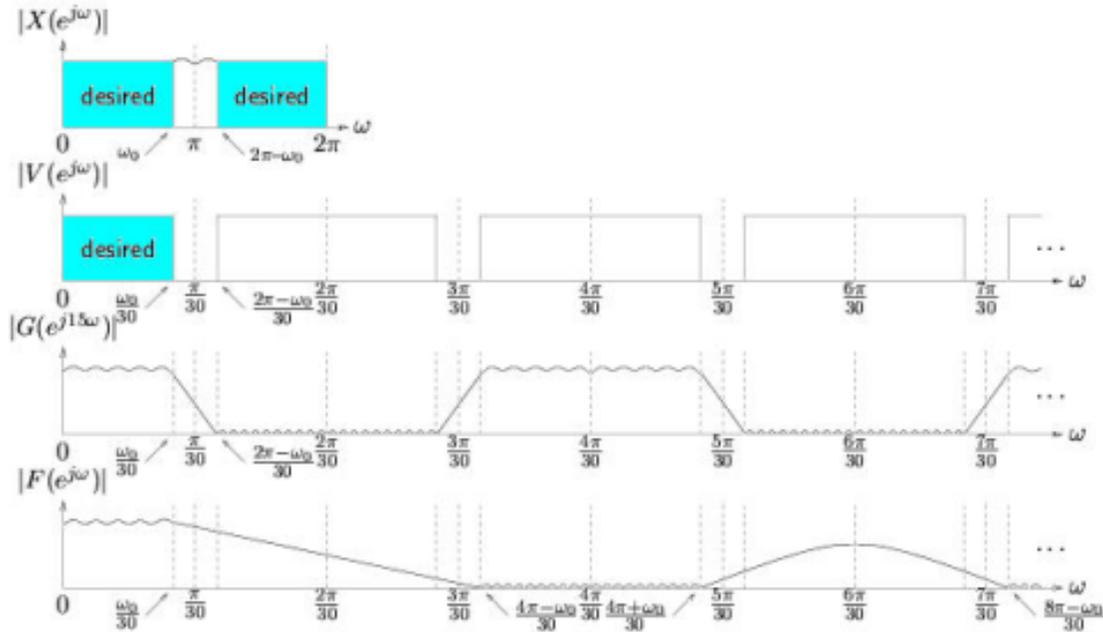
Figure 1.35

The computational savings of the multi-stage structure result from the fact that the transition bands in both $F(z)$ and $G(z)$ are much wider than the transition bands in $H(z)$. From the block diagram (Figure 1.35), we can infer that the transition band in $G(z)$ is centered at $\omega = \frac{\pi}{2}$ with width $\pi - \omega_0 = 0.1\pi$rad. Likewise, the transition bands in $F(z)$ have width $\frac{4\pi - 2\omega_0}{30} = \frac{2.2}{30}\pi$rad. Plugging these specifications into the Kaiser length approximation, we obtain

$$N_g \simeq 64$$

and

$$N_f \simeq 88$$

Already we see that it will be much easier, computationally, to design two filters of lengths 64 and 88 than it would be to design one 900-tap filter.

As we now show, the computational savings also carry over to the operation of the two-stage structure. As a point of reference, recall that a polyphase implementation of the one-stage interpolator would require $N_h \simeq 900$ multiplications per input point. Using a cascade of two single-stage polyphase interpolators to implement the two-stage scheme, we find that the first interpolator would require $N_g \simeq 64$ per input point $x[n]$, while the second would require $N_f \simeq 88$ multiplies per output of $G(z)$. Since $G(z)$ outputs two points per input $x[n]$, the two-stage structure would require a total of

$$\simeq 64 + 2 \times 88 = 240$$

multiplies per input. Clearly this is a significant savings over the 900 multiplies required by the one-stage structure. Note that it was advantageous to choose the first upsampling ratio (L_1) as small as possible, so that the second stage of interpolation operates at a low rate.

Multi-stage decimation can be formulated in a very similar way. Using the same example quantities as we did for the case of multi-stage interpolation, we have the block diagrams and filter-design methodology illustrated in Figure 1.36 and Figure 1.37.
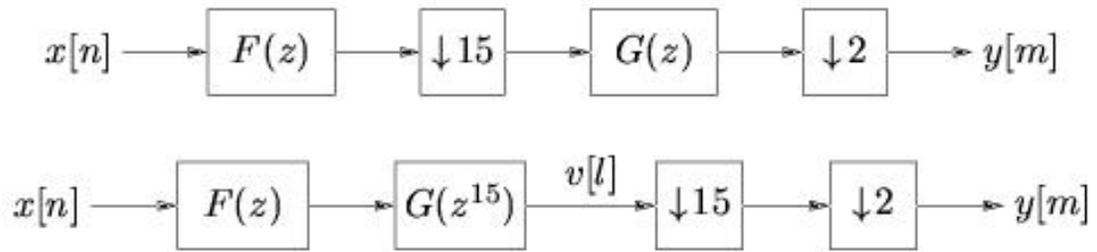
Figure 1.36
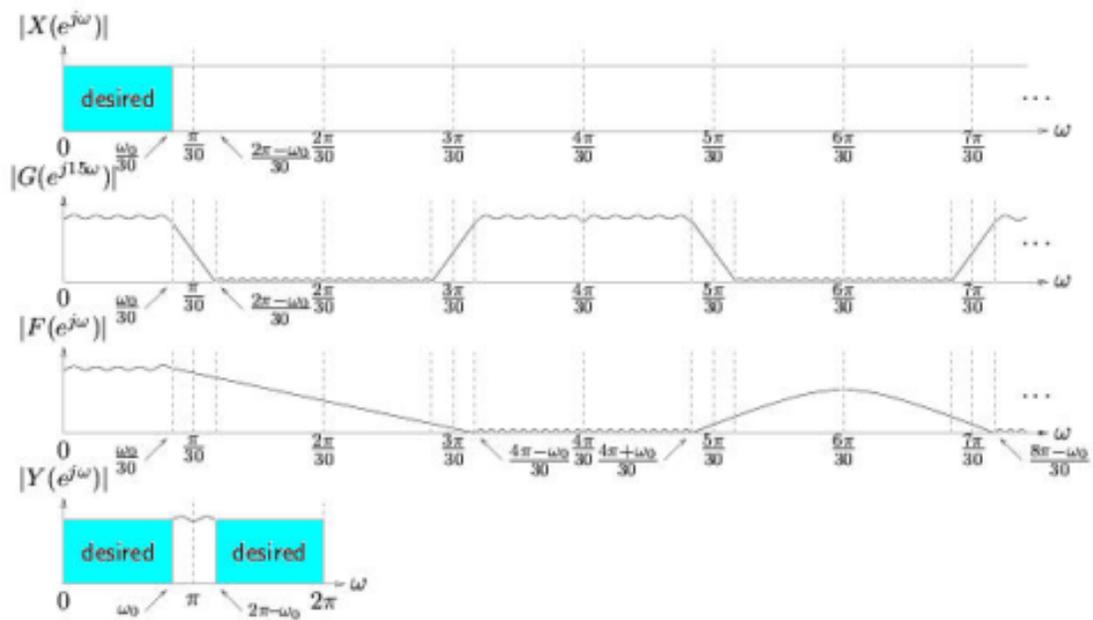
Figure 1.37

Chapter 2

Filterbanks

2.1 Why Filterbanks?

2.1.1 Sub-Band Processing[1]

2.1.1.1 Why Filterbanks?

2.1.1.1.1 Sub-band Processing

There exist many applications in modern signal processing where it is advantageous to separate a signal into different frequency ranges called **sub-bands**. The spectrum might be partitioned in the uniform manner illustrated in Figure 2.1, where the sub-band width $\Delta_k = \frac{2\pi}{M}$ is identical for each sub-band and the band centers are uniformly spaced at intervals of $\frac{2\pi}{M}$.
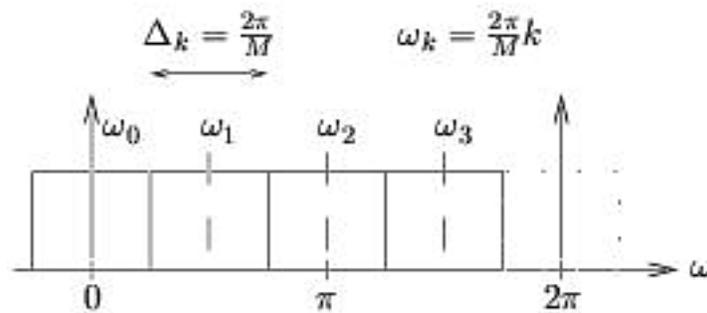
Figure 2.1

Alternatively, the sub-bands might have a logarithmic spacing like that shown in Figure 2.2.

---

[1] This content is available online at $<$http://cnx.org/content/m10423/2.14/$>$.

$$\Delta_k = \pi 2^{-(k+1)}$$
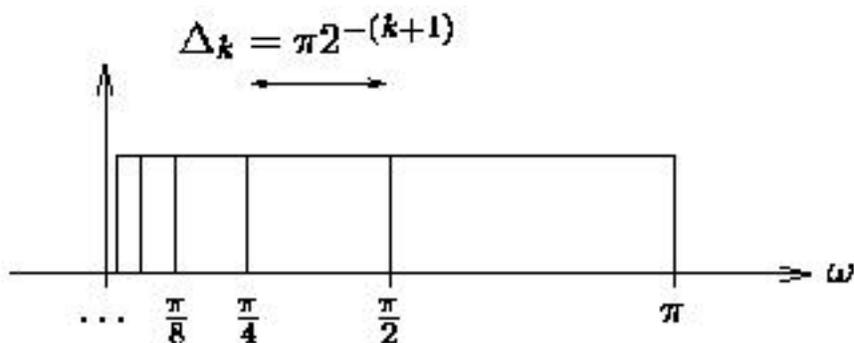
Figure 2.2

For most of our discussion, we will focus on uniformly spaced sub-bands.

The separation into sub-band components is intended to make further processing more convenient. Some of the most popular applications for sub-band decomposition are audio and video source coding (with the goal of efficient storage and/or transmission).

Figure 2.3 illustrates the use of sub-band processing in MPEG audio coding. There a psychoacoustic model is used to decide how much quantization error can be tolerated in each sub-band while remaining below the hearing threshold of a human listener. In the sub-bands that can tolerate more error, less bits are used for coding. The quantized sub-band signals can then be decoded and recombined to reconstruct (an approximate version of) the input signal. Such processing allows, on average, a 12-to-1 reduction in bit rate while still maintaining "CD quality" audio. The psychoacoustic model takes into account the **spectral masking** phenomenon of the human ear, which says that high energy in one spectral region will limit the ear's ability to hear details in nearby spectral regions. Therefore, when the energy in one sub-band is high, nearby sub-bands can be coded with less bits without degrading the perceived quality of the audio signal. The MPEG standard specifies 32-channels of sub-band filtering. Some psychoacoustic models also take into account "temporal masking" properties of the human ear, which say that a loud burst of sound will temporarily overload the ear for short time durations, making it possible to hide quantization noise in the time interval after a loud sound burst.
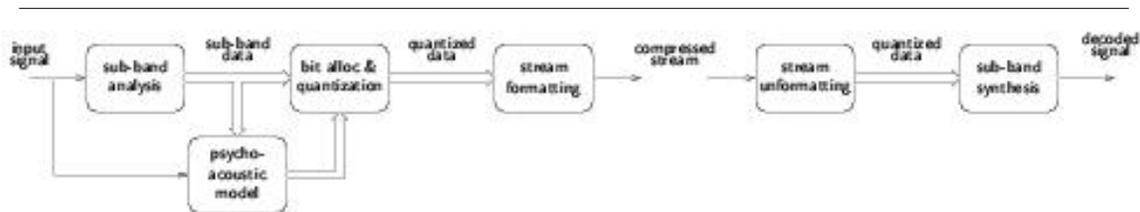
Figure 2.3

In typical applications, non-trivial signal processing takes place between the bank of analysis filters and the bank of synthesis filters, as shown in Figure 2.4. We will focus, however, on filterbank design rather than on the processing that occurs between the filterbanks.
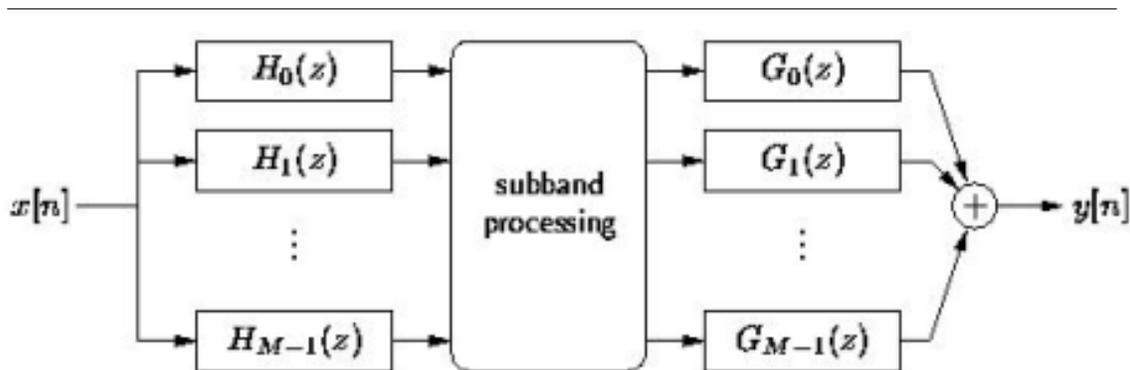
Figure 2.4

Our goals in filter design are:

1. Good sub-band frequency separation (*i.e.*, good "frequency selectivity").
2. Good reconstruction (*i.e.*, $y[n] \simeq x[n-d]$ for some integer delay d) when the sub-band processing is lossless.

The first goal is driven by the assumption that the sub-band processing works **best** when it is given access to cleanly separated sub-band signals, while the second goal is motivated by the idea that the sub-band filtering should not limit the reconstruction performance when the sub-band processing (*e.g.*, the coding/decoding) is lossless or nearly lossless.

2.1.2 Uniform Filterbanks[2]

2.1.2.1 Uniform Filterbanks

With M uniformly spaced sub-bands, the sub-band width is $\frac{2\pi}{M}$ radians, implying that the sub-band signal can be downsampled by a factor M (but not more than M) without loss of information. This is referred to as a "critically sampled" filterbank. This maximal level of downsampling is advantageous when storing or further processing the sub-band signals. With critical sampling, the total number of downsampled sub-band output samples equals the total number of input samples. Assuming lossless sub-band processing, the critically-sampled synthesis/analysis procedure is illustrated in Figure 2.5:

---

[2]This content is available online at $<$http://cnx.org/content/m10990/2.3/$>$.
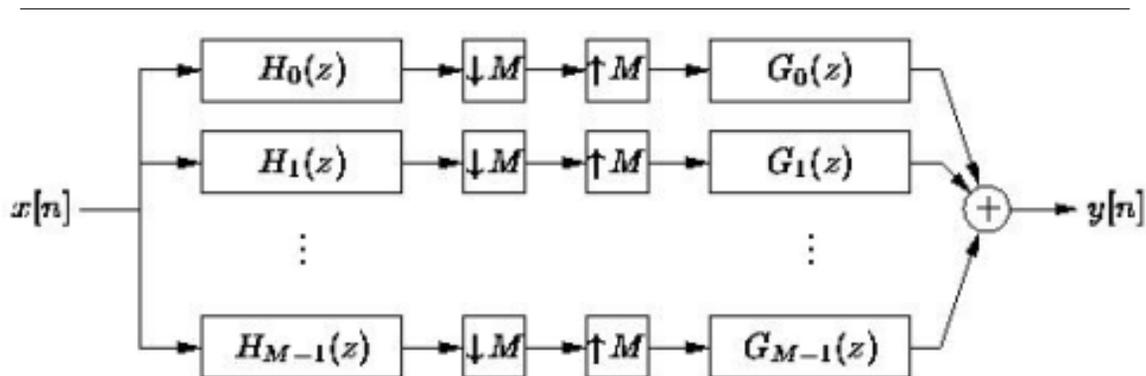
Figure 2.5

Recall that one of our goals in filter design is to ensure that $y[n] \simeq x[n-d]$ for some integer delay d. From the block diagram above (Figure 2.5), one can see that imperfect analysis filtering will contribute aliasing errors to the sub-band signals. This aliasing distortion will degrade $y[n]$ if it is not cancelled by the synthesis filterbank. Though ideal brick-wall filters $H_k(z)$ and $G_k(z)$ could easily provide perfect reconstruction (i.e., $y[n] = x[n-d]$), they would be unimplementable due to their doubly-infinite impulse responses. Thus, we are interested in the design of causal FIR filters that give near-perfect reconstruction or, if possible, perfect reconstruction.

There are two principle approaches to the design of filterbanks:

1. Classical: Approximate ideal brick wall filters to ensure good sub-band isolation (i.e., frequency selectivity) and accept (a hopefully small amount of) aliasing and thus reconstruction error.
2. Modern: Constrain the filters to give perfect (or near-perfect) reconstruction and hope for good sub-band isolation.

2.2 Classical Filterbanks

2.2.1 Uniform Modulated Filterbank[3]

A modulated filterbank (Section 2.1.1) is composed of analysis branches which:

1. Modulate the input to center the desired sub-band at DC
2. Lowpass filter the modulated signal to isolate the desired sub-band
3. Downsample the lowpass signal

The synthesis branches interpolate the sub-band signals by upsampling and lowpass filtering, then modulate each sub-band back to its original spectral location (see Figure 2.6).

---

[3]This content is available online at $<$http://cnx.org/content/m10929/2.6/$>$.
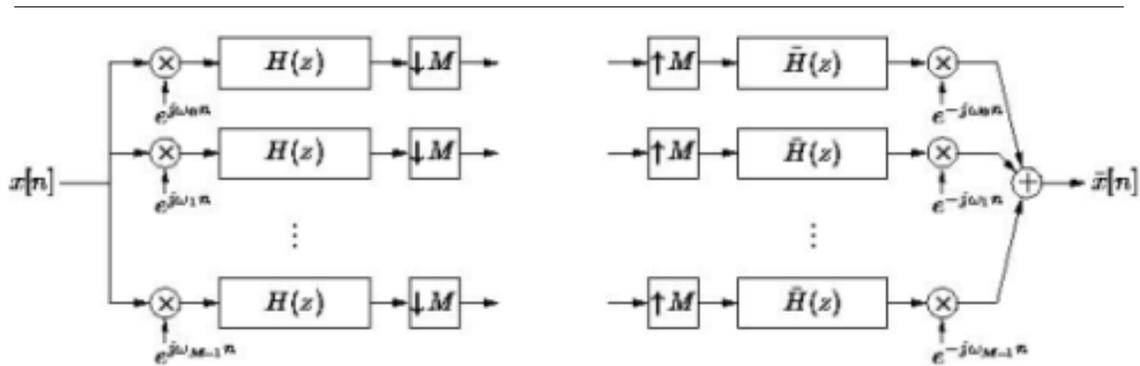
Figure 2.6

In M-branch critically-sampled uniformly-modulated filterbanks, the kth branch has a modulation frequency of $\pm (\omega_k) = \frac{2\pi}{M} k$ radians, a lowpass cutoff frequency of $\frac{\pi}{M}$ radians, and a downsampling/upsampling factor of M.

The most common way to implement this type of uniform modulated filterbank is through the use of polyphase filterbanks and DFTs (Section 2.2.2).

2.2.2 Uniformally Modulated (DFT) Filterbank[4]

The uniform modulated filterbank (Section 2.2.1) can be implemented using polyphase filterbanks and DFTs, resulting in huge computational savings. Figure 2.7 below illustrates the equivalent polyphase/DFT structures for analysis and synthesis. The impulse responses of the polyphase filters $P_l(z)$ and $\overline{P}_l(z)$ can be defined in the time domain as $\overline{p}_l[m] = \overline{h}[mM + l]$ and $p_l(m) = h[mM + l]$, where $h[n]$ and $\overline{h}[n]$ denote the impulse responses of the analysis and synthesis lowpass filters, respectively.

---

[4]This content is available online at <http://cnx.org/content/m10424/2.13/>.
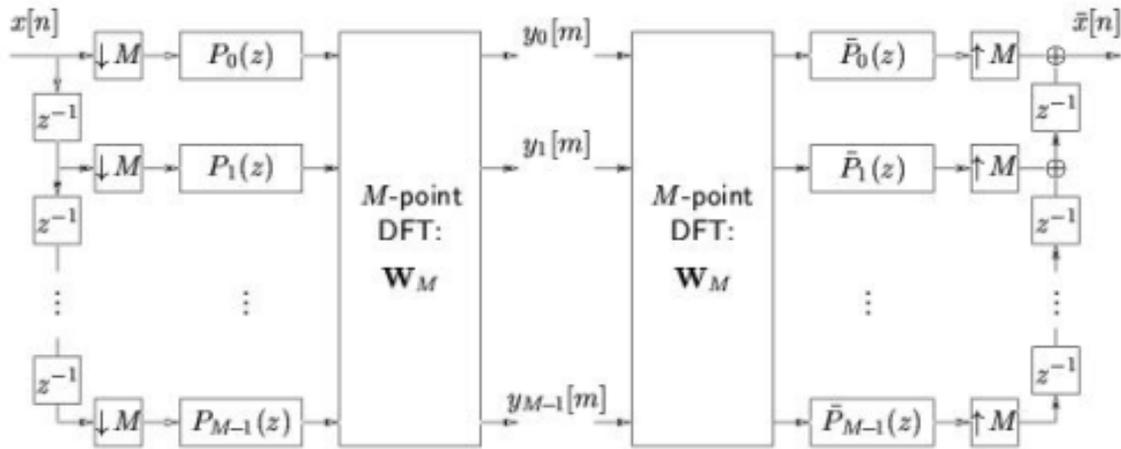
Figure 2.7

Recall that the standard implementation performs modulation, filtering, and downsampling, in that order. The polyphase/DFT implementation reverses the order of these operations; it performs downsampling, then filtering, then modulation (if we interpret the DFT as a two-dimensional bank of "modulators"). We derive the polyphase/DFT implementation below, starting with the standard implementation and exchanging the order of modulation, filtering, and downsampling.

2.2.2.1 Polyphase/DFT Implementation Derivation

We start by analyzing the kth filterbank branch, analyzed in Figure 2.8:
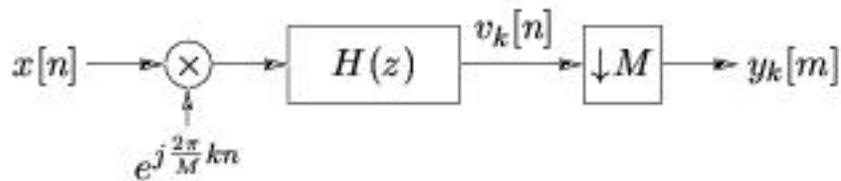
Figure 2.8: kth filterbank branch

The first step is to reverse the modulation and filtering operations. To do this, we define a "modulated filter" $H_k(z)$:

$$
\begin{aligned}
v_k[n] &= \sum_i h[i]\, x[n-i]\, e^{j\frac{2\pi}{M}k(n-i)} \\
&= \left(\sum_i h[i]\, e^{(-j)\frac{2\pi}{N}ki} x[n-i]\right) e^{j\frac{2\pi}{M}kn} \\
&= \left(\sum_i h_k[i]\, x[n-i]\right) e^{j\frac{2\pi}{M}kn}
\end{aligned}
\tag{2.1}
$$

The equation above indicated that $x[n]$ is convolved with the modulated filter and that the filter output is modulated. This is illustrated in Figure 2.9:
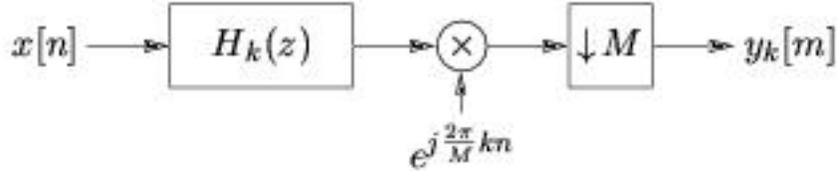
Figure 2.9

Notice that the only modulator outputs not discarded by the downsampler are those with time index $n = mM$ for $m \in \mathbb{Z}$. For these outputs, the modulator has the value $e^{j\frac{2\pi}{M}kmM} = 1$, and thus it can be ignored. The resulting system is portrayed by:

Figure 2.10

Next we would like to reverse the order of filtering and downsampling. To apply the Noble identity, we must decompose $H_k(z)$ into a bank of upsampled polyphase filters. The technique used to derive polyphase decimation can be employed here:

$$
\begin{aligned}
H_k(z) &= \sum_{n=-\infty}^{\infty} h_k[n] z^{-n} \\
&= \sum_{l=0}^{M-1} \sum_{m=-\infty}^{\infty} h_k[mM+l] z^{-(mM+l)}
\end{aligned}
\tag{2.2}
$$

Noting the fact that the lth polyphase filter has impulse response:

$$
h_k[mM+l] = h[mM+l] e^{(-j)\frac{2\pi}{M}(k(mM+l))} = h[mM+l] e^{(-j)\frac{2\pi}{M}kl} = p_l[m] e^{(-j)\frac{2\pi}{M}kl}
$$

where $p_l[m]$ is the lth polyphase filter defined by the original (unmodulated) lowpass filter $H(z)$, we obtain

$$
\begin{aligned}
H_k(z) &= \sum_{l=0}^{M-1} \sum_{m=-\infty}^{\infty} p_l[m] e^{(-j)\frac{2\pi}{M}kl} z^{-(mM-l)} \\
&= \sum_{l=0}^{M-1} e^{(-j)\frac{2\pi}{M}kl} z^{-l} \sum_{m=-\infty}^{\infty} p_l[m] \left(z^M\right)^{-m} \\
&= \sum_{l=0}^{M-1} e^{(-j)\frac{2\pi}{M}kl} z^{-l} P_l\left(z^M\right)
\end{aligned}
\tag{2.3}
$$

The kth filterbank branch (now containing M polyphase branches) is in Figure 2.11:
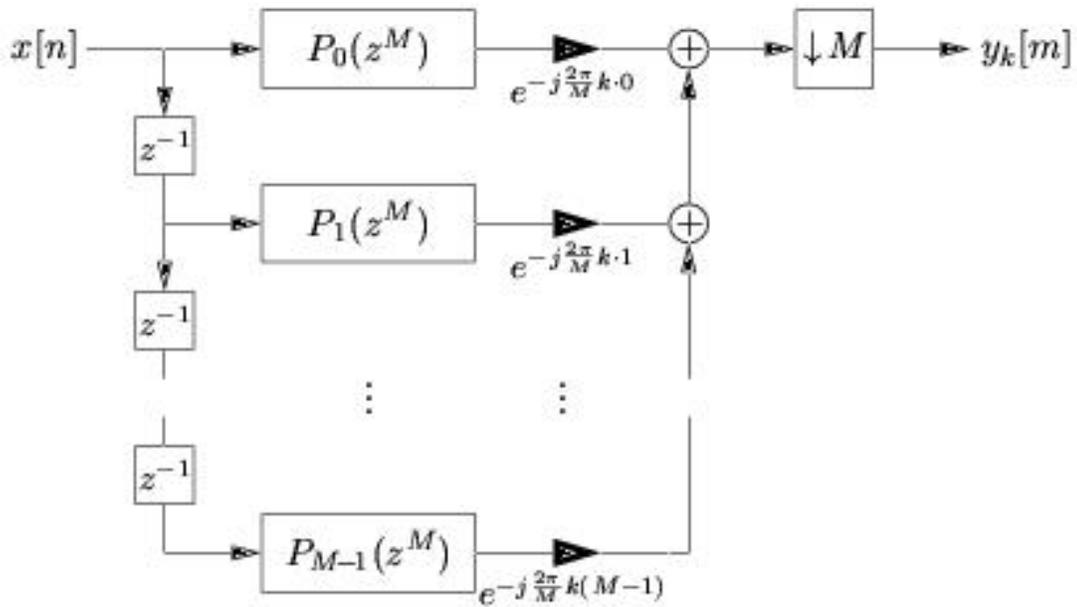
Figure 2.11: kth filterbank branch containing M polyphase branches.

