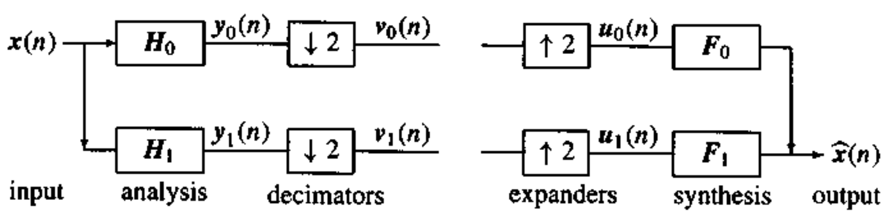
Chapter 4

Filter Banks

4.1 Perfect Reconstruction

A filter bank is a set of filters, linked by sampling operators and sometimes by delays. The downsampling 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 $\boldsymbol{H_0}$ and $\boldsymbol{H_1}$ at the start of the following filter bank:



This structure was introduced in the 1980's. It gradually become clear how to choose $\boldsymbol{H_0}$, $\boldsymbol{H_1}$, $\boldsymbol{F_0}$, $\boldsymbol{F_1}$ to get perfect reconstruction: $\widehat{x}(n) = x(n-l)$. 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 $\boldsymbol{H_0}$ is lowpass and $\boldsymbol{H_1}$ 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 $\boldsymbol{F_0}$ and $\boldsymbol{F_1}$ must be specially adapted to the analysis filters $\boldsymbol{H_0}$ and $\boldsymbol{H_1}$, 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 $\boldsymbol{F_0}$ and $\boldsymbol{F_1}$ and $\uparrow 2$, is the inverse of the analysis bank. Inverse matrices automatically involve biorthogonality. (The rows of $\boldsymbol{T}$ and the columns of $\boldsymbol{T}^{-1}$ 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 $\boldsymbol{F_0 H_0} + \boldsymbol{F_1 H_1} = \boldsymbol{I}$. A perfect reconstruction with an l-step delay would mean (in the z-domain) that

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

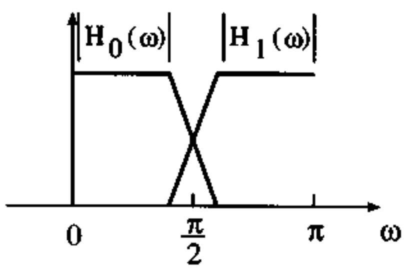


Figure 4.1: Rough sketch of frequency responses. Do not expect $|H_0(\omega)| + |H_1(\omega)| = 1$.

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 $H_0(z)X(z)$:

$$\text{The transform of } (\uparrow 2)(\downarrow 2)\boldsymbol{H_0 x} \text{ is } \tfrac{1}{2}(H_0(z)X(z) + H_0(-z)X(-z)).$$

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:

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

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

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

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 $\boldsymbol{C} = \sqrt{2}\boldsymbol{H_0}$ and $\boldsymbol{D} = \sqrt{2}\boldsymbol{H_1}$ to the analysis filters, with no need for subscripts. The sum of lowpass coefficients $\boldsymbol{c}(n) = \sqrt{2}\boldsymbol{h}(n)$ 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 $\boldsymbol{x}(n)$. The lowpass analysis filter is $\boldsymbol{H_0}$. In the z-domain this produces $H_0(z)X(z)$. Now downsample and upsample:

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

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

$(\uparrow 2)\,(\downarrow 2)\,\boldsymbol{H_0x}$ *has zeros in its odd-numbered components.* Those zeros are produced by averaging $\boldsymbol{H_0}x(n)$ with its alternating alias $(-1)^n\boldsymbol{H_0}x(n)$. 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):

$$\begin{aligned} \text{lowpass output} &= \tfrac{1}{2}F_0(z)\,[H_0(z)X(z) + H_0(-z)\,X(-z)] \\ \text{highpass output} &= \tfrac{1}{2}F_1(z)\,[H_1(z)X(z) + H_1(-z)\,X(-z)]. \end{aligned}$$

Now add. The filter bank combines the channels to get $\widehat{x}(n)$. In the z-domain this is $\widehat{X}(z)$. 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 $\widehat{X}(z)$ 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*

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

$$F_0(z)H_0(-z) + F_1(z)H_1(-z) = 0. \tag{4.5}$$

In vector-matrix form these two conditions involve the *modulation matrix* $\boldsymbol{H}_m(z)$:

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

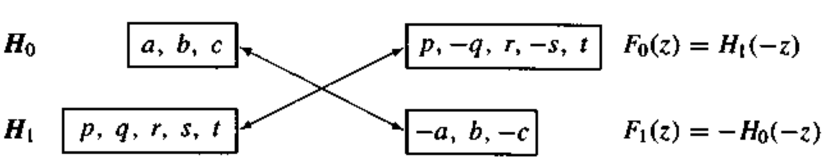
This matrix $\boldsymbol{H}_m(z)$ will play a very important role. It involves the responses $H_k(z)$ and their alias terms $H_k(-z)$. For an M-channel bank the matrix will be $M \times M$. But the real problem is clearly identified by the separate conditions (4.4) and (4.5) — *how to design filters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

Alias Cancellation and the Product Filter $P_0 = F_0H_0$

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

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

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 $\boldsymbol{F_0}$ to $\boldsymbol{H_1}$ and of $\boldsymbol{F_1}$ to $\boldsymbol{H_0}$ 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 $P_1(z) = -P_0(-z)$:

$$P_1(z) = -H_0(-z)H_1(z) = -H_0(-z)F_0(-z) = -P_0(-z). \tag{4.8}$$

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

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

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:

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

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:

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"

$$P(z) + P(-z) = 2. \tag{4.11}$$

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

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

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:

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

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.

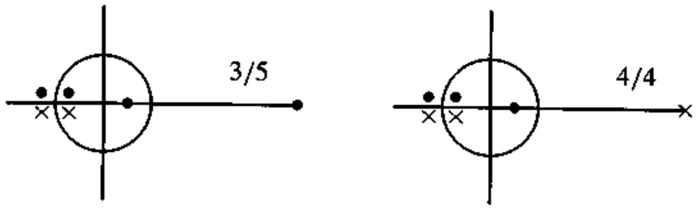
Notice the variety of factorizations into $P_0(z) = F_0(z)H_0(z)$. The polynomial $P_0(z)$ has *six roots*. The two roots from $Q(z)$ are at $c = 2 - \sqrt{3}$ and $\frac{1}{c} = 2 + \sqrt{3}$. The other four roots from $(1 + z^{-1})^4$ are at $z = -1$. Each factor normally has at least *one* root at $z = -1$. (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



• = roots of $H_0(z)$ and × = roots of $F_0(z)$

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

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

The synthesis filters are

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

The product filters are

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

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

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

Modulation Matrices

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

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

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

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

The second row of equations follows from the first, when $-z$ replaces z. Note that $F_1(z)$ is in the $(1,2)$ position, but $H_1(z)$ is in the $(2,1)$ position. This *transpose convention* between analysis and synthesis will appear again for polyphase matrices. The reader sees why it is necessary.

The reconstruction condition (4.13) applies to $\boldsymbol{F}_m(z)\boldsymbol{H}_m(z)$. If we "center" the filter coefficients around the zero position, the right side becomes the identity matrix! This is so desirable and memorable that we do it. It is the same normalization that centered $P_0(z)$ into $P(z)$, and it is especially clear when the filters are linear phase (and l is odd):

Theorem 4.2 *If all filters are symmetric (or antisymmetric) around zero, as in* $H(z) = H(z^{-1})$ *and* $\boldsymbol{h}(k) = \boldsymbol{h}(-k)$, *then the condition for perfect reconstruction becomes a statement about inverse matrices:*

$$\boldsymbol{F}_m(z)\boldsymbol{H}_m(z) = 2\boldsymbol{I}. \tag{4.14}$$

The H's determine the F's. The analysis bank is inverted by the synthesis bank. When we express it that way, equation (4.14) becomes almost obvious.

A Brief History of H_1

The reader understands that the filters $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$ are still to be chosen. These choices are connected. Historically, designers chose the lowpass filter coefficients $\boldsymbol{h}(0), \ldots \boldsymbol{h}(N)$ and then constructed $\boldsymbol{H}_1$ from $\boldsymbol{H}_0$. Here are two possibilities that produce *equal length filters*. $\boldsymbol{H}_1$ will be highpass whenever $\boldsymbol{H}_0$ is lowpass:

Alternating signs : $\quad H_1(z) = H_0(-z)$ comes from $(\boldsymbol{h}(0), -\boldsymbol{h}(1), \boldsymbol{h}(2), -\boldsymbol{h}(3), \ldots)$

Alternating flip : $\quad H_1(z) = -z^{-N}H_0(-z^{-1})$ comes from $(\boldsymbol{h}(N), -\boldsymbol{h}(N-1), \ldots)$.

For convenience we are assuming real coefficients. The number N is odd in the alternating flip. The perfect reconstruction condition is *still to be imposed*. When that is satisfied, the overall system delay is $l = N$.

Early choice. Croisier–Estaban–Galand (1976) chose alternating signs $H_1(z) = H_0(-z)$. The resulting filter bank was called QMF (Quadrature Mirror Filter). The highpass response $|H_1(e^{j\omega})|$ is a mirror image of the lowpass magnitude $|H_0(e^{j\omega})|$ with respect to the middle frequency $\frac{\pi}{2}$ — the quadrature frequency. Note that IIR filters $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$ are allowed (and needed for PR, except for Haar!). This name QMF has since been extended to a larger class of filter banks, allowing M channels.

Better choice. Smith and Barnwell (1984–6) and Mintzer (1985) chose the alternating flip $H_1(z) = -z^{-N}H_0(-z^{-1})$. This leads to orthogonal filter banks, when $\boldsymbol{H}_0$ is correctly chosen. The Daubechies filters will fit this pattern.

