

Chapter 3

Downsampling and Upsampling

3.1 The Matrices for Downsampling and Upsampling

The previous chapter was about individual filters. The next chapter combines two or more filters into a *filter bank*. The idea is to separate the incoming signal into frequency bands. Often we will use a bank of two filters to explain a point clearly. A lowpass filter C and a highpass filter D will split the signal into two parts. Those parts can be compressed and coded separately (and efficiently). Then they can be transmitted and the signal can be recovered.

When the synthesis bank recovers the input signal exactly (apart from a time delay), it is a *perfect reconstruction* filter bank. These PR banks interest us most.

One problem to overcome. We don't want two full length signals in place of one, if we can help it. It is not desirable to double the volume of data. The information has not doubled; the outputs Cx and Dx from the two filters must be redundant.

The solution to this problem is beautiful. We keep only half of the outputs Cx and Dx . More precisely, we *downsample* those vectors by removing every other component. This operation is called *decimation*. Downsampling is represented by the symbol $(\downarrow 2)$ (pronounced "down two"):

$$(\downarrow 2) \begin{bmatrix} \cdot \\ y(-1) \\ y(0) \\ y(1) \\ y(2) \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ y(-2) \\ y(0) \\ y(2) \\ y(4) \\ \cdot \end{bmatrix}.$$

Make no mistake. This operation is not invertible. Most vectors y cannot be recovered from $(\downarrow 2)y$. The odd-numbered components are lost. Normally the even-numbered components of both $y = Cx$ and $y = Dx$ will be needed to recover all components of x .

We note that recovery of x from $(\downarrow 2)x$ is possible if the transform $X(\omega)$ is zero over a half-band of frequencies. Such a signal is "band-limited". It may be limited to the upper half-band or the lower half-band:

$$X(\omega) = 0 \text{ for } 0 \leq |\omega| < \frac{\pi}{2} \quad \text{or} \quad X(\omega) = 0 \text{ for } \frac{\pi}{2} \leq |\omega| < \pi.$$

It is amazing that for band-limited signals we can recover the odd-numbered components from the even-numbered components. This is achieved by the *Shannon Sampling Theorem* in Section 2.3.

Example 3.1. Downsampling a vector of alternating signs produces a vector of constant sign:

$$(\downarrow 2)(\dots, 1, -1, 1, -1, \dots) = (\dots, 1, 1, \dots).$$

The -1 's in the odd-numbered positions have disappeared. You will quickly think of other inputs that give the same output. One in particular is the input vector $(\dots, 1, 1, 1, 1, \dots)$ of all ones. Downsampling produces the same vector of ones. It is an eigenvector of the linear operator $(\downarrow 2)$, with eigenvalue $\lambda = 1$.

Another eigenvector is $\delta = (\dots, 0, 0, 1, 0, 0, \dots)$. This also has $\lambda = 1$, and $(\downarrow 2)\delta = \delta$.

It is valuable to compare the alternating and constant inputs. Vector $(\dots, 1, -1, 1, -1, \dots)$ is at the frequency $\omega = \pi$. That is the highest possible frequency in discrete time. No vector can oscillate faster. Its components are $x(n) = e^{i\pi n}$, which is $(-1)^n$. By contrast, the input of all ones is at the lowest frequency $\omega = 0$. Its components are $x(n) = e^{i0n} = 1$.

This is an example of *aliasing*. Two different frequencies, after downsampling, cannot be distinguished. The "alias" of the high frequency $\omega = \pi$ is the low frequency $\omega = 0$.

Certainly we cannot recover the input (which input would it be?) from the common output $(\downarrow 2)x = (\dots, 1, 1, 1, \dots)$. If we know that all frequencies in the input satisfy $|\omega| < \frac{\pi}{2}$, then we can "invert" the downsampling operation. The only possible input would be the constant vector, the DC input with $\omega = 0$.

Example 3.2. When the impulse response $C\delta = (\dots, c(0), c(1), c(2), \dots)$ is downsampled, the odd-numbered responses are lost:

$$(\downarrow 2)C\delta = (\dots, c(0), c(2), c(4), \dots).$$

This is the impulse response of a "decimation filter" — *filtering followed by downsampling*. The impulse δ came at time zero, and downsampling leaves the even coefficients. The response is entirely different if the impulse is delayed — we compute that now. *Downsampling is not time-invariant*. The delayed impulse is $S\delta = (\dots, 0, 1, 0, \dots)$. The response from the time-invariant filter C is similarly delayed:

$$CS\delta = (\dots, 0, c(0), c(1), \dots). \quad (3.1)$$

Because of the delay, downsampling now destroys the even-numbered components $c(0), c(2)$. We pick up the odd components:

$$(\downarrow 2)CS\delta = (\dots, 0, c(1), c(3), \dots).$$

First conclusion: Complete information may still be there if we downsample *two* vectors. Here the vectors were $C\delta$ and its shift $SC\delta = CS\delta$. Those outputs are not as "separated" as we want — the two pieces are probably far from orthogonal. Downsampling $C\delta$ and $CS\delta$ may be a waste of time, but we do see that the whole vector $(c(0), c(1), c(2), c(3), \dots)$ is recoverable from the two samples. Upsampling will be the way to recover it.

A better filter bank will use two different filters C and D . Part of the design problem is to ensure that $(\downarrow 2)Cx$ and $(\downarrow 2)Dx$ have all the information to recover x . Even better if those vectors are orthogonal, which is not true for this easy choice $D = \text{shift of } C$.

Definition 3.1 The n th component of $v = (\downarrow 2)x$ is the $(2n)$ th component of x :

$$v(n) = x(2n). \quad (3.2)$$

We will now identify the "downsampling matrix". *It is the identity matrix with odd-numbered rows removed.* You could say that the identity matrix has been downsampled! The matrix form of downsampling is

$$\begin{bmatrix} \cdot & 1 & & & & \\ & 0 & 0 & 1 & & \\ & & & 0 & 0 & 1 \\ & & & & & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ x(-2) \\ x(-1) \\ x(0) \\ x(1) \\ x(2) \\ \cdot \end{bmatrix} = \begin{bmatrix} x(-2) \\ x(0) \\ x(2) \\ \cdot \end{bmatrix}. \quad (3.3)$$

We are defining the linear operator $(\downarrow 2)$. Notice that the rows of the downsampling matrix are orthonormal. They are unit vectors and their dot products are all zero. The matrix multiplication $(\downarrow 2)(\downarrow 2)^T$ contains all these dot products, so it must be the identity matrix. The matrix product $(\downarrow 2)^T(\downarrow 2)$ is different, as we see below.

The operator $(\downarrow 2)^T$ is important. *The transpose of downsampling is upsampling:*

$$(\downarrow 2)^T = (\uparrow 2). \quad (3.4)$$

Upsampling places zeros into the odd-numbered components:

$$\text{Definition of } (\uparrow 2): \quad (\uparrow 2) \begin{bmatrix} \cdot \\ v(-1) \\ v(0) \\ v(1) \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ v(-1) \\ 0 \\ v(0) \\ 0 \\ v(1) \\ 0 \\ \cdot \end{bmatrix}.$$

We need a two-part formula to describe those components of $u = (\uparrow 2)v$:

$$u(n) = \begin{cases} v(k) & \text{if } n = 2k \\ 0 & \text{if } n = 2k + 1. \end{cases} \quad (3.5)$$

To do this with a matrix, insert zero rows between the rows of I . You could say that *the identity matrix has been upsampled*. The matrix form of upsampling is

$$\begin{bmatrix} \cdot & & & & \\ 1 & & & & \\ & 0 & & & \\ & 0 & & & \\ & & 1 & & \\ & & 0 & & \\ & & 0 & & \\ & & 1 & & \\ & & & 0 & \\ & & & & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ v(-1) \\ v(0) \\ v(1) \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ v(-1) \\ 0 \\ v(0) \\ 0 \\ v(1) \\ 0 \\ \cdot \end{bmatrix} \quad (3.6)$$

The matrix in (3.6) is the transpose of the matrix in (3.3). If we multiply them we do get the

identity matrix, verifying $(\downarrow 2)(\uparrow 2) = I$:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.7)$$

If we upsample and then downsample, we put in zeros and take them out. The original x is recovered. This is not the usual order of the steps!

Normally we downsample and then upsample. *This product is not the identity matrix.* The output $(\uparrow 2)(\downarrow 2)x$ is different from x . Watch the steps:

$$\text{Down: } (\downarrow 2)x = \begin{bmatrix} x(-2) \\ x(0) \\ x(2) \\ \vdots \end{bmatrix}. \quad \text{Then up: } (\uparrow 2)(\downarrow 2)x = \begin{bmatrix} x(-2) \\ 0 \\ x(0) \\ 0 \\ x(2) \\ 0 \\ \vdots \end{bmatrix}. \quad (3.8)$$

In this usual order, $(\downarrow 2)$ removes the odd-numbered components and $(\uparrow 2)$ puts in zeros. The product $(\uparrow 2)(\downarrow 2)$ replaces the lost components by zeros:

$$(\uparrow 2)(\downarrow 2) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

Thus $(\downarrow 2)$ and $(\uparrow 2)$ are one-sided inverses but not two-sided inverses. This cannot happen for square matrices of finite size. For n by n matrices, $AB = I$ means that $BA = I$. For rectangular matrices and infinite matrices, a *one-sided inverse* is possible. Downsampling is a very important example.

The matrices for $(\downarrow 2)$ and $(\uparrow 2)$ came from downsampling and upsampling the columns of I . Here is another way to describe those same matrices:

- The $(\downarrow 2)$ matrix has the rows of I with a **double shift**.
- The $(\uparrow 2)$ matrix has the columns of I with a **double shift**.

Problem Set 3.1

1. Downsampling the alternating vector $x = (\dots, 1, -1, 1, -1, 1, \dots)$ produces the vector $(\downarrow 2)x = (\dots, 1, 1, 1, \dots)$. If you delay x to $Sx(n) = x(n-1)$, what is $(\downarrow 2)Sx$?
2. Describe the operator $(\uparrow 2)(\downarrow 2)(\uparrow 2)(\downarrow 2)$.
3. Describe the operators $(\downarrow 3)$ and $(\uparrow 3)$ and $(\uparrow 3)(\downarrow 3)$.
4. What are the components of $(\uparrow 3)(\downarrow 2)x$ and $(\downarrow 2)(\uparrow 3)x$?

3.2 Subsampling in the Frequency Domain

For a pure exponential signal $x(k) = e^{ik\omega}$, the effect of downsampling is particularly clear. The k th component of $v = (\downarrow 2)x$ is $v(k) = e^{i2k\omega}$. This is a pure exponential with frequency 2ω . Frequencies are doubled by downsampling.

It looks as if $V(2\omega) = X(\omega)$, in other words $V(\omega) = X(\frac{\omega}{2})$. *That is wrong.*

Suppose x' is another pure exponential, but with frequency $\omega + \pi$. Then $x'(k) = e^{ik(\omega+\pi)}$ is downsampled to $v'(k) = e^{i2k(\omega+\pi)}$. The factor $e^{i2k\pi}$ is always 1. This output is the same $e^{i2k\omega}$ as before! Adding π to the frequency only reversed the signs of the *odd*-numbered components of x' . Those components disappear anyway in downsampling, and $v'(k) = v(k)$. The frequency 2ω appears by doubling ω , and it *also* appears by doubling $\omega + \pi$.

In other words, ω enters the downsampled vector v not only by doubling $\frac{\omega}{2}$ but also by doubling $\frac{\omega}{2} + \pi$. There are two sources, not just one, contributing to $V(\omega)$:

Theorem 3.1 *The transform of $v = (\downarrow 2)x$ is $V(\omega) = \frac{1}{2} [X(\frac{\omega}{2}) + X(\frac{\omega}{2} + \pi)]$.*

Proof. Downsampling in the time domain is very direct:

$$v = (\downarrow 2)x \text{ means } v(k) = x(2k). \quad (3.10)$$

The Fourier transform of v is

$$V(\omega) = \sum v(k)e^{-ik\omega} = \sum x(2k)e^{-ik\omega}. \quad (3.11)$$

It is natural to set $n = 2k$, but do it carefully. The reverse direction $k = \frac{n}{2}$ is not safe. You might think that our sum is $\sum x(n)e^{-in\omega/2}$, and the frequency variable ω is just halved. *But the sum is only over even $n = 2k$.* So we start again.

Let u be the vector x with its odd-numbered components set to zero:

$$u(n) = \begin{cases} x(n), & n \text{ even} \\ 0, & n \text{ odd} \end{cases} \text{ so that } u = (\dots, x(0), 0, x(2), 0, \dots). \quad (3.12)$$

Certainly $(\downarrow 2)u$ equals $(\downarrow 2)x$. The transform of u has two terms. The second term includes $(-1)^n$ or $e^{-in\pi}$ so that addition knocks out odd n :

$$u(n) = \sum_{n \text{ even}} x(n)e^{-in\omega} = \frac{1}{2} \sum_{\text{all } n} x(n)e^{-in\omega} + \frac{1}{2} \sum_{\text{all } n} x(n)e^{-in(\omega+\pi)}. \quad (3.13)$$

In frequency, this says that

$$U(\omega) = \frac{1}{2} [X(\omega) + X(\omega + \pi)]. \quad (3.14)$$

Now $V(\omega)$ comes safely from $U(\omega)$ by halving the frequency, because $u(n)$ only involves even n :

$$V(\omega) = U\left(\frac{\omega}{2}\right). \quad (3.15)$$

Equations (3.15)–(3.16) mean that the downsampled signal $v = (\downarrow 2)x$ has transform

$$V(\omega) = \frac{1}{2} [X\left(\frac{\omega}{2}\right) + X\left(\frac{\omega}{2} + \pi\right)]. \quad (3.16)$$

This is downsampling in the frequency domain. It is not time-invariant, because $X(\omega)$ is not just multiplied by $C(\omega)$. The rate changes and a new term appears.

Figures 3.1 and 3.2 and 3.3 show extreme aliasing and no aliasing and typical aliasing. You can see the two terms in Figure 3.1, where $X(\omega)$ is a sawtooth function. The average in $U(\omega)$ is a constant. That second term is responsible for *aliasing*: $X(\frac{\omega}{2})$ overlaps $X(\frac{\omega}{2} + \pi)$ in $V(\omega)$. Their sum happens to be constant, so this special input x gives $v = (\downarrow 2)x = \frac{1}{2}\delta$. All its even-numbered samples are zero, except $x(0) = \frac{1}{2}$.

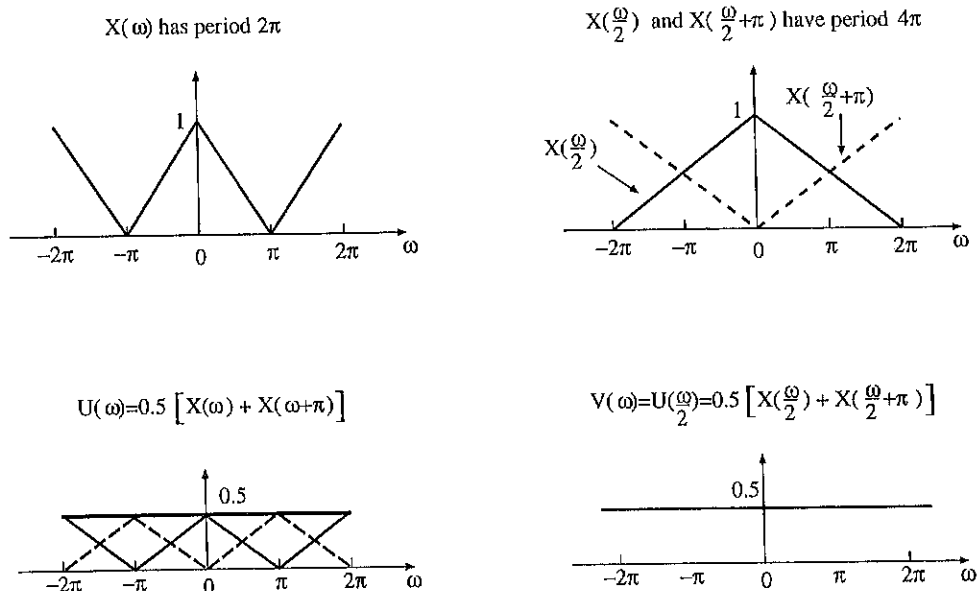


Figure 3.1: Downsampling stretches $X(\omega)$ to $X(\frac{\omega}{2})$ and it also creates the alias $X(\frac{\omega}{2} + \pi)$. Because the alias overlaps, we cannot recover $X(\omega)$ from $U(\omega)$ or from the downsampled $V(\omega)$.

In Figure 3.2, downsampling does not lead to aliasing. The input $X(\omega)$ is properly band-limited. The stretched transform $X(\frac{\omega}{2})$ does not overlap the aliasing term $X(\frac{\omega}{2} + \pi)$. The outputs $V(\omega)$ and $U(\omega)$ from downsampling and upsampling still contain this aliasing part! But a non-overlapping alias can be filtered away.

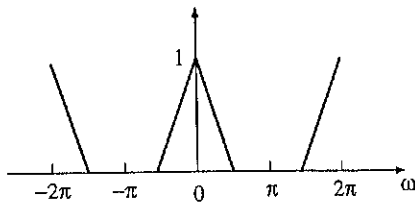
It is striking that the adjective “halfband” is used in both of these extreme cases, but in different ways:

x in Figure 3.1 is a *halfband impulse response* ($x(2n) = 0$ for $n \neq 0$).

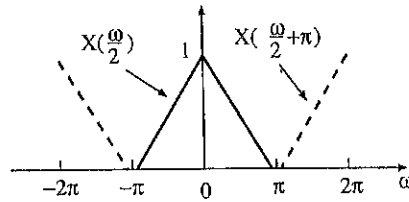
x in Figure 3.2 is a *halfband signal* ($X(\omega) = 0$ for $\frac{\pi}{2} \leq |\omega| < \pi$).

Figure 3.3 is somewhere between Figure 3.1 and Figure 3.2. *Aliasing is expected*. The original x cannot be recovered from $(\downarrow 2)x$. But we don't expect the unusual result in Figure 3.1, where downsampling produced a pure impulse.

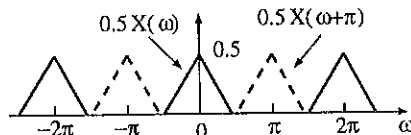
$X(\omega)$ is bandlimited to $|\omega| < \frac{\pi}{2}$



$X(\frac{\omega}{2})$ and $X(\frac{\omega}{2} + \pi)$ have period 4π



$$U(\omega) = 0.5 [X(\omega) + X(\omega + \pi)]$$



$$V(\omega) = U(\frac{\omega}{2}) = 0.5 [X(\frac{\omega}{2}) + X(\frac{\omega}{2} + \pi)]$$

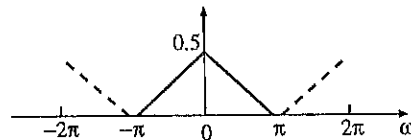


Figure 3.2: Downsampling this bandlimited signal stretches its transform to $[-\pi, \pi]$, no aliasing.

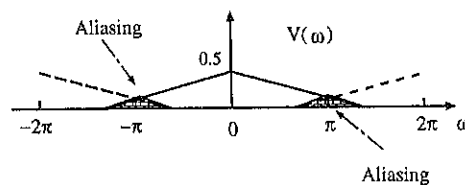
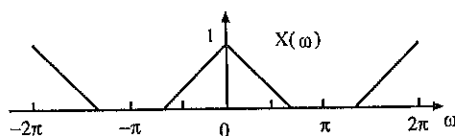


Figure 3.3: Typical aliasing from $(\downarrow 2)$. The stretched $X(\frac{\omega}{2})$ overlaps $X(\frac{\omega}{2} + \pi)$.

Upsampling in the Frequency Domain

In the time domain, upsampling needs separate formulas for even $n = 2k$ and odd $n = 2k + 1$:

$$u = (\uparrow 2)v \text{ means } \begin{cases} u(2k) = v(k) \\ u(2k + 1) = 0. \end{cases} \quad (3.17)$$

In the frequency domain this becomes simple. Where downsampling $v(k) = x(2k)$ goes from one term in the time domain to two terms in frequency, upsampling goes from two terms to one. The zeros in $u(2k + 1) = 0$ contribute nothing to $U(\omega)$. We now establish the upsampling formula in the ω domain:

Theorem 3.2 The transform of $u = (\uparrow 2)v$ is

$$U(\omega) = V(2\omega). \quad (3.18)$$

Proof. Only the even indices $n = 2k$ enter because $u(2k + 1) = 0$:

$$\begin{aligned} U(\omega) &= \sum u(n)e^{-in\omega} = \sum u(2k)e^{-i2k\omega} \\ &= \sum v(k)e^{-i2k\omega} = V(2\omega). \end{aligned} \quad (3.19)$$

Where $V(\omega)$ has period 2π , this new function $V(2\omega)$ has period π . The original graph is compressed into $|\omega| \leq \frac{\pi}{2}$. An image of that compressed graph appears next to it. Upsampling creates an imaging effect! Figure 3.4 shows how two compressed copies of the graph of $V(\omega)$ appear in the spectrum of $U(\omega)$.

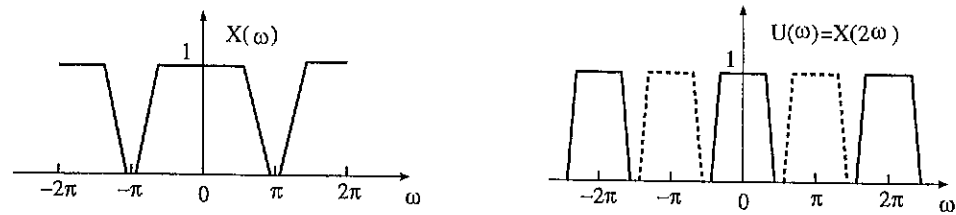


Figure 3.4: The upsampled spectrum $U(\omega) = V(2\omega)$ has compressed images of $V(\omega)$.

This imaging is the *opposite* of aliasing. In aliasing, two input frequencies ω and $\omega + \pi$ give the same output. In imaging, one input frequency ω is responsible for two outputs—at frequency $\frac{\omega}{2}$ and also at $\frac{\omega}{2} + \pi$. Upsampling produces imaging where downsampling produces aliasing.

Upsampling after Downsampling

The analysis bank includes downsampling. Then the synthesis bank includes upsampling. It is very common to see those two operations together:

$$v = (\downarrow 2)x \quad \text{and then} \quad u = (\uparrow 2)v \quad \text{give} \quad u = (\uparrow 2)(\downarrow 2)x.$$

Theorem 3.3 The transform of $u = (\uparrow 2)(\downarrow 2)x$ is

$$U(\omega) = \frac{1}{2} [X(\omega) + X(\omega + \pi)]. \quad (3.20)$$

This was equation (3.14). The aliasing $X(\omega + \pi)$ comes from $(-1)^n x(n)$. When n is odd, this cancels $x(n)$. The average of $x(n)$ and $(-1)^n x(n)$ is

$$u(n) = \begin{cases} x(n), & n \text{ even} \\ 0, & n \text{ odd.} \end{cases} \quad (3.21)$$

In the frequency domain the proof has two steps:

$$\begin{aligned} \text{down:} \quad V(\omega) &= \frac{1}{2} \left[X\left(\frac{\omega}{2}\right) + X\left(\frac{\omega}{2} + \pi\right) \right] \\ \text{then up:} \quad U(\omega) &= V(2\omega) = \frac{1}{2} [X(\omega) + X(\omega + \pi)]. \end{aligned}$$

Conclusion: $(\uparrow 2)(\downarrow 2)$ produces *aliasing and also imaging*. Figure 3.5 shows them both; the image of the alias overlaps the original. We can't determine the original $X(\omega)$ from $U(\omega)$.

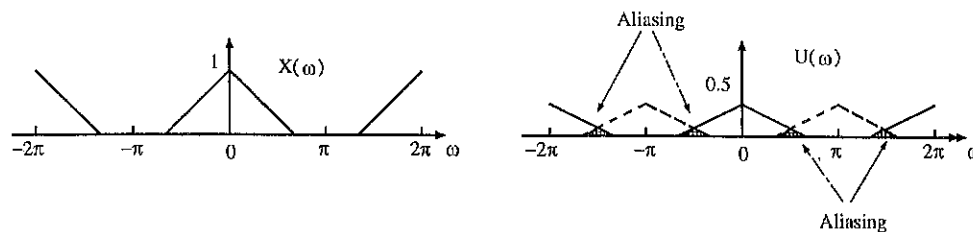
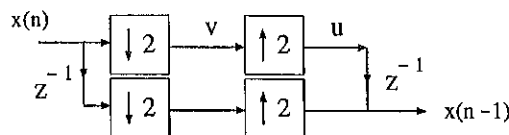


Figure 3.5: The spectrum is stretched by downsampling and compressed by upsampling. The alias from downsampling generally overlaps the image from upsampling.

Filter Bank without Filters

The idea of a filter bank without filters sounds absurd. But this is a very useful example. The filters come from the identity matrix and don't do anything (except possibly a delay). All the activity comes from downsampling and upsampling. It is good to see how the original signal x is reconstructed.

Here is the block form of the simple filter bank, with no filters:



In words, the upper channel downsamples and upsamples. It produces the vector u described above, with all odd components of x replaced by zeros. Then this u is delayed before output.

The lower channel delays x at the start. Then downsampling followed by upsampling removes the *even*-numbered components $x(2k)$. The combined output is just a delay of the input:

$$\begin{aligned} \text{delay of } (\uparrow 2)(\downarrow 2)x &= \begin{bmatrix} 0 \\ x(0) \\ 0 \\ x(2) \\ \vdots \end{bmatrix} + \begin{bmatrix} x(-1) \\ 0 \\ x(1) \\ 0 \\ \vdots \end{bmatrix} = \text{delay of } x. \\ (\uparrow 2)(\downarrow 2)(\text{delay of } x) & \end{aligned}$$

A delay multiplies the transform by $e^{-i\omega}$. The delay in a filter bank is indicated by z^{-1} . Instead of the Fourier transform we more often use the z -transform. A delay multiplies by z^{-1} , and a two-step delay multiplies by z^{-2} . An advance (non-causal!) multiplies by z .

We will next express $(\downarrow 2)$ and $(\uparrow 2)$ in the z -domain. And we take this opportunity to do the same for $(\downarrow M)$ and $(\uparrow M)$. Downsampling and upsampling involve every M th component of the signals in an M -channel filter bank.

Problem Set 3.2

1. Draw $V(\omega)$ and $U(\omega)$ when $X(\omega)$ is a constant function. What vector x has this transform? What is the result of downsampling that x ?
2. What are the usual periods of the functions $X(\omega)$, $X(\frac{\omega}{2})$, and $V(\omega)$?

3. For downsampling by 3 instead of 2, write down the equations corresponding to (3.11) through (3.16). In equation (3.10), $v = (\downarrow 3)x$ means $v(k) = x(3k)$.
4. What is the original input x in Figure 3.1, whose transform is the sawtooth function? To invert transforms use $x(n) = \frac{1}{2\pi} \int X(\omega) e^{-jn\omega} d\omega$.
5. By inverting $X(\omega)$ in Figure 3.2 find the original band-limited signal $x(n)$. What is $v(n) = (\downarrow 2)x(n)$? Compare with the input signal in Figure 3.1.
6. (Review) Verify that the transform of $(\downarrow 2)(\uparrow 2)v$ is still $V(\omega)$. Then $(\downarrow 2)(\uparrow 2) = I$.
7. Draw the block form of an M -channel filter bank without filters. Imitating the $M = 2$ case above, the output equals the input after $M - 1$ delays.

3.3 Sampling Operations in the z -Domain

This section includes a conversion from the frequency variable ω to the complex variable z . The connection between them is always $e^{j\omega} = e^{j\omega} = z$. Real frequencies ω correspond to points on the unit circle $|z| = 1$. The transform is extended beyond this special circle of complex numbers, to include all z in the complex plane.

In the process we overcome one inconvenience. The frequencies ω and $\omega + 2\pi$ and $\omega + 4\pi$ are impossible to distinguish. In the z -domain we don't attempt to distinguish them. The complex numbers $z = e^{j\omega}$ and $z = e^{j(\omega+2\pi)}$ and $z = e^{j(\omega+4\pi)}$ are absolutely identical. This remains just as true when the "frequency" ω is not real. It depends only on the fact that $e^{2\pi j} = 1$.

We repeat the definition of the z -transform of x :

$$\text{The signal } x(n) \text{ is transformed to } X(z) = \sum_n x(n)z^{-n}.$$

What we earlier called $X(\omega)$ is now $X(e^{j\omega})$. You will know immediately from the letter z that $X(z)$ refers to the z -transform rather than the Fourier transform.

At present, these infinite series are "formal power series". We are not worrying about their convergence. In reality they probably converge for some z and diverge for other z . When positive and negative powers are both included, the region of convergence is essentially an *annulus*. This is the region $r_{\text{inner}} < |z| < r_{\text{outer}}$ between two circles. Convergence may or may not occur on the circles $|z| = r_{\text{inner}}$ and $|z| = r_{\text{outer}}$. There is also the extreme but quite likely possibility that $r_{\text{inner}} = 0$ or $r_{\text{outer}} = \infty$ or $r_{\text{inner}} = r_{\text{outer}}$.

What is the effect on our down and up formulas? The vectors are $v = (\downarrow 2)x$ and $u = (\uparrow 2)v$. In frequency, their transforms are

$$V(\omega) = \frac{1}{2} \left[X\left(\frac{\omega}{2}\right) + X\left(\frac{\omega}{2} + \pi\right) \right] \quad \text{and} \quad U(\omega) = V(2\omega).$$

We now derive and explain these rules for converting to the z -transform:

1. Halved frequency $\frac{\omega}{2}$ leads in the z -domain to $z^{1/2}$.
2. Doubled frequency 2ω leads in the z -domain to z^2 .
3. Shifted frequency by π leads in the z -domain to $-z$.

These rules follow directly from $z = e^{i\omega}$. Immediately $e^{i\omega/2}$ is $z^{1/2}$ and $e^{i2\omega}$ is z^2 and $e^{i(\omega+\pi)}$ is $-z$. The rules produce the correct down and up formulas in the z-domain:

Theorem 3.4 The z-transforms of $v = (\downarrow 2)x$ and $u = (\uparrow 2)v$ are

$$V(z) = \frac{1}{2} [X(z^{1/2}) + X(-z^{1/2})] \quad \text{and} \quad U(z) = V(z^2). \quad (3.22)$$

The transform of $u = (\uparrow 2)(\downarrow 2)x$ is

$$U(z) = \frac{1}{2} (X(z) + X(-z)). \quad (3.23)$$

The extension from two channels to M channels is straightforward and important. In the time domain, $v = (\downarrow M)x$ and $u = (\uparrow M)v$ have components

$$v(n) = x(Mn) \quad \text{and} \quad u(k) = \begin{cases} v(\frac{k}{M}), & M \text{ divides } k \\ 0, & \text{otherwise.} \end{cases}$$

Then $(\uparrow M)(\downarrow M)x$ leaves every M th component unchanged (the components for which M divides k). The other components become zeros.

In the frequency domain there are $M - 1$ aliases in downsampling and $M - 1$ images from upsampling (it is very instructive to graph $V(\omega)$ and $U(\omega)$):

$$\begin{aligned} \text{down: } V(\omega) &= \frac{1}{M} \left[X\left(\frac{\omega}{M}\right) + X\left(\frac{\omega + 2\pi}{M}\right) + \cdots + X\left(\frac{\omega + (M-1)2\pi}{M}\right) \right] \\ \text{up: } U(\omega) &= V(M\omega). \end{aligned} \quad (3.24)$$

Eliminating V leaves $U(\omega) = \frac{1}{M} [X(\omega) + X(\omega + 2\pi) + \cdots]$ for the upsampled $u = (\uparrow M)(\downarrow M)x$. Those aliases overlap the original (not if it is properly bandlimited). To transfer into the z-domain, extend the three rules from $M = 2$ to any M :

1. Dividing ω by M leads to $z^{1/M}$.
2. Multiplying ω by M leads to z^M .
3. Shifting ω by $2\pi/M$ leads to $ze^{2\pi i/M}$.

From this correspondence we can express $(\downarrow M)$ and $(\uparrow M)$ in the z-domain:

Theorem 3.5 The z-transforms of $v = (\downarrow M)x$ and $u = (\uparrow M)v$ are

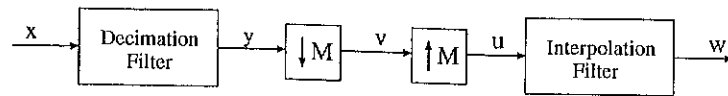
$$V(z) = \frac{1}{M} \sum_{k=0}^{M-1} X(z^{1/M} e^{2\pi i k/M}) \quad \text{and} \quad U(z) = V(z^M). \quad (3.25)$$

The transform of $u = (\uparrow M)(\downarrow M)x$, down and up, is by eliminating V :

$$U(z) = \frac{1}{M} [X(z) + X(ze^{2\pi i/M}) + \cdots + X(ze^{2\pi i(M-1)/M})]. \quad (3.26)$$

The *decimator* (downsampling operator) by itself creates aliases. The *expander* (upsampling operator) by itself creates images. To prevent the aliases and to remove the images, we filter the signal. A filter comes before the decimator, to band-limit the input. The aliasing terms become

zero. A filter comes *after* the expander, to wipe out the images. This filter removes (or nearly removes) the frequencies beyond $|\omega| = \frac{\pi}{M}$. The block forms are clear:



The decimation filter suppresses aliasing. The interpolation filter suppresses imaging.

It is easiest to see the decimation circuit in the frequency domain. The band-limiting filter cuts off $X(\omega)$ outside $[-\frac{\pi}{M}, \frac{\pi}{M}]$, to give $Y(\omega)$. Downsampling stretches that band to the full interval $[-\pi, \pi]$. But no aliases appear, as shown in Figure 3.6.

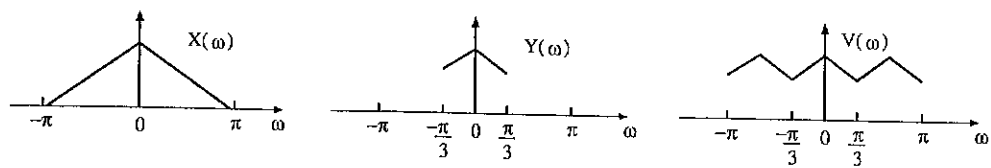
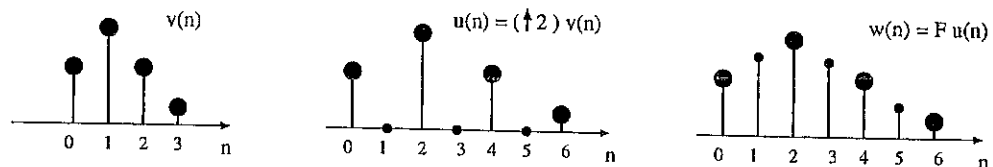


Figure 3.6: Spectra of $X(\omega)$, $Y(\omega)$, and $V(\omega)$ with $M = 3$.

From $V(\omega)$ we can recover $Y(\omega)$. Unless the original signal was band-limited, we cannot recover $X(\omega)$. If it was, we can. The interpolation circuit would do the recovery!

It is important to see the interpolation circuit in the time domain. The upsampling step inserts zeros into the signal. The interpolation filter F changes those zero values to smoother values $w(n)$. Remember that a lowpass filter reduces oscillations to give a smoother output. Here $M = 2$ and the interpolation circuit produces w from v :



Fractional Sampling Rates

To understand fractional decimation, study the product $(\uparrow 2)(\downarrow 3)$ and also $(\downarrow 3)(\uparrow 2)$. What are the outputs? In some way these should produce an output at a fractional rate, which is $\frac{2}{3}$ of the original rate. The transform $X(\omega)$ should be stretched by a factor $\frac{3}{2}$. Let us see.

The direct approach is in the time domain. When up follows down, we reach $x(0)$, 0 , $x(3)$, 0 :

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ \vdots \end{bmatrix} \xrightarrow{\downarrow 3} \begin{bmatrix} x(0) \\ x(3) \\ \vdots \end{bmatrix} \xrightarrow{\uparrow 2} \begin{bmatrix} x(0) \\ 0 \\ x(3) \\ 0 \\ \vdots \end{bmatrix}$$

When down follows up, the remarkable thing is that the result is the same:

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \xrightarrow{\uparrow 2} \begin{bmatrix} x(0) \\ 0 \\ x(1) \\ 0 \\ x(2) \\ 0 \\ x(3) \end{bmatrix} \xrightarrow{\downarrow 3} \begin{bmatrix} x(0) \\ 0 \\ 0 \\ x(3) \\ 0 \end{bmatrix}$$

Thus $(\uparrow 2)$ commutes with $(\downarrow 3)$. In either order, every third component appears in every second position. Remember that $(\uparrow 2)$ does *not* commute with $(\downarrow 2)$.

A natural question arises. *When does $(\uparrow L)$ commute with $(\downarrow M)$?* The output will come at $\frac{L}{M}$ times the original rate. It is best to reduce this fraction to lowest terms. The numbers L and M should be *relatively prime*, meaning that they have no common factors. The fraction $\frac{2L}{2M}$ is equal to $\frac{L}{M}$, but it is a mistake to use $(\uparrow 2L)$ and $(\downarrow 2M)$. Those operators do not commute and they are inefficient. The natural choice is the good choice.

Theorem 3.6 *The operators $(\downarrow L)$ and $(\uparrow M)$ commute if and only if L and M are relatively prime. Their greatest common divisor is $(L, M) = 1$. In that case $(\uparrow L)(\downarrow M)x = (\downarrow M)(\uparrow L)x$ has components*

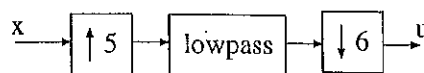
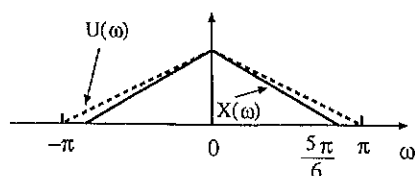
$$u(n) = \begin{cases} x(Mk) & \text{if } \frac{n}{L} = k \text{ is an integer} \\ 0 & \text{if } \frac{n}{L} \text{ is not an integer.} \end{cases} \quad (3.27)$$

This formula is always correct for the order $(\uparrow L)(\downarrow M)x$. With $L = 2$ and $M = 3$ it yields $u(1) = 0$ and $u(2) = x(3)$ as in the example above. With $L = 2$ and $M = 2$ it yields $u(1) = 0$ and $u(2) = x(2)$. This is the familiar output from $(\uparrow 2)(\downarrow 2)x$.

The formula (3.27) can be wrong if down follows up. In that order, the test is whether $\frac{Mn}{L}$ is an integer. For $M = L = 2$ this is true for every n . The output from $(\downarrow 2)(\uparrow 2)$ is the same as the input, and $(\downarrow 2)(\uparrow 2) = I$. But for $M = 3$ and $L = 2$, the fraction $\frac{3n}{2}$ is only an integer when n is even. In this $\frac{2}{3}$ case, up-down agrees with down-up.

When M and L are relatively prime, the output in either order has $u(n) = 0$ unless L divides n . If $\frac{n}{L}$ is not an integer, then $\frac{Mn}{L}$ is not an integer.

Why do we alter the rate by a fraction? The rate may need to change from 60 Hertz to 50 Hertz. The choices $L = 5$ and $M = 6$ will do it. (Not 50 and 60; they are not relatively prime.) We will get five samples instead of six samples, and 50 samples per second instead of 60 per second. There is no aliasing when the input signal is band-limited to $|\omega| < \frac{5\pi}{6}$. Then, stretching by $\frac{6}{5}$ will create no problems:



Problem Set 3.3

1. Find $U(\omega)$ from $X(\omega)$ if $u = (\uparrow L)(\downarrow M)x$. Use (3.25) for $(\downarrow M)$. Similarly find $V(\omega)$ if $v = (\downarrow M)(\uparrow L)x$. Show that $U(\omega) \equiv V(\omega)$ if (and only if) L and M are relatively prime.
2. Verify formula (3.27) directly. What happens to the odd-numbered components of x when we compute $(\uparrow 2)(\downarrow 2)x$? What happens to the odd part of $X(z)$ in equation (3.24)?
3. If a band-limited signal v goes through an expander, how is $(\uparrow 2)v$ related to v ?
4. For a band-limited signal x , draw in the frequency domain the steps of decimation and recovery by interpolation. Graph $X(\omega)$, $Y(\omega)$, $V(\omega)$, $U(\omega)$, and $W(\omega)$.
5. Draw $H(\omega)$ for an ideal decimator that prevents aliasing by $(\downarrow 6)$. Draw an ideal $F(\omega)$ that removes images created by $(\uparrow 5)$.
6. What lowpass filter placed between $(\uparrow 5)$ and $(\downarrow 6)$ avoids images and also aliasing?
7. In smoothing $u(n)$ to get the final output $w = Fu$, which filters F will *interpolate* and not change the even samples: $w(2k) = u(2k)$?

3.4 Filters Interchanged with Samplers

Our notation for the filter coefficients is $h(n)$. When the input x is the impulse δ , the responses are $h(0), \dots, h(N)$. The vector h is the impulse response (in the time domain). The response in the z -domain (we are now using this form systematically and often) is $H(z) = \sum h(n)z^{-n}$.

As an operator, the filter is a combination of delays (and possibly advances). *Note especially that a delay corresponds to z^{-1} .* The filter is linear and time-invariant. Expressed as an infinite matrix, H has the number $h(n)$ along its n th diagonal (counting down from the main diagonal). The i, j entry is $h(i - j)$.

First Noble Identity

In the standard form, downsampling follows filtering. We apply H and then $(\downarrow 2)$ or $(\downarrow M)$. Taken literally, this would be highly inefficient. We are apparently computing all components of Hx , and then throwing away many of them. No sensible engineer would do that. The polyphase form of $H(z)$ will tell us how to reorganize the calculation and do the sampling first.

We cannot just reverse the order of $(\downarrow M)$ and H (of course!). Here we note a special and important case, when the sampling can be done first. *The filter has $M - 1$ zero coefficients between each pair of nonzeros.* The nonzero coefficients $h(0), h(M), h(2M), \dots$ will be renamed $g(0), g(1), g(2), \dots$. Then $H(z)$ is the same as $G(z^M)$. For filters of this special type we can move the decimator outside the filter:

$$x \rightarrow \boxed{\downarrow M} \rightarrow \boxed{G(z)} \rightarrow v \equiv x \rightarrow \boxed{G(z^M)} \rightarrow \boxed{\downarrow M} \rightarrow v$$

The new order starting with $(\downarrow M)$ cuts the sampling rate immediately. The old order applies H to the full vector x . Here is an example with $M = 2$ and $h = (1, 0, 2)$:

$$\begin{array}{ccc}
 (\downarrow 2)x & G(\downarrow 2)x & Hx \\
 \left[\begin{array}{c} x(0) \\ x(2) \\ \vdots \end{array} \right] & \left[\begin{array}{c} x(0) + 2x(-2) \\ x(2) + 2x(0) \\ \vdots \end{array} \right] & \text{or} \quad \left[\begin{array}{c} x(0) + 2x(-2) \\ x(1) + 2x(-1) \\ x(2) + 2x(0) \\ x(3) + 2x(1) \\ \vdots \end{array} \right] \\
 & & (\downarrow 2)Hx = G(\downarrow 2)x \\
 & & \left[\begin{array}{c} x(0) + 2x(-2) \\ x(2) + 2x(0) \\ \vdots \end{array} \right]
 \end{array}$$

In the z -domain, this example yields the *even powers* in $(1 + 2z^{-2})X(z)$. The odd powers in $X(z)$ don't matter! So the quick way is to pick out the even powers by downsampling first. This "Noble Identity" represents a simple but very useful fact of algebra:

$$\text{First Noble Identity: } G(z)(\downarrow M) = (\downarrow M)G(z^M). \quad (3.28)$$

Proof Downsampling $x(n)$ leaves $x(Mn)$. Then the filter G produces

$$v(n) = \sum g(k)x(M(n-k)). \quad (3.29)$$

The other way (the slow way), $H(z) = G(z^M)$ uses every M th sample of the input. The filter gives $\sum g(k)x(n-Mk)$. Then the sampling operator replaces n by Mn . We reach the same $v(n)$.

You may find it useful to follow an impulse $x = \delta$ through both systems, to see that the impulse response is the same. On the left we get the vector $v = g$ of filter coefficients. On the right we keep all the zeros. Then downsampling ends with the same g :

$$\delta \rightarrow \boxed{G(z^M)} \rightarrow (\dots, g(0), 0, \dots, 0, g(1), \dots) \rightarrow \boxed{\downarrow M} \rightarrow (\dots, g(0), g(1), \dots).$$

We used G instead of the basic symbol H to emphasize that the *Noble Identities cannot be directly applied to a typical filter*. The identity applies only when $H(z)$ has the form $G(z^M)$. Only filters that have special impulse responses, with $M-1$ zeros between the nonzeros, fit this form. The idea of the polyphase matrix is to decompose a typical H into a set of M filters with these special responses. Then the Noble Identities apply to each polyphase component of H , even if they don't apply to the whole filter.

Second Noble Identity

The second identity concerns *upsampling*. It is a transpose of the first identity. The standard form of a filter bank has $(\uparrow M)$ before a synthesis filter $F(z)$. For filters of the special form $F(z) = G(z^M)$, that order can be reversed. But again we must change $G(z^M)$ to $G(z)$:

$$x \rightarrow \boxed{G(z)} \rightarrow \boxed{\uparrow M} \rightarrow u \equiv x \rightarrow \boxed{\uparrow M} \rightarrow \boxed{G(z^M)} \rightarrow u$$

This time we verify the equivalence in the z -domain. The output $(\uparrow M)Gx$ comes from upsampling $G(z)X(z)$:

$$U(z) = G(z^M)X(z^M).$$

In the other order, the upsampled signal $X(z^M)$ is filtered by the stretched-out $G(z^M)$. Again $U(z) = G(z^M)X(z^M)$, which proves the identity:

$$\text{Second Noble Identity: } (\uparrow M)G(z) = G(z^M)(\uparrow M). \quad (3.30)$$

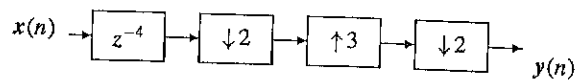
As before, this identity does not apply to most filters. We still need $M-1$ zeros between the nonzeros. For a simple delay, $G(z^M) = z^{-1}$ is not possible.

The Noble Identities are derived for systems with one input and one output. They also hold for systems with multiple inputs and outputs, as long as the decimation (or interpolation) factors are the same for all channels.

We soon move to the polyphase decomposition of a filter (and a filter bank). The phases have $M - 1$ zeros and the special form involving z^M . In the polyphase form, *the downsampling can come first*. And upsampling comes last.

Problem Set 3.4

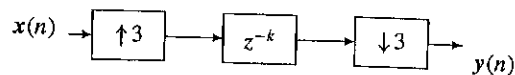
1. Verify the first Noble Identity $G(z)(\downarrow M) = (\downarrow M)G(z^M)$ in the z -domain.
2. Verify the second Noble Identity $(\uparrow M)G(z) = G(z^M)(\uparrow M)$ in the time domain.
3. Simplify the following system:



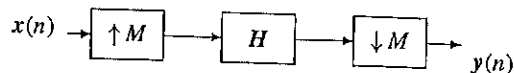
What is $Y(z)$ in terms of $X(z)$? Find $y(n)$ for the following inputs:

$$x(n) = \delta(n), \quad x(n) = (\dots, 1, 1, 1, \dots), \quad x(n) = (\dots, 1, -1, 1, -1, \dots).$$

4. What is $Y(z)$ in terms of $X(z)$? What is $y(n)$ in terms of $x(n)$?



5. If $H(z) = G(z^M)$ has $M - 1$ zeros between $h(0), h(M), \dots$, find $y(n)$ in terms of $x(n)$ and $h(n)$ for this system:



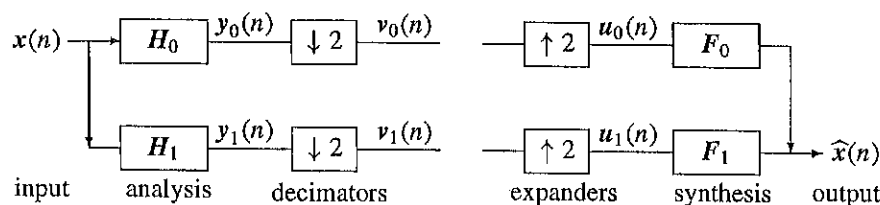
6. What identity would you use to compute $v = (\downarrow 2)Hx$ efficiently, if H has only *odd-numbered* coefficients $h = (0, h(1), 0, h(3))$?

Chapter 4

Filter Banks

4.1 Perfect Reconstruction

A filter bank is a set of filters, linked by sampling operators and sometimes by delays. The down-sampling operators are decimators, the upsampling operators are expanders. In a two-channel filter bank, the analysis filters are normally lowpass and highpass. Those are the filters H_0 and H_1 at the start of the following filter bank:



This structure was introduced in the 1980's. It gradually became clear how to choose H_0, H_1, F_0, F_1 to get perfect reconstruction: $\hat{x}(n) = x(n - l)$. The gap in the figure indicates where the downsampled signals might be coded for storage or transmission. At that point we may compress the signal and destroy information. Perfect reconstruction assumes no compression, so the gap is closed.

To indicate that H_0 is lowpass and H_1 is highpass, we often sketch the frequency responses. Figure 4.1 shows that they are not ideal brick wall filters. The responses overlap. *There is aliasing in each channel.* There is also amplitude distortion and phase distortion (our drawing does not show the phase). The synthesis filters F_0 and F_1 must be specially adapted to the analysis filters H_0 and H_1 , in order to cancel the errors in this analysis bank.

The goal of this section is to discover the conditions for perfect reconstruction. This means that the filter bank is **biorthogonal**. The synthesis bank, from F_0 and F_1 and $\uparrow 2$, is the inverse of the analysis bank. Inverse matrices automatically involve biorthogonality. (The rows of T and the columns of T^{-1} are by definition biorthogonal.) This will extend in Chapter 6 to biorthogonal scaling functions and wavelets.

Perfect reconstruction is a crucial property. If the sampling operators ($\downarrow 2$) and ($\uparrow 2$) were not present, a reconstruction without delay would mean that $F_0 H_0 + F_1 H_1 = I$. A perfect reconstruction with an l -step delay would mean (in the z -domain) that

$$\text{without } (\downarrow 2) \text{ and } (\uparrow 2): \quad F_0(z)H_0(z) + F_1(z)H_1(z) = z^{-l}. \quad (4.1)$$

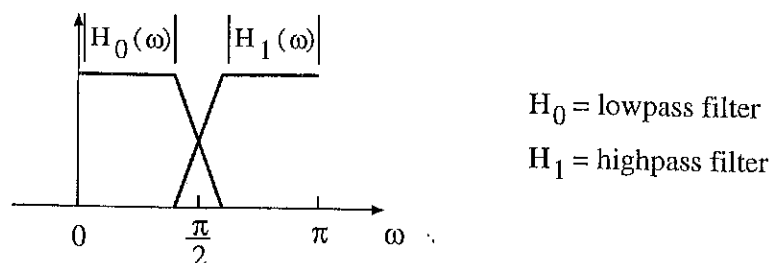


Figure 4.1: Rough sketch of frequency responses. Do not expect $|H_0(\omega)| + |H_1(\omega)| = 1$.

We do expect an overall delay z^{-l} , because each individual filter is causal.

Now take account of the sampling operators, which introduce *aliasing*. We recognize aliasing by the appearance of $-z$ as well as z (and $\omega + \pi$ as well as ω). The combination of $(\downarrow 2)$ followed by $(\uparrow 2)$ zeros out the odd-numbered components. In the z -domain it keeps only the even powers of $H_0(z)X(z)$:

The transform of $(\uparrow 2)(\downarrow 2)H_0x$ is $\frac{1}{2}(H_0(z)X(z) + H_0(-z)X(-z))$.

This is an even function, because the odd components are gone. The aliasing term $H_0(-z)X(-z)$ is multiplied by $F_0(z)$ at the synthesis step. This alias has to cancel the alias $F_1(z)H_1(-z)X(-z)$ from the other channel. So there is an alias cancellation condition in addition to a reconstruction condition:

$$\text{Alias cancellation } F_0(z)H_0(-z) + F_1(z)H_1(-z) = 0. \quad (4.2)$$

Correction The sampling operators also produce a change in equation (4.1). The right side has an extra factor 2. You can see this by considering a simple set of filters: $H_0(z) = 1$ and $H_1(z) = z^{-1}$, $F_0(z) = z^{-1}$ and $F_1(z) = 1$. This satisfies (4.2) and cancels aliasing. The left side of equation (4.1) equals $2z^{-1}$ rather than z^{-1} . The overall delay is $l = 1$ for this filter bank, and its perfect reconstruction comes from

$$\text{No distortion } F_0(z)H_0(z) + F_1(z)H_1(z) = 2z^{-l}. \quad (4.3)$$

The next page establishes these two conditions (4.2–4.3) for perfect reconstruction.

One further point. In a genuine filter bank, the highpass filter has $H_1 = 0$ at $z = 1$ (or $\omega = 0$). Equation (4.3) becomes $F_0(1)H_0(1) = 2$. That equation is more natural if we include an extra factor $\sqrt{2}$ in the filter coefficients. For a similar reason the highpass filters can be normalized by an extra $\sqrt{2}$. We take this opportunity to assign single letters $C = \sqrt{2}H_0$ and $D = \sqrt{2}H_1$ to the analysis filters, with no need for subscripts. The sum of lowpass coefficients $c(n) = \sqrt{2}h(n)$ is $\sqrt{2}$.

No Aliasing and No Distortion Conditions (4.2) and (4.3) for perfect reconstruction come directly from following a signal through the filter bank. The original signal is $x(n)$. The lowpass analysis filter is H_0 . In the z -domain this produces $H_0(z)X(z)$. Now downsample and upsample:

$$\text{First } (\downarrow 2) \text{ produces } \frac{1}{2}[H_0(z^{\frac{1}{2}})X(z^{\frac{1}{2}}) + H_0(-z^{\frac{1}{2}})X(-z^{\frac{1}{2}})]$$

$$\text{Then } (\uparrow 2) \text{ produces } \frac{1}{2}[H_0(z)X(z) + H_0(-z)X(-z)].$$

($\uparrow 2$) ($\downarrow 2$) H_0x has zeros in its odd-numbered components. Those zeros are produced by averaging $H_0x(n)$ with its alternating alias $(-1)^n H_0x(n)$. In the z -domain this is the average of $H_0(z)X(z)$ with $H_0(-z)X(-z)$. The aliasing term has entered the filter bank.

The final filter multiplies by $F_0(z)$. This yields the output from the lowpass channel. Below it we write the corresponding output from the highpass channel (same formula with subscripts changed to 1):

$$\text{lowpass output} = \frac{1}{2} F_0(z) [H_0(z)X(z) + H_0(-z)X(-z)],$$

$$\text{highpass output} = \frac{1}{2} F_1(z) [H_1(z)X(z) + H_1(-z)X(-z)].$$

Now add. The filter bank combines the channels to get $\hat{x}(n)$. In the z -domain this is $\hat{X}(z)$. Half the terms involve $X(z)$ and half involve $X(-z)$:

$$\frac{1}{2} [F_0(z)H_0(z) + F_1(z)H_1(z)]X(z) + \frac{1}{2} [F_0(z)H_0(-z) + F_1(z)H_1(-z)]X(-z).$$

For perfect reconstruction with l time delays, this $\hat{X}(z)$ must be $z^{-l}X(z)$. So the "distortion term" must be z^{-l} and the "alias term" must be zero:

Theorem 4.1 A 2-channel filter bank gives perfect reconstruction when

$$F_0(z)H_0(z) + F_1(z)H_1(z) = 2z^{-l} \quad (4.4)$$

$$F_0(z)H_0(-z) + F_1(z)H_1(-z) = 0. \quad (4.5)$$

In vector-matrix form these two conditions involve the modulation matrix $H_m(z)$:

$$\begin{bmatrix} F_0(z) & F_1(z) \end{bmatrix} \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} = [2z^{-l} \quad 0]. \quad (4.6)$$

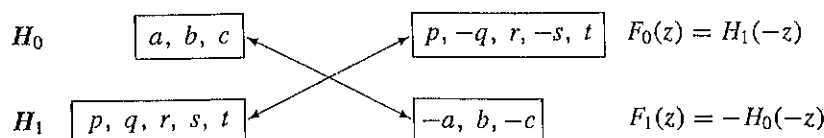
This matrix $H_m(z)$ will play a very important role. It involves the responses $H_k(z)$ and their alias terms $H_k(-z)$. For an M -channel bank the matrix will be $M \times M$. But the real problem is clearly identified by the separate conditions (4.4) and (4.5)—*how to design filters that meet those conditions?*

Alias Cancellation and the Product Filter $P_0 = F_0H_0$

At this point we have four filters H_0, H_1, F_0, F_1 to design. They must satisfy (4.4) and (4.5). It is almost irresistible to determine some of the filters from the others:

$$\text{For alias cancellation choose } F_0(z) = H_1(-z) \text{ and } F_1(z) = -H_0(-z) \quad (4.7)$$

Important: This choice automatically satisfies $F_0(z)H_0(-z) + F_1(z)H_1(-z) = 0$. Aliasing is removed; it cancels itself! This relation of F_0 to H_1 and of F_1 to H_0 gives the *alternating signs* pattern of a 2-channel filter bank:



Now comes a definition that allows us to rewrite equation (4.4) for no distortion:

Define the *product filter* by $P_0(z) = F_0(z)H_0(z)$.

This is a lowpass filter. The highpass product filter is $P_1(z) = F_1(z)H_1(z)$. These products P_0 and P_1 are exactly the terms in (4.4). The crucial point is the relation between $P_0(z)$ and $P_1(z)$, when the synthesis filters are determined by $F_0(z) = H_1(-z)$ and $F_1(z) = -H_0(-z)$. We substitute directly to find that $P_1(z) = -P_0(-z)$:

$$P_1(z) = -H_0(-z)H_1(z) = -H_0(-z)F_0(-z) = -P_0(-z). \quad (4.8)$$

The reconstruction equation $F_0(z)H_0(z) + F_1(z)H_1(z) = 2z^{-l}$ simplifies to

$$F_0(z)H_0(z) - F_0(-z)H_0(-z) = P_0(z) - P_0(-z) = 2z^{-l}. \quad (4.9)$$

The design of a 2-channel PR filter bank is reduced to two steps:

Step 1. Design a lowpass filter P_0 satisfying (4.9).

Step 2. Factor P_0 into F_0H_0 . Then use (4.7) to find F_1 and H_1 .

The length of P_0 determines the sum of the lengths of F_0 and H_0 . There are many ways to design P_0 in Step 1. And there are many ways to factor it in Step 2. Experiments are going on as this book is written, and undoubtedly they are going on as the book is read, to find the best factors F_0 and H_0 of the best product filter P_0 .

Note that (4.9) is a condition on the *odd powers* in $P_0(z) = F_0(z)H_0(z)$. Those odd powers must have coefficient zero, except z^{-l} has coefficient one. The shift to $P(z)$ in equation (4.11) below will remove the even powers except z^0 .

A look forward To help the reader find the specific filters that are coming, we point to an outstanding choice for the product filter:

$$P_0(z) = (1 + z^{-1})^{2p} Q(z). \quad (4.10)$$

The polynomial $Q(z)$ of degree $2p - 2$ is chosen so that (4.9) is satisfied. There are $2p - 1$ odd powers in $P_0(z)$, and $2p - 1$ coefficients to choose in $Q(z)$. Then $Q(z)$ is *unique*. This is the Daubechies construction, with a history that we will outline in Section 5.5. Since the construction starts with the special factor $(1 + z^{-1})^{2p}$, these filters are called *binomial* or *maxflat*. The binomial factor gives a maximum number of zeros at $z = -1$, which means that the frequency response is maximally flat at $\omega = \pi$. The binomial by itself, without $Q(z)$, represents a "spline filter". $Q(z)$ is needed to give perfect reconstruction.

Splitting P_0 into F_0H_0 can give linear phase filters (symmetry in F_0 and H_0 separately). It can give orthogonal filters (symmetry between F_0 and H_0). It *cannot* give both, except in the Haar case $p = 1$. Section 5.4 discusses the factorization, and Chapter 11 reports some comparisons for image processing.

Simplification The equation $P_0(z) - P_0(-z) = 2z^{-l}$ can be made a little more convenient. The left side is an odd function, so l is odd. Normalize $P_0(z)$ by z^l to center it:

$$\text{The normalized product filter is } P(z) = z^l P_0(z).$$

Then $P(-z) = (-z)^l P_0(-z)$. Since l is odd, this is $-z^l P_0(-z)$. The reconstruction equation $P_0(z) - P_0(-z) = 2z^{-l}$ takes an extremely simple form when we multiply by z^l . The factor z^{-l} disappears and the minus sign becomes plus:

Perfect Reconstruction Condition. $P(z)$ must be a "halfband filter"

$$P(z) + P(-z) = 2. \quad (4.11)$$

This means that *all even powers in $P(z)$ are zero, except the constant term (which is 1)*. The odd powers cancel when $P(z)$ combines with $P(-z)$ —so the coefficients of odd powers in $P(z)$ are design variables in 2-channel PR filter banks.

Example 4.1. The maxflat product filter $P_0(z) = (1 + z^{-1})^4 Q(z)$ happens to be

$$P_0(z) = \frac{1}{16}(-1 + 9z^{-2} + 16z^{-3} + 9z^{-4} - z^{-6}).$$

The center term is $z^{-3} = z^{-l}$. The requirement $P_0(z) - P_0(-z) = 2z^{-l}$ is quickly verified, because the even powers in P_0 cancel in the difference. The function $Q(z) = -1 + 4z^{-1} - z^{-2}$ was chosen so that the odd powers z^{-1} and z^{-5} are absent from $P_0(z)$.

A centering operation gives the normalized product filter $P(z)$. Multiply by $z^l = z^3$ to symmetrize the polynomial around the constant term z^0 :

$$P(z) = \frac{1}{16}(-z^3 + 9z + 16 + 9z^{-1} - z^{-3}).$$

This is halfband, because the only even power is z^0 and its coefficient is 1. The perfect reconstruction requirement $P(z) + P(-z) = 2$ is verified. The odd powers in P cancel in the sum.

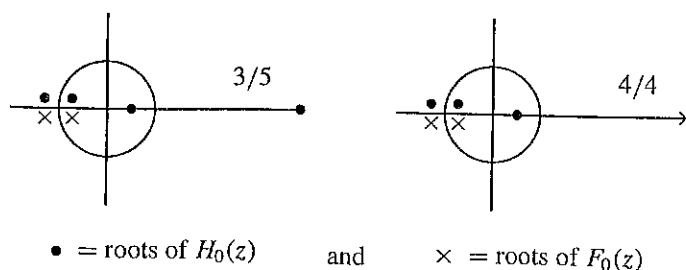
Notice the variety of factorizations into $P_0(z) = F_0(z)H_0(z)$. The polynomial $P_0(z)$ has *six roots*. The two roots from $Q(z)$ are at $c = 2 - \sqrt{3}$ and $\frac{1}{c} = 2 + \sqrt{3}$. The other four roots from $(1 + z^{-1})^4$ are at $z = -1$. Each factor normally has at least *one* root at $z = -1$. (But the factorization into $H_0 = 1$ and $F_0 = P_0$ is quite interesting.) Thus F_0 or H_0 (*either order is possible!*) could be

(a)	1	degree	$N = 0$
(b)	$(1 + z^{-1})$	degree	$N = 1$
(c)	$(1 + z^{-1})^2$ or $(1 + z^{-1})(c - z^{-1})$	degree	$N = 2$
(d)	$(1 + z^{-1})^3$ or $(1 + z^{-1})^2(c - z^{-1})$	degree	$N = 3$

The number $N + 1$ of filter coefficients is one greater than the degree. Thus (c) can produce a 5/3 filter, with $H_0(z) = \frac{1}{8}(-1 + 2z^{-1} + 6z^{-2} + 2z^{-3} - z^{-4})$ and $F_0(z) = \frac{1}{2}(1 + 2z^{-1} + z^{-2})$. The analysis length is given first. This is a possible choice for compression, with symmetric filters of very low complexity. The binomial $(1 + z^{-1})^2$ in the synthesis filter means that its continuous-time scaling function will be a hat function. Reversing F_0 and H_0 gives a 3/5 pair, not as successful in practice.

The choice (b) is also of interest. One factor is $1 + z^{-1}$ so one scaling function is the box function. Experiments indicate better performance when this short lowpass filter is in the analysis bank. Where 5/3 was preferred to 3/5 for odd-length filters, it seems that 2/6 is preferred to 6/2. *Five roots go into $F_0(z)$.*

The other outstanding choices are length 4/4 from the factorizations in (d). We get linear phase from $(1 + z^{-1})^3$ and $(-1 + 3z^{-1} + 3z^{-2} - z^{-3})$. The orthonormal Daubechies filter comes



from $(1+z^{-1})^2(c-z^{-1})$ and $(1+z^{-1})^2(\frac{1}{c}-z^{-1})$. These are not linear phase. (Problem 7 shows that linear phase requires $\frac{1}{c}$ to be a root when c is a root.) For the Daubechies orthogonal choice, the roots $-1, -1, c$ of one factor are the reciprocals of the roots $-1, -1, \frac{1}{c}$ of the other factor. Section 5.4 demonstrates that this balanced splitting (*spectral factorization*) of a halfband filter produces an orthogonal filter bank and orthogonal wavelets.

Example 4.2. (Haar filter bank) The average-difference analysis filter has

$$H_m(z) = \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1+z^{-1} & 1-z^{-1} \\ 1-z^{-1} & 1+z^{-1} \end{bmatrix}.$$

The synthesis filters are

$$\begin{aligned} F_0(z) &= H_1(-z) = (1+z^{-1})/\sqrt{2} \\ F_1(z) &= -H_0(-z) = -(1-z^{-1})/\sqrt{2} \end{aligned}$$

The product filters are

$$\begin{aligned} P_0(z) &= F_0(z)H_0(z) = \frac{1}{2}(1+z^{-1})^2 \\ P_1(z) &= F_1(z)H_1(z) = -\frac{1}{2}(1-z^{-1})^2 = -P_0(-z). \end{aligned}$$

Both P_0 and P_1 contain $+z^{-1}$. Perfect reconstruction $P_0(z) - P_0(-z) = 2z^{-1}$ means that $l = 1$. The normalized product filter (symmetric halfband) is

$$P(z) = z^l P_0(z) = \frac{1}{2}z(1+z^{-1})^2 = \frac{1}{2}z + 1 + \frac{1}{2}z^{-1}.$$

Modulation Matrices

The conditions for perfect reconstruction are expressed in (4.6) by

$$[F_0(z) \ F_1(z)] \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} = [2z^{-l} \ 0]. \quad (4.12)$$

This displays the *analysis modulation matrix* $H_m(z)$ —which is central to filter bank theory. With no extra effort we can also produce the synthesis modulation matrix $F_m(z)$. The two matrices should play matching (and even reversible) roles. This balance between F_m and H_m is achieved by expanding (4.12) into a matrix equation:

$$\begin{bmatrix} F_0(z) & F_1(z) \\ F_0(-z) & F_1(-z) \end{bmatrix} \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} = \begin{bmatrix} 2z^{-l} & 0 \\ 0 & 2(-z)^{-l} \end{bmatrix}. \quad (4.13)$$