Because it is a linear operator, the downsampler can be moved through the adders and the (time-invariant) scalings $e^{(-j)\frac{2\pi}{M}kl}$. Finally, the Noble identity is employed to exchange the filtering and downsampling. The kth filterbank branch becomes:
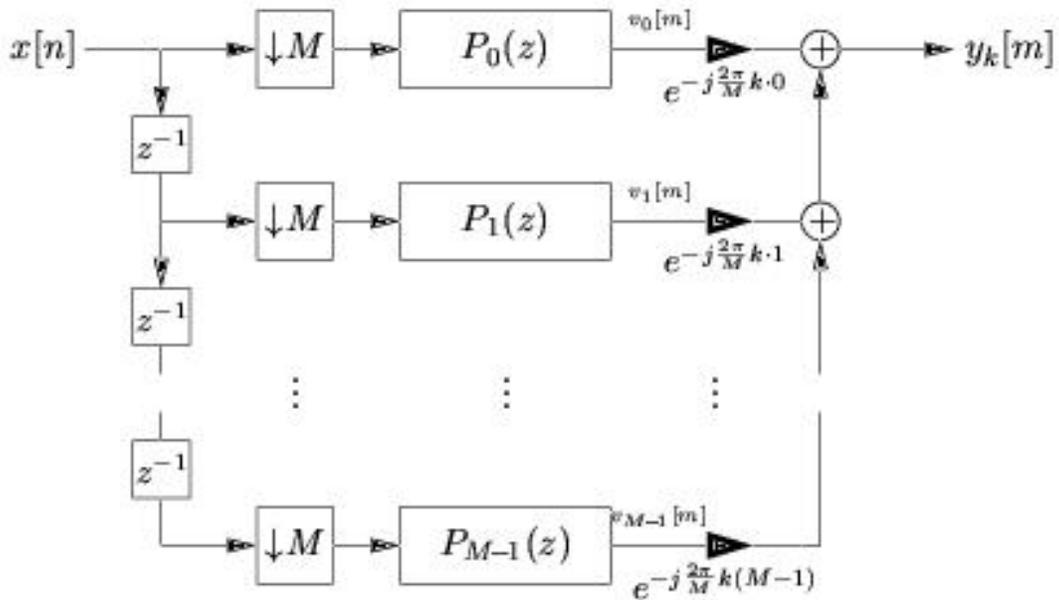
Figure 2.12

Observe that the polyphase outputs $\{\, v_l\,[m]\,,v_l\,[m]\mid l=\{0,\dots,M-1\}\,\}$ are identical for each filterbank branch, while the scalings $\left\{\, e^{(-j)\frac{2\pi}{M}kl}\mid l=\{0,\dots,M-1\}\,\right\}$ once. Using these outputs we can compute the branch outputs via

$$y_k\,[m]=\sum_{l=0}^{M-1}v_l\,[m]\,e^{(-j)\frac{2\pi}{M}kl} \tag{2.4}$$

From the previous equation it is clear that $y_k\,[m]$ corresponds to the kth DFT output given the M-point input sequence $\{\, v_l\,[m]\mid l=\{0,\dots,M-1\}\,\}$. Thus the M filterbank branches can be computed in parallel by taking an M-point DFT of the M polyphase outputs (see Figure 2.13).
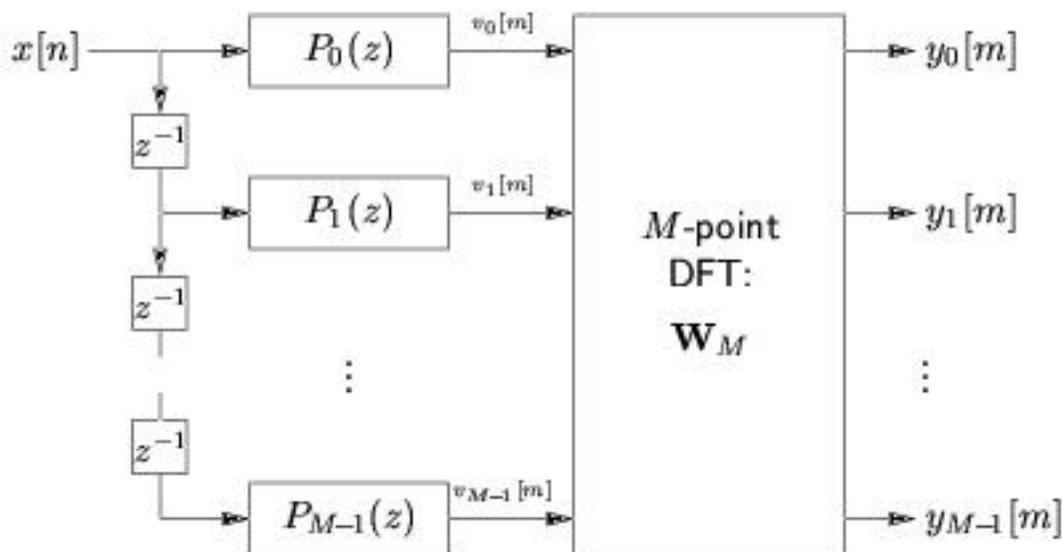
Figure 2.13

The polyphase/DFT synthesis bank can be derived in a similar manner.

2.2.3 Computational Savings of Polyphase/DFT Filterbanks[5]

This module will briefly take a look a the computational savings of the polyphase/DFT modulated filterbank implementation. We will start by looking at our standard analysis bank and then move on to compare this with our other implementation approaches.

Assume that the lowpass filter in the standard analysis bank, $H(z)$, has impulse response length N. To calculate the sub-band output vector $\{ y_k[m], y_k[m] \mid k = \{0, \dots, M-1\} \}$ using the standard structure, we have

$$M \frac{\text{decimator outputs}}{\text{vector}} \times M \frac{\text{filter outputs}}{\text{decimator outputs}} \times (N+1) \frac{\text{multiplications}}{\text{filter output}} = M^2 (N+1) \frac{\text{multiplications}}{\text{vector}}$$
(2.5)

where we have included one multiply for the modulator. The calculations above pertain to standard (*i.e.*, not polyphase) decimation. If we implement the lowpass/downsampler in each filterbank branch with a polyphase decimator,

$$M \frac{\text{branch outputs}}{\text{vector}} \times (N+1) \frac{\text{multiplications}}{\text{branch output}} = M (N+1) \frac{\text{multiplications}}{\text{vector}}$$
(2.6)

To calculate the same output vector for the polyphase/DFT structure, we have approximately

$$1 \frac{\text{DFT}}{\text{vector}} \times \left(\frac{M}{2} \log_2 M \frac{\text{multiplications}}{\text{DFT}} \times M \frac{\text{polyphase outputs}}{\text{DFT}} \times \frac{N}{M} \frac{\text{multiplications}}{\text{polyphase output}} \right)$$
(2.7)

---

[5]This content is available online at <http://cnx.org/content/m10930/2.3/>.

$$= \left(N + \frac{M}{2}\log_2 M \right) \frac{\text{multiplications}}{\text{vector}}$$

The table below gives some typical numbers. Recall that the filter length N will be linearly proportional to the decimation factor M, so that the ratio $\frac{N}{M}$ determines the passband and stopband performance.

	$M = 32$, $\frac{N}{M} = 10$	$M = 128$, $\frac{N}{M} = 10$
standard	328,704	20,987,904
standard with polyphase	10,272	163,968
polyphase/DFT	400	1,728

Table 2.1

2.2.4 Drawbacks of Classical Filterbank Designs[6]

To begin with, the reference to "classical" filterbank designs, generally refers to the filterbank types that you should have seen up to now. These include the following types, that can be reviewed if necessary:

- Uniform Modulated Filterbanks (Section 2.2.1)
- Polyphase/DFT Implementation of Uniform Modulated Filterbanks (Section 2.2.2)

2.2.4.1 Drawbacks to Classical Implementation

The classical filterbanks that we have considered so far (those listed above) give perfect reconstruction performance only when the analysis and synthesis filters are ideal. With non-ideal (*i.e.*, implementable) filters, aliasing will result from the downsampling/upsampling operation and corrupt the output signal. Since aliasing distortion is inherently non-linear, it may be very undesirable in certain applications. Thus, long analysis/synthesis filters might be required to force aliasing distortion down to tolerable levels. The cost of long filters is somewhat offset by the efficient polyphase implementation, though.

That said, clever fliter designs have been proposed which prevent aliasing in **neighboring** sub-bands. These designs include the following references: *Rothweiler*[8], *Crochiere and Rabiner*[2], and *Vaidyanathan*[9]. As neighboring-subband aliasing typically constitutes the bulk of aliasing distortion, these designs give significant performance gains. In fact, such filter designs are used in MPEG high-performance audio compression standards.

2.3 Modern Filterbanks

2.3.1 Aliasing-Cancellation Conditions of Filterbanks[7]

2.3.1.1 Introduction

It is possible to design combinations of analysis and synthesis filters such that the aliasing from downsampling/upsampling is completely cancelled. Below we derive aliasing-cancellation conditions for two-channel filterbanks. Though the results can be extended to M-channel filterbanks in a rather straightforward manner, the two-channel case offers a more lucid explanation of the principle ideas (see Figure 2.14).

---

[6]This content is available online at <http://cnx.org/content/m10931/2.4/>.

[7]This content is available online at <http://cnx.org/content/m10425/2.11/>.
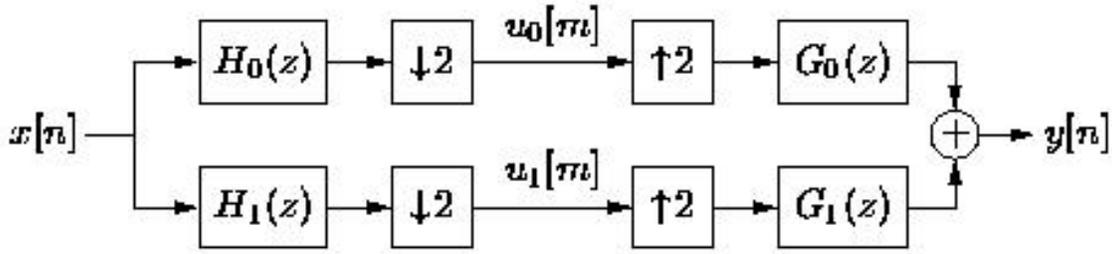
Figure 2.14

2.3.1.2 Aliasing Cancellation Conditions

The aliasing cancellation conditions follow directly from the input/output equations derived below. Let $i \in \{0, 1\}$ denote the filterbank branch index. Then

$$U_i(z) = \frac{1}{2} \sum_{p=0}^{1} H_i\left(z^{\frac{1}{2}} e^{(-j)\pi p}\right) X\left(z^{\frac{1}{2}} e^{(-j)\pi p}\right) \tag{2.8}$$

$$
\begin{array}{rcl}
Y(z) & = & \sum_{i=0}^{1} G_i(z) U_i\left(z^2\right) \\
& = & \sum_{i=0}^{1} G_i(z) \frac{1}{2} \sum_{p=0}^{1} H_i\left(ze^{(-j)\pi p}\right) X\left(ze^{(-j)\pi p}\right) \\
& = & \frac{1}{2} \sum_{i=0}^{1} G_i(z) \left(H_i(z) X(z) + H_i(-z) X(-z)\right) \\
& = & \frac{1}{2} \left(\begin{array}{cc} X(z) & X(-z) \end{array} \right) \left(\begin{array}{cc} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{array} \right) \left(\begin{array}{c} G_0(z) \\ G_1(z) \end{array} \right)
\end{array}
\tag{2.9}
$$

where $H(z) = \left(\begin{array}{cc} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{array} \right)$. $H(z)$ is often called the **aliasing component matrix**. For aliasing cancellation, we need to ensure that $X(-z)$ does not contribute to the output $Y(z)$. This requires that

$$\left(\begin{array}{cc} H_0(-z) & H_1(-z) \end{array} \right) \left(\begin{array}{c} G_0(z) \\ G_1(z) \end{array} \right) = H_0(-z) G_0(z) + H_1(-z) G_1(z) = 0$$

which is guaranteed by

$$\frac{G_0(z)}{G_1(z)} = -\frac{H_1(-z)}{H_0(-z)} \tag{2.10}$$

or by the following pair of conditions for any rational $C(z)$

$$G_0(z) = C(z) H_1(-z)$$

$$G_1(z) = (-C(z)) H_0(-z)$$

Under these aliasing-cancellation conditions, we get the input/output relation

$$Y(z) = \frac{1}{2}\left(H_0(z)H_1(-z) - H_1(z)H_0(-z)\right)C(z)X(z) \tag{2.11}$$

where $T(z) = \frac{1}{2}\left(H_0(z)H_1(-z) - H_1(z)H_0(-z)\right)C(z)$ represents the system transfer function. We say that "perfect reconstruction" results when $y[n] = x[n-l]$ for some $l \in \mathbb{N}$, or equivalently when $T(z) = z^{-l}$.

> NOTE: The aliasing-cancellation conditions remove one degree of freedom from our filterbank design; originally, we had the choice of four transfer functions $\{H_0(z), H_1(z), G_0(z), G_1(z)\}$, whereas now we can choose three: $\{H_0(z), H_1(z), C(z)\}$.

2.3.2 Two-Branch Quadvalue Mirror Filterbank (QMF)[8]

2.3.2.1 Quadrature Mirror Filterbanks

The quadrature mirror filterbank (QMF) is an aliasing-cancellation filterbank with the additional design choices:

- $H_0(z)$: causal real-coefficient FIR
- $H_1(z) = H_0(-z)$
- $C(z) = 2$

Combining the various design rules, it is easy to see that all filters will be causal, real-coefficient, and FIR. The QMF choices yield the system transfer function

$$\begin{aligned} T(z) &= H_0{}^2(z) - H_1{}^2(z) \\ &= H_0{}^2(z) - H_0{}^2(-z) \end{aligned} \tag{2.12}$$

The name "QMF" is appropriate for the following reason. Note that

$$|H_1\left(e^{j\omega}\right)| = |H_0\left(e^{-(j\omega)}\right)| = |H_0\left(e^{j(\omega-\pi)}\right)| = |H_0\left(e^{j(\pi-\omega)}\right)|$$

where the last step follows from the DTFT[9] conjugate-symmetry of real-coefficient filters. This implies that the magnitude responses $|H_0\left(e^{j\omega}\right)|$ and $|H_1\left(e^{j\omega}\right)|$ from a mirror-image pair, symmetric around $\omega = \frac{\pi}{2} = \frac{2\pi}{4}$ (the "quadrature frequency"), as illustrated in Figure 2.15.

---

[8]This content is available online at $<$http://cnx.org/content/m10426/2.14/$>$.

[9]"Discrete-Time Fourier Transform" $<$http://cnx.org/content/m10461/latest/$>$
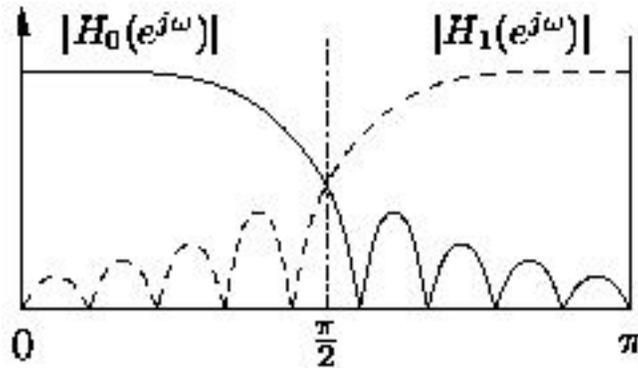
Figure 2.15

The QMF design rules imply that all filters in the bank are directly related to the "prototype" filter $H_0(z)$, and thus we might suspect a polyphase implementation. In fact, one exists. Using the standard polyphase decomposition of $H_0(z)$, we have

$$H_0(z) = P_0(z^2) + z^{-1}P_1(z^2) \tag{2.13}$$

so that

$$H_1(z) = H_0(-z) = P_0(z^2) - z^{-1}P_1(z^2)$$

$$G_0(z) = 2H_1(-z) = 2P_0(z^2) + 2z^{-1}P_1(z^2)$$

$$G_1(z) = -2H_0(-z) = 2P_0(z^2) + 2z^{-1}P_1(z^2)$$

Application of the Noble identity results in the polyphase structure in Figure 2.15:
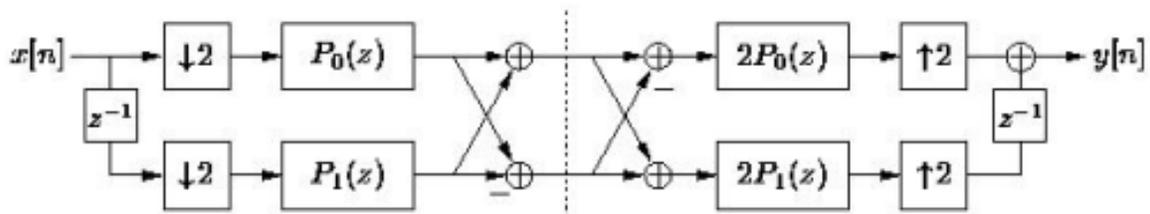
Figure 2.16

The QMF choice $C(z) = 2$ implies that the synthesis filters have twice the DC gain of the corresponding analysis filters. Recalling that decimation by 2 involves anti-alias lowpass filtering with DC gain equal to one, while interpolation by 2 involves anti-image lowpass filtering with DC gain equal to 2, Figure 2.16 suggests an explanation for the choice $C(z) = 2$.

2.3.3 Perfect Reconstruction QMF[10]

2.3.3.1 Perfect Reconstruction QMF

The system transfer function for a QMF bank is

$$
\begin{aligned}
T\left(z\right) &= {H_0}^2\left(z\right) - {H_1}^2\left(z\right) \\
&= 4z^{-1}P_0\left(z^2\right)P_1\left(z^2\right)
\end{aligned}
\tag{2.14}
$$

For perfect reconstruction, we need $T\left(z\right) = z^{-l}$ for some $l \in \mathbb{N}$, which implies the equivalent conditions

$$
4z^{-1}P_0\left(z^2\right)P_1\left(z^2\right) = z^{-l}
$$

$$
P_0\left(z^2\right)P_1\left(z^2\right) = \frac{1}{4}z^{-(l-1)}
$$

$$
P_0\left(z\right)P_1\left(z\right) = \frac{1}{4}z^{-\frac{l-1}{2}}
$$

For FIR polyphase filters, this can only be satisfied by

$$
P_0\left(z\right) = \beta_0 z^{-n_0}
$$

$$
P_1\left(z\right) = \beta_1 z^{-n_1}
$$

where we have $n_0 + n_1 = \frac{l-1}{2}$ and $\beta_0\beta_1 = \frac{1}{4}$.

In other words, the polyphase filters are trivial, so that the prototype filter $H_0\left(z\right)$ has a two-tap response. With only two taps, $H_0\left(z\right)$ cannot be a very good lowpass filter, meaning that the sub-band signals will not be spectrally well-separated. From this we conclude that two-channel [11] perfect reconstruction QMF banks exist but are not very useful.

2.3.4 Johnston's QMF Banks[12]

Two-channel perfect-reconstruction QMF banks (Section 2.3.2) are not very useful because the analysis filters have poor frequency selectivity. The selectivity characteristics can be improved, however, if we allow the system response $T\left(e^{j\omega}\right)$ to have magnitude-response ripples while keeping its linear phase.

Say that $H_0\left(z\right)$ is causal, linear-phase, and has impulse response[13] length N. Then it is possible to write $H_0\left(e^{j\omega}\right)$ in terms of a real-valued zero-phase response $\widetilde{H}_0\left(e^{j\omega}\right)$, so that

$$
H_0\left(e^{j\omega}\right) = e^{(-j)\omega\frac{N-1}{2}}\widetilde{H}_0\left(e^{j\omega}\right)
\tag{2.15}
$$

$$
\begin{aligned}
T\left(e^{j\omega}\right) &= {H_0}^2\left(e^{j\omega}\right) - {H_0}^2\left(e^{j(\omega-\pi)}\right) \\
&= e^{(-j)\omega(N-1)}{\widetilde{H}_0}^2\left(e^{j\omega}\right) - e^{(-j)(\omega-\pi)(N-1)}{\widetilde{H}_0}^2\left(e^{j(\omega-\pi)}\right) \\
&= e^{(-j)\omega(N-1)}\left({\widetilde{H}_0}^2\left(e^{j\omega}\right) - e^{j\pi(N-1)}{\widetilde{H}_0}^2\left(e^{j(\omega-\pi)}\right)\right)
\end{aligned}
\tag{2.16}
$$

Note that if N is odd, $e^{j\omega(N-1)} = 1$,

$$
T\left(e^{j\omega}\right)\big|_{\omega=\frac{\pi}{2}} = 0
\tag{2.17}
$$

---

[10]This content is available online at $<$http://cnx.org/content/m11002/2.2/$>$.

[11]It turns out that M-channel perfect reconstruction QMF banks have more useful responses for larger values of M.

[12]This content is available online at $<$http://cnx.org/content/m10932/2.4/$>$.

[13]"Continuous Time Impulse Function" $<$http://cnx.org/content/m10059/latest/$>$

A null in the system response would be very undesirable, and so we restrict N to be an even number. In that case,

$$
\begin{aligned}
T\left(e^{j\omega}\right) &= e^{(-j)\omega(N-1)}\left(\widetilde{H}_0{}^2\left(e^{j\omega}\right) + \widetilde{H}_0{}^2\left(e^{j(\omega-\pi)}\right)\right) \\
&= e^{(-j)\omega(N-1)}\left(\left(\left|H_0\left(e^{j\omega}\right)\right|\right)^2 + \left(\left|H_0\left(e^{j(\omega-\pi)}\right)\right|\right)^2\right)
\end{aligned}
\tag{2.18}
$$

NOTE: The system response is linear phase, but will have amplitude distortion if $\left(\left|H_0\left(e^{j\omega}\right)\right|\right)^2 +$ $\left(\left|H_0\left(e^{j(\omega-\pi)}\right)\right|\right)^2$ is not equal to a constant.

Johnston's idea was to assign a cost function that penalizes deviation from perfect reconstruction as well as deviation from an ideal lowpass filter with cutoff ω_0. Specifically, real symmetric coefficients $h_0[n]$ are chosen to minimize

$$
J = \lambda \int_{\omega_0}^{\pi}\left(\left|H_0\left(e^{j\omega}\right)\right|\right)^2 d\omega - 1\int_0^{\infty} 1 - \left(\left|H_0\left(e^{j\omega}\right)\right|\right)^2 - \left(\left|H_0\left(e^{j(\pi-\omega)}\right)\right|\right)^2 d\omega
\tag{2.19}
$$

where $0 < \lambda < 1$ balances between the two conflicting objectives. Numerical optimization techniques can be used to determine the coefficients, and a number of popular coefficient sets have been tabulated. (See *Crochiere and Rabiner*[7], *Johnston*[5], and *Ansari and Liu*[1])

Example 2.1: "12B" Filter

As an example, consider the "12B" filter from *Johnston*[5]:

$$
h_0[0] = -0.006443977 = h_0[11]
$$

$$
h_0[1] = 0.02745539 = h_0[10]
$$

$$
h_0[2] = -0.00758164 = h_0[9]
$$

$$
h_0[3] = -0.0913825 = h_0[8]
$$

$$
h_0[4] = 0.09808522 = h_0[7]
$$

$$
h_0[5] = 0.4807962 = h_0[6]
$$

which gives the following DTFT magnitudes (Figure 2.17).
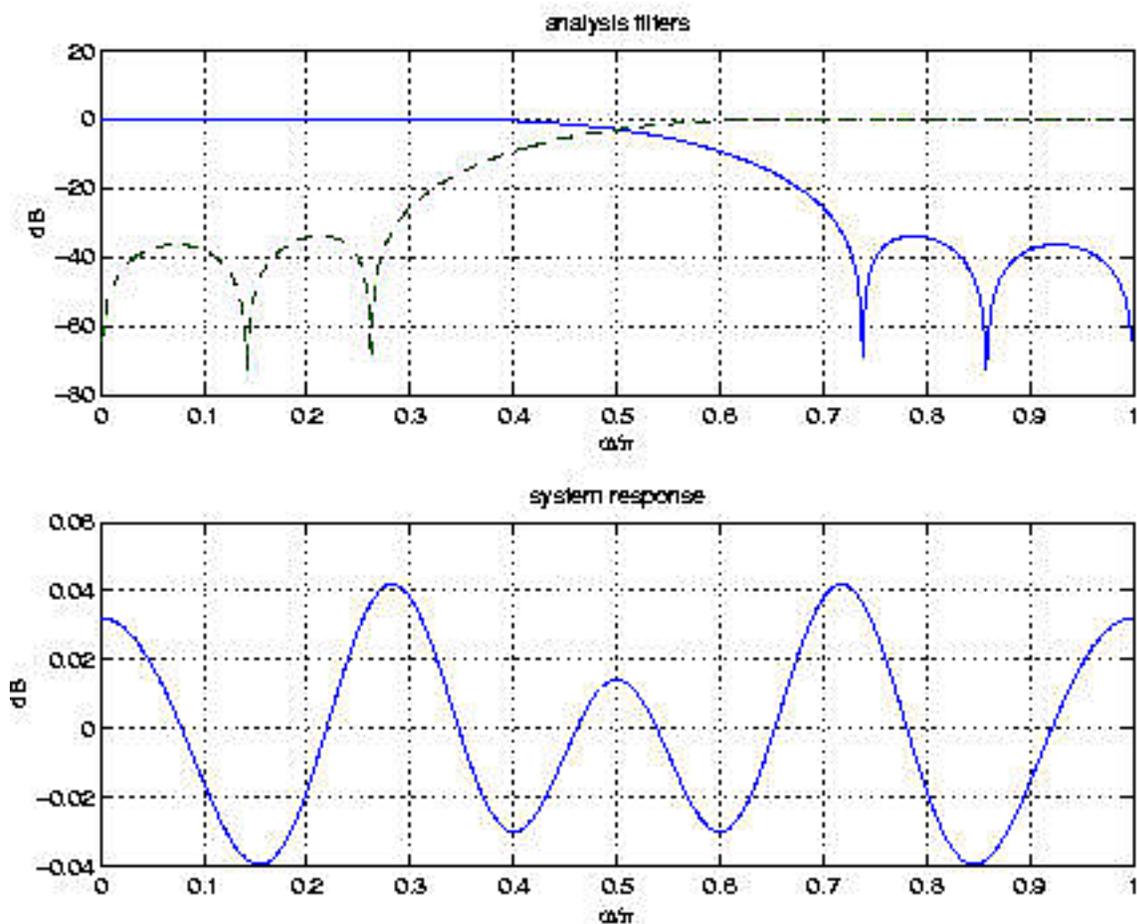
Figure 2.17: The top plot shows the analysis filters and the bottom one shows the system response.

2.3.5 Perfect Reconstruction FIR Filter Banks[14]

2.3.5.1 FIR Perfect-Reconstruction Conditions

The QMF (Section 2.3.2) design choices prevented the design of a useful (*i.e.*, frequency selective) perfect-reconstruction (PR) FIR filterbank. This motivates us to re-examine PR filterbank design without the overly-restrictive QMF conditions. However, we will still require causal FIR filters with real coefficients.

Recalling that the two-channel filterbank (Section 2.3.1) (Figure 2.18),

---

[14]This content is available online at <http://cnx.org/content/m10412/2.14/>.
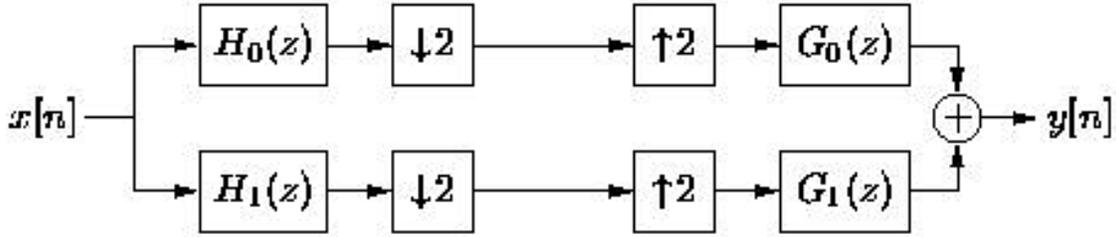
Figure 2.18

has the input/output relation:

$$Y(z) = \frac{1}{2} \begin{pmatrix} X(z) & X(-z) \end{pmatrix} \begin{pmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{pmatrix} \begin{pmatrix} G_0(z) \\ G_1(z) \end{pmatrix} \tag{2.20}$$

we see that the delay-l perfect reconstruction requires

$$\begin{pmatrix} 2z^{-l} \\ 0 \end{pmatrix} = \begin{pmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{pmatrix} \begin{pmatrix} G_0(z) \\ G_1(z) \end{pmatrix} \tag{2.21}$$

where

$$H(z) = \begin{pmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{pmatrix}$$

or, equivalently, that

$$
\begin{aligned}
\begin{pmatrix} G_0(z) \\ G_1(z) \end{pmatrix} &= H^{-1}(z) \begin{pmatrix} 2z^{-l} \\ 0 \end{pmatrix} \\
&= \frac{1}{\det H(z)} \begin{pmatrix} H_1(-z) & -H_1(z) \\ -H_0(-z) & H_0(z) \end{pmatrix} \begin{pmatrix} 2z^{-l} \\ 0 \end{pmatrix} \\
&= \frac{2}{\det H(z)} \begin{pmatrix} z^{-l} H_1(-z) \\ -\left(z^{-l} H_0(-z) \right) \end{pmatrix}
\end{aligned}
\tag{2.22}
$$

where

$$\det H(z) = H_0(z) H_1(-z) - H_0(-z) H_1(z) \tag{2.23}$$

For FIR $G_0(z)$ and $G_1(z)$, we require [15] that

$$\det H(z) = cz^{-k} \tag{2.24}$$

for $c \in \mathbb{R}$ and $k \in \mathbb{Z}$. Under this determinant condition, we find that

$$\begin{pmatrix} G_0(z) \\ G_1(z) \end{pmatrix} = \frac{2z^{-(l-k)}}{c} \begin{pmatrix} H_1(-z) \\ -H_0(-z) \end{pmatrix} \tag{2.25}$$

Assuming that $H_0(z)$ and $H_1(z)$ are causal with non-zero initial coefficient, we choose $k = l$ to keep $G_0(z)$ and $G_1(z)$ causal and free of unnecessary delay.

_____

[15]Since we cannot assume that FIR $H_0(z)$ and $H_1(z)$ share a common root.