General choice. The product $F_0(z)H_0(z)$ is a halfband filter. This gives biorthogonality, when aliasing is cancelled by the relation of $\boldsymbol{F}_0$ to $\boldsymbol{H}_1$ and $\boldsymbol{F}_1$ to $\boldsymbol{H}_0$.

Actually the synthesis bank has little freedom. Alias cancellation requires $F_0(z)H_0(-z) + F_1(z)H_1(-z) = 0$. Croisier-Estaban-Galand wrote each F_k directly in terms of H_k by

$$F_0(z) = H_0(z) \quad \text{and} \quad F_1(z) = -H_1(z).$$

With alternating signs $H_1(z) = H_0(-z)$ inside the analysis bank, their synthesis construction agrees (as it must) with the anti-aliasing equations

$$F_0(z) = H_1(-z) \text{ and } F_1(z) = -H_0(-z). \tag{4.15}$$

Smith-Barnwell also made the anti-aliasing choice (4.15). With the alternating flip in $H_1(z)$, their synthesis filters (remembering that N is odd) are

$$\begin{aligned} F_0(z) &= H_1(-z) &&= z^{-N}H_0(z^{-1}) \text{ comes from } (\boldsymbol{h}(N), \boldsymbol{h}(N-1), \ldots, \boldsymbol{h}(0)). \\ F_1(z) &= -H_0(-z) &&= z^{-N}H_1(z^{-1}) \text{ comes from } (-\boldsymbol{h}(0), \boldsymbol{h}(1), -\boldsymbol{h}(2), \ldots, \boldsymbol{h}(N)). \end{aligned}$$

Notice! Each $\boldsymbol{F_k}$ in Figure 4.1 has become the ordinary flip of the corresponding $\boldsymbol{H_k}$. In matrix language the synthesis matrices are the *transposes* of the analysis matrices. A shift by N delays makes them causal. When we flip to get $\boldsymbol{F_0}$ and then alternate signs to get $\boldsymbol{H_1}$, we have the alternating flip from $\boldsymbol{H_0}$ to $\boldsymbol{H_1}$.

The alternating flip automatically gives double-shift orthogonality between highpass and lowpass (to be explained). *Conclusion*: When the design of $\boldsymbol{H_0}$ leads to perfect reconstruction in the alternating flip filter bank, it also leads to orthogonality.

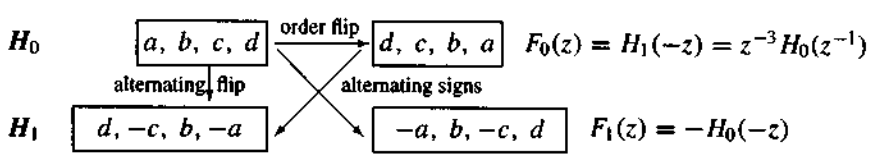


Figure 4.2: Relations between the filters allowing orthogonality when $N = 3$.

With aliasing cancelled, we now look at the PR condition

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

The early choice was alternating signs $H_1(z) = H_0(-z)$. With $F_0(z) = H_1(-z)$ and $F_1(z) = -H_0(-z)$, PR requires

$$H_0^2(z) - H_1^2(z) = H_0^2(z) - H_0^2(-z) = 2z^{-l}. \tag{4.16}$$

Therefore $H_0^2(z)$ has exactly one odd power z^{-l}. This is not easy for the square of a polynomial. An FIR filter is restricted to *two coefficients* (not good). Problem 4 asks for the reasoning, which forces the filters to be IIR.

The better choice is alternating flip. Perfect reconstruction is definitely possible. Product filters $\boldsymbol{F_0H_0}$ and $\boldsymbol{F_1H_1}$ become $P_0(z) = z^{-N}H_0(z^{-1})H_0(z)$ and $P_1(z) = -z^{-N}H_0(-z^{-1})H_0(-z)$. Multiply by $z^l = z^N$ to center these filters. The normalized product filter is $P(z)$ and the reconstruction condition is (4.11):

$$P(z) + P(-z) = 2 \text{ with } P(z) = H_0(z^{-1})H_0(z). \tag{4.17}$$

This is spectral factorization of a halfband filter! On the unit circle $z = e^{j\omega}$, the product $H_0(e^{-j\omega})H_0(e^{j\omega})$ is a magnitude squared:

$$P(e^{j\omega}) = \sum_{-N}^{N} p(n)e^{-jn\omega} = \left|\sum_{0}^{N} h(n)e^{-jn\omega}\right|^2. \tag{4.18}$$

The halfband coefficients are $p(n) = p(-n)$ for odd n and $p(n) = 0$ for even n (except $p(0) = 1$). *We design $P(z)$ and factor to find $H_0(z)$.* This symmetric factorization coincides with the Smith-Barnwell alternating flip. *It yields orthogonal banks with perfect reconstruction.* The flattest $P(z)$ will lead us to the Daubechies wavelets.

A note on biorthogonality (PR) with linear phase

Theorem 4.3 *In a biorthogonal linear-phase filter bank with two channels, the filter lengths are all odd or all even. The analysis filters can be*

(*a*) *both symmetric, of odd length*

(*b*) *one symmetric and the other antisymmetric, of even length.*

Proof: Odd and even lengths behave differently when we alternate signs:

odd length:	$a\ b\ c\ b\ a$	$\rightarrow\ a\ -b\ c\ -b\ a$	(remains symmetric)
even length:	$a\ b\ b\ a$	$\rightarrow\ a\ -b\ b\ -a$	(becomes antisymmetric).

To cancel aliasing, there is sign alternation in $F_0(z) = H_1(-z)$. There is also alternation in $F_1(z) = -H_0(-z)$. The extra minus sign does not change the symmetry type. The two successful combinations are

$$\begin{matrix} H_0 = \text{symm} & & F_0 = \text{symm} \\ & \times & \\ H_1 = \text{symm} & & F_1 = \text{symm} \end{matrix} \qquad\qquad \begin{matrix} H_0 = \text{symm} & & F_0 = \text{symm} \\ & \times & \\ H_1 = \text{anti} & & F_1 = \text{anti} \end{matrix}$$

odd lengths *even lengths*

The other possibilities are excluded by the PR condition: $F_0(z)\,H_0(z)$ has to be a ***halfband filter***. It must have an odd number of coefficients, and the center coefficient must be 1. For $F_0(z)\,H_0(z)$ to have odd length (which means even degree), the factors $F_0(z)$ and $H_0(z)$ must be both odd length or both even length. If one is symmetric and the other antisymmetric, the product $F_0(z)\,H_0(z)$ will be antisymmetric with zero at the center — not allowed. We conclude that $F_0(z)$ and $H_0(z)$ must match: both odd length or both even length, both symmetric or both antisymmetric.

This leaves the two successful possibilities shown above, and two more: H_0 and F_0 both antisymmetric. But the sum of lowpass coefficients cannot be zero. So antisymmetry of H_0 is ruled out.

Perfect Reconstruction with M Channels In reality a filter bank can have M channels. Although $M = 2$ is standard in many applications, we often see $M > 2$. There are M analysis filters $H_0, H_1, \ldots, H_{M-1}$. The sampling is done at the critical rate by $(\downarrow M)$ and $(\uparrow M)$. There

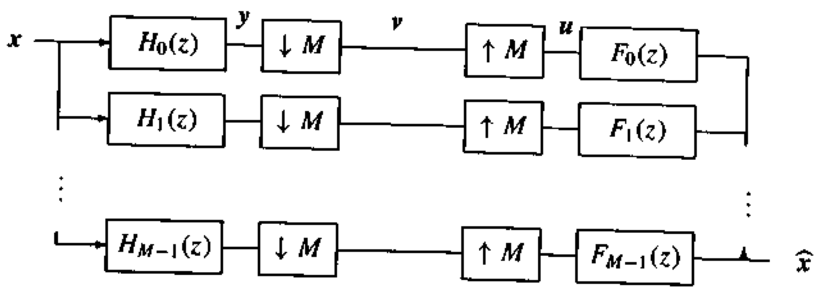


Figure 4.3: Standard form of an M-channel filter bank (maximally decimated).

are M filters $F_0, F_1, \ldots, F_{M-1}$ in the synthesis bank. The outputs from all channels are combined into a single output $\hat{x}$. Our standard picture of this implementation is Figure 4.3.

For this M-band case, the theory of perfect reconstruction was developed over several years. When we follow each channel from H_k through $(\downarrow M)$ and $(\uparrow M)$ and F_k, and add, we find M conditions for perfect reconstruction. These equations involve the $M \times M$ *modulation matrix* $\boldsymbol{H}_m(z)$ — corresponding exactly to the 2-channel case. The modulations are by multiples of $2\pi/M$ in frequency, and by powers of $W = e^{-2\pi j/M}$ in the z-domain. We put that matrix on record here.

Modulation matrix

$$\boldsymbol{H}_m(z) = \begin{bmatrix} H_0(z) & H_0(zW) & \cdots & H_0(zW^{M-1}) \\ H_1(z) & H_1(zW) & \cdots & H_1(zW^{M-1}) \\ \cdots & \cdots & \cdots & \cdots \\ H_{M-1}(z) & H_{M-1}(zW) & \cdots & H_{M-1}(zW^{M-1}) \end{bmatrix}. \quad (4.19)$$

The last $M-1$ columns represent the $M-1$ aliases created by $(\downarrow M)$, just as $H(-z)$ represented the one alias $(W = -1)$ created when $M = 2$. *The transpose of* $\boldsymbol{H}_m(z)$ *is the* ***alias component matrix***.

Another matrix will play an equally central role — in fact an interchangeable role, because it is very closely linked to $\boldsymbol{H}_m(z)$. This new matrix is the *polyphase matrix* $\boldsymbol{H}_p(z)$. It is developed and explained in the following section, as the natural way to follow the "phases" when a signal is subsampled. We put on record the 2-phase matrix — *this is the polyphase matrix when* $M = 2$:

$$\boldsymbol{H}_p(z) = \begin{bmatrix} H_{0,\text{even}}(z) & H_{0,\text{odd}}(z) \\ H_{1,\text{even}}(z) & H_{1,\text{odd}}(z) \end{bmatrix}.$$

Still looking ahead, we mention especially the orthogonal case. *The polyphase matrix is then unitary for* $|z| = 1$ (this makes it *paraunitary*). $\boldsymbol{H}_m(z)$ is also paraunitary, after dividing by $\sqrt{2}$ (Section 5.1). The analysis of paraunitary matrices was led by Vaidyanathan. He and others built onto the early theory (and nearly indecipherable exposition) of Belevitch. For $M = 2$, the paraunitary matrix leads back to the Smith-Barnwell construction of an orthogonal PR filter bank.

Several perfect reconstruction filter banks deserve special mention. Simplest is the average-difference pair from Chapter 1. This is a useful example but a poor filter. That is the first in a family of "maxflat filters", corresponding to the Daubechies wavelets. The others in the family are orthogonal but not linear phase — since those two properties conflict.

Different factorizations of the product $P_0(z)$ lead to linear phase (not orthogonality). Those filters have become favorites for compression.

For $M > 2$, the design of separate filters $\boldsymbol{H}_0, \boldsymbol{H}_1, \ldots, \boldsymbol{H}_{M-1}$ can become unwieldy. We look for constructions in which these all come from one prototype filter. A particular class is the *cosine-modulated filter banks* in Chapter 9. A phase change (= modulation) is the key to their construction. Those are efficient in every way.

Problem Set 4.1

1. If $(\boldsymbol{H}_m(z))^{-1}$ is also a polynomial, the synthesis bank as well as the analysis bank is FIR. Why must the determinant $H_0(z)H_1(-z) - H_1(z)H_0(-z)$ in the denominator of $\boldsymbol{H}_m^{-1}$ be a monomial cz^{-l}? This determinant is an odd function, $\det \boldsymbol{H}_m(z) = -\det \boldsymbol{H}_m(-z)$. Then the exponent l must be odd.

2. The solution of *both* equations (4.4–5) for F_0 and F_1 involves $(\boldsymbol{H}_m(z))^{-1}$:

$$[F_0(z) \quad F_1(z)] = [2z^{-l} \quad 0](\boldsymbol{H}_m(z))^{-1} = \frac{2z^{-l}}{\det \boldsymbol{H}_m(z)}[H_1(-z) \quad H_0(-z)].$$

If $\det \boldsymbol{H}_m(z) = 2z^{-l}$, as normal, this yields $F_0(z) = H_1(-z)$ and $F_1(z) = -H_0(-z)$.

Extra credit: Verify that the key equation $P(z) + P(-z) = 2$ still holds in the IIR case with the extended definition of the product filter

$$P(z) = \frac{z^l F_0(z) H_0(z)}{\det \boldsymbol{H}_m(z)}.$$

3. Find all filters if $H_0(z) = (\frac{1+z^{-1}}{2})^3$ and $P_0(z) = \frac{1}{16}(-1 + 9z^{-2} + 16z^{-3} + 9z^{-4} - z^{-6})$.

4. If an FIR filter $H_0(z)$ has three or more coefficients, explain why $H_0^2(z)$ has at least two odd powers. Then $H_0^2(z) - H_0^2(-z) = 2z^{-l}$ is impossible. The "alternating signs" construction is not PR. (This is extended in Theorem 5.3: Symmetry prevents orthogonality except with two coefficients.)

5. If H_0 and P_0 are symmetric, why is F_0 symmetric? Why is H_1 linear phase, and when can it be antisymmetric?

6. Prove Theorem 4.3 by observing that an order flip in linear-phase filters $H_0(e^{j\omega})$ and $F_0(e^{j\omega})$ gives $H_{0,R}$ and $F_{0,R}$:

$$\begin{cases} H_0(e^{j\omega}) = e^{-j(N_0/2)\omega} H_{0,R}(e^{-j\omega}), \quad F_0(e^{j\omega}) = e^{-j(N_1/2)\omega} F_{0,R}(e^{-j\omega}) \\ H_0(e^{j\omega})F_0(e^{j\omega}) + H_0(e^{j(\omega+\pi)})F_0(e^{j(\omega+\pi)}) = 2e^{-j\ell\omega}. \end{cases}$$

N_0 and N_1 are the degrees of $H_0(z)$ and $F_0(z)$, and ℓ is odd.

7. A symmetric filter has $H(z^{-1}) = z^N$ times (___). If $H(c) = 0$ show that also $H(\frac{1}{c}) = 0$.

8. In the example with six roots, show a 4/4 linear phase pair—the roots c and $\frac{1}{c}$ go together. Find the polynomials $H(z)$ and $F(z)$.

9. The 10th degree halfband polynomial $P_0(z) = \left(1 + z^{-1}\right)^6 Q(z)$ has four complex roots r, $\bar{r}$, r^{-1}, $\bar{r}^{-1}$ in the right halfplane (roots of Q). Draw a figure to show the ten roots and how Daubechies 6/6 filters will divide them: r and $\bar{r}$ are separated from r^{-1} and $\bar{r}^{-1}$.

10. For the same 10th degree $P_0(z)$, show how the ten roots can give 6/6 filters with linear phase. One filter has 5 zeros at $z = -1$.

11. (Good problem) Find the actual 4th degree $Q(z)$ that makes $P_0(z)$ halfband. If possible compute its roots.

12. Given a PR filter bank H_0, H_1, F_0, F_1, interchange $H_k(z)$ and $F_k(z)$ (so that the synthesis bank has $H_k(z)$ and analysis bank has $F_k(z)$). Verify that the new system is PR. Define another system $\widehat{H}_k(z) = H_k(-z)$ and $\widehat{F}_k(z) = F_k(-z)$. Is this new system PR?

13. Let $H_0(z)$ be a symmetric lowpass filter with even length and $H_1(z) = H_0(-z)$. Verify that $H_1(z)$ is an antisymmetric highpass filter. Find the synthesis filters $F_k(z)$ that cancel aliasing. Can this system be PR? (Is $P(z)$ a halfband filter?)

4.2 The Polyphase Matrix

This section establishes a key idea and a valuable notation. The word "*polyphase*" has gained a certain mystique in the theory of multirate filters. Perhaps we can begin by explaining the meaning of the word, and also the purpose of the idea. Then the notation and applications will come naturally.

Meaning of polyphase: When a vector is downsampled by 2, its even-numbered components are kept. Its odd-numbered components are lost. Those are the two phases, *even* and *odd*. It is natural to follow the two phases of the input vector, $\boldsymbol{x}_{\text{even}}$ and $\boldsymbol{x}_{\text{odd}}$, as they go through the filter bank. They are acted on by the two phases $\boldsymbol{H}_{\text{even}}$ and $\boldsymbol{H}_{\text{odd}}$ of the filter.

For downsampling by M there are M phases. The ideas still apply to this "several-phase" or "polyphase" decomposition. Instead of even and odd inputs we will have M vectors (phases of $\boldsymbol{x}$). Instead of even and odd filters we will have M filters (phases of $\boldsymbol{H}$). The vector of filter coefficients $\boldsymbol{h}(n)$ is separated into phases, exactly as $\boldsymbol{x}(n)$ is separated. Then we watch those phases during downsampling.

The word "phase" is applied because the even filter with coefficients $\boldsymbol{h}(0), \boldsymbol{h}(2)$ has a different delay (phase shift) from the odd phase with coefficients $\boldsymbol{h}(1), \boldsymbol{h}(3)$.

Purpose of polyphase: The operation $(\downarrow 2)\boldsymbol{H}\boldsymbol{x}$, taken literally, is not efficient. We are computing all components of $\boldsymbol{H}\boldsymbol{x}$ and then destroying half of them. If we don't compute them, the system is still working at a fast rate (high bandwidth). The output is at half rate, because of downsampling. Each output component needs N additions and $N+1$ multiplications, to apply all the coefficients $\boldsymbol{h}(0), \ldots, \boldsymbol{h}(N)$.

The polyphase implementation works on the different phases separately. The input vector is separated into $\boldsymbol{x}_{\text{even}}$ and $\boldsymbol{x}_{\text{odd}}$. The operator $(\downarrow 2)$ comes *before the filter*! It changes one input at a high rate to two (or M) inputs at a lower rate. Then the separate phases of the filters act simultaneously (*in parallel*) on separate phases of the input.

The notation has to keep track of each phase. Often we find that "even multiplies even" and "odd multiplies odd". The *Noble Identities* justify an interchange of filtering and sampling. For the whole filter this interchange is forbidden, but it is allowed for each phase.

Polyphase in the time domain (block Toeplitz matrix): We can display the infinite matrix for a 2-channel analysis bank. Recall that $\boldsymbol{c}(n) = \sqrt{2}\,\boldsymbol{h}_0(n)$ and $\boldsymbol{d}(n) = \sqrt{2}\,\boldsymbol{h}_1(n)$. The two filters $\sqrt{2}\boldsymbol{H}_0 = \boldsymbol{C}$ and $\sqrt{2}\boldsymbol{H}_1 = \boldsymbol{D}$ are downsampled by $(\downarrow 2)$. This removes the odd-numbered rows. Then we *interleave the rows* of $\boldsymbol{L} = (\downarrow 2)\boldsymbol{C}$ and $\boldsymbol{B} = (\downarrow 2)\boldsymbol{D}$ to see the analysis bank as a *block Toeplitz matrix*:

$$\textbf{Block Toeplitz } H_b = \begin{bmatrix} \cdot & \cdot & & & & & & \\ \cdot & \cdot & & & & & & \\ c(3) & c(2) & c(1) & c(0) & & & & \\ d(3) & d(2) & d(1) & d(0) & & & & \\ & & c(3) & c(2) & c(1) & c(0) & & \\ & & d(3) & d(2) & d(1) & d(0) & & \\ & & & & \cdot & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot & \cdot \end{bmatrix}.$$

This takes the input in blocks (two samples at a time). It gives the output in blocks. It is time-invariant in blocks! By block z-transform, multiplication by the infinite matrix H_b (which is block convolution) becomes multiplication by the *polyphase matrix*:

$$\textbf{Polyphase matrix} \quad H_p(z) = \begin{bmatrix} c(0) & c(1) \\ d(0) & d(1) \end{bmatrix} + z^{-1} \begin{bmatrix} c(2) & c(3) \\ d(2) & d(3) \end{bmatrix} \tag{4.20}$$

The polyphase matrix is nothing but *the z-transform of a block of filters*. There are 2^2 or M^2 filters, from M phases of M original filters. Here those filters have four coefficients and their phases have two coefficients.

Notice especially how the block matrix H_b relates to the two separate downsampled filters $(\downarrow 2)C$ and $(\downarrow 2)D$:

The efficient form downsamples the input *first* (to make blocks for H_b)

The inefficient form downsamples *last* (after the filters C and D)

The Noble Identities prove the equivalence. It is just a removal of useless odd-numbered rows and an interleaving of the remaining rows. Next we discuss the algebra and the implementation.

Key identity in the z-domain: The even part of $X(z)$ is $\frac{1}{2}(X(z) + X(-z))$. The odd part is $\frac{1}{2}(X(z) - X(-z))$. The first has even powers $1, z^2, z^4$; the second has z, z^3, z^5. The original X is the sum of even plus odd (obviously). The same splitting holds for $C(z)$, and furthermore for $C(z)X(z)$. *The key is to find the even part of $C(z)X(z)$.* It is the even coefficients of Cx that survive downsampling and appear in $(\downarrow 2)Cx$. In most of this section the lowpass filter is denoted by C, to avoid the subscripts on H.

A simple and important identity shows how the even part of $C(z)X(z)$ comes from even times even plus odd times odd:

$$\begin{aligned} \tfrac{1}{2}[C(z)X(z) + C(-z)X(-z)] &= \tfrac{1}{4}[C(z) + C(-z)][X(z) + X(-z)] \\ &\quad + \tfrac{1}{4}[C(z) - C(-z)][X(z) - X(-z)]. \end{aligned} \tag{4.21}$$

In multiplying numbers, odd times odd is odd. But we are *adding exponents*, as in $(z^3)(z^5) = z^8$. So it is really odd *plus* odd, and even *plus* even, that yield the even part of the z-transform. This is the part that downsampling picks out, when Cx is decimated.

The importance of the key identity is this. The left side involves *all* coefficients of $C(z)$ and $X(z)$. Each product on the right involves only *half* the coefficients. The multiplication in the z-domain, which is $(\downarrow 2)Cx$ in the time domain, becomes computationally efficient. We don't want all of $C(z)X(z)$, only the even half. The right side shows how to do half the work. Better still, it shows how even-even and odd-odd can be executed in parallel at half the rate.

Downsampling an even function effectively replaces z by $z^{1/2}$. It "closes the gaps" in 1, z^2, z^4 by changing to $1, z, z^2$. For an odd function we will need a delay or an advance. We cannot change z and z^3 to $z^{1/2}$ and $z^{3/2}$. You will see how the coefficients of z^{-1}, z^{-3}, z^{-5} in $C(z)$ become coefficients of $1, z^{-1}, z^{-2}$ in the odd phase $C_{\text{odd}}(z)$. There is a delay for the odd phase and a "delay chain" when there are multiple phases.

This chapter works out the polyphase notation. We concentrate most on $M = 2$; the phases are even and odd. Then the polyphase forms of the analysis and synthesis banks lead quickly to a main goal of the theory. We find the perfect reconstruction condition on the polyphase matrices, when the filters are centered:

$$\boldsymbol{F}_p(z)\boldsymbol{H}_p(z) = \boldsymbol{I}.$$

This tells us, clearly and directly, what is required:

1. At a minimum, $\boldsymbol{H}_p(z)$ must be ***invertible***. (biorthogonality)
2. Better than that, its inverse $\boldsymbol{F}_p(z)$ should be a ***polynomial***. (FIR)
3. Better still, $\boldsymbol{F}_p(z)$ might be the ***transpose*** of $\boldsymbol{H}_p(z)$. (orthogonality)

In case **3**, the polyphase matrices are "paraunitary". The analysis and synthesis banks are orthogonal. In the more general case **1**, the banks are "biorthogonal". In case **2**, the synthesis bank is biorthogonal and also FIR.

The rows of a matrix are always biorthogonal to the columns of its inverse. When the rows of one are identical to the columns of the other, the matrix is self-orthogonal. Then it is an *orthogonal* matrix if real, a *unitary* matrix if complex, and a *paraunitary* matrix if it is a function of a complex parameter z.

Polyphase for Vectors

Any input vector $\boldsymbol{x}$ and any filter vector $\boldsymbol{c}$ or $\boldsymbol{h}$ can be separated into even and odd:

$$\boldsymbol{x} = (\ldots, x(0), 0, x(2), 0, \ldots) + (\ldots, 0, x(1), 0, x(3), 0, \ldots).$$

The z-transform is separated into even powers and odd powers, as in

$$X(z) \quad = \quad [x(0) + x(2)z^{-2} + \cdots] + z^{-1}[x(1) + x(3)z^{-2} + \cdots]. \tag{4.22}$$

The even part has powers of z^2. So has the odd part, when we factor out z^{-1}. This is the polyphase decomposition of $\boldsymbol{x}$ in the z-domain:

$$X(z) = X_{\text{even}}(z^2) + z^{-1}X_{\text{odd}}(z^2) \tag{4.23}$$

Each phase has its own z-transform:

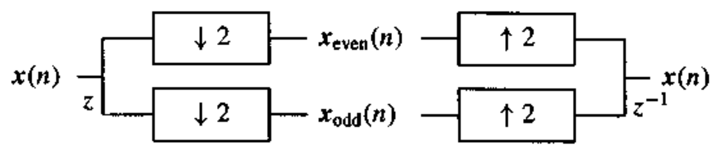
$$\boldsymbol{x}_{\text{even}} = \begin{bmatrix} x(0) \\ x(2) \\ \cdot \end{bmatrix} \leftrightarrow X_0(z) = \sum x(2k)z^{-k}$$

$$\boldsymbol{x}_{\text{odd}} = \begin{bmatrix} x(1) \\ x(3) \\ \cdot \end{bmatrix} \leftrightarrow X_1(z) = \sum x(2k+1)z^{-k}.$$

Because of the z^2 in the definition, the in-between zeros are gone from $X_0(z)$ and $X_1(z)$. Please verify that the phases of $X(z) = z^{-1} + z^{-2} + z^{-3}$ are $X_{\text{even}} = z^{-1}$ and $X_{\text{odd}} = 1 + z^{-1}$.

Now reverse the process, to recover $\boldsymbol{x}$. Upsampling puts zeros back into $\boldsymbol{x}_{\text{even}}$ and $\boldsymbol{x}_{\text{odd}}$. *Those zeros change z to z^2.* The odd phase is delayed by z^{-1}, to move $x(1)$ from position 0 to position 1. Then addition reconstructs equation (4.25).

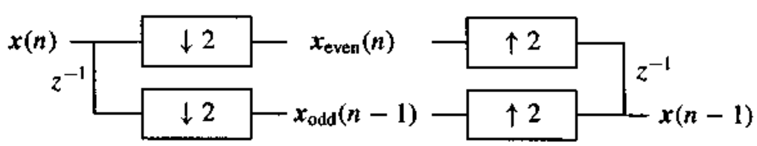
Here is the splitting and the reconstruction in block form. Notice that so far the filters are not included, and $(\downarrow 2)\boldsymbol{x}$ is exactly $\boldsymbol{x}_{\text{even}}$:



Important! The z at the start of the odd channel is because the odd phase has $x(1)$ in its zeroth position. We have to advance the signal to achieve that. (See Problem 1.) But advances look bad in our flow diagram. So the advance can be replaced by a delay, if we make up for it at the end by delaying the even part too.

Here is the "delay form" that we use in later sections. Please go through that form:

Polyphase with Delay

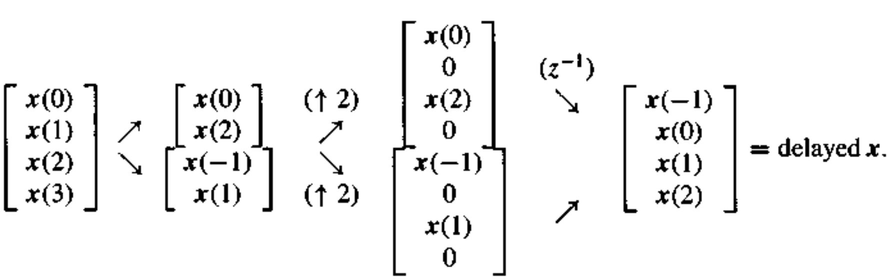


Only delays are involved! This is its advantage. Its disadvantage is that the output $\widehat{x}(n)$ is $x(n-1)$. The whole system equals a delay, where previously the system reproduced $\boldsymbol{x}$. The delay form in the z-domain produces $z^{-1}X(z)$ by delaying the even term:

$$z^{-1}X(z) = z^{-1}X_0(z^2) + z^{-2}X_1(z^2). \tag{4.24}$$

This *polyphase with delay* is just the original definition multiplied by z^{-1}.

Your eye will pick out this delay form. The output $\widehat{\boldsymbol{x}}$ comes later than the input. We still call this perfect reconstruction. Here is the same delay form in the time domain:



Polyphase Matrices for Filters

The polyphase form of a filter C comes directly from the polyphase form of c (the vector of filter coefficients). That vector separates into c_{even} and c_{odd}. Its z-transform $C(z)$ separates into phases exactly as $X(z)$ did:

$$C(z) = C_0(z^2) + z^{-1}C_1(z^2). \tag{4.25}$$

The filtering step is $C(z)X(z)$. This is ordinary filtering Cx, where even mixes with odd. But when downsampling picks out the even part of the product $C(z)X(z)$, it comes from even times even plus odd times odd. The transform of $(\downarrow 2)Cx$ is

$$(C(z)X(z))_{\text{even}} = C_0(z)X_0(z) + z^{-1}C_1(z)X_1(z). \tag{4.26}$$

The direct multiplication of $C(z)$ times $X(z)$ will have even parts from $C_0(z^2)X_0(z^2)$ and from $z^{-2}C_1(z^2)X_1(z^2)$. Those give the even part of $C(z)X(z)$. Downsampling picks out those terms. It changes z^2 to z in their transform. The result is the z-transform of $(\downarrow 2)Cx$.

Example 4.3. The moving average filter, downsampled.
Chapter 1 introduced the lowpass filter $\frac{1}{2}x(n) + \frac{1}{2}x(n-1)$. The coefficients are $h(0) = \frac{1}{2}$ and $h(1) = \frac{1}{2}$. Thus $\frac{1}{2}$ is the leading and only coefficient in H_{even} and H_{odd}. Both matrices have $\frac{1}{2}$ on *one diagonal*—the main diagonal. Here is the two-phase form of $(\downarrow 2)Hx$:

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & & & \\ & & \frac{1}{2} & \frac{1}{2} & \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} \\ x \\ \\ \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & & \\ & \frac{1}{2} & \\ & & \ddots \end{bmatrix} \begin{bmatrix} x(0) \\ x(2) \\ \cdot \end{bmatrix} + \begin{bmatrix} \frac{1}{2} & & \\ & \frac{1}{2} & \\ & & \ddots \end{bmatrix} \begin{bmatrix} x(-1) \\ x(1) \\ \cdot \end{bmatrix}$$

The polyphase components of $\frac{1}{2} + \frac{1}{2}z^{-1}$ are constants: $H_{\text{even}}(z) = H_{\text{odd}}(z) = \frac{1}{2}$.

Polyphase for one filter. The downsampling operator $(\downarrow 2)$ follows the filter C. The outputs are Cx and then $(\downarrow 2)Cx$. This is the normal order for filter banks, but it can be made more efficient. A close look at the product $(\downarrow 2)C$ shows that *the filter coefficients $c(n)$ are completely separated into even n and odd n.*

Thus C has two parts (or phases), which have their own z-transforms $C_0(z) = C_{\text{even}}(z)$ and $C_1(z) = C_{\text{odd}}(z)$. The 1 by 2 matrix $C_p(z)$ is the two-phase or polyphase form of $C(z)$:

$$C_p(z) = [C_0(z) \quad C_1(z)].$$

Warning. $C_0(z)$ is not $\frac{1}{2}(C(z)+C(-z))$. That involves $1, z^2, z^4, \ldots$ with zeros between. This is $C_0(z^2)$. By definition, $C_0(z)$ closes the zero gaps and has coefficients $c(0), c(2), c(4)$. Similarly $C_1(z)$ has coefficients $c(1), c(3), c(5)$. An advance or delay is involved. We don't keep the zeros from the even powers in $\frac{1}{2}(C(z) - C(-z))$.

The same splitting occurs for the highpass filter D. Its even and odd phases are represented by $D_0(z)$ and $D_1(z)$. Those go into the 1 by 2 polyphase matrix $D_p(z)$. Then the whole analysis bank comes together when we combine the polyphase matrices for C and D into a single polyphase matrix $H_p(z)$:

$$\textbf{Polyphase Matrix} \quad H_p(z) = \begin{bmatrix} C_p(z) \\ D_p(z) \end{bmatrix} = \begin{bmatrix} C_0(z) & C_1(z) \\ D_0(z) & D_1(z) \end{bmatrix} \tag{4.27}$$

This shows the matrix that we are aiming for. Now we go back for the close look at $(\downarrow 2)C$. This operator is fundamental in the theory of multirate filters and wavelets.

Polyphase in the time domain. When downsampling follows the filter C, we get the crucial matrix $L = (\downarrow 2)C$. This has to display the separation of even and odd, and I would like to show how this happens. Most of polyphase theory is developed in the z-domain, and we will do that too. But first, look at the filter matrix as it produces $y = Cx$:

$$\begin{bmatrix} \cdot \\ y(0) \\ y(1) \\ y(2) \\ y(3) \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & & & & & \\ \cdot & c(0) & & & & \\ \cdot & c(1) & c(0) & & & \\ \cdot & c(2) & c(1) & c(0) & & \\ \cdot & c(3) & c(2) & c(1) & c(0) & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ x(0) \\ x(1) \\ x(2) \\ x(3) \\ \cdot \end{bmatrix}. \tag{4.28}$$

Downsampling leaves the even-numbered components $y(2n)$. To reach $v = (\downarrow 2)y$, we throw away the odd-numbered rows. This leaves the matrix $L = (\downarrow 2)C$:

$$\begin{bmatrix} \cdot \\ y(0) \\ y(2) \\ y(4) \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & & & & & & \\ \cdot & c(1) & c(0) & & & & \\ \cdot & c(3) & c(2) & c(1) & c(0) & & \\ \cdot & c(5) & c(4) & c(3) & c(2) & c(1) & c(0) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ x(-1) \\ x(0) \\ x(1) \\ x(2) \\ \cdot \end{bmatrix}. \tag{4.29}$$

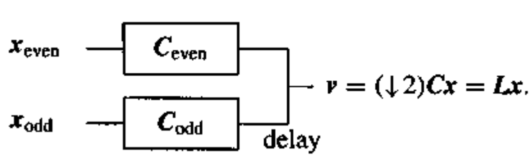
For polyphase here is the important point. *Only the even-numbered coefficients* $c(2n)$ *are multiplying the even-numbered coefficients* $x(2n)$. The even and odd c's are in separate columns. The even-numbered $x(0)$ is multiplying the column that starts with $c(0)$. The odd-numbered component $x(1)$ is multiplying the column containing $c(1), c(3), \ldots$. We can separate the matrix multiplication $(\downarrow 2)Cx$ into *even times even and odd times odd*:

$$\begin{bmatrix} \cdot \\ y(0) \\ y(2) \\ y(4) \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot & & & \\ \cdot & c(0) & & \\ \cdot & c(2) & c(0) & \\ \cdot & c(4) & c(2) & c(0) \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ x(0) \\ x(2) \\ \cdot \end{bmatrix} + \begin{bmatrix} \cdot & & & \\ \cdot & c(1) & & \\ \cdot & c(3) & c(1) & \\ \cdot & c(5) & c(3) & c(1) \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot \\ x(-1) \\ x(1) \\ \cdot \end{bmatrix} \tag{4.30}$$

This is a matrix display of equation (4.21):

$$(\downarrow 2)Cx = C_{\text{even}}\, x_{\text{even}} + (\text{delay})\, C_{\text{odd}}\, x_{\text{odd}}. \tag{4.31}$$

The two phases x_{even} and x_{odd} are filtered by the two polyphase components C_{even} and C_{odd}. We need a delay in the odd phase, because $c(1)x(1)$ contributes to $y(2)$ and not to $y(0)$. Then $(\downarrow 2)Cx$ is the sum from the two phases:



Notice something nice. The two matrices in equation (4.30) have constant diagonals. *The two operators* C_{even} *and* C_{odd} *are time-invariant filters.* They have frequency responses $C_0(z) =$

$C_{\text{even}}(z)$ and $C_1(z) = C_{\text{odd}}(z)$. The delay in the odd channel can go before or after C_1, because it commutes with C_1. (C_1 is time-invariant!) The two filters involve even coefficients $c(2n)$ and odd coefficients $c(2n+1)$, without zeros in between. They can operate in parallel, more efficiently.

Summary: Polyphase form of $L = (\downarrow 2)C$

The matrix C_{even} multiplies x_{even} in equation (4.31). The matrix C_{odd} multiplies x_{odd} (with a delay). We repeat this time-domain multiplication so you can compare it with the z-domain:

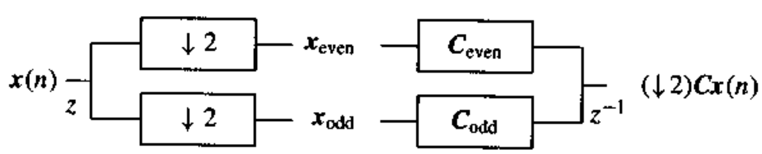
$$(\downarrow 2)Cx = [C_{\text{even}} \quad (delay)C_{\text{odd}}] \begin{bmatrix} x_{\text{even}} \\ x_{\text{odd}} \end{bmatrix}. \tag{4.32}$$

This polyphase form has two ordinary filter matrices side by side. The z-domain polyphase form has two ordinary transfer functions side by side:

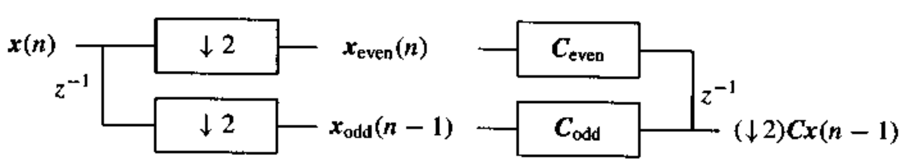
$$(C(z)X(z))_{\text{even}} = [C_{\text{even}}(z) \quad z^{-1}C_{\text{odd}}(z)] \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix}. \tag{4.33}$$

If C = identity then $C_{\text{even}}(z) = 1$ and the output is $X_0(z)$. If C = delay then $C_{\text{odd}}(z) = 1$ and the output is $X_1(z)$. All straightforward, but the block form shows something remarkable: *Downsampling comes before filtering* !

The block form with an advance to compute x_{odd} is ***Polyphase with Advance***:



An advance has slipped into this form to get $x(1)$ as the zeroth component of x_{odd}. If we prefer to have only delays (and we do), that is possible. Delay x in the odd channel so that $(\downarrow 2)$ produces $x_{\text{odd}}(n-1)$. Then after filtering, add a delay in the even channel. The result is to delay the whole signal by one time step in ***Polyphase with Delay***:



Key point: The polyphase form puts $(\downarrow 2)$ *before* the filters. This order is more efficient. It was possible to use the Noble Identity on each phase separately, because $C_{\text{even}}(z^2)$ and also $C_{\text{odd}}(z^2)$ appeared in the right place:

$$\text{(direct)} \quad (\downarrow 2)C_{\text{even}}(z) = C_{\text{even}}(z^2)(\downarrow 2) \quad \text{(polyphase)}.$$

Note on our convention. It would be very satisfying to avoid the delays completely. We would prefer to have

$$V(z) = [C_0(z) \quad C_1(z)] \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix}. \tag{4.34}$$

To achieve this we can alter the definition of $C_1(z)$. The polyphase decomposition of X would stay the same, but the decomposition of C would change to

$$C(z) = C_0(z^2) + zC_1(z^2). \tag{4.35}$$

Note z in the odd term where we had z^{-1}. This is the convention chosen by [VK]. It is just as good as ours. The zeroth component of C_{odd} becomes $c(-1)$. It multiplies $x(1)$. But I am afraid that in later chapters you would forget (I would too) this convention for $\boldsymbol{c}$, different from $\boldsymbol{x}$. So we keep the same even-odd decomposition of $\boldsymbol{c}$ and $\boldsymbol{x}$, yielding parallel decompositions of $C(z)$ and $X(z)$ — and requiring the delay.

Problem Set 4.2

1. Find $X_{\text{even}}(z)$ and $X_{\text{odd}}(z)$ when $X(z) = 1 + 2z^{-5} + z^{-10}$. Verify that $X_{\text{even}}(z^2) = \frac{1}{2}(X(z) + X(-z))$ and $X_{\text{odd}}(z^2) = \frac{z}{2}(X(z) - X(-z))$. The odd definition involves an advance!

2. Express the z-transform of $\uparrow 2(\downarrow 2)\boldsymbol{x}$ in terms of X_{odd} and/or X_{even}.
What operations on $\boldsymbol{x}$ would produce the vector whose transform is $X_1(z^2)$?

3. The phases C_0, C_1, D_0, D_1 are all time-invariant, so they commute with delays. Does it follow that
$$\begin{bmatrix} 1 & \\ & \text{delay} \end{bmatrix} \begin{bmatrix} C_0 & C_1 \\ D_0 & D_1 \end{bmatrix} = \begin{bmatrix} C_0 & C_1 \\ D_0 & D_1 \end{bmatrix} \begin{bmatrix} 1 & \\ & \text{delay} \end{bmatrix}?$$

4. *Polyphase Representation of an IIR Transfer Function*
Let $H(z) = \frac{1}{1-az^{-1}}$ where $0 < a < 1$. Its impulse response is $h(n) = a^n$ for $n \geq 0$ (and zero for negative n). The phases are $h_{\text{even}}(n) = (1, a^2, a^4, \ldots)$ and $h_{\text{odd}}(n) = (a, a^3, a^5, \ldots)$. The z-transforms are $H_{\text{even}}(z) = 1/(1-a^2z^{-1})$ and $H_{\text{odd}}(z) = a/(1-a^2z^{-1})$. This method is very cumbersome. One has to find the impulse response $h(n)$, then its even and odd parts $h_{\text{even}}(n)$ and $h_{\text{odd}}(n)$, then the z-transforms.
An alternate method is to write $H(z) = \frac{1}{1-az^{-1}}$ directly as $H(z) = H_{\text{even}}(z^2) + z^{-1}H_{\text{odd}}(z^2)$. The denominator must be a function of z^2. So multiply above and below by $1 + az^{-1}$:
$$H(z) = \frac{1}{1-az^{-1}}\,\frac{1+az^{-1}}{1+az^{-1}} = \frac{1+az^{-1}}{1-a^2z^{-2}} = \frac{1}{1-a^2z^{-2}} + z^{-1}\frac{a}{1-a^2z^{-2}}.$$
This displays $H_{\text{even}}(z)$ and $H_{\text{odd}}(z)$. An Nth order filter can be factored as a cascade of first-order sections, and this method applies to each section.
(a) Let $H(z) = \frac{1}{1-\frac{5}{6}z^{-1}+\frac{1}{6}z^{-2}}$. Factor $H(z)$ into two first-order poles. Find the polyphase components of $H(z)$.
(b) Let $H(z) = \frac{1+2z^{-1}+5z^{-2}}{1-\frac{5}{6}z^{-1}+\frac{1}{6}z^{-2}}$. What are its polyphase components?

5. For $M = 4$ channels, we want the four polyphase components of $H(z)$:
$$H(z) = H_0(z^4) + z^{-1}H_1(z^4) + z^{-2}H_2(z^4) + z^{-3}H_3(z^4).$$
(a) What polynomial multiplies $1 - az^{-1}$ to produce $1 - a^4z^{-4}$?
(b) Find the four components for $\frac{1}{1-az^{-1}}$ and $\frac{1+2z^{-1}+5z^{-2}}{1-\frac{5}{6}z^{-1}+\frac{1}{6}z^{-2}}$.

6. If H is a symmetric filter, how many of its phases are symmetric filters?
7. Let $H(z) = 1+2z^{-1}+3z^{-2}+4z^{-3}+4z^{-4}+3z^{-5}+2z^{-6}+z^{-7}$. Find the polyphase components $H_{even}(z)$ and $H_{odd}(z)$. What is the relation between $H_{even}(z)$ and $H_{odd}(z)$ for antisymmetric filters of even length and symmetric filters of odd length?
8. What are the two polyphase components of a symmetric halfband filter? Generalize to an M-th band filter.

4.3 Efficient Filter Banks

Let us repeat the key idea of the polyphase decomposition. The input is x, the filter is C, and we intend to downsample Cx. We want the *even powers of* z in the product $C(z)X(z)$:

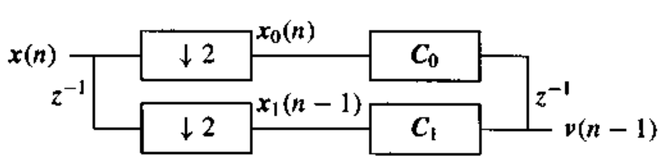
$$\text{even powers come from} \begin{cases} \text{even powers in } C(z) \text{ times even powers in } X(z) \\ \text{odd powers in } C(z) \text{ times odd powers in } X(z). \end{cases}$$

The polyphase decomposition is exactly this separation. Even times even is $C_0(z^2)X_0(z^2)$. Odd times odd starts with $c(1)z^{-1}$ times $x(1)z^{-1}$. The product $c(1)x(1)$ enters $Cx(2)$ because indices add. Then $Cx(2)$ after downsampling is $v(1)$. Since $c(1)$ and $x(1)$ are the *zeroth components* of the odd phases, we need a delay to put their product into $v(1)$. The algebra with $X(z) = X_0(z^2) + z^{-1}X_1(z^2)$ and $C(z) = C_0(z^2) + z^{-1}C_1(z^2)$ is

$$\begin{aligned} (CX)_0(z^2) &= C_0(z^2)X_0(z^2) + z^{-2}C_1(z^2)X_1(z^2) \\ (CX)_0(z) &= C_0(z)X_0(z) + z^{-1}C_1(z)X_1(z). \end{aligned} \tag{4.36}$$

We reached this answer by filtering and then subsampling (the step when z^2 becomes z). But now we see a better way. *Sample first* to get $x_0 = x_{\text{even}}$ and $x_1 = x_{\text{odd}}$. Filter those separately (and in parallel) by C_0 and C_1. Then combine the outputs with a suitable delay — the step down on the right:

Polyphase with Delay

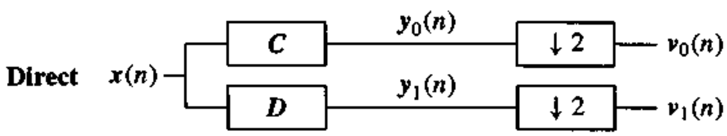


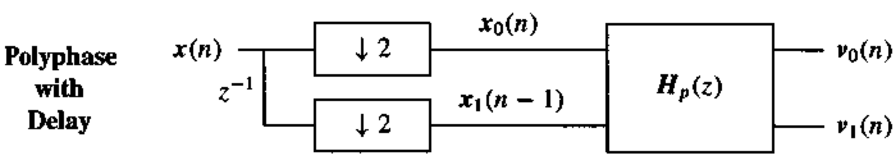
In this polyphase form, *the filters* C_0 and C_1 *are half as long as* C. The output can be computed twice as fast, if all operations are done at the same speed. We could also use two cheaper low bandwidth processors, both working full time. They do $\frac{N}{2}$ multiplications and additions per unit time instead of N. By downsampling first, we reduce the input rate to the filters. The bandwidth is halved.

Example 4.4. The averaging filter in its direct form computes $\frac{1}{2}x(0)+\frac{1}{2}x(-1)$. Then it sits idle for one clock step, before computing $\frac{1}{2}x(2) + \frac{1}{2}x(1)$. The polyphase form has H_{even} and H_{odd} multiplying separately by $\frac{1}{2}$. They do one multiplication each, in parallel. An easy-to-write code will execute the polyphase form.

Polyphase for Filter Banks

The polyphase idea extends from one filter to a bank of filters. The direct form of the analysis bank does the downsampling last. *The polyphase form does the downsampling first.* In the block diagram of the filter bank, the decimators move *outside* the filters. We can write $\boldsymbol{C}$ and $\boldsymbol{D}$ or $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$ for the lowpass and highpass filters (0 and 1 do not mean even and odd!):





The *polyphase matrix* multiplies $X_0(z)$ and $z^{-1}X_1(z)$ to produce $V_0(z)$ and $V_1(z)$:

$$\begin{bmatrix} V_0(z) \\ V_1(z) \end{bmatrix} = \begin{bmatrix} C_0(z) & C_1(z) \\ D_0(z) & D_1(z) \end{bmatrix} \begin{bmatrix} X_0(z) \\ z^{-1}X_1(z) \end{bmatrix} = \boldsymbol{H}_p(z) \begin{bmatrix} X_0(z) \\ z^{-1}X_1(z) \end{bmatrix} \tag{4.37}$$

This defines and displays $\boldsymbol{H}_p(z)$. For FIR causal filters, the kind we expect to use, the polyphase components are polynomials in z^{-1}. When the input $\boldsymbol{x}$ is also causal, the outputs are causal.

Another point about notation. The indices in X_0 and X_1 refer to even and odd. The indices in V_0 and V_1 refer to the two channels. This is normal for matrix multiplication, when H_{ij} multiplies X_j and contributes to V_i. *Rows of $\boldsymbol{H}_p(z)$ go with channels, and columns of $\boldsymbol{H}_p(z)$ go with phases.*

In an M-channel bank, i is the channel index and j is the phase index in $H_{ij}(z)$. Then V_i is the output from channel i, and X_j is the jth phase of the input. We often reorganize a filter bank into its polyphase form.

Example 4.5. Average-difference filter bank in polyphase form.

The averaging filter $\boldsymbol{H}_0\boldsymbol{x}(n) = \frac{1}{2}\boldsymbol{x}(n) + \frac{1}{2}\boldsymbol{x}(n-1)$ has polyphase components $\boldsymbol{H}_{0,\text{even}} = \boldsymbol{H}_{0,\text{odd}} = \frac{1}{2}(\text{identity})$. The differencing filter $\boldsymbol{H}_1\boldsymbol{x}(n) = \frac{1}{2}\boldsymbol{x}(n) - \frac{1}{2}\boldsymbol{x}(n-1)$ has components $\boldsymbol{H}_{1,\text{even}} = \frac{1}{2}\boldsymbol{I}$ and $\boldsymbol{H}_{1,\text{odd}} = -\frac{1}{2}\boldsymbol{I}$. Note that $H_0(z)$ and $H_1(z)$ are linear but the polyphase matrix is constant — typical of a *block transform*:

$$\boldsymbol{H}_p(z) = \begin{bmatrix} H_{00}(z) & H_{01}(z) \\ H_{10}(z) & H_{11}(z) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}.$$

Example 4.6. Four-tap filters yield two taps for each phase. The even phase $\boldsymbol{C}_{\text{even}}$ has two coefficients $\boldsymbol{c}(0)$ and $\boldsymbol{c}(2)$. The odd phase has $\boldsymbol{C}_{\text{odd}} = \boldsymbol{c}(1) + \boldsymbol{c}(3)z^{-1}$. The same pattern holds for $\boldsymbol{D}$. The polyphase matrix for the filter bank is

$$\boldsymbol{H}_p(z) = \begin{bmatrix} \boldsymbol{c}(0) + \boldsymbol{c}(2)z^{-1} & \boldsymbol{c}(1) + \boldsymbol{c}(3)z^{-1} \\ \boldsymbol{d}(0) + \boldsymbol{d}(2)z^{-1} & \boldsymbol{d}(1) + \boldsymbol{d}(3)z^{-1} \end{bmatrix}. \tag{4.38}$$

Even is separated from odd. This reflects what happens in the filter bank, when Cx and Dx are downsampled:

$$\begin{bmatrix} v_0 \\ v_1 \end{bmatrix} = \begin{bmatrix} (\downarrow 2)Cx \\ (\downarrow 2)Dx \end{bmatrix} = \begin{bmatrix} C_{\text{even}} & C_{\text{odd}} \\ D_{\text{even}} & D_{\text{odd}} \end{bmatrix} \begin{bmatrix} 1 & \\ & \text{delay} \end{bmatrix} \begin{bmatrix} x_{\text{even}} \\ x_{\text{odd}} \end{bmatrix}$$

In case you like matrices, we are going to write the time-domain filter bank matrix in three ways. Downsampling is included in all three! First comes the matrix $H_d = H_{\text{direct}}$ that multiplies the input vector x in the direct form:

$$\textbf{Direct} \quad H_d = \begin{bmatrix} \cdot & \cdot & & & & & & \\ c(3) & c(2) & c(1) & c(0) & & & & \\ & & c(3) & c(2) & c(1) & c(0) & & \\ & & & & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & & & & \\ d(3) & d(2) & d(1) & d(0) & & & & \\ & & d(3) & d(2) & d(1) & d(0) & & \\ & & & & \cdot & \cdot & \cdot & \cdot \end{bmatrix}. \tag{4.39}$$

Downsampling has removed every other row. That leaves this "square" infinite matrix. Each column is completely odd or completely even.

For the second form we rearrange the *rows* of H_d. The highpass outputs are interleaved with the lowpass outputs, both downsampled by 2. This produces the block-diagonal form (or *block Toeplitz form*) $H_b = H_{\text{block}}$:

$$\textbf{Block} \quad H_b = \begin{bmatrix} \cdot & \cdot & & & & & & \\ \cdot & \cdot & & & & & & \\ c(3) & c(2) & c(1) & c(0) & & & & \\ d(3) & d(2) & d(1) & d(0) & & & & \\ & & c(3) & c(2) & c(1) & c(0) & & \\ & & d(3) & d(2) & d(1) & d(0) & & \\ & & & & \cdot & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \cdot & \cdot \end{bmatrix}. \tag{4.40}$$

Your eye will divide that matrix into 2 by 2 blocks. It is like an ordinary time-invariant constant-diagonal matrix, but the entries are blocks instead of scalars. The main diagonal block corresponds to the constants in the polyphase matrix. The subdiagonal block produces the z^{-1} terms. There are only two diagonals because the phases of C and D have *two* coefficients. The original C and D had four coefficients.

The third form is the polyphase form H_p. We are still in the time domain. For this third form we rearrange the *columns* of the direct form:

$$\textbf{Polyphase} \quad H_p = \begin{bmatrix} \cdot & & & \cdot & & \\ c(2) & c(0) & & c(3) & c(1) & \\ & c(2) & c(0) & & c(3) & c(1) \\ & & \cdot\ \cdot & & & \cdot\ \cdot \\ \cdot & & & \cdot & & \\ d(2) & d(0) & & d(3) & d(1) & \\ & d(2) & d(0) & & d(3) & d(1) \\ & & \cdot\ \cdot & & & \cdot\ \cdot \end{bmatrix} = \begin{bmatrix} C_0 & C_1 \\ D_0 & D_1 \end{bmatrix}. \tag{4.41}$$

When the columns are rearranged, the vector $\boldsymbol{x}$ must be rearranged. Here $\boldsymbol{x}_{\text{even}}$ comes above $\boldsymbol{x}_{\text{odd}}$(delayed). The transform of the time-domain matrix $\boldsymbol{H}_p$ is the z-domain polyphase matrix $\boldsymbol{H}_p(z)$. *This 2 by 2 matrix of filters becomes a 2 by 2 matrix of functions.*

The block form $\boldsymbol{H}_b$ is an infinite matrix of 2 by 2 blocks. The polyphase form $\boldsymbol{H}_p$ is a 2 by 2 matrix of infinite blocks. Each block is a time-invariant filter. Either form leads by z-transform to the 2×2 polyphase matrix $\boldsymbol{h}_p(0) + z^{-1}\boldsymbol{h}_p(1)$:

Polyphase matrix $$\boldsymbol{H}_p(z) = \begin{bmatrix} c(0) & c(1) \\ d(0) & d(1) \end{bmatrix} + z^{-1} \begin{bmatrix} c(2) & c(3) \\ d(2) & d(3) \end{bmatrix} \qquad (4.42)$$

Relation Between Modulation and Polyphase

To produce two vectors from $\boldsymbol{x}$, one way is by polyphase. The parts $\boldsymbol{x}_{\text{even}}$ and $\boldsymbol{x}_{\text{odd}}$ are half-length. The other way is by modulation. Both $\boldsymbol{x}$ and $\boldsymbol{x}_{\text{mod}}$ are full length (therefore redundant). *The modulated vector* $\boldsymbol{x}_{\text{mod}}$ *reverses the sign of odd-numbered components*:

$$x_{\text{mod}}(n) = (-1)^n x(n) \quad \text{and} \quad X_{\text{mod}}(z) = X(-z).$$

In the frequency domain, $-1 = e^{i\pi}$ and ω is modulated by π — which explains the name:

$$X_{\text{mod}}(\omega) = X(\omega + \pi). \qquad (4.43)$$

The vector $\boldsymbol{x}_{\text{mod}}$ appears naturally when upsampling follows downsampling. Remember that $\boldsymbol{u} = (\uparrow 2)(\downarrow 2)\boldsymbol{x}$ is the average of $\boldsymbol{x}$ and $\boldsymbol{x}_{\text{mod}}$:

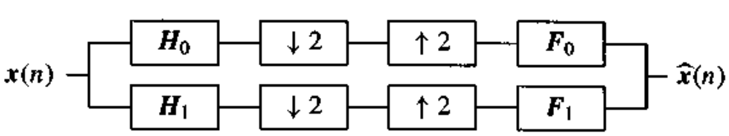
$$\boldsymbol{u} = \tfrac{1}{2}(\boldsymbol{x} + \boldsymbol{x}_{\text{mod}}) = (\ldots, x(0), 0, x(2), 0, \ldots). \qquad (4.44)$$

In the z-domain, the same fact gives the formula we use constantly:

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

For downsampling and upsampling by M, the frequency modulation is by multiples of $2\pi/M$. The z-domain equivalent is multiplication by $W = e^{-2\pi i/M}$. There are M terms $X(W^k z)$. We continue with $M = 2$ terms, $X(z)$ and $X(-z)$.

The extra term represents aliasing. This modulated signal can overlap the original signal (in the frequency domain). If it does, we cannot recover $\boldsymbol{x}$ from $\boldsymbol{u}$. Now place this step $(\uparrow 2)(\downarrow 2)$ into a complete filter bank:



The transform of $\boldsymbol{H}_0\boldsymbol{x}$ is $H_0(z)X(z)$. When this is downsampled and upsampled, its alias appears in $\frac{1}{2}[H_0(z)X(z) + H_0(-z)X(-z)]$. Multiplying by $F_0(z)$ gives the output from the top channel. The lower channel is the same with 0 replaced by 1. The final output is the sum of channels:

$$\widehat{X}(z) = \tfrac{1}{2}[F_0(z) \quad F_1(z)] \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix} \begin{bmatrix} X(z) \\ X(-z) \end{bmatrix} \qquad (4.45)$$

This is the 2×2 modulation matrix $\boldsymbol{H}_m(z)$. It contains $H_0(z)$ and $H_1(z)$ along with their aliases. *The transpose of $\boldsymbol{H}_m(z)$ is also called the alias component matrix.*

A perfect reconstruction filter bank has to avoid aliasing (and other distortions). There is no aliasing when the combination $F_0(z)H_0(-z) + F_1(z)H_1(-z)$ is zero. Section 4.1 presented the synthesis filters $F_0(z) = H_1(-z)$ and $F_1(z) = -H_0(-z)$ that cancel aliasing.

Now turn to polyphase, the separation into even and odd phases:

$$H(z) = \tfrac{1}{2}[H(z) + H(-z)] + \tfrac{1}{2}[H(z) - H(-z)]. \tag{4.46}$$

The first bracket is even. Replacing z by $-z$ leaves it unchanged. The second bracket is odd. Replacing z by $-z$ reverses its sign. Therefore (4.46) must be the polyphase decomposition $H_{\text{even}}(z^2) + z^{-1}H_{\text{odd}}(z^2)$:

$$\begin{aligned}[H_{\text{even}}(z^2) \quad z^{-1}H_{\text{odd}}(z^2)] &= \tfrac{1}{2}[H(z) + H(-z) \quad H(z) - H(-z)] \\ &= \tfrac{1}{2}[H(z) \quad H(-z)]\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.\end{aligned} \tag{4.47}$$

This connects polyphase to modulation. For one filter $\boldsymbol{H}_p$ and $\boldsymbol{H}_m$ have one row. For a bank of filters $\boldsymbol{H}_p$ and $\boldsymbol{H}_m$ are square matrices. A 2-channel analysis bank has equation (4.47) for H_0 in the top row, and the same equation for H_1 in the lower row:

$$\begin{bmatrix} H_{0,\text{even}}(z^2) & H_{0,\text{odd}}(z^2) \\ H_{1,\text{even}}(z^2) & H_{1,\text{odd}}(z^2) \end{bmatrix}\begin{bmatrix} 1 & \\ & z^{-1} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4.48}$$

Theorem 4.4 *The polyphase matrix $\boldsymbol{H}_p$ is connected to the modulation matrix $\boldsymbol{H}_m$ by a 2-point DFT and a diagonal delay matrix $D(z) = diag(1, z^{-1})$.*

This pattern extends in Section 9.2 to a bank of M filters. Each filter has M phases, and it has M modulations (by multiples of $2\pi/M$). The phases are associated with time shifts, and the modulations are frequency shifts. So the DFT connects them.

Problem Set 4.3

1. From $H(z) = H_{\text{even}}(z^2) + z^{-1}H_{\text{odd}}(z^2)$, find the corresponding formula for $H(-z)$. Write the two equations as

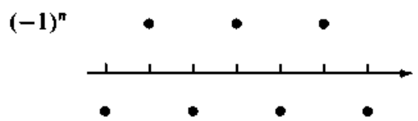
$$[H(z) \quad H(-z)] = [H_{\text{even}}(z^2) \quad H_{\text{odd}}(z^2)]\begin{bmatrix} 1 & \\ & z^{-1} \end{bmatrix}\begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}.$$

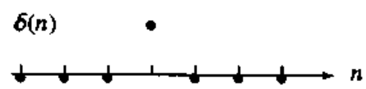
Identify that last matrix and verify that you have inverted equation (4.48). What is $\boldsymbol{H}_m(z)$ in terms of $\boldsymbol{H}_p(z)$?

2. If $\boldsymbol{H}_m^T(z^{-1})\boldsymbol{H}_m(z) = 2\boldsymbol{I}$ show from (4.48) that $\boldsymbol{H}_p^T(z^{-1})\boldsymbol{H}_p(z) = \boldsymbol{I}$.
3. Find a new example of matrices $\boldsymbol{H}_m(z)$ and $\boldsymbol{H}_p(z)$ for Problem 2.
4. Write down the equations for $M = 4$ that are analogous to (4.46)–(4.48).

Problems 5–11 develop an important example of biorthogonal filters.

5. The upper channel has responses $H_0(z) = 1$ and $F_0(z) = \frac{1}{2} + z^{-1} + \frac{1}{2}z^{-2} = \frac{1}{2}(1 + z^{-1})^2$. Follow the signals $x(n) = (-1)^n$ and $\delta(n)$ through this channel by plotting $(\uparrow 2)(\downarrow 2)H_0x(n)$.





Plot the output $w_0(n) = F_0(\uparrow 2)(\downarrow 2)H_0x(n)$ on top of these inputs. Describe the output from this channel with any input $x(n)$: The odd $w_0(2k+1)$ are ____ and the even $w_0(2k)$ are ____.

6. The lower channel has responses $H_1(z) = F_0(-z) = \frac{1}{2} - z^{-1} + \frac{1}{2}z^{-2} = \frac{1}{2}(1 - z^{-1})^2$ and $F_1(z) = -H_0(-z) = -1$. Follow the same two inputs $(-1)^n$ and $\delta(n)$ through this channel by plotting $H_1x(n)$ and $w_1(n) = F_1(\uparrow 2)(\downarrow 2)H_1x(n)$. Verify that the sum of outputs $w_0(n)+w_1(n)$ is $\widehat{x}(n) = x(n-1)$. In words, the output $w_1(n)$ from the lower channel is ____ for $n = 2k+1$ and ____ for $n = 2k$ for any input $x(n)$.

7. Verify the perfect reconstruction condition on $F_0(z)H_0(z) + F_1(z)H_1(z)$ for this filter bank. What is the delay ℓ? What is the product filter $P_0(z)$? Find the centered product filter $P(z)$, which is halfband.

8. Find the modulation matrix $H_m(z)$ for the analysis bank. Find the synthesis matrix $F_m(z)$, remembering the transpose. Compute the product $F_m(z)H_m(z)$.

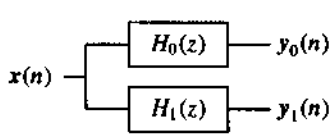
9. Reverse the filters in the upper channel by $H_0(z) = \frac{1}{2} + z^{-1} + \frac{1}{2}z^{-2}$ and $F_0(z) = 1$. Follow the same signals $x(n) = (-1)^n$ and $x(n) = \delta(n)$ through this new channel.

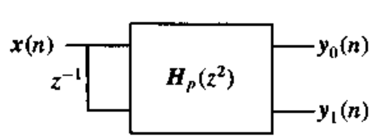
10. Construct the corresponding $H_1(z) = F_0(-z)$ and $F_1(z) = -H_0(-z)$ and follow the two inputs through the lower channel. Verify that the outputs $w_0(n)+w_1(n)$ reconstruct the impulse $x(n)$. Certainly $F_0(z)H_0(z) + F_1(z)H_1(z)$ is the same as before (still PR). The halfband filter $P(z) = \frac{1}{2}z + 1 + \frac{1}{2}z^{-1}$ was factored into $P(z) \bullet 1$ and then $1 \bullet P(z)$. Which filter bank factors $P(z)$ into $(1+z^{-1})/\sqrt{2}$ times $(1+z)/\sqrt{2}$?

    Big question: Is the more regular filter $\frac{1}{2} + z^{-1} + \frac{1}{2}z^{-2}$ better in analysis (H_0) or in synthesis (F_0)? The output w_0 should be a "good but compressed" copy of x.

11. Show that the linear signal $x(n) = n$ goes entirely through the lowpass channel. The highpass channel has $H_1x(n) =$ ____. The constant signal $x(n) = 1$ also goes through, so this filter bank (still from Problem 5) has accuracy $p = 2$.

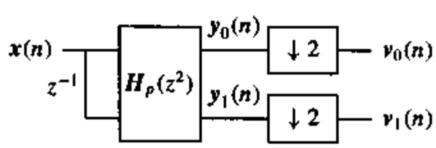
12. Explain the equivalence of these representations before downsampling:

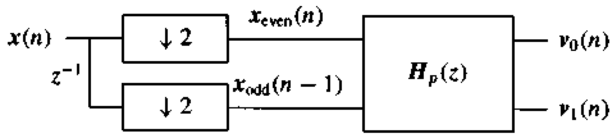




In matrix notation this is $\begin{bmatrix} H_0(z) \\ H_1(z) \end{bmatrix} = \begin{bmatrix} H_{00}(z^2) & H_{01}(z^2) \\ H_{10}(z^2) & H_{11}(z^2) \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix}$.

13. Explain the equivalence of these representations including downsampling:





Warning for downsampling by 3. The polyphase components when $M = 3$ are C_0, C_1, C_2 and X_0, X_1, X_2. We want the component Y_0 of the product, because Y_1 and Y_2 are lost in downsampling. *That component Y_0 does not involve C_1 times X_1.* Multiplying z^{-1} by z^{-1} does not give z^{-3}. The exponents $0, 3, 6, \ldots$ come from C_2 times X_1 and C_1 times X_2 (and also C_0 times X_0). To get z^{-3} in the product, we multiply z^{-1} by z^{-2} or z^0 by z^{-3}:

$$Y_0(z^3) = C_0(z^3)X_0(z^3) + z^{-3}C_1(z^3)X_2(z^3) + z^{-3}C_2(z^3)X_1(z^3).$$

Then replace z^3 by z to find the transform of $(\downarrow 3)y = (\downarrow 3)C\boldsymbol{x}$.

We save this pattern for a later discussion of $(\downarrow M)$. It uses "Type 1 polyphase" components C_0, C_1, C_2. By defining Type 2 components R_0, R_1, R_2 we could achieve that R_i multiplies X_i. (The R's are a permutation of the C's.) Our immediate interest is restricted to $(\downarrow 2)$.

14. Write down the formula for $Y_0(z^3)$ that corresponds to equation (4.46). The polyphase components of C are C_0, C_1, C_2.

15. