Elec4621:
Advanced Digital Signal Processing
Chapter 9: Multi-Rate Signal Processing

Dr. D. S. Taubman

May 16, 2011

1 Decimation

Let $x[n]$ be a sequence which has been sampled at rate 1. That is, in an appropriate time scale, $x[n]$ is obtained by sampling a Nyquist bandlimited continuous signal, $x(t)$, according to

$$x[n] = x(t)|_{t=n}$$

The decimation (or down-sampling) operator, $\boxed{\downarrow M}$, converts its input, $x[n]$, into a new sequence, $y[n]$, having rate $\frac{1}{M}$, according to

$$y[n] = x[Mn]$$

That is, $y[n]$ contains every M^{th} sample from $x[n]$. Equivalently, $y[n]$ is related to the underlying continuous time signal, $x(t)$, through

$$y[n] = x(t)|_{t=Mn}$$

The sampling rate of $y[n]$ is not generally sufficient to avoid aliasing, unless $x(t)$ and hence $x[n]$ happen to be bandlimited to the interval $-\pi/M < \omega < \pi/M$.

We can analyze the DTFT transform of $y[n]$ in one of a number of ways. Since the DTFT of $x[n]$ is identical to the true Fourier transform of $x(t)$, we could first scale the time axis of $x(t)$ in such a way that $y[n]$ becomes a unit sample sequence of the scaled signal, $s(t)$, having Fourier transform

$$\hat{s}(\omega) = \frac{1}{M}\hat{x}\left(\frac{\omega}{M}\right)$$

We can then use the familiar aliasing equation to find

$$\hat{y}(\omega) = \sum_k \hat{s}(\omega + 2\pi k) = \frac{1}{M}\sum_k \hat{x}\left(\frac{\omega + 2\pi k}{M}\right)$$

An alternate method for developing the relationship between $\hat{y}(\omega)$ and $\hat{x}(\omega)$ will prove useful for our later work. We begin by defining a useful intermediate sequence, $x'[n]$ (we shall see this again), according to

$$x'[n] = \begin{cases} x[n] & n = Mk \\ 0 & n = Mk + p,\ 0 < p < M \end{cases}$$

That is, we identify $x'[n]$ with $x[n]$ at the points which are kept by the decimator and set $x'[n]$ to zero at the points which are discarded.

Now we can write the Fourier transform of $x'[n]$ in terms of the Fourier transform of $x[n]$ by exploiting the properties of the M'th roots of unity, $W_M^m = e^{j\frac{2\pi m}{M}}$, $m = 0, 1, \ldots M-1$. These are the M distinct complex numbers which yield 1 when raised to the power of M. Moreover,

$$\sum_{m=0}^{M-1} W_M^m = 0 \quad \text{and} \quad \sum_{m=0}^{M-1} W_M^{mk} = 0$$

for any k which is not divisible by M. With this in mind, we can write

$$x'[n] = \frac{1}{M}\sum_{m=0}^{M-1} W_M^{mn} x[n]$$

and so

$$\begin{aligned} \hat{x'}(\omega) &= \frac{1}{M}\sum_{m=0}^{M-1}\sum_n e^{-j\omega n} e^{j\frac{2\pi n m}{M}} x[n] \\ &= \frac{1}{M}\sum_{m=0}^{M-1}\sum_n e^{-j(\omega - \frac{2\pi m}{M})n} x[n] \\ &= \frac{1}{M}\sum_{m=0}^{M-1} \hat{x}(\omega - \frac{2\pi m}{M}) \end{aligned}$$

Finally, we find the DTFT of $y[n]$ from

$$\begin{aligned}\hat{y}(\omega) &= \sum_n y[n]e^{-j\omega n} \\ &= \sum_n x'[n]e^{-j\omega n/M} \\ &= \frac{1}{M}\sum_{m=0}^{M-1}\hat{x}\left(\frac{\omega-2\pi m}{M}\right)\end{aligned} \tag{1}$$

This relationship involves aliasing and contraction of the original spectrum, $\hat{x}(\omega)$. Specifically, we first shift $\hat{x}(\omega)$ by multiples of $\frac{2\pi m}{M}$, finding the M shifted copies which overlap any given frequency of interest, $\omega \in \left(-\frac{\pi}{M}, \frac{\pi}{M}\right)$. We take the average of these M shifted copies and then expand the interval, $\left(-\frac{\pi}{M}, \frac{\pi}{M}\right)$, to the full Nyquist frequency range, $(-\pi, \pi)$, associated with $\hat{y}(\omega)$. These operations are depicted in Figure 1.

Suppose now that we pre-filter $x[n]$ using an ideal band-pass filter, $h_m^{(M)}[n]$, having

$$\hat{h}_m^{(M)}(\omega) = \begin{cases} 1 & |\pm\omega_m - \omega| < \frac{\pi}{2M} \\ 0 & \text{otherwise} \end{cases}$$

where

$$\omega_m = \frac{\pi}{2M} + \frac{m\pi}{M}, \quad m = 0, 1, \ldots M-1$$

is the centre-frequency of the band-pass filter and $\frac{\pi}{M}$ is the bandwidth of each half (positive and negative frequencies) of the filter's pass band. After some thought (convince yourself by drawing sketches), it should be clear that only one term is preserved in the aliasing summation.

If m is an even integer, using the anti-aliasing filter $h_m^{(M)}$ prior to down-sampling by M, moves the lower edge of the pass band centred at $\pm\omega_m$ down to DC and the upper edge of the pass band to $\omega = \pm\pi$ in the down-sampled spectrum, $\hat{y}(\omega)$. On the other hand, if m is odd, the lower edge of the pass band centred at $\pm\omega_m$ is actually moved to $\omega = \mp\pi$, while the upper edge of the pass band is moved to DC in the down-sampled spectrum, $\hat{y}(\omega)$. This spectral reversal property for odd indexed pass bands is well worth remembering.

Often we are interested in preserving only the low frequency (low resolution) portion of $\hat{x}(\omega)$ in the down-sampled sequence, so we select the low-pass anti-aliasing filter $h_0^{(M)}$. In a communication system, however, we may be interested in some portion of the spectrum other than baseband.

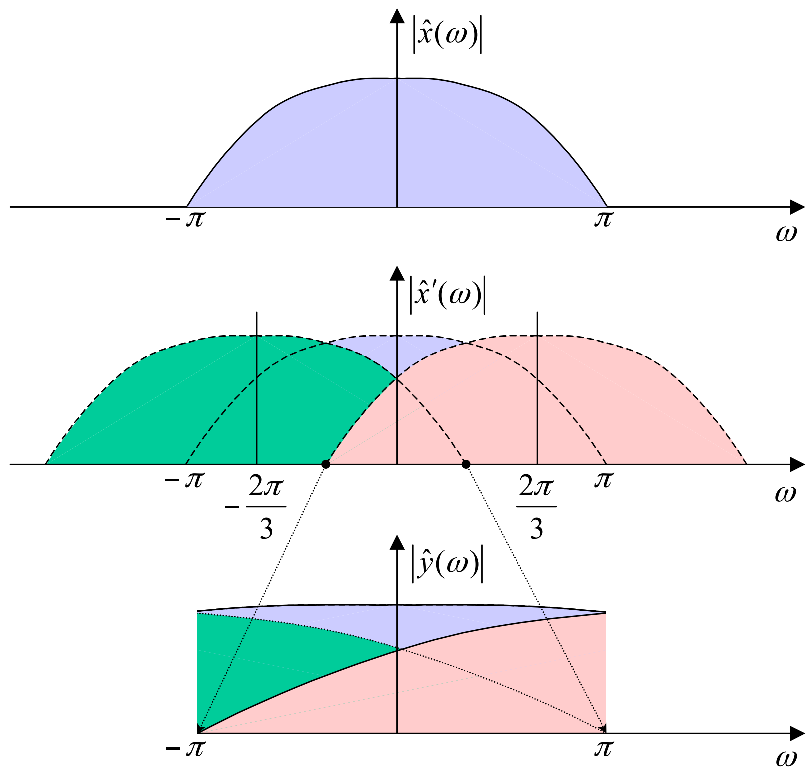

Figure 1: *Effect of downsampling by a factor of 3, in the frequency domain.*

Because we will not be able to build ideal low-pass filters, the actual filters $h_0^{(M)}$ will not be able to completely remove the effects of aliasing in practice.

2 Interpolation

Let $x[n]$ be a sequence with sampling rate 1. The interpolator (or up-sampling) operator, $\boxed{\uparrow M}$, converts its input, $x[n]$, into a sequence, $y[n]$, with rate M, according to

$$y[n] = \begin{cases} x[\frac{n}{M}] & n = Mk \\ 0 & \text{otherwise} \end{cases}$$

That is, we simply insert 0's between the samples from the original sequence to expand the sample rate. Of course, to obtain an intuitively reasonable interpolation we will have to apply some kind of smoothing filter to smoothly interpolate the missing samples, rather than just inserting 0's, but we will consider this later and as a separate operation.

The DTFT of $y[n]$ may easily be deduced from that of $x[n]$ as follows:

$$\begin{aligned} \hat{y}(\omega) &= \sum_n y[n]e^{-j\omega n} \\ &= \sum_n x[n]e^{-j\omega(Mn)} \\ &= \hat{x}(M\omega) \end{aligned}$$

Recall that the DTFT is defined over $(-\pi, \pi)$, but that the DTFT equation yields a function which is periodic with period 2π. Thus, the above equation is to be interpreted as follows. Write down the Fourier transform of $x[n]$ over the interval $(-\pi, \pi)$. Periodically extend it out to the interval $(-M\pi, M\pi)$. Finally, shrink this interval down into the interval $(-\pi, \pi)$ corresponding to $\hat{y}(\omega)$. These operations are illustrated in Figure 2.

Now suppose that we apply the ideal low-pass filter $h_m^{(M)}$, defined above, to the interpolated sequence $y[n]$. The effect is to select only a single period from the M periods present in $\hat{y}(\omega)$. The output, $\hat{z}(\omega) = \hat{y}(\omega)\hat{h}_m^{(M)}(\omega)$, can be related to $\hat{x}(\omega)$ as follows. The spectrum of $\hat{x}(\omega)$ is first shrunk down to the interval $\left(-\frac{\pi}{M}, \frac{\pi}{M}\right)$ and shifted into the passband of the filter, $\hat{h}_m^{(M)}(\omega)$. Some consideration (draw some sketches) should reveal that for even integers, m, the frequencies near DC in $\hat{x}(\omega)$ shift to frequencies at the

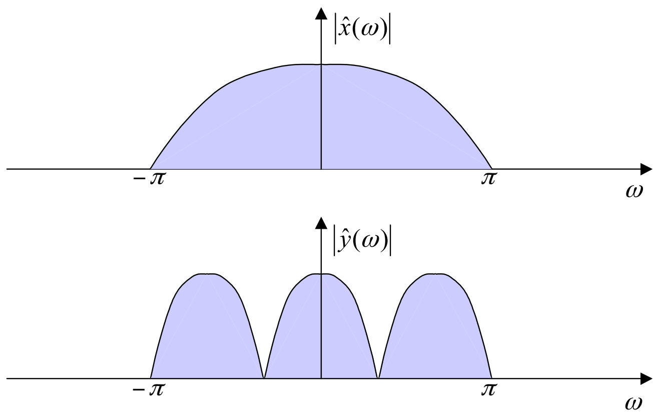

Figure 2: *Upsampling by $M = 3$ in the frequency domain.*

lower edge of the pass band centred at ω_m, while frequencies near $\omega = \pi$ in $\hat{x}(\omega)$ shift to the upper edge of the pass band. On the other hand, for odd integers, m, frequencies near DC in $\hat{x}(\omega)$ shift to the upper edge of the pass band, with frequencies near Nyquist shifting to the lower edge of the passband. These frequency shifting operations exactly complement those described above for down-sampling.

Often we are interested in avoiding frequency shifts, i.e. interpolating to baseband, and so we will use the low-pass interpolator $h_0^{(M)}$. As for decimation, in practice we cannot build ideal bandpass filters and so we must approximate them by appropriate FIR or IIR filters. The result is that the interpolated and filtered signal will generally occupy a larger fraction of the spectrum than $\frac{1}{M}$.

3 Narrow Band Sampling

Putting the ideas of the last two sections together, using only the ideal filters $h_m^{(M)}[n]$ defined above, we can build a system which represents narrow band signals using a reduced number of samples. Specifically, consider the following system:

$$x[n] \longrightarrow \boxed{\star\, h_m^{(M)}} \longrightarrow \boxed{\downarrow M} \longrightarrow y[n] \longrightarrow \boxed{\uparrow M} \longrightarrow \boxed{\star\, M \cdot h_m^{(M)}} \longrightarrow z[n]$$

It is not hard to see that $z[n]$ is simply a band-pass filtered version of $x[n]$. That is,

$$\hat{z}(\omega) = \hat{x}(\omega)\hat{h}_m^{(M)}(\omega) \tag{2}$$

The extra scale factor, M, in the interpolating output filter serves to combat the attenuation by M which occurs during down-sampling, as suggested by equation (1). It is perhaps more fruitful to understand the need for this scale factor in the time domain. All of our filters have unit magnitude response in their passband. Taking the low-pass band, with filter $h_0^{(M)}$, this means that a DC input will be passed unaffected by $h_0^{(M)}$, and its samples will appear in the decimated sequence, $y[n]$, without any scaling. After inserting zeros for the missing samples, the up-sampled signal has an average amplitude which is $\frac{1}{M}$ that of the input signal (due to the zeros), so after filtering by the low-pass filter, $h_0^{(M)}$, the result will be identical to $x[n]$ except that it has been attenuated by M.

Multiplying by M during interpolation is thus seen to be essential if the input, down-sampled, and interpolated signals are all to have identical amplitudes. Similar arguments hold for inputs within the passband of any of the filters, $h_m^{(M)}$.

It is important to bear in mind that equation (2) holds exactly only when the filters are ideal. In practice, we cannot realize ideal band-pass filters and so we will have some aliasing distortion in the recovered signal, $z[n]$.

4 Rational Sampling Rate Changes

Here we are interested in changing the sampling rate by a rational factor, $\frac{M_u}{M_d}$. This can be accomplished by first increasing the sampling rate by M_u and then decreasing the sampling rate again by M_d. It could also be accomplished by first decreasing the sampling rate and then increasing it again, but this latter approach would discard too much information, since the intermediate representation would have a lower sampling rate than either the input or the output. Because we are almost invariably interested in baseband, we will use the ideal low-pass filters, $h_0^{(M_u)}$ and $h_0^{(M_d)}$, as illustrated below:

$$x[n] \longrightarrow \boxed{\uparrow M_u} \longrightarrow \boxed{\star\ M_u \cdot h_0^{(M_u)}} \longrightarrow \boxed{\star\ h_0^{(M_d)}} \longrightarrow \boxed{\downarrow M_d} \longrightarrow z[n]$$

Using the same arguments as above, the scale factor, M_u, in the initial interpolation filter serves to ensure that the DC levels of the input, output and all intermediate signals are identical. Evidently, we can collapse the cascade of two digital filters into a single filter with impulse response

$$h[n] = M_u \cdot \left(h_0^{(M_u)} \star h_0^{(M_d)}\right)[n]$$

yielding the system illustrated below:

$$x[n] \longrightarrow \boxed{\uparrow M_u} \longrightarrow w[n] \longrightarrow \boxed{\star h} \longrightarrow y[n] \longrightarrow \boxed{\downarrow M_d} \longrightarrow z[n]$$

Now it is clear that quite a lot of computation is wasted if we implement the filtering system directly as shown. This is because the input to the filter, $h[n]$, contains mostly 0's and most of the output of the filters outputs are discarded. To reduce the computational load, it is helpful to consider the so-called "polyphase" components of the signals.

To this end, define

$$x_m^{(M)}[n] \triangleq x[nM+m], \quad \text{for } m = 0, 1, 2, \ldots, M-1$$

to be the m^{th} polyphase component of the signal, $x[n]$. Essentially, we are viewing $x[n]$ as the interleaving of M sub-sequences, $x_m^{(M)}[n]$. Consistent with our earlier notation, let $x_m^{(M)\prime}[n]$ be the sequence obtained by up-sampling $x_m^{(M)}[n]$ by the factor M. Then $x[n]$ may be written in terms of its up-sampled and delayed polyphase components as

$$x[n] = \sum_{m=0}^{M-1} x_m^{(M)\prime}[n-m]$$

With this in mind, we can consider how each up-sampled and delayed polyphase component passes through the rational sample-rate conversion system above, adding up the separate contributions to recover the output sequence. Consider the m^{th} input polyphase component, $x_m^{(M_d)}[n]$, with respect to the down-sampling factor, M_d. After up-sampling, the contribution to $w[n]$ satisfies

$$\begin{aligned} w_m[n] &= x_m^{(M_d)}[k], \quad \text{if } n = M_u(M_d k + m) \\ w_m[n] &= 0, \quad \text{for all } n \text{ not of the form } n = M_u(M_d k + m) \end{aligned}$$

After the filtering operation, the contribution to $y[n]$ then satisfies

$$y_m[n] = \sum_k x_m^{(M_d)}[k] \cdot h[n - M_u(M_d k + m)]$$

Finally, the decimation operation leaves us with a contribution to the output, $z[n]$, of

$$z_m[n] = \sum_k h[M_d n - M_u(M_d k + m)] \cdot x_m^{(M_d)}[k]$$

Summing these contributions leaves us with

$$z[n] = \sum_{m=0}^{M_d-1} \sum_k h[M_d n - M_u m - M_u M_d k] \cdot x_m^{(M_d)}[k]$$

The above system can be understood more easily if we identify the polyphase components of the output sequence, $z[n]$. Specifically, the p^{th} polyphase component of $z[n]$, with respect to the up-sampling factor, M_u, satisfies

$$\begin{aligned} z_p^{(M_u)}[n] &= z[M_u n + p] \\ &= \sum_{m=0}^{M_d-1} \sum_k h[M_d(M_u n + p) - M_u m - M_u M_d k] \cdot x_m^{(M_d)}[k] \\ &= \sum_{m=0}^{M_d-1} \sum_k h[M_u M_d (n-k) + M_d p - M_u m] \cdot x_m^{(M_d)}[k] \\ &= \sum_{m=0}^{M_d-1} \sum_k h_{p,m}[n-k] x_m^{(M_d)}[k] \end{aligned}$$

where the impulse response, $h_{p,m}[n]$, is given by

$$h_{p,m}[n] = h[M_u M_d n + M_d p - M_u m]$$

Thus, each polyphase component of the output sequence, $z[n]$, is obtained as a sum of filtered input polyphase components, where the filter coefficients are themselves polyphase components of the filter, $h_r[n]$. This convolution formulation involves no decimation or interpolation whatsoever and so it identifies the underlying computational complexity of the system. Figure 3 illustates these operations.

The above considerations may be applied to simple up-sampling or down-sampling systems with integral sampling factors, by setting M_d or M_u equal to 1, as appropriate.

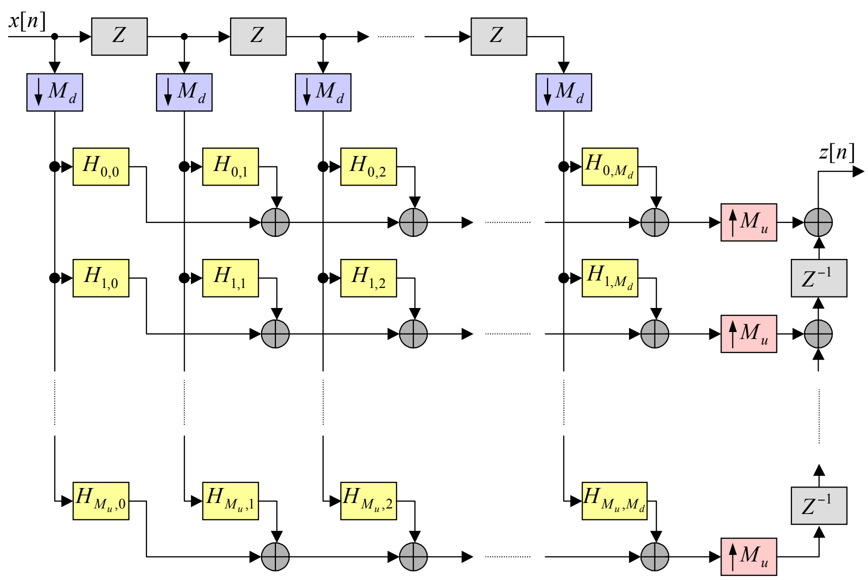

Figure 3: *Rational sampling system, implemented as a network (matrix) of polyphase filters, $h_{p,m}[n]$.*

5 Cascade Decimation/Interpolation Systems

Suppose we wish to reduce the sampling rate of some sequence $x[n]$, by a very large factor, M, e.g. $M = 256$. Following the above discussion, we would design a filter to approximate the ideal low-pass filter,

$$h_0^{(M)}[n] = \frac{1}{M}\,\mathrm{sinc}\left(\frac{n}{M}\right)$$

and then subsample the output. Following the discussion of efficient implementations above, we would divide $x[n]$ into M polyphase components and filter each of these polyphase components, $x_m^{(M)}[n] = x[nM + m]$, using the filter

$$h_m[n] = h_0^{(M)}[Mn - m]$$

summing the results to form the down-sampled output sequence.

Suppose that we build an FIR approximation to the ideal sinc filter above, whose spatial extent covers L ripples of the sinc impulse response. Then the number of taps in this filter is LM and a direct implementation of the system would require LM multiply-accumulate operations for each input sample. By using the polyphase representation, the polyphase filters each have only L taps and so we require only LM multiply-accumulate operations for each output sample, which is a saving by a factor of M.

To further reduce the cost of the system, we can implement the down-sampling operation as a cascade of two systems, each of which reduces the sampling rate by a factor of $\sqrt{M} = 16$, for a total reduction of $M = 256$. To see how this helps, suppose we design each system using the same principles. Then, the second stage requires only $\sqrt{M}L$ multiple-accumulate operations per output sample. The first stage requires $\sqrt{M}L$ multiply-accumulate operations for each of its output samples, but has $\sqrt{M}$ times as many output samples as the second stage. Consequently, this stage requires ML multiply-accumulate operations per final output sample and we would appear to be worse off than before. However, the first stage's low-pass filter can be a much poorer approximation to the desired sinc function than that of the second stage, since most of the spectrum will be discarded by the second stage. We can tolerate some aliasing in all but the lowest frequencies and we can tolerate far from an ideal passband response, except at the lowest frequencies. Let us suppose that the corresponding reduction in filter length is F. Then the overall system complexity is

$$\sqrt{M}L + \frac{M}{F}L$$

which represents a potentially large computational saving.

This idea can easily be extended to larger cascades when the sampling rate change is very large. Similar methods may be applied to the implementation of digital interpolation systems with very large upsampling factors. The same ideas may also be applied when the signal is not baseband, i.e. when we are approximating a bandpass filter, $h_m^{(M)}[n]$, for some $m \neq 0$, with large M.

6 Filter Banks

The ideas discussed in Section 3 can easily be extended to a representation of the entire signal $x[n]$ in terms of a collection of subbands, using the filter bank system illustrated in Figure 4. Provided we use the ideal band-pass filters $h_m^{(M)}[n]$, it is clear that we will recover the original signal $x[n]$ without error. Notice that the overall sampling rate, taken over all the subbands, is the same as the original sampling rate. Consequently, the subband representation is non-redundant; we also say that the filter bank system is "maximally decimated."

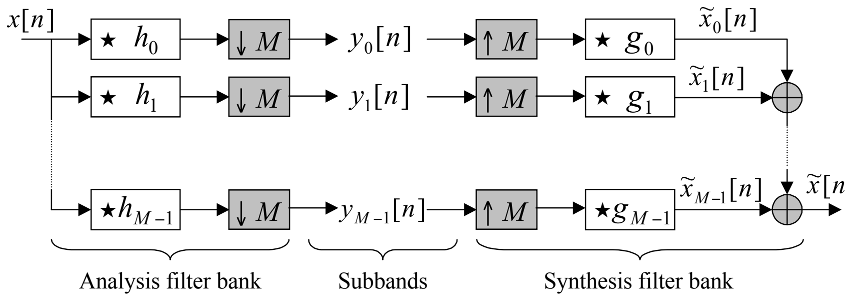

Figure 4: *M-channel subband filter bank.*

Subband systems were first introduced for coding of audio waveforms, since they allow each frequency band to be processed/coded using its own specific techniques, which are sensitive to the statistical properties of the samples in that frequency band and to the auditory significance of the particular frequency band. Since then, subband systems have found wide applicability in image and video compression algorithms. They also represent a computationally efficient framework for certain types of signal processing operations such as equalization, signal detection, and auditory scene analysis.

Since we cannot build ideal bandpass filters, each subband signal will generally contain aliasing components and so we cannot reconstruct any given frequency band without aliasing distortion. It is thus reasonable to expect that we would not be able to reconstruct the original signal without aliasing distortion. It turns out, however, that it is possible to choose the subband filters in such a way that the aliasing content from each subband channel perfectly cancels out. In this case, the final reconstructed signal, $\tilde{x}[n]$, is a true filtered version of the input signal, $x[n]$, even though the individual bandpass signals, $\tilde{x}_m[n]$, necessarily contain aliasing components. Also, the filters $g_m[n]$ used in the synthesis filter bank will not generally be identical to (or scaled versions of) the filters $h_m[n]$ used in the analysis filter bank.

It is actually possible to choose the analysis and synthesis filters in such a way as to achieve perfect reconstruction, i.e. $\tilde{x}[n] = x[n]$. We will briefly examine conditions for alias free reconstruction in the next section. Then, in the next set of lecture notes, we will consider the design and implementation of "Perfect Reconstruction" (PR) subband systems, and the closely associated Discrete Wavelet Transform (DWT).

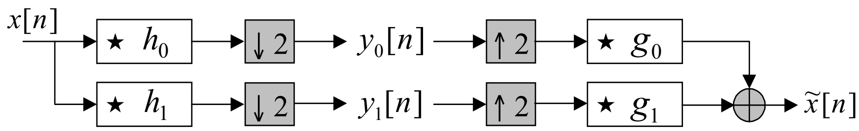

Figure 5: *Two channel subband transform and its inverse.*

6.1 Alias Free Reconstruction

We restrict our attention here to filter banks with $M = 2$ channels for convenience, using practically realizable analysis filters, $h_0[n]$ and $h_1[n]$, and synthesis filters, $g_0[n]$ and $g_1[n]$, as illustrated in Figure 5. The analysis system is often referred to as a subband transform, with the synthesis system representing the inverse transform.

Using the multi-rate relationships derived above we can write

$$\hat{y}_0(\omega) = \frac{1}{2}\left(\hat{x}(\frac{\omega}{2})\hat{h}_0(\frac{\omega}{2}) + \hat{x}(\frac{\omega}{2}-\pi)\hat{h}_0(\frac{\omega}{2}-\pi)\right)$$

and

$$\hat{y}_1(\omega) = \frac{1}{2}\left(\hat{x}(\frac{\omega}{2})\hat{h}_1(\frac{\omega}{2}) + \hat{x}(\frac{\omega}{2}-\pi)\hat{h}_1(\frac{\omega}{2}-\pi)\right)$$

Then

$$\begin{aligned}\hat{\tilde{x}}(\omega) &= \frac{1}{2}\left(\hat{x}(\omega)\hat{h}_0(\omega)\hat{g}_0(\omega) + \hat{x}(\omega-\pi)\hat{h}_0(\omega-\pi)\hat{g}_0(\omega)\right) \\ &+ \frac{1}{2}\left(\hat{x}(\omega)\hat{h}_1(\omega)\hat{g}_1(\omega) + \hat{x}(\omega-\pi)\hat{h}_1(\omega-\pi)\hat{g}_1(\omega)\right)\end{aligned} \tag{3}$$

To ensure alias free reconstruction, we need to eliminate the components involving frequency shifts. That is, we require

$$\hat{x}(\omega-\pi)\left(\hat{h}_0(\omega-\pi)\hat{g}_0(\omega) + \hat{h}_1(\omega-\pi)\hat{g}_1(\omega)\right) = 0$$

For this to work with any input sequences we will need to choose the filters to satisfy

$$\hat{h}_0(\omega-\pi)\hat{g}_0(\omega) = -\hat{h}_1(\omega-\pi)\hat{g}_1(\omega)$$

In the case of ideal filters, h_0 and g_0 were both ideal lowpass filters and so $\hat{h}_0(\omega-\pi)$ was an ideal high-pass filter; as a result, we had $\hat{h}_0(\omega-\pi)\hat{g}_0(\omega) = 0$. Similarly, $\hat{h}_1(\omega-\pi)\hat{g}_1(\omega) = 0$, and the alias-free condition is easily seen to hold.

For non-ideal filters, one choice which will satisfy the alias free condition is

$$\begin{pmatrix} \hat{g}_0(\omega) \\ \hat{g}_1(\omega) \end{pmatrix} = \hat{f}(\omega) \begin{pmatrix} \hat{h}_1(\omega - \pi) \\ -\hat{h}_0(\omega - \pi) \end{pmatrix}$$

where $\hat{f}(\omega)$ is an arbitrary frequency response.

The above condition may be written in the time domain as

$$\begin{pmatrix} g_0[n] \\ g_1[n] \end{pmatrix} = f[n] \star \begin{pmatrix} (-1)^n h_1[n] \\ (-1)^{n+1} h_0[n] \end{pmatrix}$$

or in the Z-transform domain as

$$\begin{pmatrix} G_0(z) \\ G_1(z) \end{pmatrix} = F(z) \begin{pmatrix} H_1(-z) \\ -H_0(-z) \end{pmatrix}$$

Clearly, if the analysis filters are FIR and $f[n]$ is FIR, then so are the synthesis filters with this choice. Similarly, if the analysis filters have a rational Z-transform, then so do the synthesis filters. In fact, it turns out that choices of the above form are the only way of achieving alias free reconstruction with FIR analysis and synthesis filters.

It also turns out that if both the analysis and synthesis filters are to be FIR, perfect reconstruction will not be possible unless $F(z)$ is of the form

$$F(z) = \alpha z^{-\kappa}$$

where κ is a delay term, and α is a scale factor. In view of this, we will always make the above selection, which leads to

$$\begin{pmatrix} G_0(z) \\ G_1(z) \end{pmatrix} = \alpha z^{-\kappa} \begin{pmatrix} H_1(-z) \\ -H_0(-z) \end{pmatrix} \tag{4}$$

Filter bank systems which satisfy this relationship are known as Quadrature Mirror Filter (QMF) banks. The term arises from the fact that there are four filters (hence "quadrature") with g_0 and h_1 exhibiting mirror image frequency responses and g_1 and h_0 also exhibiting mirror image frequency responses, up to a delay and a scale factor. The low-pass synthesis filter is obtained from the high-pass analysis filter by swapping low and high frequencies. The high-pass filter is obtained from the low-pass analysis filter in a similar manner.

6.2 Classical QMF Definition

The first QMF filter banks described by Croisier, Esteban and Garland in 1976 made the selection

$$G_0(z) = H_0(z) \tag{5}$$

This is a natural selection in view of the fact that the ideal analysis and synthesis filters are identical to each other. Combining with equation (4), using $\alpha = 1$ and $\kappa = 0$ yields the additional relationships

$$\begin{aligned} H_1(z) &= H_0(-z) \\ G_1(z) &= -H_1(z) \end{aligned}$$

so that $G_1(z)$ and $H_1(z)$ cannot be identical, as they are in the ideal case.

Johnson designed a family of even length linear phase subband filters using these relationships which have been quite popular in the literature. Unfortunately, these filters do not exhibit the perfect reconstruction property; they are designed to minimize the error between $x[n]$ and $\tilde{x}[n]$, while having good pass and stop band properties.

6.3 QMF Filters with Perfect Reconstruction

It turns out that it is not possible to achieve perfect reconstruction by making the choice in equation (5). A slight modification, however, does lead to a whole family of filter banks which exhibit the perfect reconstruction property. By setting

$$G_0(z) = H_0\left(z^{-1}\right) \tag{6}$$

and making the selection $\alpha = 1$ and $\kappa = 1$ in equation (4), we obtain

$$H_1(z) = -zH_0\left(-z^{-1}\right) \tag{7}$$

$$G_1(z) = H_1\left(z^{-1}\right) \tag{8}$$

In this case, the low-pass analysis and synthesis filters are mirror images of one another, as are the high-pass analysis and synthesis filters. This property is also shared by the ideal filters considered previously.

With aliasing removed, equation (3) yields

$$\tilde{X}(z) = \frac{1}{2} X(z) \left(H_0(z) G_0(z) + H_1(z) G_1(z)\right)$$

so for perfect reconstruction, we require

$$H_0(z) G_0(z) + H_1(z) G_1(z) = 2$$

Substituting equations (6), (7) and (8) into the above, we get

$$\begin{aligned} 2 &= H_0(z)\,H_0\left(z^{-1}\right) + H_1(z)\,H_1\left(z^{-1}\right) \\ &= H_0(z)\,H_0\left(z^{-1}\right) + H_0(-z)\,H_0\left(-z^{-1}\right) \end{aligned}$$

Or, in the frequency domain,

$$2 = \left|\hat{h}_0(\omega)\right|^2 + \left|\hat{h}_0(\omega - \pi)\right|^2 \tag{9}$$

This tells us that perfect reconstruction can be achieved by designing the low-pass analysis filter, $h_0[n]$, to satisfy the so-called "power complementary" property in equation (9). In the next set of lecture notes we shall look at one strategy for designing filters with this property.