2.3.5.1.1 Summary of Two-Channel FIR-PR Conditions

Summarizing the two-channel FIR-PR conditions:

$$H_0\left(z\right) \quad H_1\left(z\right) = \quad \text{causal real} - \text{coefficient} \ \ \text{FIR}$$

$$\det H\left(z\right) = cz^{-l} \quad , \quad c \in \mathbb{R} \ \ l \in \mathbb{Z}$$

$$G_0\left(z\right) = \frac{2}{c}H_1\left(-z\right)$$

$$G_1\left(z\right) = \frac{-2}{c}H_0\left(-z\right)$$

2.3.6 Orthogonal Perfect Reconstruction FIR Filterbank[16]

2.3.6.1 Orthogonal PR Filterbanks

The FIR perfect-reconstruction (PR) (Section 2.3.5) conditions leave some freedom in the choice of $H_0\left(z\right)$ and $H_1\left(z\right)$. **Orthogonal PR filterbanks** are defined by causal real-coefficient even-length-N analysis filters that satisfy the following two equations:

$$1 = H_0\left(z\right)H_0\left(z^{-1}\right) + H_0\left(-\text{z}\right)H_0\left(-z^{-1}\right) \tag{2.26}$$

$$H_1\left(z\right) = \left(\pm\left(z\right)\right)^{-(N-1)}H_0\left(-z^{-1}\right) \tag{2.27}$$

To verify that these design choices satisfy the FIR-PR requirements for $H_0\left(z\right)$ and $H_1\left(z\right)$, we evaluate $\det\mathbf{H}\left(z\right)$ under the second condition above. This yields

$$
\begin{aligned}
\det\mathbf{H}\left(z\right) &= \pm\left(H_0\left(z\right)H_1\left(z^{-1}\right) - H_0\left(-\text{z}\right)H_1\left(z\right)\right) \\
&= \left(-z^{-(N-1)}\right)\left(H_0\left(z\right)H_0\left(z^{-1}\right) + H_0\left(-\text{z}\right)H_0\left(-z^{-1}\right)\right) \\
&= z^{-(N-1)}
\end{aligned}
\tag{2.28}
$$

which corresponds to $c = -1$ and $l = N - 1$ in the FIR-PR determinant condition $\det\mathbf{H}\left(z\right) = cz^{-l}$. The remaining FIR-PR conditions then imply that the synthesis filters are given by

$$
\begin{aligned}
G_0\left(z\right) &= -2H_1\left(z^{-1}\right) \\
&= 2z^{-(N-1)}H_0\left(z^{-1}\right)
\end{aligned}
\tag{2.29}
$$

$$
\begin{aligned}
G_1\left(z\right) &= 2H_0\left(-z\right) \\
&= 2z^{-(N-1)}H_1\left(z^{-1}\right)
\end{aligned}
\tag{2.30}
$$

The orthogonal PR design rules imply that $H_0\left(e^{j\omega}\right)$ is "power symmetric" and that $\left\{H_0\left(e^{j\omega}\right), H_1\left(e^{j\omega}\right)\right\}$ form a "power complementary" pair. To see the power symmetry, we rewrite the first design rule using $z = e^{j\omega}$ and $-1 = e^{\pm j\pi}$, which gives

$$
\begin{aligned}
1 &= H_0\left(e^{j\omega}\right)H_0\left(e^{-(j\omega)}\right) + H_0\left(e^{j(\omega-\pi)}\right)H_0\left(e^{(-j)(\omega-\pi)}\right) \\
&= \left(|H_0\left(e^{j\omega}\right)|\right)^2 + \left(|H_0\left(e^{j(\omega-\pi)}\right)|\right)^2 \\
&= \left(|H_0\left(e^{j\omega}\right)|\right)^2 + \left(|H_0\left(e^{j(\pi-\omega)}\right)|\right)^2
\end{aligned}
\tag{2.31}
$$

---

[16]This content is available online at <http://cnx.org/content/m10413/2.14/>.

The last two steps leveraged the fact that the DTFT[17] of a real-coefficient filter is conjugate-symmetric. The power-symmetry property is illustrated in Figure 2.19:
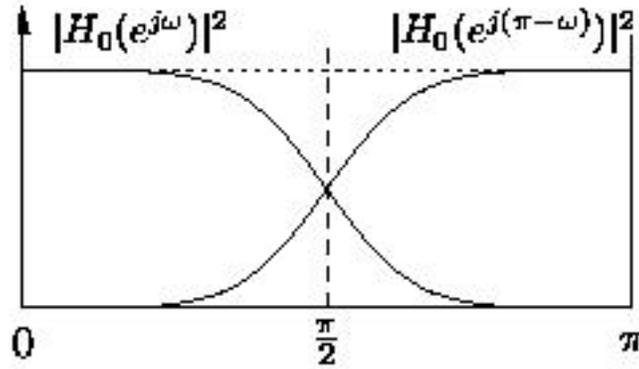
Figure 2.19

Power complementarity follows from the second orthogonal PR design rule, which implies $|H_1\left(e^{j\omega}\right)| = |H_0\left(e^{j(\pi-\omega)}\right)|$. Plugging this into the previous equation, we find

$$1 = \left(|H_0\left(e^{j\omega}\right)|\right)^2 + \left(|H_1\left(e^{j\omega}\right)|\right)^2 \tag{2.32}$$

The power-complimentary property is illustrated in Figure 2.20:
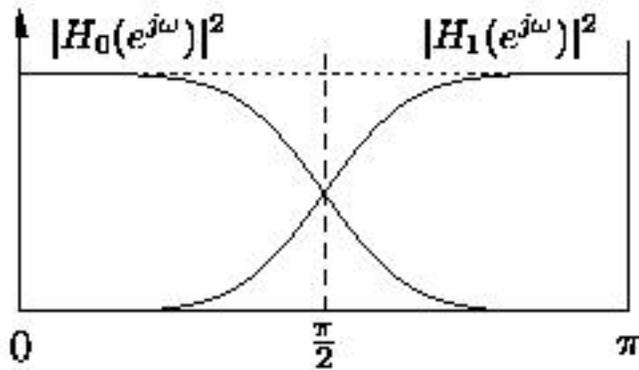
Figure 2.20

---

[17]"Discrete Time Fourier Transform (DTFT)" <http://cnx.org/content/m10108/latest/>

2.3.7 Design of Orthogonal PR-FIR Filterbanks via Halfband Spectral Factorization[18]

2.3.7.1 Design of Orthogonal PR-FIR Filterbanks via Halfband Spectral Factorization

Recall that analysis-filter design for orthogonal PR-FIR filterbanks reduces to the design of a real-coefficient causal FIR prototype filter $H_0(z)$ that satisfies the power-symmetry condition

$$\left(|H_0\left(e^{j\omega}\right)|\right)^2 + \left(|H_0\left(e^{j(\pi-\omega)}\right)|\right)^2 = 1 \qquad (2.33)$$

Power-symmetric filters are closely related to "halfband" filters. A zero-phase halfband filter is a zero-phase filter $F(z)$ with the property

$$F(z) + F(-z) = 1 \qquad (2.34)$$

When, in addition, $F(z)$ has real-valued coefficients, its DTFT is "amplitude-symmetric":

$$F\left(e^{j\omega}\right) + F\left(e^{j(\pi-\omega)}\right) = 1 \qquad (2.35)$$

The amplitude-symmetry property is illustrated in Figure 2.21:
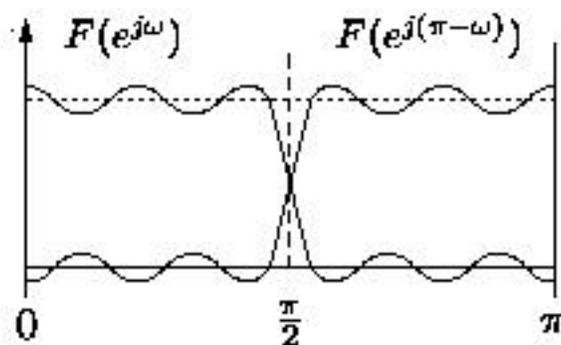
Figure 2.21

If, in addition to being real-valued, [19] $F\left(e^{j\omega}\right)$ is non-negative, then $F\left(e^{j\omega}\right)$ constitutes a valid power response. If we can find $H_0(z)$ such that $\left(|H_0\left(e^{j\omega}\right)|\right)^2 = F\left(e^{j\omega}\right)$, then this $H_0(z)$ will satisfy the desired power-symmetry property $\left(|H_0\left(e^{j\omega}\right)|\right)^2 + \left(|H_0\left(e^{j(\pi-\omega)}\right)|\right)^2 = 1$.

First, realize $F\left(e^{j\omega}\right)$ is easily modified to ensure non-negativity: construct $q[n] = f[n] + \epsilon\delta[n]$ for sufficiently large ε, which will raise $F\left(e^{j\omega}\right)$ by ϵ uniformly over ω (see Figure 2.22).

---

[18]This content is available online at <http://cnx.org/content/m11005/2.2/>.

[19]Recall that zero-phase filters have real-valued DTFTs.
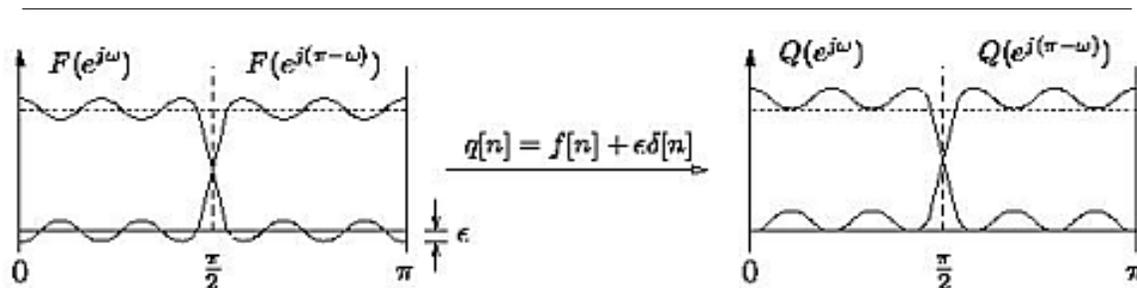
Figure 2.22

The resulting $Q(z)$ is non-negative and satisfies the amplitude-symmetry condition $Q\left(e^{j\omega}\right) + Q\left(e^{j(\pi-\omega)}\right) = 1 + 2\epsilon$. We will make up for the additional gain later. The procedure by which $H_0(z)$ can be calculated from the raised halfband $Q(z)$, known as **spectral factorization**, is described next.

Since $q[n]$ is conjugate-symmetric around the origin, the roots of $Q(z)$ come in pairs $\left\{a_i, \frac{1}{a_i^*}\right\}$. This can be seen by writing $Q(z)$ in the factored form below, which clearly corresponds to a polynomial with coefficients conjugate-symmetric around the 0th-order coefficient.

$$
\begin{aligned}
Q(z) &= \sum_{n=-(N-1)}^{N-1} q[n] z^{-n} \\
&= A \prod_{i=1}^{N-1} \left(1 - a_i z^{-1}\right)\left(1 - a_i^* z\right)
\end{aligned}
\tag{2.36}
$$

where $A \in \mathbb{R}^+$. Note that the complex numbers $\left\{a_i, \frac{1}{a_i^*}\right\}$ are symmetric across the unit circle in the z-plane[20]. Thus, for ever root of $Q(z)$ inside the unit-circle, there exists a root outside of the unit circle (see Figure 2.23).
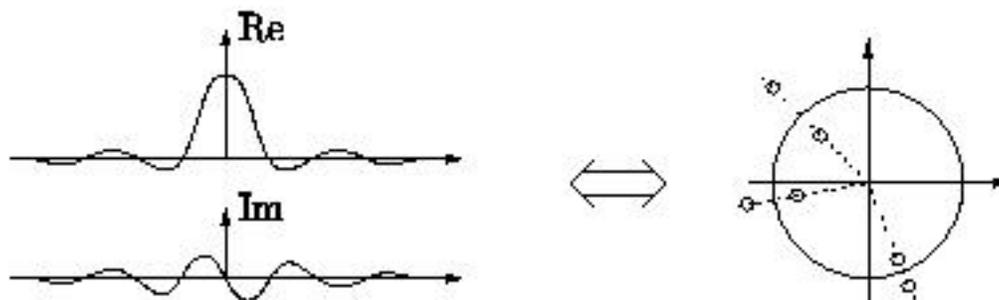
Figure 2.23

Let us assume, without loss of generality, that $|a_i| < 1$. If we form $H_0(z)$ from the roots of $Q(z)$ with

---

[20]"Understanding Pole/Zero Plots on the Z-Plane": Section The Z-Plane
<http://cnx.org/content/m10556/latest/#zplane>

magnitude less than one:

$$H_0(z) = \sqrt{A} \prod_{i=1}^{N-1} 1 - a_i z^{-1} \tag{2.37}$$

then it is apparent that $\left(|H_0\left(e^{j\omega}\right)|\right)^2 = Q\left(e^{j\omega}\right)$. This $H_0(z)$ is the so-called **minimum-phase spectral factor** of $Q(z)$.

Actually, in order to make $\left(|H_0\left(e^{j\omega}\right)|\right)^2 = Q\left(e^{j\omega}\right)$, we are not required to choose all roots inside the unit circle; it is enough to choose one root from every unit-circle-symmetric pair. However, we do want to ensure that $H_0(z)$ has real-valued coefficients. For this, we must ensure that roots come in conjugate-symmetric pairs, *i.e.*, pairs having symmetry with respect to the real axis in the complex plane (Figure 2.24).
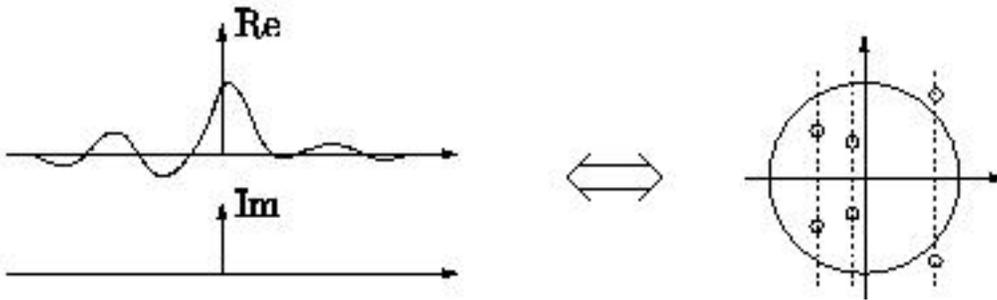
Figure 2.24

Because $Q(z)$ has real-valued coefficients, we know that its roots satisfy this conjugate-symmetry property. Then forming $H_0(z)$ from the roots of $Q(z)$ that are strictly inside (or strictly outside) the unit circle, we ensure that $H_0(z)$ has real-valued coefficients.

Finally, we say a few words about the design of the halfband filter $F(z)$. The **window design** method is one technique that could be used in this application. The window design method starts with an ideal lowpass filter, and windows its doubly-infinite impulse response using a window function with finite time-support. The ideal real-valued zero-phase halfband filter has impulse response (where $n \in \mathbb{Z}$):

$$\overline{f}[n] = \frac{\sin\left(\frac{\pi}{2}n\right)}{\pi n} \tag{2.38}$$

which has the important property that all even-indexed coefficients except $\overline{f}[0]$ equal zero. It can be seen that this latter property is implied by the halfband definition $\overline{F}(z) + \overline{F}(z^{-1}) = 1$ since, due to odd-coefficient cancellation, we find

$$\begin{aligned} 1 &= \overline{F}(z) + \overline{F}(z^{-1}) \\ &= 2\sum_{m=-\infty}^{\infty} \overline{f}[2m] z^{-(2m)} \Leftrightarrow \overline{f}(2m) = \tfrac{1}{2}\delta[m] \end{aligned} \tag{2.39}$$

Note that windowing the ideal halfband does not alter the property $\overline{f}(2m) = \frac{1}{2}\delta[m]$, thus the window design $F(z)$ is guaranteed to be halfband feature. Furthermore, a real-valued window with origin-symmetry preserves the real-valued zero-phase property of $\{\overline{f}[n]\}$ above. It turns out that many of the other popular design methods (*e.g.*, LS and equiripple) also produce halfband filters when the cutoff is specified at $\frac{\pi}{2}$ radians and all passband/stopband specifications are symmetric with respect to $\omega = \frac{\pi}{2}$.

2.3.7.1.1 Design Procedure Summary

We now summarize the design procedure for a length-N analysis lowpass filter for an orthogonal perfect-reconstruction FIR filterbank:

1. Design a zero-phase real-coefficient halfband lowpass filter $F(z) = \sum_{n=-(N-1)}^{N-1} f[n] z^{-n}$ where N is a positive even integer (via, *e.g.*, window designs, LS, or equiripple).
2. Calculate ϵ, the maximum negative value of $F(e^{j\omega})$. (Recall that $F(e^{j\omega})$ is real-valued for all ω because it has a zero-phase response.) Then create "raised halfband" $Q(z)$ via $q[n] = f[n] + \epsilon\delta[n]$, ensuring that $Q(e^{j\omega}) \geq 0$, forall ω.
3. Compute the roots of $Q(z)$, which should come in unit-circle-symmetric pairs $\left\{a_i, \frac{1}{a_i^*}\right\}$. Then collect the roots with magnitude less than one into filter $\overset{\wedge}{H_0}(z)$.
4. $\overset{\wedge}{H_0}(z)$ is the desired prototype filter except for a scale factor. Recall that we desire

$$\left(|H_0\left(e^{j\omega}\right)|\right)^2 + \left(\left|H_0\left(e^{j(\pi-\omega)}\right)\right|\right)^2 = 1$$

Using Parseval's Theorem[21], we see that $\left\{\overset{\wedge}{h_0}[n]\right\}$ should be scaled to give $\{h_0[n]\}$ for which $\sum_{n=0}^{N-1} h_0{}^2[n] = \frac{1}{2}$.

2.3.8 Bi-Orthogonal Perfect Reconstruction FIR Filterbanks[22]

2.3.8.1 Bi-Orthogonal Filter Banks

Due to the minimum-phase spectral factorization, orthogonal PR-FIR filterbanks (Section 2.3.6) will not have linear-phase analysis and synthesis filters. Non-linear phase may be undesirable for certain applications. "Bi-orthogonal" designs are closely related to orthogonal designs, yet give linear-phase filters. The analysis-filter design rules for the bi-orthogonal case are

- $F(z)$: zero-phase real-coefficient halfband such that $F(z) = \sum_{n=-(N-1)}^{N-1} f[n] z^{-n}$, where N is even.
- $z^{-(N-1)}F(z) = H_0(z) H_1(-z)$

It is straightforward to verify that these design choices satisfy the FIR perfect reconstruction condition $\det H(z) = cz^{-l}$ with $c = 1$ and $l = N - 1$:

$$\begin{aligned}
\det H(z) &= H_0(z) H_1(-z) - H_0(-z) H_1(z) \\
&= z^{-(N-1)}F(z) - (-1)^{-(N-1)}z^{-(N-1)}F(-z) \\
&= z^{-(N-1)}(F(z) + F(-z)) \\
&= z^{-(N-1)}
\end{aligned} \tag{2.40}$$

Furthermore, note that $z^{-(N-1)}F(z)$ is causal with real coefficients, so that both $H_0(z)$ and $H_1(z)$ can be made causal with real coefficients. (This was another PR-FIR requirement.) The choice $c = 1$ implies that the synthesis filters should obey

$$G_0(z) = 2H_1(-z)$$

$$G_1(z) = -2H_0(-z)$$

---

[21]"Plancharel and Parseval's Theorems": Section Parseval's Theorem: a different approach
$<$http://cnx.org/content/m10769/latest/#parsec$>$
[22]This content is available online at $<$http://cnx.org/content/m10414/2.11/$>$.

From the design choices above, we can see that bi-orthogonal analysis filter design reduces to the factorization of a causal halfband filter $z^{-(N-1)}F(z)$ into $H_0(z)$ and $H_1(z)$ that have **both** real coefficients and linear-phase. Earlier we saw that linear-phase corresponds to root symmetry across the unit circle in the complex plane, and that real-coefficients correspond to complex-conjugate root symmetry. Simultaneous satisfaction of these two properties can be accomplished by **quadruples** of roots. However, there are special cases in which a root pair, or even a single root, can simultaneously satisfy these properties. Examples are illustrated in Figure 2.25:
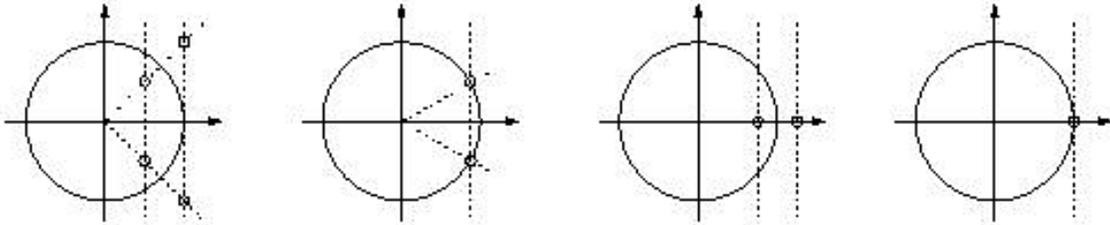
Figure 2.25

The design procedure for the analysis filters of a bi-orthogonal perfect-reconstruction FIR filterbank is summarized below:

1. Design a zero-phase real-coefficient filter $F(z) = \sum_{n=-(N-1)}^{N-1} f[n] z^{-n}$ where N is a positive even integer (via, e.g., window designs, LS, or equiripple).

2. Compute the roots of $F(z)$ and partition into a set of root groups $\{G_0, G_1, G_2, \dots\}$ that have **both** complex-conjugate and unit-circle symmetries. Thus a root group may have one of the following forms:

$$G_i = \left\{ a_i, a_i^*, \frac{1}{a_i}, \frac{1}{a_i^*} \right\}$$

$$G_i = \{a_i, a_i^*\} \quad , \quad |a_i| = 1$$

$$G_i = \left\{ a_i, \frac{1}{a_i} \right\} \quad , \quad a_i \in \mathbb{R}$$

$$G_i = \{a_i\} \quad , \quad a_i = \pm(1)$$

Choose [23] a subset of root groups and construct $\hat{H}_0(z)$ from those roots. Then construct $\hat{H}_1(-z)$ from the roots in the remaining root groups. Finally, construct $\hat{H}_1(z)$ from $\hat{H}_1(-z)$ by reversing the signs of odd-indexed coefficients.

3. $\hat{H}_0(z)$ and $\hat{H}_1(z)$ are the desired analysis filters up to a scaling. To take care of the scaling, first create $\widetilde{H}_0(z) = a\hat{H}_0(z)$ and $\widetilde{H}_1(z) = b\hat{H}_1(z)$ where a and b are selected so that $\sum_n \widetilde{h}_0[n] = 1 = \sum_n \widetilde{h}_1[n]$.

---

[23] Note that $\hat{H}_0(z)$ and $\hat{H}_1(z)$ will be real-coefficient linear-phase regardless of which groups are allocated to which filter. Their frequency selectivity, however, will be strongly influenced by group allocation. Thus, you many need to experiment with different allocations to find the best highpass/lowpass combination. Note also that the length of $H_0(z)$ may differ from the length of $H_0(z)$.

Then create $H_0(z) = c\widetilde{H}_0(z)$ and $H_1(z) = c\widetilde{H}_1(z)$ where c is selected so that the property

$$z^{-(N-1)}F(z) = H_0(z)H_1(-z)$$

is satisfied at DC (i.e., $z = e^{j0} = 1$). In other words, find c so that $\sum_n h_0[n]\sum_m h_1[n](-1)^m = 1$.

2.3.9 Filterbanks with >2 Branches[24]

2.3.9.1 Filterbanks with >2 Branches

Thus far the previous discussion on filterbanks has concentrated on "modern" filterbanks with only two branches. There are two standard ways by which the number of branches can be increased.

2.3.9.1.1 M-Band Filterbanks

The ideas used to construct two-branch PR-FIR filterbanks (Section 2.3.5) can be directly extended to the M-branch case. (See *Vaidyanathan*[10] and *Mitra*[6]) This yields, for example, a polynomial matrix $H(z)$ with M rows and M columns. For these M-band filterbanks, the sub-bands will have uniform widths $\frac{2\pi}{L}$ radians (in the ideal case) Figure 2.26.
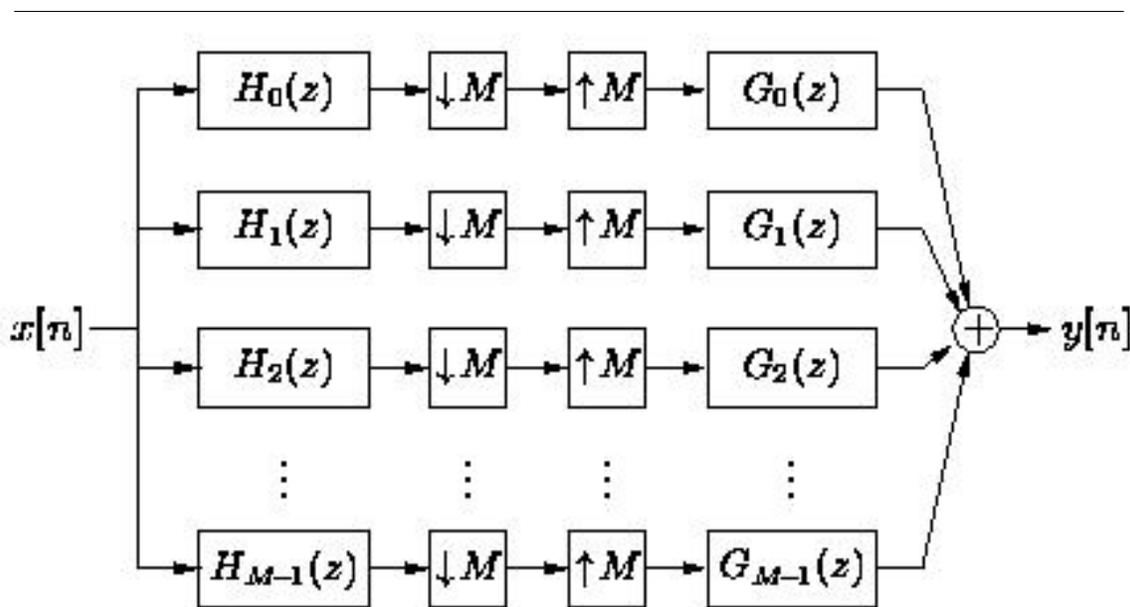
Figure 2.26: M-band Filterbank

2.3.9.1.2 Multi-Level (Cascade) Filterbanks

The two-branch PR-FIR filterbanks can be cascaded to yield PR-FIR filterbanks whose sub-band widths equal $2^{-k}\pi$ for non-negative integers k (in the ideal case). If the magnitude responses of the filters are not well behaved, however, the cascading will result in poor effective frequency-selectivity. Below (Figure 2.27)

---

[24]This content is available online at <http://cnx.org/content/m10415/2.13/>.

we show a filterbank in which the low-frequency sub-bands are narrower than the high-frequency sub-band. Note that the number of input samples equals the total number of sub-band samples.
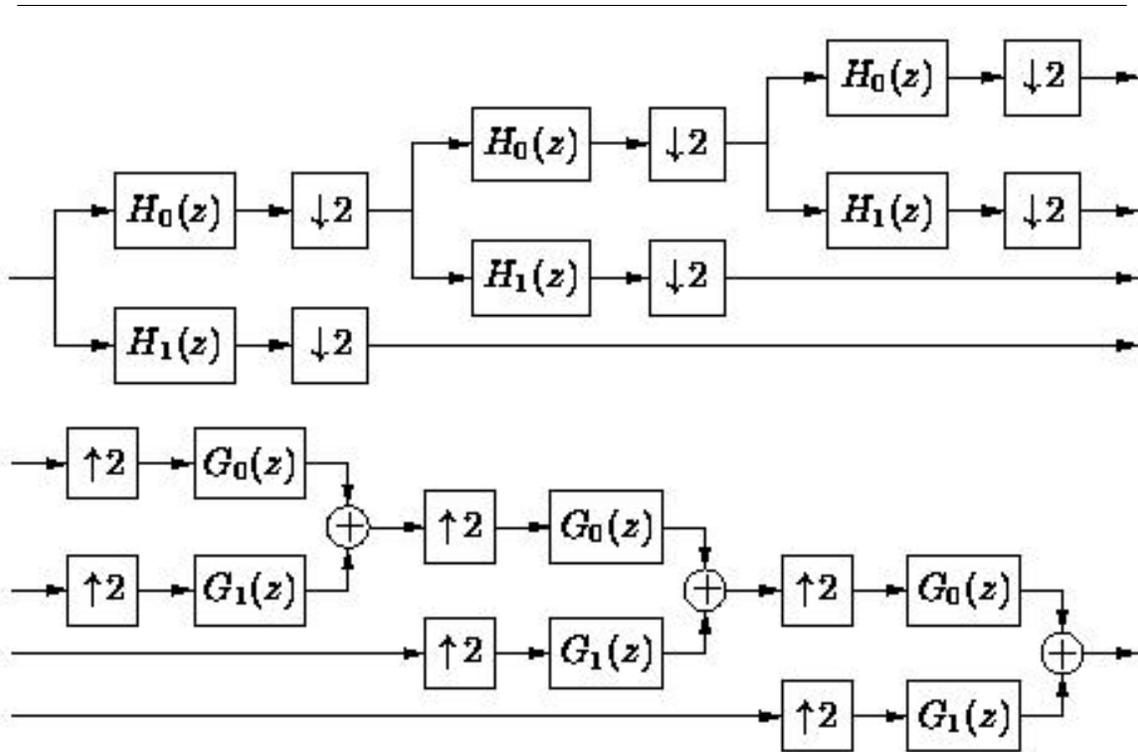
Figure 2.27: Multi-level (Cascaded) Filterbank

We shall see these structures in the context of the discrete wavelet transform.

Chapter 3

Wavelets

3.1 Time Frequency Analysis and Continuous Wavelet Transform

3.1.1 Why Transforms?[1]

In the field of signal processing we frequently encounter the use of **transforms**. Transforms are named such because they take a signal and **transform** it into another signal, hopefully one which is easier to process or analyze than the original. Essentially, transforms are used to manipulate signals such that their most important characteristics are made plainly evident. To isolate a signal's important characteristics, however, one must employ a transform that is well matched to that signal. For example, the Fourier transform, while well matched to certain classes of signal, does not efficiently extract information about signals in other classes. This latter fact motivates our development of the wavelet transform.

3.1.2 Limitations of Fourier Analysis[2]

Let's consider the Continuous-Time Fourier Transform (CTFT) pair:

$$X\left(\Omega\right) = \int_{-\infty}^{\infty} x\left(t\right) e^{-\left(j\Omega t\right)} dt$$

$$x\left(t\right) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X\left(\Omega\right) e^{j\Omega t} d\Omega$$

The Fourier transform pair supplies us with our notion of "frequency." In other words, all of our intuitions regarding the relationship between the time domain and the frequency domain can be traced to this particular transform pair.

It will be useful to view the CTFT in terms of basis elements. The inverse CTFT equation above says that the time-domain signal $x\left(t\right)$ can be expressed as a weighted **summation** of basis elements $\left\{ b_\Omega\left(t\right), b_\Omega\left(t\right) \mid -\infty < \Omega < \infty \right\}$, where $b_\Omega\left(t\right) = e^{j\Omega t}$ is the basis element corresponding to frequency Ω. In other words, the basis elements are parameterized by the variable Ω that we call **frequency**. Finally, $X\left(\Omega\right)$ specifies the weighting coefficient for $b_\Omega\left(t\right)$. In the case of the CTFT, the number of basis elements is uncountably infinite, and thus we need an integral to express the summation.

The Fourier Series (FS) can be considered as a special sub-case of the CTFT that applies when the time-domain signal is periodic. Recall that if $x\left(t\right)$ is periodic with period T, then it can be expressed as a

---

[1]This content is available online at <http://cnx.org/content/m11001/2.1/>.
[2]This content is available online at <http://cnx.org/content/m11003/2.1/>.

weighted summation of basis elements $\{b_k(t)\}|_{k=-\infty}^{\infty}$, where $b_k(t) = e^{j\frac{2\pi}{T}tk}$:

$$x(t) = \sum_{k=-\infty}^{\infty} X[k] e^{j\frac{2\pi}{T}tk}$$

$$X[k] = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) e^{-\left(j\frac{2\pi}{T}tk\right)} dt$$

Here the basis elements comes from a countably-infinite set, parameterized by the frequency index $k \in \mathbb{Z}$. The coefficients $\{X[k]\}|_{k=-\infty}^{\infty}$ specify the strength of the corresponding basis elements within signal $x(t)$.

Though quite popular, Fourier analysis is not always the best tool to analyze a signal whose characteristics vary with time. For example, consider a signal composed of a periodic component plus a sharp "glitch" at time t_0, illustrated in time- and frequency-domains, Figure 3.1.
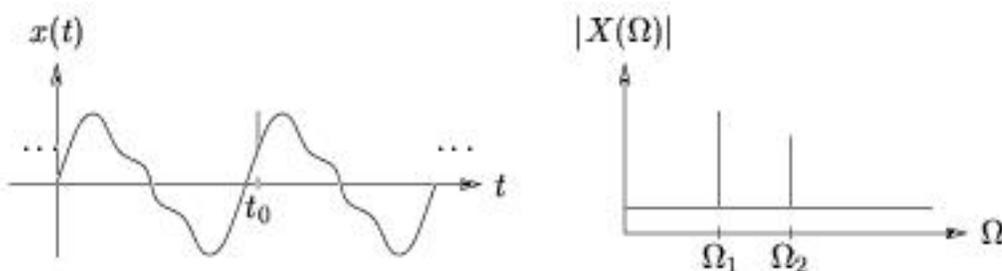
Figure 3.1

Fourier analysis is successful in reducing the complicated-looking periodic component into a few simple parameters: the frequencies $\{\Omega_1, \Omega_2\}$ and their corresponding magnitudes/phases. The glitch component, described compactly in terms of the time-domain location t_0 and amplitude, however, is not described efficiently in the frequency domain since it produces a wide spread of frequency components. Thus, neither time- nor frequency-domain representations alone give an efficient description of the glitched periodic signal: each representation distills only certain aspects of the signal.

As another example, consider the **linear chirp** $x(t) = \sin\left(\Omega t^2\right)$ illustrated in Figure 3.2.
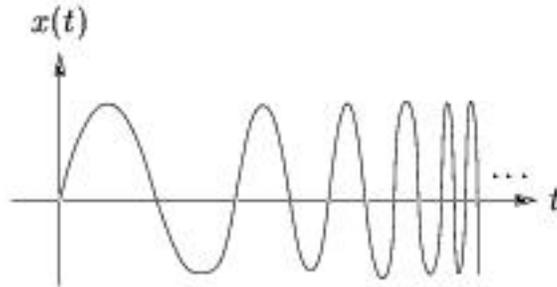
Figure 3.2

Though written using the $\sin(\cdot)$ function, the chirp is not described by a single Fourier frequency. We might try to be clever and write

$$\sin\left(\Omega t^2\right) = \sin\left(\Omega t \cdot t\right) = \sin\left(\Omega(t) \cdot t\right)$$

where it now seems that signal has an **instantaneous frequency** $\Omega(t) = \Omega t$ which grows linearly in time. But here we must be cautious! Our newly-defined instantaneous frequency $\Omega(t)$ is **not** consistent with the Fourier notion of frequency. Recall that the CTFT says that a signal can be constructed as a superposition of fixed-frequency basis elements $e^{j\Omega t}$ with time support from $-\infty$ to $+(\infty)$; these elements are evenly spread out over all time, and so there is noting instantaneous about Fourier frequency! So, while **instantaneous frequency** gives a compact description of the linear chirp, Fourier analysis is not capable of uncovering this simple structure.

As a third example, consider a sinusoid of frequency Ω_0 that is rectangularly windowed to extract only one period (Figure 3.3).
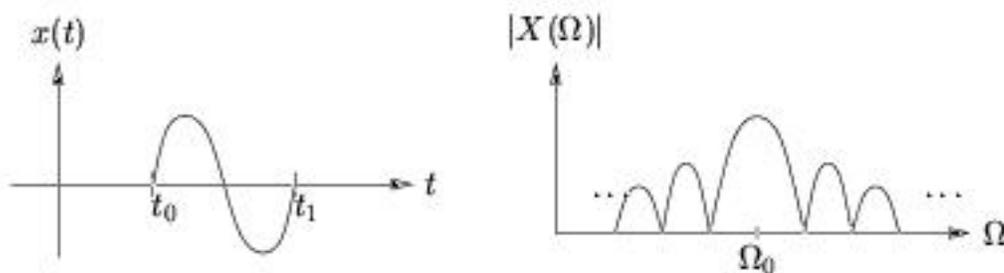
Figure 3.3

Instantaneous-frequency arguments would claim that

$$x(t) = \sin\left(\Omega(t) \cdot t\right) \quad , \quad \Omega(t) = \begin{cases} \Omega_0 & \text{if } t \in \text{window} \\ 0 & \text{if } t \notin \text{window} \end{cases}$$

where $\Omega(t)$ takes on exactly two distinct "frequency" values. In contrast, Fourier theory says that rectangular windowing induces a frequency-domain spreading by a $\frac{\sin(\Omega)}{\Omega}$ profile, resulting in a continuum of Fourier frequency components. Here again we see that Fourier analysis does not efficiently decompose signals whose "instantaneous frequency" varies with time.

3.1.3 Time-Frequency Uncertainty Principle[3]

Recall that Fourier basis elements $b_\Omega(t) = e^{j\Omega t}$ exhibit poor time localization abilities - a consequence of the fact that $b_\Omega(t)$ is evenly spread over all $t \in (-\infty, \infty)$. By **time localization** we mean the ability to clearly identify signal events which manifest during a short time interval, such as the "glitch" described in an earlier example.

At the opposite extreme, a basis composed of shifted Dirac deltas $b_\tau(t) = \Delta(t - \tau)$ would have excellent time localization but terrible "frequency localization," since every Dirac basis element is evenly spread over all Fourier frequencies $\Omega \in [-\infty, \infty]$. This can be seen via $|B_\tau(\Omega)| = |\int_{-\infty}^{\infty} b_\tau(t) e^{-(j\Omega t)} dt| = 1 \ \forall(\Omega)$, regardless of τ. By **frequency localization** we mean the ability to clearly identify signal components which are concentrated at particular Fourier frequencies, such as sinusoids.

These observations motivate the question: does there exist a basis that provides both excellent frequency localization **and** excellent time localization? The answer is "not really": there is a fundamental tradeoff between the time localization and frequency localization of any basis element. This idea is made concrete below.

Let us consider an arbitrary waveform, or basis element, $b(t)$. Its CTFT will be denoted by $B(\Omega)$. Define the energy of the waveform to be E, so that (by Parseval's theorem)

$$E = \int_{-\infty}^{\infty} (|b(t)|)^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} (|B(\Omega)|)^2 d\Omega$$

Next, define the temporal and spectral centers [4] as

$$t_c = \frac{1}{E} \int_{-\infty}^{\infty} t(|b(t)|)^2 dt$$

$$\Omega_c = \frac{1}{2\pi E} \int_{-\infty}^{\infty} \Omega(|B(\Omega)|)^2 d\Omega$$

and the temporal and spectral widths [5] as

$$\Delta_t = \sqrt{\frac{1}{E} \int_{-\infty}^{\infty} (t - t_c)^2 (|b(t)|)^2 dt}$$

$$\Delta_\Omega = \sqrt{\frac{1}{2\pi E} \int_{-\infty}^{\infty} (\Omega - \Omega_c)^2 (|B(\Omega)|)^2 d\Omega}$$

If the waveform is well-localized in time, then $b(t)$ will be concentrated at the point t_c and Δ_t will be small. If the waveform is well-localized in frequency, then $B(\Omega)$ will be concentrated at the point Ω_c and Δ_Ω will be small. If the waveform is well-localized in both time **and** frequency, then $\Delta_t \Delta_\Omega$ will be small. The quantity $\Delta_t \Delta_\Omega$ is known as the **time-bandwidth product**.

From the definitions above one can derive the fundamental properties below. When interpreting the properties, it helps to think of the waveform $b(t)$ as a prototype that can be used to generate an entire basis set. For example, the Fourier basis $\{b_\Omega(t), b_\Omega(t) \mid -\infty < \Omega < \infty\}$ can be generated by frequency shifts of $b(t) = 1$, while the Dirac basis $\{b_\tau(t) \mid -\infty < \tau < \infty\} b_\tau(t)$ can be generated by time shifts of $b(t) = \delta(t)$

---

[3] This content is available online at $<$http://cnx.org/content/m10416/2.18/$>$.

[4] It may be interesting to note that both $\frac{1}{E}(|b(t)|)^2$ and $\frac{1}{2\pi E}(|B(\Omega)|)^2$ are non-negative and integrate to one, thereby satisfying the requirements of probability density functions for random variables t and Ω. The temporal/spectral centers can then be interpreted as the **means** (*i.e.*, centers of mass) of t and Ω.

[5] The quantities $\Delta_t{}^2$ and $\Delta_\Omega{}^2$ can be interpreted as the **variances** of t and Ω, respectively.

1. Δ_t and Δ_Ω are invariant to time and frequency [6] shifts.

$$\Delta_t \left(b\left(t\right) \right) = \Delta_t \left(b\left(t - t_0\right) \right) \quad, \quad t_0 \in \mathbb{R}$$

$$\Delta_\Omega \left(B\left(\Omega\right) \right) = \Delta_\Omega \left(B\left(\Omega - \Omega_0\right) \right) \quad, \quad \Omega_0 \in \mathbb{R}$$

This implies that all basis elements constructed from time and/or frequency shifts of a prototype waveform $b\left(t\right)$ will inherit the temporal and spectral widths of $b\left(t\right)$.

2. The **time-bandwidth product** $\Delta_t \Delta_\Omega$ is invariant to time-scaling. [7]

$$\Delta_t \left(b\left(at\right) \right) = \frac{1}{|a|} \Delta_t \left(b\left(t\right) \right)$$

$$\Delta_\Omega \left(b\left(at\right) \right) = |a| \Delta_\Omega \left(b\left(t\right) \right)$$

The above two equations imply

$$\Delta_t \Delta_\Omega \left(b\left(at\right) \right) = \Delta_t \Delta_\Omega \left(b\left(t\right) \right) \quad, \quad a \in \mathbb{R}$$

Observe that time-domain expansion (*i.e.*, $|a| < 1$) increases the temporal width but decreases the spectral width, while time-domain contraction (*i.e.*, $|a| > 1$) does the opposite. This suggests that time-scaling might be a useful tool for the design of a basis element with a particular tradeoff between time and frequency resolution. On the other hand, scaling cannot simultaneously increase both time **and** frequency resolution.

3. No waveform can have time-bandwidth product less than $\frac{1}{2}$.

$$\Delta_t \Delta_\Omega \geq \frac{1}{2}$$

This is known as the **time-frequency uncertainty principle**.

4. The Gaussian pulse $g\left(t\right)$ achieves the minimum time-bandwidth product $\Delta_t \Delta_\Omega = \frac{1}{2}$.

$$g\left(t\right) = \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{1}{2}t^2\right)}$$

$$G\left(\Omega\right) = e^{-\left(\frac{1}{2}\Omega^2\right)}$$

Note that this waveform is neither bandlimited nor time-limited, but reasonable concentrated in both domains (around the points $t_c = 0$ and $\Omega_c = 0$).

Properties 1 and 2 can be easily verified using the definitions above. Properties 3 and 4 follow from the Cauchy-Schwarz inequality[8].

Since the Gaussian pulse $g\left(t\right)$ achieves the minimum time-bandwidth product, it makes for a theoretically good prototype waveform. In other words, we might consider constructing a basis from time shifted, frequency shifted, time scaled, or frequency scaled versions of $g\left(t\right)$ to give a range of spectral/temporal centers and spectral/temporal resolutions. Since the Gaussian pulse has doubly-infinite time-support, though, other windows are used in practice. Basis construction from a prototype waveform is the main concept behind Short-Time Fourier Analysis (Section 3.1.4) and the continuous Wavelet transform (Section 3.1.5) discussed later.

---

[6] Keep in mind the fact that $b\left(t\right)$ and $B\left(\Omega\right) = \int_{-\infty}^{\infty} b\left(t\right) e^{-\left(j\Omega t\right)} dt$ are alternate descriptions of the same waveform; we could have written $\Delta_\Omega \left(b\left(t\right) e^{j\Omega_0 t} \right)$ in place of $\Delta_\Omega \left(B\left(\Omega - \Omega_0\right) \right)$ above.

[7] The invariance property holds also for frequency scaling, as implied by the Fourier transform property $b\left(at\right) \Leftrightarrow \frac{1}{|a|} B\left(\frac{\Omega}{a}\right)$.

[8] "Cauchy-Schwarz Inequality" $<$http://cnx.org/content/m10757/latest/$>$

3.1.4 Short-time Fourier Transform[9]

We saw earlier that Fourier analysis is not well suited to describing local changes in "frequency content" because the frequency components defined by the Fourier transform have infinite (*i.e.*, global) time support. For example, if we have a signal with periodic components plus a glitch at time t_0, we might want accurate knowledge of both the periodic component frequencies **and** the glitch time (Figure 3.4).
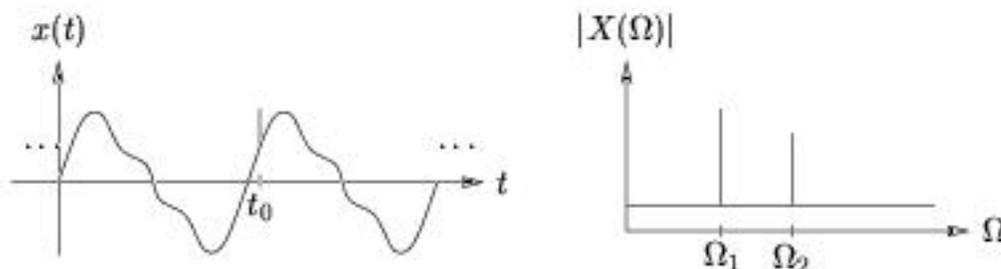
Figure 3.4

The Short-Time Fourier Transform (STFT) provides a means of joint time-frequency analysis. The STFT pair can be written

$$X_{\text{STFT}}(\Omega, \tau) = \int_{-\infty}^{\infty} x(t) w(t - \tau) e^{-(j\Omega t)} dt$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X_{\text{STFT}}(\Omega, \tau) w(t - \tau) e^{j\Omega t} d\Omega dt$$

assuming real-valued $w(t)$ for which $\int (|w(t)|)^2 dt = 1$. The STFT can be interpreted as a "sliding window CTFT": to calculate $X_{\text{STFT}}(\Omega, \tau)$, slide the center of window $w(t)$ to time τ, window the input signal, and compute the CTFT of the result (Figure 3.5 ("Sliding Window CTFT")).

---

[9]This content is available online at <http://cnx.org/content/m10417/2.14/>.

"Sliding Window CTFT"
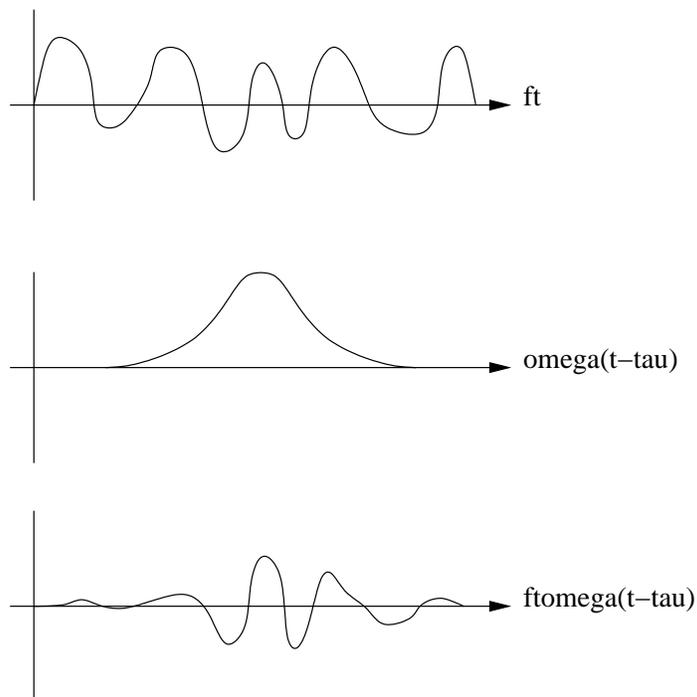
Figure 3.5

The idea is to isolate the signal in the vicinity of time τ, then perform a CTFT analysis in order to estimate the "local" frequency content at time τ.

Essentially, the STFT uses the basis elements

$$b_{\Omega,\tau}\left(t\right) = w\left(t - \tau\right) e^{j\Omega t}$$

over the range $t \in (-\infty, \infty)$ and $\Omega \in (-\infty, \infty)$. This can be understood as time and frequency shifts of the window function $w\left(t\right)$. The STFT basis is often illustrated by a tiling of the time-frequency plane, where each tile represents a particular basis element (Figure 3.6):
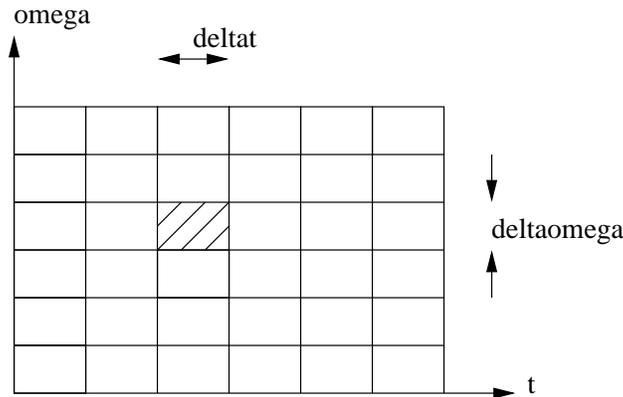
Figure 3.6

The height and width of a tile represent the spectral and temporal widths of the basis element, respectively, and the position of a tile represents the spectral and temporal centers of the basis element. Note that, while the tiling diagram (Figure 3.6) suggests that the STFT uses a discrete set of time/frequency shifts, the STFT basis is really constructed from a continuum of time/frequency shifts.

Note that we can decrease spectral width Δ_Ω at the cost of increased temporal width Δ_t by stretching basis waveforms in time, although the time-bandwidth product $\Delta_t \Delta_\Omega$ (*i.e.*, the area of each tile) will remain constant (Figure 3.7).
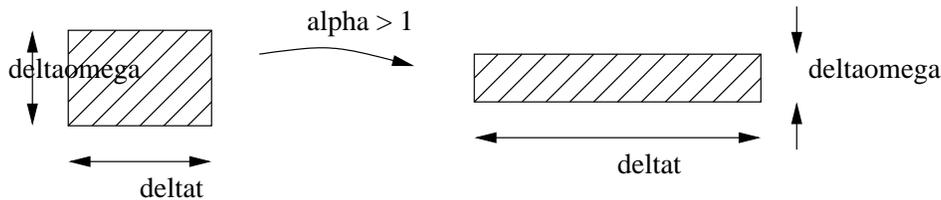
Figure 3.7

Our observations can be summarized as follows:

- the time resolutions and frequency resolutions of every STFT basis element will equal those of the window $w(t)$. (All STFT tiles have the same shape.)
- the use of a wide window will give good frequency resolution but poor time resolution, while the use of a narrow window will give good time resolution but poor frequency resolution. (When tiles are stretched in one direction they shrink in the other.)
- The combined time-frequency resolution of the basis, proportional to $\frac{1}{\Delta_t \Delta_\Omega}$, is determined not by window width but by window shape. Of all shapes, the Gaussian [10] $w(t) = \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{1}{2}t^2\right)}$ gives the

---

[10] The STFT using a Gaussian window is known as the **Gabor Transform** (1946).

highest time-frequency resolution, although its infinite time-support makes it impossible to implement. (The Gaussian window results in tiles with minimum area.)

Finally, it is interesting to note that the STFT implies a particular definition of **instantaneous frequency**. Consider the linear chirp $x(t) = \sin(\Omega_0 t^2)$. From casual observation, we might expect an instantaneous frequency of $\Omega_0 \tau$ at time τ since

$$\sin(\Omega_0 t^2) = \sin(\Omega_0 \tau t) \quad , \quad t = \tau$$

The STFT, however, will indicate a time-τ instantaneous frequency of

$$\frac{d}{dt}\left(\Omega_0 t^2\right)|_{t=\tau} = 2\Omega_0 \tau$$

CAUTION: The phase-derivative interpretation of instantaneous frequency only makes sense for signals containing exactly **one** sinusoid, though! In summary, always remember that the traditional notion of "frequency" applies only to the CTFT; we must be very careful when bending the notion to include, *e.g.*, "instantaneous frequency", as the results may be unexpected!

3.1.5 Continuous Wavelet Transform[11]

The STFT provided a means of (joint) time-frequency analysis with the property that spectral/temporal widths (or resolutions) were the same for all basis elements. Let's now take a closer look at the implications of uniform resolution.

Consider two signals composed of sinusoids with frequency 1 Hz and 1.001 Hz, respectively. It may be difficult to distinguish between these two signals in the presence of background noise unless many cycles are observed, implying the need for a many-second observation. Now consider two signals with pure frequencies of 1000 Hz and 1001 Hz-again, a 0.1% difference. Here it should be possible to distinguish the two signals in an interval of much less than one second. In other words, good frequency resolution requires longer observation times as frequency decreases. Thus, it might be more convenient to construct a basis whose elements have larger temporal width at low frequencies.

The previous example motivates a multi-resolution time-frequency tiling of the form (Figure 3.8):

---

[11]This content is available online at <http://cnx.org/content/m10418/2.14/>.
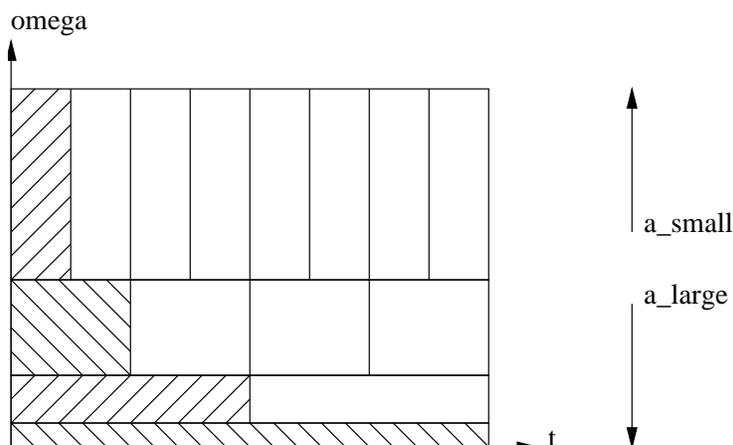
Figure 3.8

The Continuous Wavelet Transform (CWT) accomplishes the above multi-resolution tiling by time-scaling and time-shifting a prototype function $\psi(t)$, often called the **mother wavelet**. The a-scaled and τ-shifted basis elements is given by

$$\psi_{a,\tau}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-\tau}{a}\right)$$

where

$$a \ \ \tau \in \mathbb{R}$$

$$\int_{-\infty}^{\infty} \psi(t)\,dt = 0$$

$$C_\psi = \int_{-\infty}^{\infty} \frac{\left(|\psi(\Omega)|\right)^2}{|\Omega|} d\Omega < \infty$$

The conditions above imply that $\psi(t)$ is bandpass and sufficiently smooth. Assuming that $\| \psi(t) \| = 1$, the definition above ensures that $\| \psi_{a,\tau}(t) \| = 1$ for all a and τ. The CWT is then defined by the transform pair

$$X_{\text{CWT}}(a,\tau) = \int_{-\infty}^{\infty} x(t)\,\psi_{a,\tau}(t)^* dt$$

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{X_{\text{CWT}}(a,\tau)\,\psi_{a,\tau}(t)}{a^2} d\tau da$$

In basis terms, the CWT says that a waveform can be decomposed into a collection of shifted and stretched versions of the mother wavelet $\psi(t)$. As such, it is usually said that wavelets perform a "time-scale" analysis rather than a time-frequency analysis.

The **Morlet wavelet** is a classic example of the CWT. It employs a windowed complex exponential as the mother wavelet:

$$\psi(t) = \frac{1}{\sqrt{2\pi}} e^{-(j\Omega_0 t)} e^{-\frac{t^2}{2}}$$

$$\Psi(\Omega) = e^{-\frac{(\Omega-\Omega_0)^2}{2}}$$

where it is typical to select $\Omega_0 = \pi\sqrt{\frac{2}{\log 2}}$. (See illustration (Figure 3.9).) While this wavelet does not exactly satisfy the conditions established earlier, since $\Psi(0) \simeq 7 \times 10^{-7} \neq 0$, it can be corrected, though in practice the correction is negligible and usually ignored.
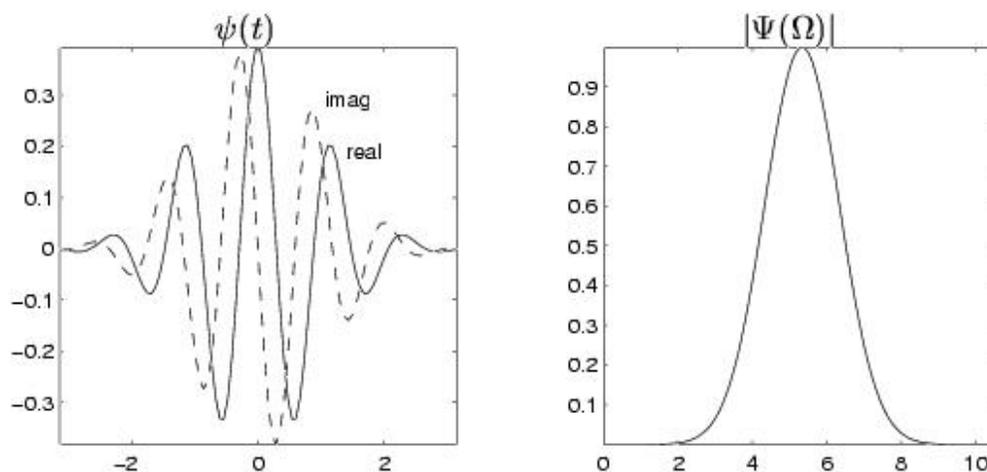
Figure 3.9

While the CWT discussed above is an interesting theoretical and pedagogical tool, the discrete wavelet transform (DWT) is much more practical. Before shifting our focus to the DWT, we take a step back and review some of the basic concepts from the branch of mathematics known as Hilbert Space theory (Vector Space (Section 3.2.2), Normed Vector Space (Section 3.2.3), Inner Product Space (Section 3.2.4), Hilbert Space (Section 3.2.5), Projection Theorem[12]). These concepts will be essential in our development of the DWT.

3.2 Hilbert Space Theory

3.2.1 Hilbert Space Theory[13]

Hilbert spaces provide the mathematical foundation for signal processing theory. In this section we attempt to clearly define some key Hilbert space concepts like vectors (Section 3.2.2), norms (Section 3.2.3), inner products (Section 3.2.4), subspaces (Section 3.2.5), orthogonality (Section 3.2.5), orthonormal bases (Section 3.2.5), and projections (Section 3.2.5). The intent is not to bury you in mathematics, but to familiarize you with the terminology, provide intuition, and leave you with a "lookup table" for future reference.

3.2.2 Vector Space[14]

- A **vector space** consists of the following four elements:

---

[12]"Projection Theorem" <http://cnx.org/content/m10435/latest/>
[13]This content is available online at <http://cnx.org/content/m11007/2.1/>.
[14]This content is available online at <http://cnx.org/content/m10419/2.13/>.

a. A set of vectors V,

b. A field of scalars $\mathbb{F}$ (where, for our purposes, $\mathbb{F}$ is either $\mathbb{R}$ or $\mathbb{C}$),

c. The operations of vector addition "+" (*i.e.*, $+ : V \times V \to V$)

d. The operation of scalar multiplication "·"(*i.e.*, $\cdot : \mathbb{F} \times V \to V$)

for which the following properties hold. (Assume $\bar{x}$ $\bar{y}$ $\bar{z} \in V$ and α $\beta \in \mathbb{F}$.)

Properties	Examples
commutativity	$\bar{x} + \bar{y} = \bar{y} + \bar{x}$
associativity	$\left(\bar{x} + \bar{y} \right) + \bar{z} = \bar{x} + \left(\bar{y} + \bar{z} \right)$
	$(\alpha\beta)\,\bar{x} = \alpha \left(\beta\,\bar{x} \right)$
distributivity	$\alpha \cdot \left(\bar{x} + \bar{y} \right) = \left(\alpha \cdot \bar{x} \right) + \left(\alpha \cdot \bar{y} \right)$
	$(\alpha + \beta)\,\bar{x} = \alpha\,\bar{x} + \beta\,\bar{x}$
additive identity	$\exists 0, 0 \in V : \left(\bar{x} + 0 = \bar{x} \right)$, $\bar{x} \in V$
additive inverse	$\exists -\mathrm{x}-, \left(-\bar{x} \right) \in V : \left(\bar{x} + -\bar{x} = 0 \right)$, $\bar{x} \in V$
multiplicative identity	$1 \cdot \bar{x} = \bar{x}$, $\bar{x} \in V$

Table 3.1

Important examples of vector spaces include

Properties	Examples
real N-vectors	$V = \mathbb{R}^N,\ \mathbb{F} = \mathbb{R}$
complex N-vectors	$V = \mathbb{C}^N,\ \mathbb{F} = \mathbb{C}$
sequences in "l_p"	$V = \left\{ x\,[n] \mid \exists n \in \mathbb{Z} : \left(\sum_{n=-\infty}^{\infty} \left(|x\,[n]| \right)^p < \infty \right) \right\},\ \mathbb{F} = \mathbb{C}$
functions in "$\mathcal{L}_p$"	$V = \left\{ f\,(t) \mid \int_{-\infty}^{\infty} \left(|f\,(t)| \right)^p dt < \infty \right\},\ \mathbb{F} = \mathbb{C}$

Table 3.2

where we have assumed the usual definitions of addition and multiplication. From now on, we will denote the arbitrary vector space $(V, \mathbb{F}, +, \cdot)$ by the shorthand V and assume the usual selection of $(\mathbb{F}, +, \cdot)$. We will also suppress the "·" in scalar multiplication, so that $\alpha \cdot x$ becomes αx.

- A **subspace** of V is a subset $M \subset V$ for which

  a. $\left(\bar{x} + \bar{y} \right) \in M$, $\bar{x} \in M$ $\bar{y} \in M$

  b. $\alpha\,\bar{x} \in M$, $\bar{x} \in M$ $\alpha \in \mathbb{F}$

  NOTE: Note that every subspace must contain 0, and that V is a subspace of itself.

- The **span** of set $S \subset V$ is the subspace of V containing all linear combinations of vectors in S. When $S = \left\{ \bar{x}_0, \ldots, \bar{x}_{N-1} \right\}$,

$$\mathrm{span}\,(S) := \left\{ \sum_{i=0}^{N-1} \alpha_i x_i \mid \alpha_i \in \mathbb{F} \right\}$$

- A subset of linearly-independent vectors $\left\{ \overline{x}_0, \ldots, \overline{x}_{N-1} \right\} \subset V$ is called a **basis for V** when its span equals V. In such a case, we say that V has **dimension** N. We say that V is **infinite-dimensional** [15] if it contains an infinite number of linearly independent vectors.

- V is a **direct sum** of two subspaces M and N, written $V = M \oplus N$, iff every $\overline{x} \in V$ has a **unique** representation $\overline{x} = \overline{m} + \overline{n}$ for $\overline{m} \in M$ and $\overline{n} \in N$.

  NOTE: Note that this requires $M \cap N = \{0\}$

3.2.3 Normed Vector Space[16]

Now we equip a vector space V with a notion of "size".

- A **norm** is a function ($\| \cdot \| : V \rightarrow \mathbb{R}$) such that the following properties hold (, $\overline{x} \in V$ $\overline{y} \in V$ and , $\alpha \in \mathbb{F}$):

  a. $\| \overline{x} \| \geq 0$ with equality iff $\overline{x} = 0$

  b. $\| \alpha \overline{x} \| = |\alpha| \cdot \| \overline{x} \|$

  c. $\| \overline{x} + \overline{y} \| \leq \| \overline{x} \| + \| \overline{y} \|$, (the **triangle inequality**).

  In simple terms, the norm measures the size of a vector. Adding the norm operation to a vector space yields a **normed vector space**. Important example include:

  a. $V = \mathbb{R}^N$, $\| (x_0, \ldots, x_{N-1})^T \| := \sqrt{\sum_{i=0}^{N-1} x_i^2} := \sqrt{\overline{x}^T \overline{x}}$

  b. $V = \mathbb{C}^N$, $\| (x_0, \ldots, x_{N-1})^T \| := \sqrt{\sum_{i=0}^{N-1} (|x_i|)^2} := \sqrt{\overline{x}^H \overline{x}}$

  c. $V = l_p$, $\| \{x[n]\} \| := \left(\sum_{n=-\infty}^{\infty} (|x[n]|)^p \right)^{\frac{1}{p}}$

  d. $V = \mathcal{L}_p$, $\| f(t) \| := \left(\int_{-\infty}^{\infty} (|f(t)|)^p dt \right)^{\frac{1}{p}}$

3.2.4 Inner Product Space[17]

Next we equip a normed vector space V with a notion of "direction".

- An **inner product** is a function ($(< \cdot, \cdot >: V \times V) \rightarrow \mathbb{C}$) such that the following properties hold (, $\overline{x} \in V$ $\overline{y} \in V$ $\overline{z} \in V$ and , $\alpha \in \mathbb{F}$):

  a. $< \overline{x}, \overline{y} > = < \left(\overline{y}, \overline{x} \right) >^*$

  b. $< \overline{x}, \alpha \overline{y} > = \alpha < \left(\overline{x}, \overline{y} \right) >$ ...implying that $< \alpha \overline{x}, \overline{y} > = \alpha^* < \left(\overline{x}, \overline{y} \right) >$

  c. $< \overline{x}, \overline{y} + \overline{z} > = < \overline{x}, \overline{y} > + < \overline{x}, \overline{z} >$

  d. $< \overline{x}, \overline{x} > \geq 0$ with equality iff $\overline{x} = 0$

  In simple terms, the inner product measures the relative alignment between two vectors. Adding an inner product operation to a vector space yields an **inner product space**. Important examples include:

---

[15] The definition of an infinite-dimensional basis would be complicated by issues relating to the convergence of infinite series. Hence we postpone discussion of infinite-dimensional bases until the Hilbert Space (Section 3.2.5) section.

[16] This content is available online at <http://cnx.org/content/m10428/2.14/>.

[17] This content is available online at <http://cnx.org/content/m10430/2.13/>.

a. $V = \mathbb{R}^N$, $<\bar{x}, \bar{y}> := \bar{x}^T \bar{y}$

b. $V = \mathbb{C}^N$, $<\bar{x}, \bar{y}> := \bar{x}^H \bar{y}$

c. $V = l_2$, $< \{x[n]\}, \{y[n]\} > := \sum_{n=-\infty}^{\infty} x[n]^* y[n]$

d. $V = \mathcal{L}_2$, $< f(t), g(t) > := \int_{-\infty}^{\infty} f(t)^* g(t)\, dt$

The inner products above are the "usual" choices for those spaces.

The inner product naturally defines a norm:

$$\|\bar{x}\| := \sqrt{<\bar{x}, \bar{x}>}$$

though not every norm can be defined from an inner product. [18]   Thus, an inner product space can be considered as a normed vector space with additional structure. Assume, from now on, that we adopt the inner-product norm when given a choice.

- The **Cauchy-Schwarz inequality** says

$$| <\bar{x}, \bar{y}> | \leq \|\bar{x}\| \|\bar{y}\|$$

with equality iff $\exists \alpha \in \mathbb{F} : \left(\bar{x} = \alpha\, \bar{y} \right)$.

When $< \left(\bar{x}, \bar{y} \right) > \in \mathbb{R}$, the inner product can be used to define an "angle" between vectors:

$$\cos(\theta) = \frac{<\bar{x}, \bar{y}>}{\|\bar{x}\| \|\bar{y}\|}$$

- Vectors $\bar{x}$ and $\bar{y}$ are said to be **orthogonal**, denoted as $\bar{x} \perp \bar{y}$, when $<\bar{x}, \bar{y}> = 0$. The **Pythagorean theorem** says:

$$\left(\|\bar{x} + \bar{y}\| \right)^2 = \left(\|\bar{x}\| \right)^2 + \left(\|\bar{y}\| \right)^2 \quad , \quad \bar{x} \perp \bar{y}$$

  Vectors $\bar{x}$ and $\bar{y}$ are said to be **orthonormal** when $\bar{x} \perp \bar{y}$ and $\|\bar{x}\| = \|\bar{y}\| = 1$.

- $\bar{x} \perp S$ means $\bar{x} \perp \bar{y}$ for all $\bar{y} \in S$. S is an **orthogonal set** if $\bar{x} \perp \bar{y}$ for all $\bar{x}$ $\bar{y} \in S$ s.t. $\bar{x} \neq \bar{y}$. An orthogonal set S is an **orthonormal set** if $\|\bar{x}\| = 1$ for all $\bar{x} \in S$. Some examples of orthonormal sets are

a. $\mathbb{R}^3 : S = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\}$

b. $\mathbb{C}^N$: Subsets of columns from unitary matrices

c. l_2 : Subsets of shifted Kronecker delta functions $S \subset \{ \{\delta[n-k]\} \mid k \in \mathbb{Z} \}$

d. $\mathcal{L}_2 : S = \left\{ \frac{1}{\sqrt{T}} f(t - nT) \mid n \in \mathbb{Z} \right\}$ for unit pulse $f(t) = u(t) - u(t-T)$, unit step $u(t)$

where in each case we assume the usual inner product.

---

[18] An example for inner product space $\mathcal{L}_2$ would be any norm $\|\bar{f}\| := \sqrt[p]{\int_{-\infty}^{\infty} (|f(t)|)^p dt}$ such that $p > 2$.

3.2.5 Hilbert Spaces[19]

Now we consider inner product spaces with nice convergence properties that allow us to define countably-infinite orthonormal bases.

- A **Hilbert space** is a **complete** inner product space. A **complete** [20] space is one where all **Cauchy** sequences **converge** to some vector within the space. For sequence $\{x_n\}$ to be **Cauchy**, the distance between its elements must eventually become arbitrarily small:

$$\exists N \; \epsilon : \left(\parallel \bar{x}_n - \bar{x}_m \parallel < \epsilon \;\; , \;\; (n \geq N_\epsilon) \;\; (m \geq N_\epsilon) \;\; \right) \;\; , \;\; \epsilon > 0$$

  For a sequence $\{x_n\}$ to be **convergent to x**, the distance between its elements and $\bar{x}$ must eventually become arbitrarily small:

$$\exists N \; \epsilon : \left(\parallel \bar{x}_n - x \parallel < \bar{\epsilon} \;\; , \;\; n \geq N_\epsilon \;\; \right) \;\; , \;\; \epsilon > 0$$

  Examples are listed below (assuming the usual inner products):

  a. $V = \mathbb{R}^N$
  b. $V = \mathbb{C}^N$
  c. $V = l_2$ (*i.e.*, square summable sequences)
  d. $V = \mathcal{L}_2$ (*i.e.*, square integrable functions)

- We will always deal with **separable** Hilbert spaces, which are those that have a countable [21] orthonormal (ON) basis. A countable **orthonormal basis** for V is a countable orthonormal set $S = \left\{ \bar{x}_k \right\}$ such that every vector in V can be represented as a linear combination of elements in S:

$$\exists \alpha \; k : \left(\bar{y} = \sum_k \alpha_k \bar{x}_k \right) \;\; , \;\; \bar{y} \in V$$

  Due to the orthonormality of S, the basis coefficients are given by

$$\alpha_k = < \bar{x}_k, \bar{y} >$$

  We can see this via:

$$< \bar{x}_k, \bar{y} > = < \bar{x}_k, \lim_{n \to \infty} \sum_{i=0}^{n} \alpha_i \bar{x}_i > = \lim_{n \to \infty} < \bar{x}_k, \sum_{i=0}^{n} \alpha_i \bar{x}_i > = \lim_{n \to \infty} \sum_{i=0}^{n} \alpha_i < \left(\bar{x}_k, \bar{x}_i \right) > = \alpha_k$$

  where $\delta[k - i] = < \bar{x}_k, \bar{x}_i >$ (where the second equality invokes the continuity of the inner product). In finite n-dimensional spaces (*e.g.*, $\mathbb{R}^n$ or $\mathbb{C}^n$), any n-element ON set constitutes an ON basis. In infinite-dimensional spaces, we have the following **equivalences**:

  a. $\left\{ \bar{x}_0, \bar{x}_1, \bar{x}_2, \dots \right\}$ is an ON basis

  b. If $< \bar{x}_i, \bar{y} > = 0$ for all i, then $\bar{y} = 0$

  c. $\left(\parallel \bar{y} \parallel \right)^2 = \sum_i \left(| < \bar{x}_i, \bar{y} > | \right)^2 \;\; , \;\; \bar{y} \in V$ (Parseval's theorem)

---

[19]This content is available online at $<$http://cnx.org/content/m10434/2.11/$>$.

[20]The rational numbers provide an example of an incomplete set. We know that it is possible to construct a sequence of rational numbers which approximate an irrational number arbitrarily closely. It is easy to see that such a sequence will be Cauchy. However, the sequence will not converge to any **rational** number, and so the rationals cannot be complete.

[21]A countable set is a set with at most a countably-infinite number of elements. Finite sets are countable, as are any sets whose elements can be organized into an infinite list. Continuums (*e.g.*, intervals of $\mathbb{R}$) are uncountably infinite.

d. Every $\bar{y} \in V$ is a limit of a sequence of vectors in span $\left(\left\{ \bar{x}_0, \bar{x}_1, \bar{x}_2, \ldots \right\} \right)$

Examples of countable ON bases for various Hilbert spaces include:

a. $\mathbb{R}^n$: $\left\{ \bar{e}_0, \ldots, \bar{e}_{N-1} \right\}$ for $\bar{e}_i = \left(\begin{array}{ccccccc} 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \end{array} \right)^T$ with "1" in the ith position

b. $\mathbb{C}^n$: same as $\mathbb{R}^n$

c. l_2: $\left\{ \left\{ \delta_i \left[n \right] \right\} \mid i \in \mathbb{Z} \right\}$, for $\left\{ \delta_i \left[n \right] \right\} := \left\{ \delta \left[n - i \right] \right\}$ (all shifts of the Kronecker sequence)

d. $\mathcal{L}_2$: to be constructed using wavelets ...

- Say S is a subspace of Hilbert space V. The **orthogonal complement of S in V**, denoted $S^\perp$, is the subspace defined by the set $\left\{ \bar{x} \in V \mid \bar{x} \perp S \right\}$. When S is closed, we can write $V = S \oplus S^\perp$

- The **orthogonal projection of y onto S**, where S is a closed subspace of V, is

$$\hat{y} = \sum_{ii} < \left(\bar{x}_i, \bar{y} \right) > \bar{x}_i$$

s.t. $\left\{ \bar{x}_i \right\}$ is an ON basis for S. Orthogonal projection yields the best approximation of $\bar{y}$ in S:

$$\hat{y} = \underset{\bar{x} \in S}{argmin} \left\| \bar{y} - \bar{x} \right\|$$

The approximation error $\bar{e} := \bar{y} - \hat{y}$ obeys the **orthogonality principle**:

$$\bar{e} \perp S$$

We illustrate this concept using $V = \mathbb{R}^3$ (Figure 3.10) but stress that the same geometrical interpretation applies to any Hilbert space.
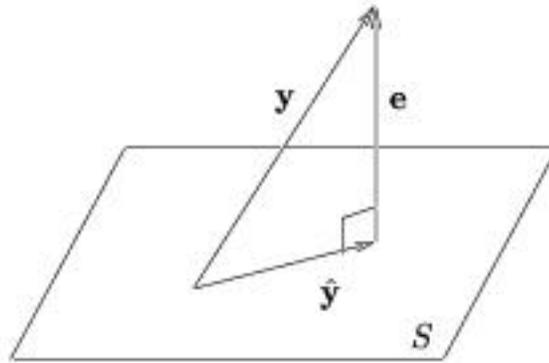
Figure 3.10

A proof of the orthogonality principle is:

$$\bar{e} \perp S \Leftrightarrow < \bar{e}, \bar{x}_i > = 0$$

$$< \overline{y} - \hat{y}, \overline{x}_i > = 0$$

$$
\begin{aligned}
< \overline{y}, \overline{x}_i > \quad &= \quad < \hat{y}, \overline{x}_i > \\
&= \quad < \sum_{jj} < \left(\overline{x}_j, \overline{y} \right) > \overline{x}_j, \overline{x}_i > \\
&= \quad \sum_{jj} < \left(\overline{x}_j, \overline{y} \right) >^* < \left(\overline{x}_j, \overline{x}_i \right) > \\
&= \quad \sum_{jj} < \left(\overline{y}, \overline{x}_j \right) > \delta_{i-j} \\
&= \quad < \overline{y}, \overline{x}_i >
\end{aligned}
\tag{3.1}
$$

3.3 Discrete Wavelet Transform

3.3.1 Discrete Wavelet Transform: Main Concepts[22]

3.3.1.1 Main Concepts

The **discrete wavelet transform** (DWT) is a representation of a signal $x(t) \in \mathcal{L}_2$ using an orthonormal basis consisting of a countably-infinite set of **wavelets**. Denoting the wavelet basis as $\{ \psi_{k,n}(t) \mid k \in \mathbb{Z} \ \ n \in \mathbb{Z} \}$, the DWT transform pair is

$$x(t) = \sum_{k=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} d_{k,n} \psi_{k,n}(t) \tag{3.2}$$

$$
\begin{aligned}
d_{k,n} \quad &= \quad < \psi_{k,n}(t), x(t) > \\
&= \quad \int_{-\infty}^{\infty} \psi_{k,n}(t)^* x(t) \, dt
\end{aligned}
\tag{3.3}
$$

where $\{ d_{k,n} \}$ are the wavelet coefficients. Note the relationship to Fourier series and to the sampling theorem: in both cases we can perfectly describe a continuous-time signal $x(t)$ using a countably-infinite (*i.e.*, discrete) set of coefficients. Specifically, Fourier series enabled us to describe **periodic** signals using Fourier coefficients $\{ X[k] \mid k \in \mathbb{Z} \}$, while the sampling theorem enabled us to describe **bandlimited** signals using signal samples $\{ x[n] \mid n \in \mathbb{Z} \}$. In both cases, signals within a limited class are represented using a coefficient set with a single countable index. The DWT can describe **any** signal in $\mathcal{L}_2$ using a coefficient set parameterized by two countable indices: $\{ d_{k,n} \mid k \in \mathbb{Z} \ \ n \in \mathbb{Z} \}$.

Wavelets are orthonormal functions in $\mathcal{L}_2$ obtained by shifting and stretching a **mother wavelet**, $\psi(t) \in \mathcal{L}_2$. For example,

$$\psi_{k,n}(t) = 2^{-\frac{k}{2}} \psi \left(2^{-k} t - n \right) \quad , \quad k \ \ n \in \mathbb{Z} \tag{3.4}$$

defines a family of wavelets $\{ \psi_{k,n}(t) \mid k \in \mathbb{Z} \ \ n \in \mathbb{Z} \}$ related by power-of-two stretches. As k increases, the wavelet stretches by a factor of two; as n increases, the wavelet shifts right.

NOTE: When $\| \psi(t) \| = 1$, the normalization ensures that $\| \psi_{k,n}(t) \| = 1$ for all $k \in \mathbb{Z}$, $n \in \mathbb{Z}$.

Power-of-two stretching is a convenient, though somewhat arbitrary, choice. In our treatment of the discrete wavelet transform, however, we will focus on this choice. Even with power-of two stretches, there are various possibilities for $\psi(t)$, each giving a different flavor of DWT.

Wavelets are constructed so that $\{ \psi_{k,n}(t) \mid n \in \mathbb{Z} \}$ (*i.e.*, the set of all shifted wavelets at fixed scale k), describes a particular level of 'detail' in the signal. As k becomes smaller (*i.e.*, closer to $-\infty$), the wavelets

---

[22]This content is available online at <http://cnx.org/content/m10436/2.12/>.

become more "fine grained" and the level of detail increases. In this way, the DWT can give a **multi-resolution** description of a signal, very useful in analyzing "real-world" signals. Essentially, the DWT gives us a **discrete multi-resolution description of a continuous-time signal in** $\mathcal{L}_2$.

In the modules that follow, these DWT concepts will be developed "from scratch" using Hilbert space principles. To aid the development, we make use of the so-called **scaling function** $\phi\left(t\right) \in \mathcal{L}_2$, which will be used to approximate the signal **up to a particular level of detail**. Like with wavelets, a family of scaling functions can be constructed via shifts and power-of-two stretches

$$\phi_{k,n}\left(t\right) = 2^{-\frac{k}{2}}\phi\left(2^{-k}t - n\right) \quad , \quad k \ \ n \in \mathbb{Z} \tag{3.5}$$

given mother scaling function $\phi\left(t\right)$. The relationships between wavelets and scaling functions will be elaborated upon later via theory (Section 3.3.5) and example (Section 3.3.2).

> NOTE: The inner-product expression for $d_{k,n}$, (3.3) is written for the general complex-valued case. In our treatment of the discrete wavelet transform, however, we will assume real-valued signals and wavelets. For this reason, we omit the complex conjugations in the remainder of our DWT discussions

3.3.2 The Haar System as an Example of DWT[23]

The Haar basis is perhaps the simplest example of a DWT basis, and we will frequently refer to it in our DWT development. Keep in mind, however, that **the Haar basis is only an example**; there are many other ways of constructing a DWT decomposition.

For the Haar case, the mother **scaling function** is defined by (3.6) and Figure 3.11.

$$\phi\left(t\right) = \begin{cases} 1 & \text{if } 0 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$
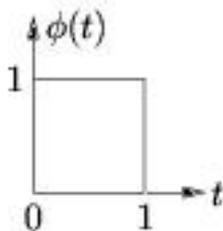
Figure 3.11

From the mother scaling function, we define a family of shifted and stretched scaling functions $\{\phi_{k,n}\left(t\right)\}$ according to (3.7) and Figure 3.12

$$\begin{aligned} \phi_{k,n}\left(t\right) &= 2^{-\frac{k}{2}}\phi\left(2^{-k}t - n\right) \quad , \quad k \in \mathbb{Z} \ \ n \in \mathbb{Z} \\ &= 2^{-\frac{k}{2}}\phi\left(\tfrac{1}{2^k}\left(t - n2^k\right)\right) \end{aligned} \tag{3.7}$$
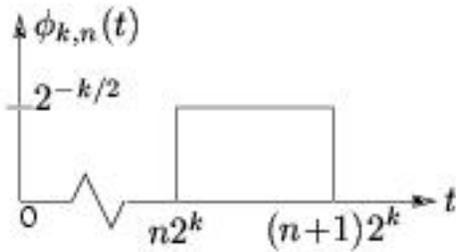
Figure 3.12

which are illustrated in Figure 3.13 for various k and n. (3.7) makes clear the principle that incrementing n by one shifts the pulse one place to the right. Observe from Figure 3.13 that $\{\phi_{k,n}(t) \mid n \in \mathbb{Z}\}$ is orthonormal for each k (*i.e.*, along each row of figures).
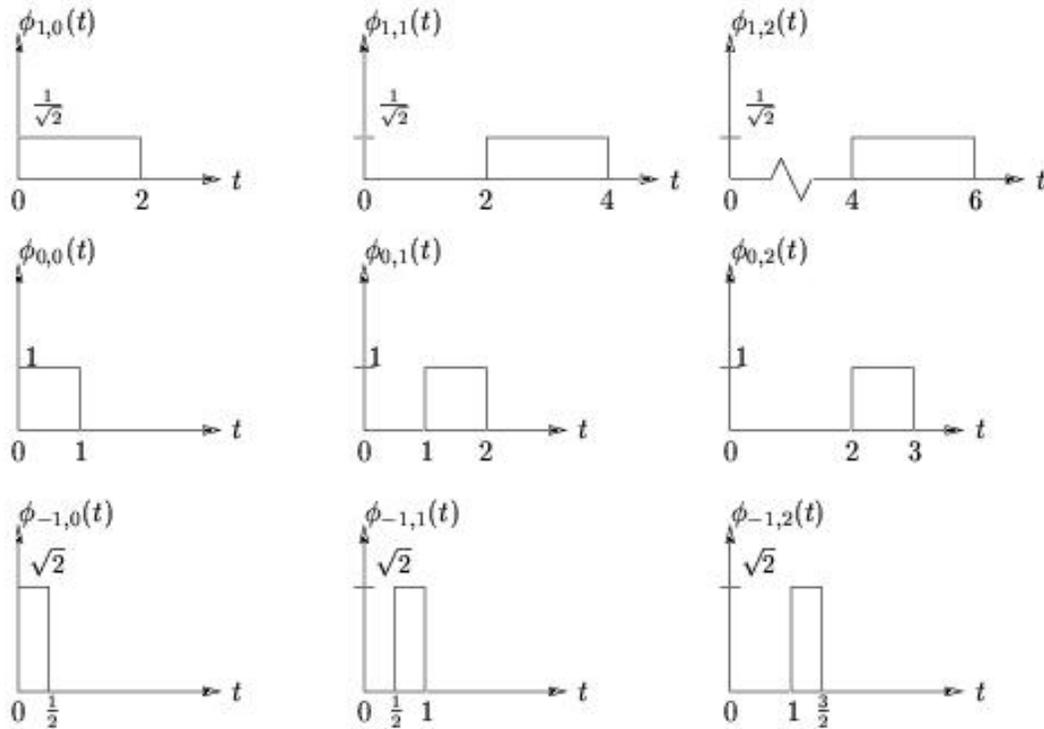
Figure 3.13

3.3.3 A Hierarchy of Detail in the Haar System[24]

Given a mother scaling function $\phi(t) \in \mathcal{L}_2$ — the choice of which will be discussed later — let us construct scaling functions at "coarseness-level-k" and "shift-n" as follows:

$$\phi_{k,n}(t) = 2^{-\frac{k}{2}} \phi\left(2^{-k}t - n\right).$$

Let us then use V_k to denote the subspace defined by linear combinations of scaling functions at the kth level:

$$V_k = \text{span}\left(\{\,\phi_{k,n}(t) \mid n \in \mathbb{Z}\,\}\right).$$

In the Haar system, for example, V_0 and V_1 consist of signals with the characteristics of $x_0(t)$ and $x_1(t)$ illustrated in Figure 3.14.
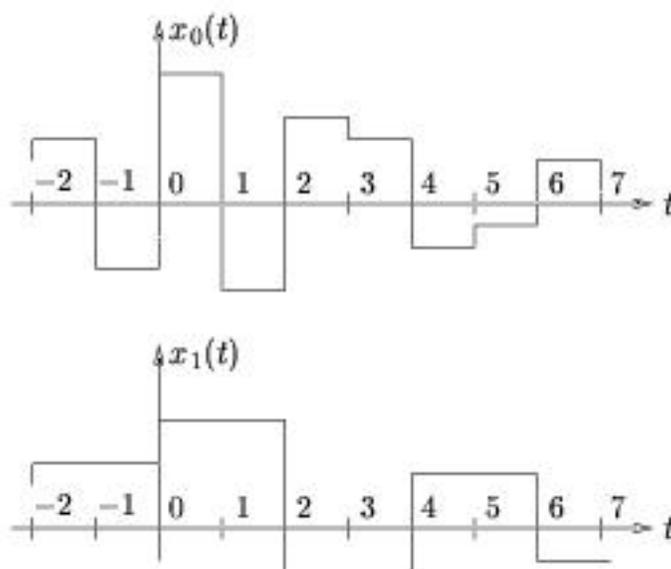
Figure 3.14

We will be careful to choose a scaling function $\phi(t)$ which ensures that the following nesting property is satisfied:

$$\cdots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots$$

$$\text{coarse} \longleftarrow \quad \longrightarrow \text{detailed}$$

In other words, any signal in V_k can be constructed as a linear combination of **more detailed** signals in V_{k-1} . (The Haar system gives proof that at least one such $\phi(t)$ exists.)

The nesting property can be depicted using the set-theoretic diagram, Figure 3.15, where V_{-1} is represented by the contents of the largest egg (which includes the smaller two eggs), V_0 is represented by the contents of the medium-sized egg (which includes the smallest egg), and V_1 is represented by the contents of the smallest egg.

---

[24]This content is available online at <http://cnx.org/content/m11012/2.3/>.
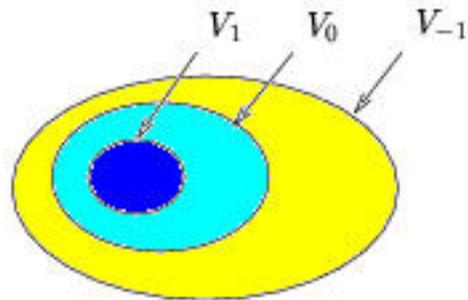
Figure 3.15

Going further, we will assume that $\phi(t)$ is designed to yield the following three important properties:

1. $\{\phi_{k,n}(t) \mid n \in \mathbb{Z}\}$ constitutes an orthonormal basis for V_k,
2. $V_\infty = \{0\}$ (contains no signals). [25]
3. $V_{-\infty} = \mathcal{L}_2$ (contains all signals).

Because $\{\phi_{k,n}(t) \mid n \in \mathbb{Z}\}$ is an orthonormal basis, the best (in $\mathcal{L}_2$ norm) approximation of $x(t) \in \mathcal{L}_2$ at coarseness-level-k is given by the orthogonal projection, Figure 3.16

$$x_k(t) = \sum_{n=-\infty}^{\infty} c_{k,n} \phi_{k,n}(t) \tag{3.8}$$

$$c_{k,n} = \; < \phi_{k,n}(t), x(t) > \tag{3.9}$$

---

[25]While at first glance it might seem that V_∞ should contain non-zero constant signals (e.g., $x(t) = a$ for $a \in \mathbb{R}$), the only constant signal in $\mathcal{L}_2$, the space of square-integrable signals, is the zero signal.
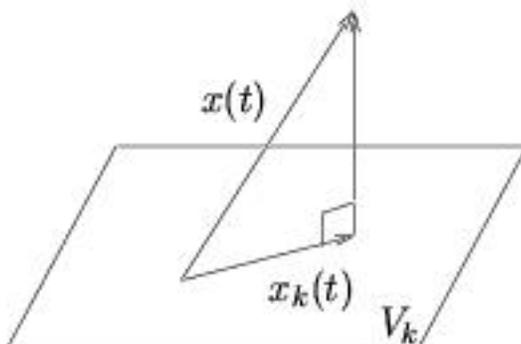
Figure 3.16

We will soon derive conditions on the scaling function $\phi(t)$ which ensure that the properties above are satisfied.

3.3.4 Haar Approximation at the kth Coarseness Level[26]

It is instructive to consider the approximation of signal $x(t) \in \mathcal{L}_2$ at coarseness-level-k of the Haar system. For the Haar case, projection of $x(t) \in \mathcal{L}_2$ onto V_k is accomplished using the basis coefficients

$$
\begin{aligned}
c_{k,n} &= \int_{-\infty}^{\infty} \phi_{k,n}(t)\, x(t)\, dt \\
&= \int_{n2^k}^{(n+1)2^k} 2^{-\frac{k}{2}} x(t)\, dt
\end{aligned}
\tag{3.10}
$$

giving the approximation

$$
\begin{aligned}
x_k(t) &= \sum_{n=-\infty}^{\infty} c_{k,n} \phi_{k,n}(t) \\
&= \sum_{n=-\infty}^{\infty} \int_{n2^k}^{(n+1)2^k} 2^{-\frac{k}{2}} x(t)\, dt\, \phi_{k,n}(t) \\
&= \sum_{n=-\infty}^{\infty} \frac{1}{2^k} \int_{n2^k}^{(n+1)2^k} x(t)\, dt \left(2^{\frac{k}{2}} \phi_{k,n}(t) \right)
\end{aligned}
\tag{3.11}
$$

where

$$
\frac{1}{2^k} \int_{n2^k}^{(n+1)2^k} x(t)\, dt = \text{average value of x(t) in interval}
$$

$$
2^{\frac{k}{2}} \phi_{k,n}(t) = \text{height} = 1
$$

This corresponds to taking the average value of the signal in each interval of width 2^k and approximating the function by a constant over that interval (see Figure 3.17).

---

[26]This content is available online at $<$http://cnx.org/content/m11013/2.2/$>$.
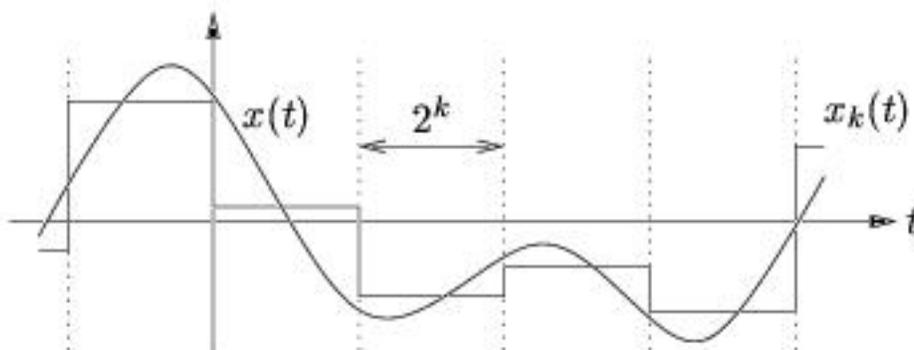
Figure 3.17

3.3.5 The Scaling Equation[27]

Consider the level-1 subspace and its orthonormal basis:

$$V_1 = \text{span}\left(\left\{\,\phi_{1,n}\left(t\right)\mid n \in \mathbb{Z}\,\right\}\right) \tag{3.12}$$

$$\phi_{1,n}\left(t\right) = \frac{1}{\sqrt{2}}\phi\left(\frac{1}{2}t - n\right) \tag{3.13}$$

Since $V_1 \subset V_0$ (i.e., V_0 is more detailed than V_1) and since $\phi_{1,0}\left(t\right) \in V_0$, there must exist coefficients $\left\{\,h\left[n\right]\mid n \in \mathbb{Z}\right\}$ such that

$$\phi_{1,0}\left(t\right) = \sum_{n=-\infty}^{\infty} h\left[n\right]\phi_{0,n}\left(t\right) \tag{3.14}$$

$$\Leftrightarrow \frac{1}{\sqrt{2}}\phi\left(\frac{1}{2}t\right) = \sum_{n=-\infty}^{\infty} h\left[n\right]\phi\left(t-n\right) \tag{3.15}$$

Scaling Equation

$$\phi\left(t\right) = \sqrt{2}\sum_{n=-\infty}^{\infty} h\left[n\right]\phi\left(2t-n\right) \tag{3.16}$$

To be a valid scaling function, $\phi\left(t\right)$ must obey the scaling equation for some coefficient set $\left\{h\left[n\right]\right\}$.

3.3.6 The Wavelet Scaling Equation[28]

The **difference** in detail between V_k and V_{k-1} will be described using W_k , the orthogonal complement of V_k in V_{k-1}:

$$V_{k-1} = V_k \oplus W_k \tag{3.17}$$

---

[27]This content is available online at <http://cnx.org/content/m10476/2.7/>.
[28]This content is available online at <http://cnx.org/content/m11014/2.2/>.

At times it will be convenient to write $W_k = V_k^{\perp}$. This concept is illustrated in the set-theoretic diagram, Figure 3.18.
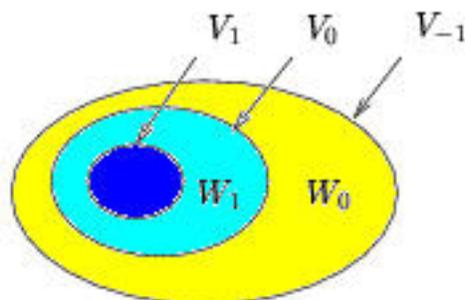
Figure 3.18

Suppose now that, for each $k \in \mathbb{Z}$, we construct an orthonormal basis for W_k and denote it by $\{\psi_{k,n}(t) \mid n \in \mathbb{Z}\}$. It turns out that, because every V_k has a basis constructed from shifts and stretches of a mother scaling function (i.e., $\phi_{k,n}(t) = 2^{-\frac{k}{2}}\phi\left(2^{-k}t - n\right)$, every W_k has a basis that can be constructed from shifts and stretches of a "mother wavelet" $\psi(t) \in \mathcal{L}_2$:

$$\psi_{k,n}(t) = 2^{-\frac{k}{2}}\psi\left(2^{-k}t - n\right).$$

The Haar system will soon provide us with a concrete example .

Let's focus, for the moment, on the specific case $k = 1$. Since $W_1 \subset V_0$, there must exist $\{g[n] \mid n \in \mathbb{Z}\}$ such that:

$$\psi_{1,0}(t) = \sum_{n=-\infty}^{\infty} g[n]\phi_{0,n}(t) \tag{3.18}$$

$$\Leftrightarrow \frac{1}{\sqrt{2}}\psi\left(\frac{1}{2}t\right) = \sum_{n=-\infty}^{\infty} g[n]\phi(t-n)$$

Wavelet Scaling Equation

$$\psi(t) = \sqrt{2}\sum_{n=-\infty}^{\infty} g[n]\phi(2t-n) \tag{3.19}$$

To be a valid scaling-function/wavelet pair, $\phi(t)$ and $\psi(t)$ must obey the wavelet scaling equation for some coefficient set $\{g[n]\}$.

3.3.7 Conditions on h[n] and g[n][29]

Here we derive sufficient conditions on the coefficients used in the scaling equation and wavelet scaling equation that ensure, for every $k \in \mathbb{Z}$, that the sets $\{\phi_{k,n}(t) \mid n \in \mathbb{Z}\}$ and $\{\psi_{k,n}(t) \mid n \in \mathbb{Z}\}$ have the orthonormality properties described in The Scaling Equation (Section 3.3.5) and The Wavelet Scaling Equation (Section 3.3.6).

---

[29]This content is available online at $<$http://cnx.org/content/m11015/2.2/$>$.

For $\{\phi_{k,n}(t) \mid n \in \mathbb{Z}\}$ to be orthonormal at all k, we certainly need orthonormality when $k = 1$. This is equivalent to

$$
\begin{aligned}
\delta[m] &= <\phi_{1,0}(t), \phi_{1,m}(t)> \\
&= <\sum_n h[n]\phi(t-n), \sum_\ell h[\ell]\phi(t-\ell-2m)> \\
&= \sum_n h[n] \sum_\ell h[\ell] < (\phi(t-n), \phi(t-\ell-2m))>
\end{aligned}
\tag{3.20}
$$

where $\delta[n-\ell+2m] = <\phi(t-n), \phi(t-\ell-2m)>$

$$
\delta[m] = \sum_{n=-\infty}^{\infty} h[n]h[n-2m]
\tag{3.21}
$$

There is an interesting frequency-domain interpretation of the previous condition. If we define

$$
\begin{aligned}
p[m] &= h[m] * h[-m] \\
&= \sum_n h[n]h[n-m]
\end{aligned}
\tag{3.22}
$$

then we see that our condition is equivalent to $p[2m] = \delta[m]$. In the z-domain, this yields the pair of conditions

Power-Symmetry Property

$$
P(z) = H(z)H(z^{-1})
\tag{3.23}
$$

$$
1 = 1/2 \sum_{p=0}^{1} P\left(z^{1/2} e^{j\frac{2\pi}{2}p}\right) = 1/2 P\left(z^{1/2}\right) + 1/2 P\left(-z^{1/2}\right)
$$

Putting these together,

$$
2 = H\left(z^{1/2}\right) H\left(z^{-1/2}\right) + H\left(-z^{1/2}\right) H\left(-z^{-1/2}\right)
\tag{3.24}
$$

$$
\Leftrightarrow 2 = H(z)H(z^{-1}) + H(-z)H(-z^{-1})
$$

$$
\Leftrightarrow 2 = \left(|H(e^{j\omega})|\right)^2 + \left(|H\left(e^{j(\pi-\omega)}\right)|\right)^2
$$

where the last property invokes the fact that $h[n] \in \mathbb{R}$ and that real-valued impulse responses yield conjugate-symmetric DTFTs. Thus we find that $h[n]$ are the impulse response coefficients of a power-symmetric filter. Recall that this property was also shared by the analysis filters in an orthogonal perfect-reconstruction FIR filterbank.

Given orthonormality at level $k = 0$, we have now derived a condition on $h[n]$ which is necessary and sufficient for orthonormality at level $k = 1$. Yet the same condition is necessary and sufficient for orthonormality at level $k = 2$:

$$
\begin{aligned}
\delta[m] &= <\phi_{2,0}(t), \phi_{2,m}(t)> \\
&= <\sum_n h[n]\phi_{1,n}(t), \sum_\ell h[\ell]\phi_{1,\ell+2m}(t)> \\
&= \sum_n h[n] \sum_\ell h[\ell] < (\phi_{1,n}(t), \phi_{1,\ell+2m}(t))> \\
&= \sum_{n=-\infty}^{\infty} h[n]h[n-2m]
\end{aligned}
\tag{3.25}
$$

where $\delta[n-\ell+2m] = <\phi_{1,n}(t), \phi_{1,\ell+2m}(t)>$. Using induction, we conclude that the previous condition will be necessary and sufficient for orthonormality of $\{\phi_{k,n}(t) \mid n \in \mathbb{Z}\}$ for all $k \in \mathbb{Z}$.

To find conditions on $\{g[n]\}$ ensuring that the set $\{\psi_{k,n}(t) \mid n \in \mathbb{Z}\}$ is orthonormal at every k, we can repeat the steps above but with $g[n]$ replacing $h[n]$, $\psi_{k,n}(t)$ replacing $\phi_{k,n}(t)$, and the wavelet-scaling equation replacing the scaling equation. This yields

$$\delta[m] = \sum_{n=-\infty}^{\infty} g[n]\, g[n-2m] \tag{3.26}$$

$$\Leftrightarrow 2 = G(z)\, G\left(z^{-1}\right) + G(-z)\, G\left(-z^{-1}\right)$$

Next derive a condition which guarantees that $W_k \perp V_k$, as required by our definition $W_k = V_k^\perp$, for all $k \in \mathbb{Z}$. Note that, for any $k \in \mathbb{Z}$, $W_k \perp V_k$ is guaranteed by $\{\psi_{k,n}(t) \mid n \in \mathbb{Z}\} \perp \{\phi_{k,n}(t) \mid n \in \mathbb{Z}\}$ which is equivalent to

$$
\begin{aligned}
0 &= \ < \psi_{k+1,0}(t), \phi_{k+1,m}(t) > \\
&= \ < \sum_n g[n]\,\phi_{k,n}(t), \sum_\ell h[\ell]\,\phi_{k,\ell+2m}(t) > \\
&= \ \sum_n g[n] \sum_\ell h[\ell] < (\phi_{k,n}(t), \phi_{k,\ell+2m}(t)) > \\
&= \ \sum_n g[n]\, h[n-2m]
\end{aligned}
\tag{3.27}
$$

for all m where $\delta[n-\ell+2m] =< \phi_{k,n}(t), \phi_{k,\ell+2m}(t) >$. In other words, a 2-downsampled version of $g[n] * h[-n]$ must consist only of zeros. This necessary and sufficient condition can be restated in the frequency domain as

$$0 = 1/2 \sum_{p=0}^{1} G\left(z^{1/2} e^{-\left(j\frac{2\pi}{2}p\right)}\right) H\left(z^{-1/2} e^{j\frac{2\pi}{2}p}\right) \tag{3.28}$$

$$\Leftrightarrow 0 = G\left(z^{1/2}\right) H\left(z^{-1/2}\right) + G\left(-z^{1/2}\right) H\left(-z^{-1/2}\right)$$

$$\Leftrightarrow 0 = G(z)\, H\left(z^{-1}\right) + G(-z)\, H\left(-z^{-1}\right)$$

The choice

$$G(z) = \pm\left(z^{-P} H\left((-z)^{-1}\right)\right) \tag{3.29}$$

satisfies our condition, since

$$G(z)\, H\left(z^{-1}\right) + G(-z)\, H\left(-z^{-1}\right) = \pm\left(z^{-P} H\left((-z)^{-1}\right) H\left(z^{-1}\right)\right) \mp z^{-P} H\left(z^{-1}\right) H\left((-z)^{-1}\right) = 0$$

In the time domain, the condition on $G(z)$ and $H(z)$ can be expressed

$$g[n] = \pm\left(-1^n h[P-n]\right) \quad . \tag{3.30}$$

Recall that this property was satisfied by the analysis filters in an orthogonal perfect reconstruction FIR filterbank.

Note that the two conditions

$$G(z) = \pm\left(z^{-P} H\left((-z)^{-1}\right)\right)$$

$$2 = H(z)\, H\left(z^{-1}\right) + H(-z)\, H\left(-z^{-1}\right)$$

are sufficient to ensure that both $\{\phi_{k,n}(t) \mid n \in \mathbb{Z}\}$ and $\{\psi_{k,n}(t) \mid n \in \mathbb{Z}\}$ are orthonormal for all k and that $W_k \perp V_k$ for all k, since they satisfy the condition $2 = G(z)\, G\left(z^{-1}\right) + G(-z)\, G\left(-z^{-1}\right)$ automatically.

3.3.8 Values of g[n] and h[n] for the Haar System[30]

The coefficients $\{h[n]\}$ were originally introduced at describe $\phi_{1,0}(t)$ in terms of the basis for V_0:

$$\phi_{1,0}(t) = \sum_n h[n]\phi_{0,n}(t).$$

From the previous equation we find that

$$
\begin{aligned}
<\phi_{0,m}(t), \phi_{1,0}(t)> & = <\phi_{0,m}(t), \sum_n h[n]\phi_{0,n}(t)> \\
& = \sum_n h[n] < (\phi_{0,m}(t), \phi_{0,n}(t))> \\
& = h[m]
\end{aligned}
\tag{3.31}
$$

where $\delta[n-m] = <\phi_{0,m}(t), \phi_{0,n}(t)>$, which gives a way to calculate the coefficients $\{h[m]\}$ when we know $\phi_{k,n}(t)$.

In the Haar case

$$
\begin{aligned}
h[m] & = \int_{-\infty}^{\infty} \phi_{0,m}(t)\phi_{1,0}(t)\,dt \\
& = \int_m^{m+1} \phi_{1,0}(t)\,dt \\
& = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } m \in \{0,1\} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{3.32}
$$

since $\phi_{1,0}(t) = \frac{1}{\sqrt{2}}$ in the interval $[0,2)$ and zero otherwise. Then choosing $P = 1$ in $g[n] = -1^n h(P-n)$, we find that

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

for the Haar system. From the wavelet scaling equation

$$\psi(t) = \sqrt{2} \sum_n g[n]\phi(2t-n) = \phi(2t) - \phi(2t-1)$$

we can see that the Haar mother wavelet and scaling function look like in Figure 3.19:
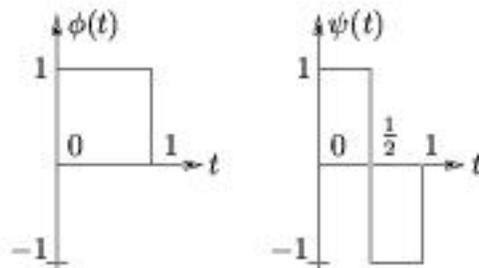
Figure 3.19

It is now easy to see, in the Haar case, how integer shifts of the mother wavelet describe the differences between signals in V_{-1} and V_0 (Figure 3.20):
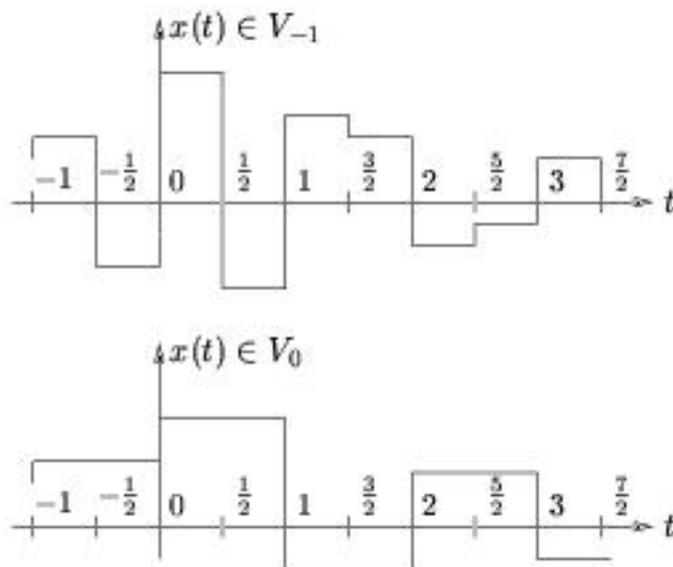
Figure 3.20

We expect this because $V_{-1} = V_0 \oplus W_0$.

3.3.9 Wavelets: A Countable Orthonormal Basis for the Space of Square-Integrable Functions[31]

Recall that $V_k = W_{k+1} \oplus V_{k+1}$ and that $V_{k+1} = W_{k+2} \oplus V_{k+2}$. Putting these together and extending the idea yields

$$
\begin{aligned}
V_k &= W_{k+1} \oplus W_{k+2} \oplus V_{k+2} \\
&= W_{k+1} \oplus W_{k+2} \oplus \cdots \oplus W_\ell \oplus V_\ell \\
&= W_{k+1} \oplus W_{k+2} \oplus W_{k+3} \oplus \ldots \\
&= \overset{\infty}{\underset{i=k+1}{\oplus}} (W_i)
\end{aligned}
\tag{3.33}
$$

If we take the limit as $k \to -\infty$, we find that

$$
\begin{aligned}
\mathcal{L}_2 &= V_{-\infty} \\
&= \overset{\infty}{\underset{i=-\infty}{\oplus}} (W_i)
\end{aligned}
\tag{3.34}
$$

Moreover,

$$
(W_1 \perp V_1) \quad (W_{k \geq 2} \subset V_1) \Rightarrow (W_1 \perp W_{k \geq 2})
\tag{3.35}
$$

---

[31] This content is available online at $<$http://cnx.org/content/m11017/2.2/$>$.

$$(W_2 \perp V_2) \quad (W_{k\geq 3} \subset V_2) \Rightarrow (W_2 \perp W_{k\geq 3}) \tag{3.36}$$

from which it follows that

$$W_k \perp W_{j\neq k} \tag{3.37}$$

or, in other words, all subspaces W_k are orthogonal to one another. Since the functions $\{\psi_{k,n}(t) \mid n \in \mathbb{Z}\}$ form an orthonormal basis for W_k, the results above imply that

$$\{\psi_{k,n}(t) \mid n \ k \in \mathbb{Z}\} \text{ constitutes an orthonormal basis for } \mathcal{L}_2 \tag{3.38}$$

This implies that, for any $f(t) \in \mathcal{L}_2$, we can write

$$f(t) = \sum_{k=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} d_k[m] \psi_{k,m}(t) \tag{3.39}$$

$$d_k[m] = <\psi_{k,m}(t), f(t)> \tag{3.40}$$

This is the key idea behind the orthogonal wavelet system that we have been developing!

3.3.10 Filterbanks Interpretation of the Discrete Wavelet Transform[32]

Assume that we start with a signal $x(t) \in \mathcal{L}_2$. Denote the best approximation at the 0th level of coarseness by $x_0(t)$. (Recall that $x_0(t)$ is the orthogonal projection of $x(t)$ onto V_0.) Our goal, for the moment, is to decompose $x_0(t)$ into scaling coefficients and wavelet coefficients at higher levels. Since $x_0(t) \in V_0$ and $V_0 = V_1 \oplus W_1$, there exist coefficients $\{c_0[n]\}$, $\{c_1[n]\}$, and $\{d_1[n]\}$ such that

$$\begin{aligned} x_0(t) &= \sum_{nn} c_0[n] \phi_{0,n}(t) \\ &= \sum_{nn} c_1[n] \phi_{1,n}[t] + \sum_{nn} d_1[n] \psi_{1,n}[t] \end{aligned} \tag{3.41}$$

Using the fact that $\{\phi_{1,n}(t) \mid n \in \mathbb{Z}\}$ is an orthonormal basis for V_1, in conjunction with the scaling equation,

$$\begin{aligned} c_1[n] &= <x_0(t), \phi_{1,n}(t)> \\ &= <\sum_{mm} c_0[m] \phi_{0,m}(t), \phi_{1,n}(t)> \\ &= \sum_{mm} c_0[m] < (\phi_{0,m}(t), \phi_{1,n}(t))> \\ &= \sum_{mm} c_0[m] < (\phi(t-m), \sum_{\ell\ell} h[\ell] \phi(t-\ell-2n))> \\ &= \sum_{mm} c_0[m] \sum_{\ell\ell} h[\ell] < (\phi(t-m), \phi(t-\ell-2n))> \\ &= \sum_{mm} c_0[m] h[m-2n] \end{aligned} \tag{3.42}$$

where $\delta[t-\ell-2n] = <\phi(t-m), \phi(t-\ell-2n)>$. The previous expression ((3.42)) indicates that $\{c_1[n]\}$ results from convolving $\{c_0[m]\}$ with a time-reversed version of $h[m]$ then downsampling by factor two (Figure 3.21).
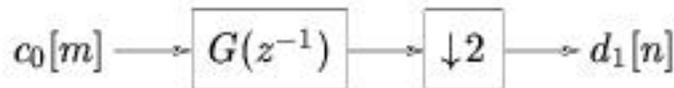
$$c_0[m] \longrightarrow \boxed{G(z^{-1})} \longrightarrow \boxed{\downarrow 2} \longrightarrow d_1[n]$$

Figure 3.21

Using the fact that $\{\psi_{1,n}(t) \mid n \in \mathbb{Z}\}$ is an orthonormal basis for W_1 , in conjunction with the wavelet scaling equation,

$$
\begin{aligned}
d_1[n] &= \; <x_0(t), \psi_{1,n}(t)> \\
&= \; <\textstyle\sum_{mm} c_0[m]\,\phi_{0,m}(t), \psi_{1,n}(t)> \\
&= \; \textstyle\sum_{mm} c_0[m] < (\phi_{0,m}(t), \psi_{1,n}(t))> \\
&= \; \textstyle\sum_{mm} c_0[m] < (\phi(t-m), \sum_{\ell\ell} g[\ell]\,\phi(t-\ell-2n))> \\
&= \; \textstyle\sum_{mm} c_0[m] \sum_{\ell\ell} g[\ell] < (\phi(t-m), \phi(t-\ell-2n))> \\
&= \; \textstyle\sum_{mm} c_0[m]\, g[m-2n]
\end{aligned}
\tag{3.43}
$$

where $\delta[t-\ell-2n] = <\phi(t-m), \phi(t-\ell-2n)>$.

The previous expression ((3.43)) indicates that $\{d_1[n]\}$ results from convolving $\{c_0[m]\}$ with a time-reversed version of $g[m]$ then downsampling by factor two (Figure 3.22).
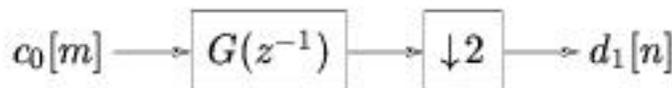
Figure 3.22

Putting these two operations together, we arrive at what looks like the analysis portion of an FIR filterbank (Figure 3.23):
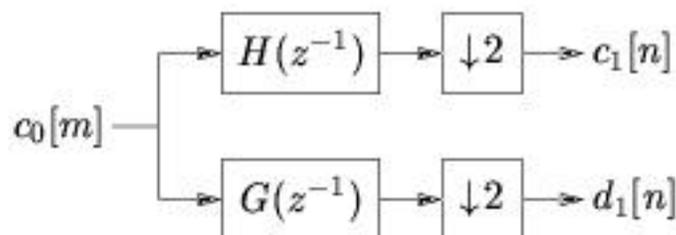
Figure 3.23

We can repeat this process at the next higher level. Since $V_1 = W_2 \oplus V_2$, there exist coefficients $\{c_2[n]\}$ and $\{d_2[n]\}$ such that

$$
\begin{aligned}
x_1(t) &= \; \textstyle\sum_{nn} c_1[n]\,\phi_{1,n}(t) \\
&= \; \textstyle\sum_{nn} d_2[n]\,\psi_{2,n}(t) + \sum_{nn} c_2[n]\,\phi_{2,n}(t)
\end{aligned}
\tag{3.44}
$$

Using the same steps as before we find that

$$c_2 [n] = \sum_{mm} c_1 [m] \, h [m - 2n] \qquad (3.45)$$

$$d_2 [n] = \sum_{mm} c_1 [m] \, g [m - 2n] \qquad (3.46)$$

which gives a cascaded analysis filterbank (Figure 3.24):
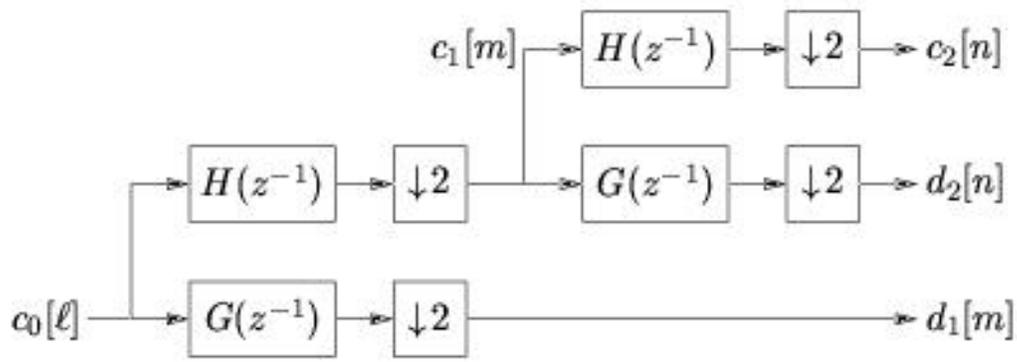
Figure 3.24

If we use $V_0 = W_1 \oplus W_2 \oplus W_3 \oplus \cdots \oplus W_k \oplus V_k$ to repeat this process up to the kth level, we get the iterated analysis filterbank (Figure 3.25).
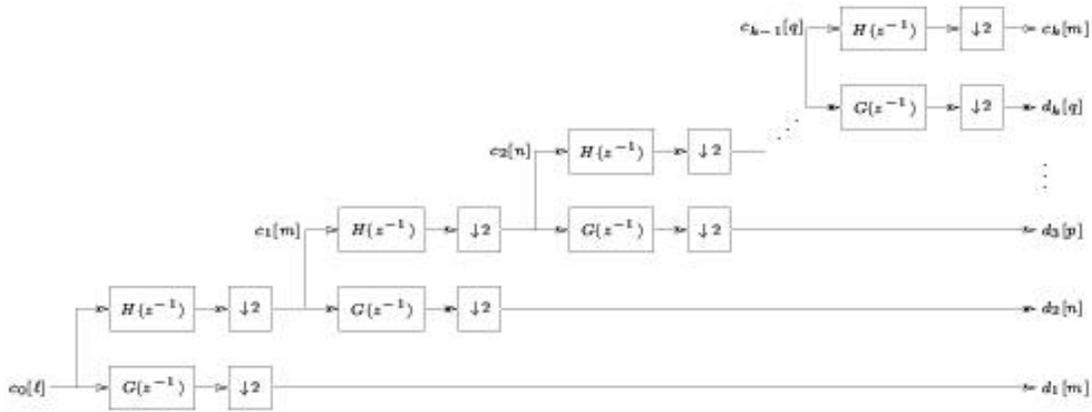
Figure 3.25

As we might expect, signal reconstruction can be accomplished using cascaded two-channel synthesis filterbanks. Using the same assumptions as before, we have:

$$
\begin{aligned}
c_0\left[m\right] &= <x_0\left(t\right),\phi_{0,m}\left(t\right)> \\
&= <\sum_{nn} c_1\left[n\right]\phi_{1,n}\left(t\right)+\sum_{nn} d_1\left[n\right]\psi_{1,n}\left(t\right),\phi_{0,m}\left(t\right)> \\
&= \sum_{nn} c_1\left[n\right]<\left(\phi_{1,n}\left(t\right),\phi_{0,m}\left(t\right)\right)>+\sum_{nn} d_1\left[n\right]<\left(\psi_{1,n}\left(t\right),\phi_{0,m}\left(t\right)\right)> \\
&= \sum_{nn} c_1\left[n\right]h\left[m-2n\right]+\sum_{nn} d_1\left[n\right]g\left[m-2n\right]
\end{aligned}
\tag{3.47}
$$

$$\text{where}\quad h\left[m-2n\right]=<\phi_{1,n}\left(t\right),\phi_{0,m}\left(t\right)>$$

$$\text{and}\quad g\left[m-2n\right]=<\psi_{1,n}\left(t\right),\phi_{0,m}\left(t\right)>$$

which can be implemented using the block diagram in Figure 3.26.
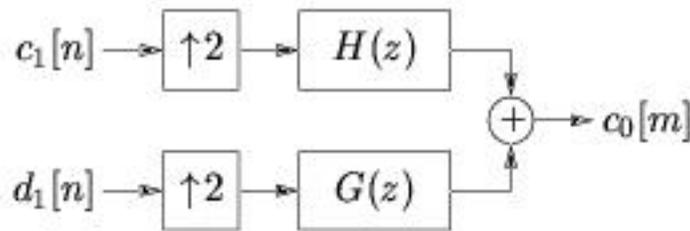
Figure 3.26

The same procedure can be used to derive

$$c_1[m] = \sum_{nn} c_2[n] \, h[m-2n] + \sum_{nn} d_2[n] \, g[m-2n] \qquad (3.48)$$

from which we get the diagram in Figure 3.27.
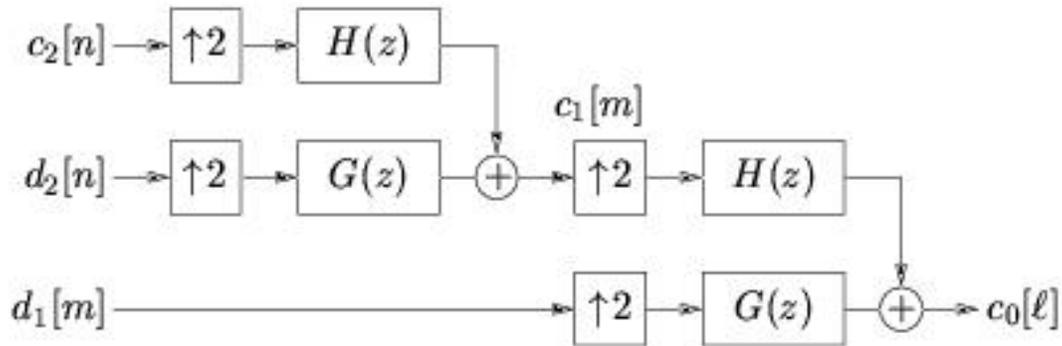
Figure 3.27

To reconstruct from the kth level, we can use the iterated synthesis filterbank (Figure 3.28).
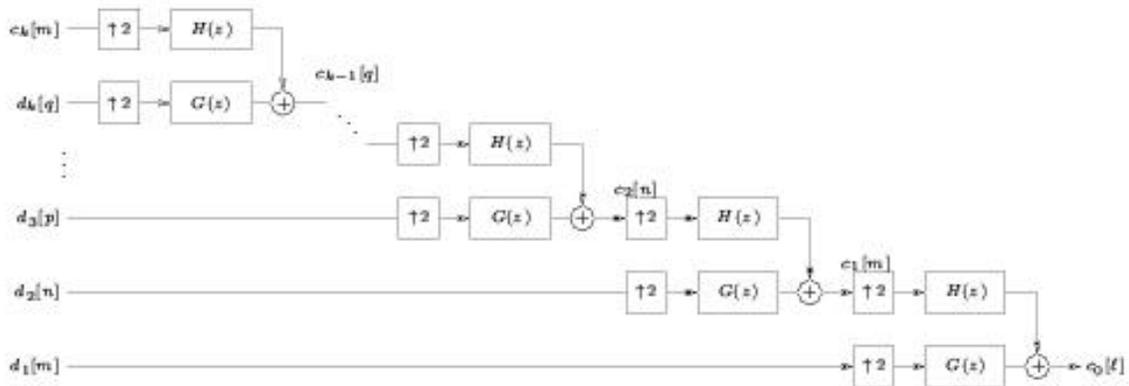
Figure 3.28

The table (Table 3.3) makes a direct comparison between wavelets and the two-channel orthogonal PR-FIR filterbanks.

	Discrete Wavelet Transform	2-Channel Orthogonal PR-FIR Filterbank
Analysis-LPF	$H\left(z^{-1}\right)$	$H_0\left(z\right)$
Power Symmetry	$H\left(z\right)H\left(z^{-1}\right)$ $+$ $H\left(-z\right)H\left(-z^{-1}\right)=2$	$H_0\left(z\right)H_0\left(z^{-1}\right)$ $+$ $H_0\left(-z\right)H_0\left(-z^{-1}\right)=1$
Analysis HPF	$G\left(z^{-1}\right)$	$H_1\left(z\right)$
Spectral Reverse	$G\left(z\right)$ $=$ $\pm\left(z^{-P}H\left(-z^{-1}\right)\right)$, P is odd	$H_1\left(z\right)$ $=$ $\pm\left(z^{-(N-1)}H_0\left(-z^{-1}\right)\right)$, N is even
Synthesis LPF	$H\left(z\right)$	$G_0\left(z\right)=2z^{-(N-1)}H_0\left(z^{-1}\right)$
Synthesis HPF	$G\left(z\right)$	$G_1\left(z\right)=2z^{-(N-1)}H_1\left(z^{-1}\right)$

Table 3.3

From the table, we see that the discrete wavelet transform that we have been developing is identical to two-channel orthogonal PR-FIR filterbanks in all but a couple details.

1. Orthogonal PR-FIR filterbanks employ synthesis filters with twice the gain of the analysis filters, whereas in the DWT the gains are equal.
2. Orthogonal PR-FIR filterbanks employ causal filters of length N, whereas the DWT filters are not constrained to be causal.

For convenience, however, the wavelet filters $H\left(z\right)$ and $G\left(z\right)$ are usually chosen to be causal. For both to have even impulse response length N, we require that $P = N - 1$.

3.3.11 Initialization of the Wavelet Transform[33]

The filterbanks developed in the module on the filterbanks interpretation of the DWT (Section 3.3.10) start with the signal representation $\{c_0\left[n\right] \mid n \in \mathbb{Z}\}$ and break the representation down into wavelet coefficients and scaling coefficients at lower resolutions (*i.e.*, higher levels k). The question remains: how do we get the initial coefficients $\{c_0\left[n\right]\}$?

From their definition, we see that the scaling coefficients can be written using a convolution:

$$\begin{aligned} c_0\left[n\right] &= <\phi\left(t-n\right),x\left(t\right)> \\ &= \int_{-\infty}^{\infty}\phi\left(t-n\right)x\left(t\right)dt \\ &= \phi\left(-t\right)*x\left(t\right)|_{t=n'} \end{aligned} \tag{3.49}$$

which suggests that the proper initialization of wavelet transform is accomplished by passing the continuous-time input $x\left(t\right)$ through an analog filter with impulse response $\phi\left(-t\right)$ and sampling its output at integer times (Figure 3.29).

---

[33]This content is available online at <http://cnx.org/content/m11018/2.2/>.
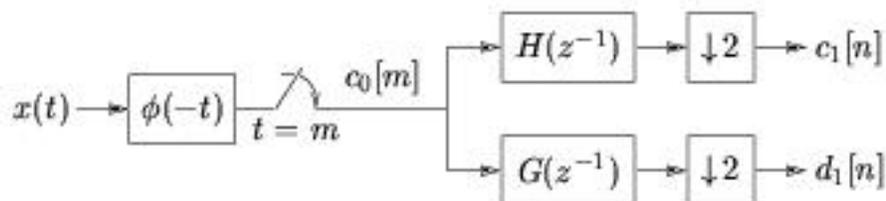
Figure 3.29

Practically speaking, however, it is very difficult to build an analog filter with impulse response $\phi\left(-t\right)$ for typical choices of scaling function.

The most often-used approximation is to set $c_0\left[n\right] = x\left[n\right]$. The sampling theorem implies that this would be exact if $\phi\left(t\right) = \frac{\sin(\pi t)}{\pi t}$, though clearly this is not **correct** for general $\phi\left(t\right)$. Still, this technique is somewhat justified if we adopt the view that the principle advantage of the wavelet transform comes from the multi-resolution capabilities implied by an iterated perfect-reconstruction filterbank (with **good** filters).

3.3.12 Regularity Conditions, Compact Support, and Daubechies' Wavelets[34]

Here we give a quick description of what is probably the most popular family of filter coefficients $h\left[n\right]$ and $g\left[n\right]$ — those proposed by Daubechies.

Recall the iterated synthesis filterbank. Applying the Noble identities, we can move the up-samplers before the filters, as illustrated in Figure 3.30.

---

[34]This content is available online at <http://cnx.org/content/m10495/2.8/>.
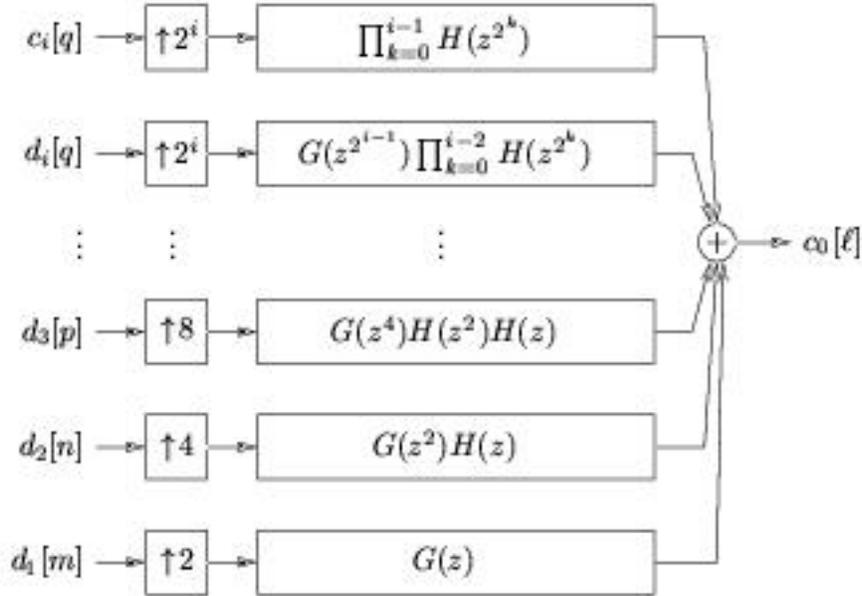
Figure 3.30

The properties of the i-stage cascaded lowpass filter

$$H^{(i)}(z) = \prod_{k=0}^{i-1} H\left(z^{2^k}\right) \quad , \quad i \geq 1 \tag{3.50}$$

in the limit $i \to \infty$ give an important characterization of the wavelet system. But how do we know that $\lim_{i \to \infty} H^{(i)}\left(e^{j\omega}\right)$ converges to a response in $\mathcal{L}_2$? In fact, there are some rather strict conditions on $H\left(e^{j\omega}\right)$ that must be satisfied for this convergence to occur. Without such convergence, we might have a finite-stage perfect reconstruction filterbank, but we will **not** have a countable wavelet basis for $\mathcal{L}_2$. Below we present some "regularity conditions" on $H\left(e^{j\omega}\right)$ that ensure convergence of the iterated synthesis lowpass filter.

NOTE: The convergence of the lowpass filter implies convergence of all other filters in the bank.

Let us denote the impulse response of $H^{(i)}(z)$ by $h^{(i)}[n]$. Writing

$$H^{(i)}(z) = H\left(z^{2^{i-1}}\right) H^{(i-1)}(z)$$

in the time domain, we have

$$h^{(i)}[n] = \sum_{kk} h[k]\, h^{(i-1)}\left[n - 2^{i-1}k\right]$$

Now define the function

$$\phi^{(i)}(t) = 2^{\frac{i}{2}} \sum_{nn} h^{(i)}[n]\, \mathcal{I}_{[n/2^i,\, n+1/2^i)}(t)$$

where $\mathcal{I}_{[a,b)}(t)$ denotes the indicator function over the interval $[a,b)$:

$$\mathcal{I}_{[a,b)}(t) = \begin{cases} 1 & \text{if } t \in [a,b) \\ 0 & \text{if } t \notin [a,b) \end{cases}$$

The definition of $\phi^{(i)}(t)$ implies

$$h^{(i)}[n] = 2^{-\frac{i}{2}} \phi^{(i)}(t) \quad , \quad t \in \left[\frac{n}{2^i}, \frac{n+1}{2^i}\right) \tag{3.51}$$

$$h^{(i-1)}\left[n - 2^{i-1}k\right] = 2^{-\frac{i-1}{2}} \phi^{(i-1)}(2t - k) \quad , \quad t \in \left[\frac{n}{2^i}, \frac{n+1}{2^i}\right) \tag{3.52}$$

and plugging the two previous expressions into the equation for $h^{(i)}[n]$ yields

$$\phi^{(i)}(t) = \sqrt{2} \sum_{kk} h[k] \phi^{(i-1)}[2t - k]. \tag{3.53}$$

Thus, if $\phi^{(i)}(t)$ converges pointwise to a continuous function, then it must satisfy the scaling equation, so that $\lim_{i \to \infty} \phi^{(i)}(t) = \phi(t)$. $Daubechies$[3] showed that, for pointwise convergence of $\phi^{(i)}(t)$ to a continuous function in $\mathcal{L}_2$, it is sufficient that $H\left(e^{j\omega}\right)$ can be factored as

$$H\left(e^{j\omega}\right) = \sqrt{2}\left(\frac{1 + e^{j\omega}}{2}\right)^P R\left(e^{j\omega}\right) \quad , \quad P \geq 1 \tag{3.54}$$

for $R\left(e^{j\omega}\right)$ such that

$$\sup_{\omega}\left(\left|R\left(e^{j\omega}\right)\right|\right) < 2^{P-1} \tag{3.55}$$

Here P denotes the number of zeros that $H\left(e^{j\omega}\right)$ has at $\omega = \pi$. Such conditions are called **regularity** conditions because they ensure the regularity, or **smoothness** of $\phi(t)$. In fact, if we make the previous condition stronger:

$$\sup_{\omega}\left(\left|R\left(e^{j\omega}\right)\right|\right) < 2^{P-1-\ell} \quad , \quad \ell \geq 1 \tag{3.56}$$

then $\lim_{i \to \infty} \phi^{(i)}(t) = \phi(t)$ for $\phi(t)$ that is ℓ-times continuously differentiable.

There is an interesting and important by-product of the preceding analysis. If $h[n]$ is a causal length-N filter, it can be shown that $h^{(i)}[n]$ is causal with length $N^i = 2^i(N-1) + 1$. By construction, then, $\phi^{(i)}[t]$ will be zero outside the interval $\left[0, \frac{2^i(N-1)+1}{2^i}\right)$. Assuming that the regularity conditions are satisfied so that $\lim_{i \to \infty} \phi^{(i)}(t) = \phi(t)$, it follows that $\phi(t)$ must be zero outside the interval $[0, N-1]$. In this case we say that $\phi(t)$ has **compact support**. Finally, the wavelet scaling equation implies that, when $\phi(t)$ is compactly supported on $[0, N-1]$ and $g[n]$ is length N, $\psi(t)$ will also be compactly supported on the interval $[0, N-1]$.

Daubechies constructed a family of $H(z)$ with impulse response lengths $N \in \{4, 6, 8, 10, \dots\}$ which satisfy the regularity conditions. Moreover, her filters have the maximum possible number of zeros at $\omega = \pi$, and thus are maximally regular (i.e., they yield the smoothest possible $\phi(t)$ for a given support interval). It turns out that these filters are the **maximally flat** filters derived by $Herrmann$[4] long before filterbanks and wavelets were in vogue. In Figure 3.31 and Figure 3.32 we show $\phi(t)$, $\Phi(\Omega)$, $\psi(t)$, and $\Psi(\Omega)$ for various members of the Daubechies' wavelet system.

See $Vetterli$ and $Kovacivi\acute{c}$[11] for a more complete discussion of these matters.
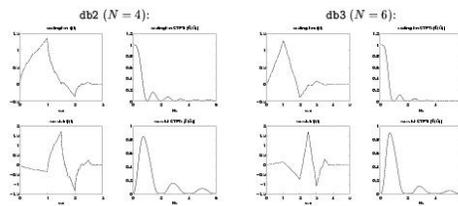
(a)
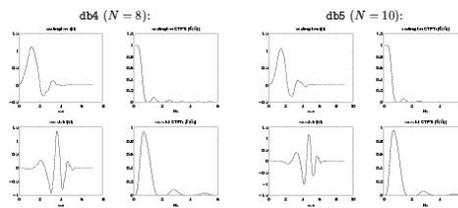
(b)

Figure 3.31
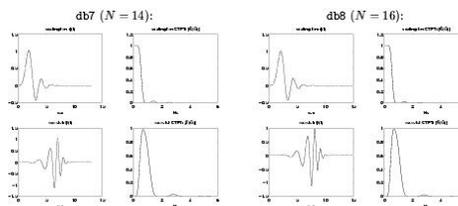
(a)
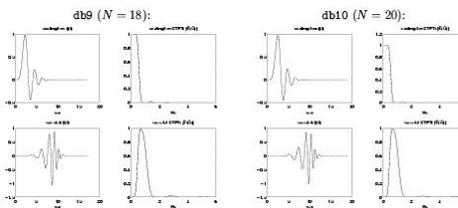
(b)

Figure 3.32

3.3.13 Computing the Scaling Function: The Cascade Algorithm[35]

Given coefficients $\{h[n]\}$ that satisfy the regularity conditions, we can iteratively calculate samples of $\phi(t)$ on a fine grid of points $\{t\}$ using the **cascade algorithm**. Once we have obtained $\phi(t)$, the wavelet scaling equation can be used to construct $\psi(t)$.

In this discussion we assume that $H(z)$ is causal with impulse response length N. Recall, from our discussion of the regularity conditions (p. 94), that this implies $\phi(t)$ will have compact support on the interval $[0, N-1]$. The cascade algorithm is described below.

1. Consider the scaling function at integer times $t = m \in \{0, \ldots, N-1\}$:

$$\phi(m) = \sqrt{2} \sum_{n=0}^{N-1} h(n) \phi(2m-n)$$

Knowing that $\phi(t) = 0$ for $t \notin [0, N-1]$, the previous equation can be written using an NxN matrix. In the case where $N = 4$, we have

$$
\begin{pmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \end{pmatrix} = \sqrt{2} \begin{pmatrix} h[0] & 0 & 0 & 0 \\ h[2] & h[1] & h[0] & 0 \\ 0 & h[3] & h[2] & h[1] \\ 0 & 0 & 0 & h[3] \end{pmatrix} \begin{pmatrix} \phi(0) \\ \phi(1) \\ \phi(2) \\ \phi(3) \end{pmatrix} \tag{3.57}
$$

$$
\text{where } H = \begin{pmatrix} h[0] & 0 & 0 & 0 \\ h[2] & h[1] & h[0] & 0 \\ 0 & h[3] & h[2] & h[1] \\ 0 & 0 & 0 & h[3] \end{pmatrix}
$$

The matrix H is structured as a **row-decimated convolution matrix**. From the matrix equation above ((3.57)), we see that $(\phi(0), \phi(1), \phi(2), \phi(3))^T$ must be (some scaled version of) the eigenvector of H corresponding to eigenvalue $(\sqrt{2})^{-1}$. In general, the nonzero values of $\{\phi(n) \mid n \in \mathbb{Z}\}$, i.e., $(\phi(0), \phi(1), \ldots, \phi(N-1))^T$, can be calculated by appropriately scaling the eigenvector of the NxN row-decimated convolution matrix H corresponding to the eigenvalue $(\sqrt{2})^{-1}$. It can be shown that this eigenvector must be scaled so that $\sum_{n=0}^{N-1} \phi(n) = 1$.

2. Given $\{\phi(n) \mid n \in \mathbb{Z}\}$, we can use the scaling equation to determine $\{\phi\left(\frac{n}{2}\right) \mid n \in \mathbb{Z}\}$:

$$\phi\left(\frac{m}{2}\right) = \sqrt{2} \sum_{n=0}^{N-1} h[n] \phi(m-n) \tag{3.58}$$

This produces the $2N-1$ non-zero samples $\{\phi(0), \phi(1/2), \phi(1), \phi(3/2), \ldots, \phi(N-1)\}$.

3. Given $\{\phi\left(\frac{n}{2}\right) \mid n \in \mathbb{Z}\}$, the scaling equation can be used to find $\{\phi\left(\frac{n}{4}\right) \mid n \in \mathbb{Z}\}$:

$$
\begin{aligned}
\phi\left(\frac{m}{4}\right) &= \sqrt{2} \sum_{n=0}^{N-1} h[n] \phi\left(\frac{m}{2} - n\right) \\
&= \sqrt{2} \sum_{p\,p \text{ even}} h\left[\frac{p}{2}\right] \phi\left(\frac{m-p}{2}\right) \\
&= \sqrt{2} \sum_{p\,p} h_{\uparrow 2}[p] \phi_{\frac{1}{2}}[m-p]
\end{aligned} \tag{3.59}
$$

where $h_{\uparrow 2}[p]$ denotes the impulse response of $H(z^2)$, i.e., a 2-upsampled version of $h[n]$, and where $\phi_{\frac{1}{2}}[m] = \phi\left(\frac{m}{2}\right)$. Note that $\{\phi\left(\frac{n}{4}\right) \mid n \in \mathbb{Z}\}$ is the result of convolving $h_{\uparrow 2}[n]$ with $\{\phi_{\frac{1}{2}}[n]\}$.

---

[35]This content is available online at $<$http://cnx.org/content/m10486/2.6/$>$.

4. Given $\left\{ \phi\left(\frac{n}{4}\right) \mid n \in \mathbb{Z} \right\}$, another convolution yields $\left\{ \phi\left(\frac{n}{8}\right) \mid n \in \mathbb{Z} \right\}$:

$$
\begin{aligned}
\phi\left(\tfrac{m}{8}\right) &= \sqrt{2} \sum_{n=0}^{N-1} h\left[n\right] \phi\left(\tfrac{m}{4} - n\right) \\
&= \sqrt{2} \sum_{pp} h_{\uparrow 4}\left[p\right] \phi_{\frac{1}{4}}\left[m - p\right]
\end{aligned}
\tag{3.60}
$$

where $h_{\uparrow 4}\left[n\right]$ is a 4-upsampled version of $h\left[n\right]$ and where $\phi_{\frac{1}{4}}\left[m\right] = \phi\left(\tfrac{m}{4}\right)$.

5. At the ℓth stage, $\left\{ \phi\left(\frac{n}{2^\ell}\right) \right\}$ is calculated by convolving the result of the $\ell - 1^{\text{th}}$ stage with a $2^{\ell-1}$-upsampled version of $h\left[n\right]$:

$$
\phi_{\frac{1}{2^\ell}}\left(m\right) = \sqrt{2} \sum_{pp} h_{\uparrow 2^{\ell-1}}\left[p\right] \phi_{\frac{1}{2^{\ell-1}}}\left[m - p\right]
\tag{3.61}
$$

For $\ell \simeq 10$, this gives a very good approximation of $\phi\left(t\right)$. At this point, you could verify the key properties of $\phi\left(t\right)$, such as orthonormality and the satisfaction of the scaling equation.

In Figure 3.33 we show steps 1 through 5 of the cascade algorithm, as well as step 10, using Daubechies' db2 coefficients (for which $N = 4$).
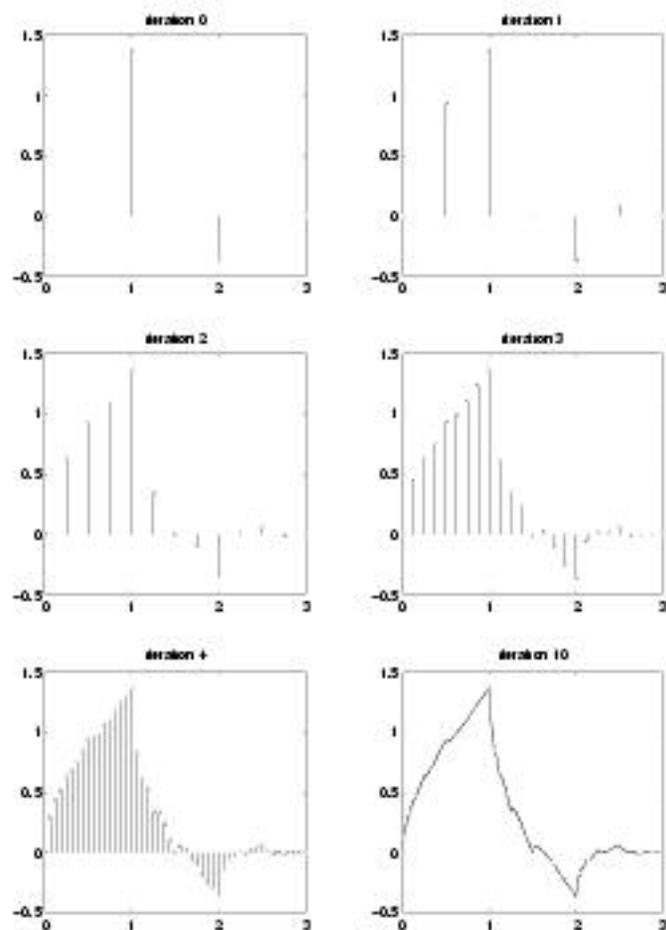
Figure 3.33

3.3.14 Finite-Length Sequences and the DWT Matrix[36]

The wavelet transform, viewed from a filterbank perspective, consists of iterated 2-channel analysis stages like the one in Figure 3.34.

---

[36]This content is available online at <http://cnx.org/content/m10459/2.6/>.
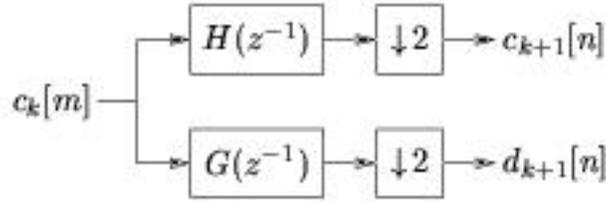
Figure 3.34

First consider a very long (i.e., practically infinite-length) sequence $\{c_k[m] \mid m \in \mathbb{Z}\}$. For every pair of input samples $\{c_k[2n], c_k[2n-1]\}$ that enter the kth filterbank stage, exactly one pair of output samples $\{c_{k+1}[n], d_{k+1}[n]\}$ are generated. In other words, the number of output equals the number of input during a fixed time interval. This property is convenient from a real-time processing perspective.

For a short sequence $\{c_k[m] \mid m \in \{0, \ldots, M-1\}\}$, however, linear convolution requires that we make an assumption about the tails of our finite-length sequence. One assumption could be

$$c_k[m] = 0 \quad, \quad m \in \{0, \ldots, M-1\} \tag{3.62}$$

In this case, the linear convolution implies that M nonzero inputs yield $\frac{M+N}{2} - 1$ outputs from each branch, for a total of $2\left(\frac{M+N}{2} - 1\right) = M + N - 2 > M$ outputs. Here we have assumed that both $H\left(z^{-1}\right)$ and $G\left(z^{-1}\right)$ have impulse response lengths of $N > 2$, and that M and N are both even. The fact that each filterbank stage produces more outputs than inputs is very disadvantageous in many applications.

A more convenient assumption regarding the tails of $\{c_k[m] \mid m \in \{0, \ldots, M-1\}\}$ is that the data outside of the time window $\{0, \ldots, M-1\}$ is a cyclic extension of data inside the time window. In other words, given a length-M sequence, the points **outside the sequence** are related to points **inside the sequences** via

$$c_k[m] = c_k[m + M] \tag{3.63}$$

Recall that a linear convolution with an M-cyclic input is equivalent to a circular convolution with one M-sample period of the input sequences. Furthermore, the output of this circular convolution is itself M-cyclic, implying our 2-downsampled branch outputs are cyclic with period $\frac{M}{2}$. Thus, given an M-length input sequence, the total filterbank output consists of exactly M values.

It is instructive to write the circular-convolution analysis filterbank operation in matrix form. In (3.64) we give an example for filter length $N = 4$, sequence length $N = 8$, and causal synthesis filters $H(z)$ and $G(z)$.

$$
\begin{pmatrix}
c_{k+1}[0] \\
c_{k+1}[1] \\
c_{k+1}[2] \\
c_{k+1}[3] \\
d_{k+1}[0] \\
d_{k+1}[1] \\
d_{k+1}[2] \\
d_{k+1}[3]
\end{pmatrix}
=
\begin{pmatrix}
h[0] & h[1] & h[2] & h[3] & 0 & 0 & 0 & 0 \\
0 & 0 & h[0] & h[1] & h[2] & h[3] & 0 & 0 \\
0 & 0 & 0 & 0 & h[0] & h[1] & h[2] & h[3] \\
h[2] & h[3] & 0 & 0 & 0 & 0 & h[0] & h[1] \\
g[0] & g[1] & g[2] & g[3] & 0 & 0 & 0 & 0 \\
0 & 0 & g[0] & g[1] & g[2] & g[3] & 0 & 0 \\
0 & 0 & 0 & 0 & g[0] & g[1] & g[2] & g[3] \\
g[2] & g[3] & 0 & 0 & 0 & 0 & g[0] & g[1]
\end{pmatrix}
\begin{pmatrix}
c_k[0] \\
c_k[1] \\
c_k[2] \\
c_k[3] \\
c_k[4] \\
c_k[5] \\
c_k[6] \\
c_k[7]
\end{pmatrix}
\tag{3.64}
$$

where

$$\begin{pmatrix} c_{k+1} \\ d_{k+1} \end{pmatrix} = \begin{pmatrix} c_{k+1}[0] \\ c_{k+1}[1] \\ c_{k+1}[2] \\ c_{k+1}[3] \\ d_{k+1}[0] \\ d_{k+1}[1] \\ d_{k+1}[2] \\ d_{k+1}[3] \end{pmatrix}$$

$$\begin{pmatrix} H_M \\ G_M \end{pmatrix} = \begin{pmatrix} h[0] & h[1] & h[2] & h[3] & 0 & 0 & 0 & 0 \\ 0 & 0 & h[0] & h[1] & h[2] & h[3] & 0 & 0 \\ 0 & 0 & 0 & 0 & h[0] & h[1] & h[2] & h[3] \\ h[2] & h[3] & 0 & 0 & 0 & 0 & h[0] & h[1] \\ g[0] & g[1] & g[2] & g[3] & 0 & 0 & 0 & 0 \\ 0 & 0 & g[0] & g[1] & g[2] & g[3] & 0 & 0 \\ 0 & 0 & 0 & 0 & g[0] & g[1] & g[2] & g[3] \\ g[2] & g[3] & 0 & 0 & 0 & 0 & g[0] & g[1] \end{pmatrix}$$

$$c_k = \begin{pmatrix} c_k[0] \\ c_k[1] \\ c_k[2] \\ c_k[3] \\ c_k[4] \\ c_k[5] \\ c_k[6] \\ c_k[7] \end{pmatrix}$$

The matrices H_M and G_M have interesting properties. For example, the conditions

$$\delta[m] = \sum_{nn} h[n]\, h[n-2m]$$

$$g[n] = -1^n h[N-1-n]$$

imply that

$$\begin{pmatrix} H_M \\ G_M \end{pmatrix}^T \begin{pmatrix} H_M \\ G_M \end{pmatrix} = \begin{pmatrix} H_M \\ G_M \end{pmatrix} \begin{pmatrix} H_M \\ G_M \end{pmatrix}^T = I_M$$

where I_M denotes the MxM identity matrix. Thus, it makes sense to define the MxM **DWT matrix** as

$$T_M = \begin{pmatrix} H_M \\ G_M \end{pmatrix} \tag{3.65}$$

whose transpose constitutes the MxM **inverse DWT matrix**:

$$T_M{}^{-1} = T_M{}^T \tag{3.66}$$

Since the synthesis filterbank (Figure 3.35)
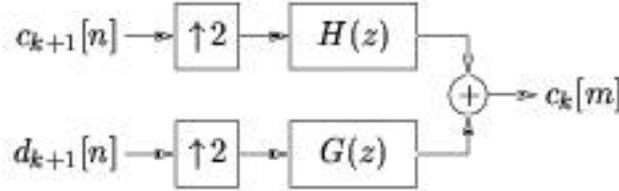
Figure 3.35

gives perfect reconstruction, and since the cascade of matrix operations $T_M{}^T T_M$ also corresponds to perfect reconstruction, we expect that the matrix operation $T_M{}^T$ describes the action of the synthesis filterbank. This is readily confirmed by writing the upsampled circular convolutions in matrix form:

$$
\begin{pmatrix} c_k[0] \\ c_k[1] \\ c_k[2] \\ c_k[3] \\ c_k[4] \\ c_k[5] \\ c_k[6] \\ c_k[7] \end{pmatrix} =
\begin{pmatrix}
h[0] & 0 & 0 & h[2] & g[0] & 0 & 0 & g[2] \\
h[1] & 0 & 0 & h[3] & g[1] & 0 & 0 & g[3] \\
h[2] & h[0] & 0 & 0 & g[2] & g[0] & 0 & 0 \\
h[3] & h[1] & 0 & 0 & g[3] & g[1] & 0 & 0 \\
0 & h[2] & h[0] & 0 & 0 & g[2] & g[0] & 0 \\
0 & h[3] & h[1] & 0 & 0 & g[3] & g[1] & 0 \\
0 & 0 & h[2] & h[0] & 0 & 0 & g[2] & g[0] \\
0 & 0 & h[3] & h[1] & 0 & 0 & g[3] & g[1]
\end{pmatrix}
\begin{pmatrix} c_{k+1}[0] \\ c_{k+1}[1] \\ c_{k+1}[2] \\ c_{k+1}[3] \\ d_{k+1}[0] \\ d_{k+1}[1] \\ d_{k+1}[2] \\ d_{k+1}[3] \end{pmatrix} \tag{3.67}
$$

where

$$
\begin{pmatrix} H_M{}^T \\ G_M{}^T \end{pmatrix} = T_M{}^T =
\begin{pmatrix}
h[0] & 0 & 0 & h[2] & g[0] & 0 & 0 & g[2] \\
h[1] & 0 & 0 & h[3] & g[1] & 0 & 0 & g[3] \\
h[2] & h[0] & 0 & 0 & g[2] & g[0] & 0 & 0 \\
h[3] & h[1] & 0 & 0 & g[3] & g[1] & 0 & 0 \\
0 & h[2] & h[0] & 0 & 0 & g[2] & g[0] & 0 \\
0 & h[3] & h[1] & 0 & 0 & g[3] & g[1] & 0 \\
0 & 0 & h[2] & h[0] & 0 & 0 & g[2] & g[0] \\
0 & 0 & h[3] & h[1] & 0 & 0 & g[3] & g[1]
\end{pmatrix}
$$

So far we have concentrated on one stage in the wavelet decomposition; a two-stage decomposition is illustrated in Figure 3.36.
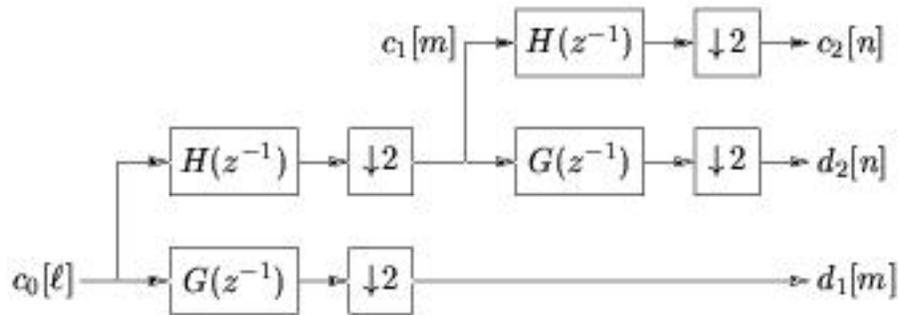
Figure 3.36

The two-stage analysis operation (assuming circular convolution) can be expressed in matrix form as

$$
\begin{pmatrix} c_{k+2} \\ d_{k+2} \\ d_{k+1} \end{pmatrix} = \begin{pmatrix} T_{\frac{M}{2}} & 0 \\ 0 & I_{\frac{M}{2}} \end{pmatrix} \begin{pmatrix} c_{k+1} \\ d_{k+1} \end{pmatrix}
$$
$$
= \begin{pmatrix} T_{\frac{M}{2}} & 0 \\ 0 & I_{\frac{M}{2}} \end{pmatrix} \begin{pmatrix} T_M \end{pmatrix} \begin{pmatrix} c_k \end{pmatrix}
$$
(3.68)

Similarly, a three-stage analysis could be implemented via

$$
\begin{pmatrix} c_{k+3} \\ d_{k+3} \\ d_{k+2} \\ d_{k+1} \end{pmatrix} = \begin{pmatrix} T_{\frac{M}{4}} & 0 & 0 \\ 0 & I_{\frac{M}{4}} & 0 \\ 0 & 0 & I_{\frac{M}{2}} \end{pmatrix} \begin{pmatrix} T_{\frac{M}{2}} & 0 \\ 0 & I_{\frac{M}{2}} \end{pmatrix} \begin{pmatrix} T_M \end{pmatrix} \begin{pmatrix} c_k \end{pmatrix}
$$
(3.69)

It should now be evident how to extend this procedure to 3 stages. As noted earlier, the corresponding synthesis operations are accomplished by transposing the matrix products used in the analysis.

3.3.15 DWT Implementation using FFTs[37]

Finally, we say a few words about DWT implementation. Here we focus on a single DWT stage and assume circular convolution, yielding an MxM DWT matrix T_M . In the general case, MxM matrix multiplication requires M^2 multiplications. The DWT matrices, however, have a circular-convolution structure which allows us to implement them using significantly less multiplies. Below we present some simple and reasonably efficient approaches for the implementation of T_M and $T_M{}^T$.

We treat the inverse DWT first. Recall that in the lowpass synthesis branch, we upsample the input before circularly convolving with $H(z)$. Denoting the upsampled coefficient sequence by $a[n]$, fast circular convolution $a[n] * h[n]$ can be described as follows (using Matlab notation)

```
ifft( fft(a).*fft(h,length(a)) )
```

where we have assumed that `length(a)` $\geq$ `length(h)`. [38] The highpass branch is handled similarly using $G(z)$, after which the two branch outputs are summed.

Next we treat the forward DWT. Recall that in the lowpass analysis branch, we circularly convolve the input with $H(z^{-1})$ and then downsample the result. The fast circular convolution $a[n] * h[-n]$ can be implemented using

```
wshift('1', ifft(fft(a).*fft(flipud(h),length(a))), length(h)-1 )
```

where `wshift` accomplishes a circular shift of the `ifft` output that makes up for the unwanted delay of `length(h)-1` samples imposed by the `flipud` operation. The highpass branch is handled similarly but with filter $G(z^{-1})$. Finally, each branch is downsampled by factor two.

We note that the proposed approach is not totally efficient because downsampling is performed after circular convolution (and upsampling before circular convolution). Still, we have outlined this approach because it is easy to understand and still results in major saving when M is large: it converts the $O(M^2)$ matrix multiply into an $O(M\log_2 M)$ operation.

3.3.16 DWT Applications - Choice of phi(t)[39]

Transforms are signal processing tools that are used to give a clear view of essential signal characteristics. Fourier transforms are ideal for infinite-duration signals that contain a relatively small number of sinusoids: one can completely describe the signal using only a few coefficients. Fourier transforms, however, are not well-suited to signals of a non-sinusoidal nature (as discussed earlier in the context of time-frequency analysis (Section 3.1.2)). The multi-resolution DWT is a more general transform that is well-suited to a larger class of signals. For the DWT to give an efficient description of the signal, however, we must choose a wavelet $\psi(t)$ from which the signal can be constructed (to a good approximation) using only a few stretched and shifted copies.

We illustrate this concept in Figure 3.37 using two examples. On the left, we analyze a step-like waveform, while on the right we analyze a chirp-like waveform. In both cases, we try DWTs based on the Haar and Daubechies `db10` wavelets and plot the log magnitudes of the transform coefficients $\left[c_k^T, d_k^T, d_{k-1}^T, d_{k-2}^T, \ldots, d_1^T\right]$.

---

[38] When implementing the multi-level transform, you must ensure that the data length does not become shorter than the filter length!

[39] This content is available online at $<$http://cnx.org/content/m11004/2.1/$>$.
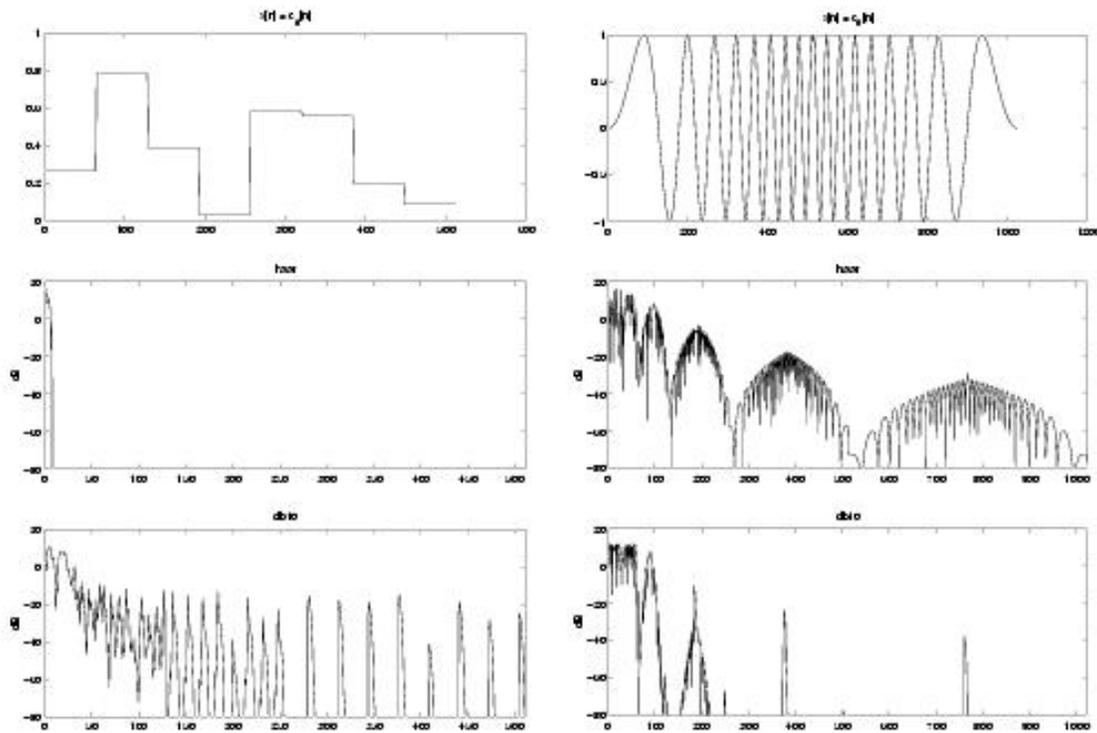
Figure 3.37

Observe that the Haar DWT yields an extremely efficient representation of the step-waveform: only a few of the transform coefficients are nonzero. The db10 DWT does not give an efficient representation: many coefficients are sizable. This makes sense because the Haar scaling function is well matched to the step-like nature of the time-domain signal. In contrast, the Haar DWT does not give an efficient representation of the chirp-like waveform, while the db10 DWT does better. This makes sense because the sharp edges of the Haar scaling function do not match the smooth chirp signal, while the smoothness of the db10 wavelet yields a better match.

3.3.17 DWT Application - De-noising[40]

Say that the DWT for a particular choice of wavelet yields an efficient representation of a particular signal class. In other words, signals in the class are well-described using a few large transform coefficients.

Now consider unstructured **noise**, which cannot be eifficiently represented by any transform, including the DWT. Due to the orthogonality of the DWT, such noise sequences make, on average, equal contributions to all transform coefficients. Any given noise sequence is expected to yield many small-valued transform coefficients.

Together, these two ideas suggest a means of **de-noising** a signal. Say that we perform a DWT on a signal from our **well-matched** signal class that has been corrupted by additive noise. We expect that large transform coefficients are composed mostly of signal content, while small transform coefficients should be

---

[40]This content is available online at <http://cnx.org/content/m11000/2.1/>.

composed mostly of noise content. Hence, throwing away the transform coefficients whose magnitude is less than some small threshold should improve the signal-to-noise ratio. The de-noising procedure is illustrated in Figure 3.38.
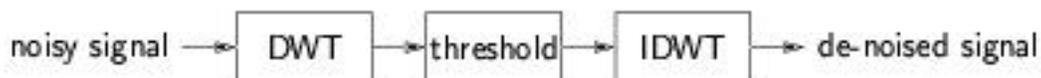
Figure 3.38

Now we give an example of denoising a step-like waveform using the Haar DWT. In Figure 3.39, the top right subplot shows the noisy signal and the top left shows it DWT coefficients. Note the presence of a few large DWT coefficients, expected to contain mostly signal components, as well as the presence of many small-valued coefficients, expected to contain noise. (The bottom left subplot shows the DWT for the original signal before any noise was added, which confirms that all signal energy is contained within a few large coefficients.) If we throw away all DWT coefficients whose magnitude is less than 0.1, we are left with only the large coefficients (shown in the middle left plot) which correspond to the de-noised time-domain signal shown in the middle right plot. The difference between the de-noised signal and the original noiseless signal is shown in the bottom right. Non-zero error results from noise contributions to the large coefficients; there is no way of distinguishing these noise components from signal components.
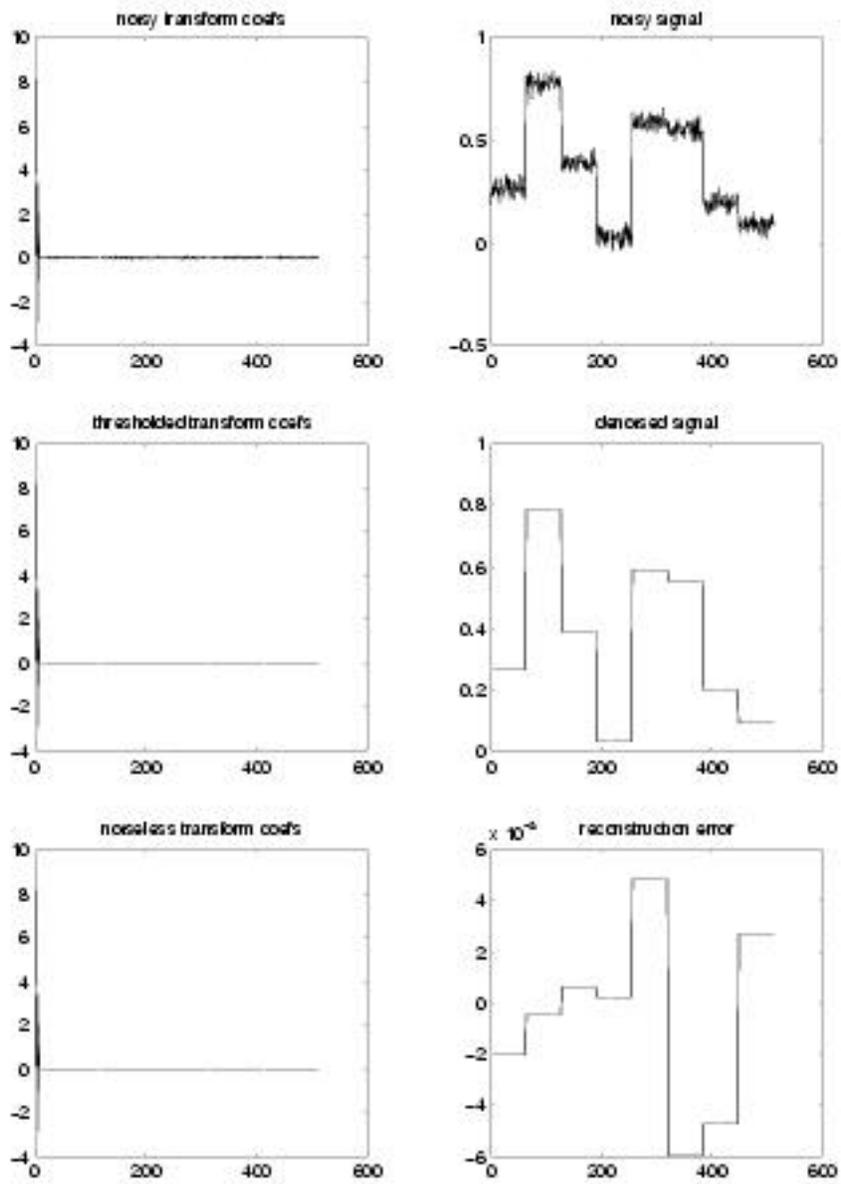
Figure 3.39

Bibliography

[1] R. Ansari and B. Liu. Multirate signal processing. In S.K. Mitra and J.F. Kaiser, editors, *Handbook for Digital Signal Processing*. Wiley Interscience, New York. chp. 14 pp. 981-1084.

[2] R.E Crochiere and L.R. Rabiner. *Multirate Digital Signal Processing*. 1983.

[3] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Commun. on Pure and Applied Math*, 41:909–996, Nov 1988.

[4] O. Herrmann. On the approximation problem in nonrecursive digital filter design. *IEEE Trans. on Circuit Theory*, 18:411–413, 1971.

[5] J.D. Johnston. A filter family designed for use in quadrature mirror filterbanks. In *Proc IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing*, pages 291–294, April 1980.

[6] S.K. Mitra. *Digital Signal Processing*. McGraw-Hill, New York, 2nd edition edition, 2001.

[7] R.E.Crochiere and L.B. Rabiner. *Multirate Digital Signal Processing*. Prentice Hall, Englewood Cliffs, NJ, 1983.

[8] J.H. Rothweiler. *Polyphase quadrature filters - A new subband coding technique*. 1983.

[9] P.P. Vaidyanathan. *Multirate Systems and Filter Banks*. 1993.

[10] P.P. Vaidyanathan. *Multirate Systems and Filterbanks*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[11] M. Vetterli and J. Kovacivi263;. *Wavelets and Subband Coding*. Prentice Hall, Englewood Cliffs, NJ, 1995.

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

Attributions

Module: "Computational Savings of Polyphase Resampling"
By: Phil Schniter
URL: http://cnx.org/content/m11009/2.1/
Pages: 21-22
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "CD to DAT Conversion"
By: Phil Schniter
URL: http://cnx.org/content/m11010/2.2/
Page: 22
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Polyphase Resampling with Arbitrary (Non-Rational or Time-Varying) Rate"
By: Phil Schniter
URL: http://cnx.org/content/m10456/2.9/
Pages: 22-25
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Multi-stage Interpolation and Decimation"
By: Phil Schniter
URL: http://cnx.org/content/m10420/2.13/
Pages: 25-29
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Sub-Band Processing"
By: Phil Schniter
URL: http://cnx.org/content/m10423/2.14/
Pages: 31-33
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Uniform Filterbanks"
By: Phil Schniter
URL: http://cnx.org/content/m10990/2.3/
Pages: 33-34
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Uniform Modulated Filterbank"
By: Phil Schniter
URL: http://cnx.org/content/m10929/2.6/
Pages: 34-35
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Uniformally Modulated (DFT) Filterbank"
By: Phil Schniter
URL: http://cnx.org/content/m10424/2.13/
Pages: 35-40
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Computational Savings of Polyphase/DFT Filterbanks"
By: Phil Schniter
URL: http://cnx.org/content/m10930/2.3/
Pages: 40-41
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Drawbacks of Classical Filterbank Designs"
By: Phil Schniter
URL: http://cnx.org/content/m10931/2.4/
Page: 41
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Aliasing-Cancellation Conditions of Filterbanks"
By: Phil Schniter
URL: http://cnx.org/content/m10425/2.11/
Pages: 41-43
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Two-Branch Quadvalue Mirror Filterbank (QMF)"
By: Phil Schniter
URL: http://cnx.org/content/m10426/2.14/
Pages: 43-44
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Perfect Reconstruction QMF"
By: Phil Schniter
URL: http://cnx.org/content/m11002/2.2/
Pages: 44-45
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Johnston's QMF Banks"
By: Phil Schniter
URL: http://cnx.org/content/m10932/2.4/
Pages: 45-47
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Perfect Reconstruction FIR Filter Banks"
By: Phil Schniter
URL: http://cnx.org/content/m10412/2.14/
Pages: 47-49
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Orthogonal Perfect Reconstruction FIR Filterbank"
By: Phil Schniter
URL: http://cnx.org/content/m10413/2.14/
Pages: 49-50
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Design of Orthogonal PR-FIR Filterbanks via Halfband Spectral Factorization"
By: Phil Schniter
URL: http://cnx.org/content/m11005/2.2/
Pages: 50-54
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Bi-Orthogonal Perfect Reconstruction FIR Filterbanks"
By: Phil Schniter
URL: http://cnx.org/content/m10414/2.11/
Pages: 54-56
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Filterbanks with >2 Branches"
By: Phil Schniter
URL: http://cnx.org/content/m10415/2.13/
Pages: 56-57
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Why Transforms?"
By: Phil Schniter
URL: http://cnx.org/content/m11001/2.1/
Page: 59
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Limitations of Fourier Analysis"
By: Phil Schniter
URL: http://cnx.org/content/m11003/2.1/
Pages: 59-62
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Time-Frequency Uncertainty Principle"
By: Phil Schniter
URL: http://cnx.org/content/m10416/2.18/
Pages: 62-63
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "The Haar System as an Example of DWT"
By: Phil Schniter
URL: http://cnx.org/content/m10437/2.10/
Pages: 76-78
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "A Hierarchy of Detail in the Haar System"
By: Phil Schniter
URL: http://cnx.org/content/m11012/2.3/
Pages: 78-80
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Haar Approximation at the kth Coarseness Level"
By: Phil Schniter
URL: http://cnx.org/content/m11013/2.2/
Pages: 80-81
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "The Scaling Equation"
By: Phil Schniter
URL: http://cnx.org/content/m10476/2.7/
Page: 81
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "The Wavelet Scaling Equation"
By: Phil Schniter
URL: http://cnx.org/content/m11014/2.2/
Pages: 81-82
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Conditions on h[n] and g[n]"
By: Phil Schniter
URL: http://cnx.org/content/m11015/2.2/
Pages: 82-84
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "Values of g[n] and h[n] for the Haar System"
By: Phil Schniter
URL: http://cnx.org/content/m11016/2.2/
Pages: 84-86
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "DWT Applications - Choice of phi(t)"
By: Phil Schniter
URL: http://cnx.org/content/m11004/2.1/
Pages: 104-105
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

Module: "DWT Application - De-noising"
By: Phil Schniter
URL: http://cnx.org/content/m11000/2.1/
Pages: 105-107
Copyright: Phil Schniter
License: http://creativecommons.org/licenses/by/1.0

About Connexions

Since 1999, Connexions has been pioneering a global system where anyone can create course materials and make them fully accessible and easily reusable free of charge. We are a Web-based authoring, teaching and learning environment open to anyone interested in education, including students, teachers, professors and lifelong learners. We connect ideas and facilitate educational communities.

Connexions's modular, interactive courses are in use worldwide by universities, community colleges, K-12 schools, distance learners, and lifelong learners. Connexions materials are in many languages, including English, Spanish, Chinese, Japanese, Italian, Vietnamese, French, Portuguese, and Thai. Connexions is part of an exciting new information distribution system that allows for **Print on Demand Books**. Connexions has partnered with innovative on-demand publisher QOOP to accelerate the delivery of printed course materials and textbooks into classrooms worldwide at lower prices than traditional academic publishers.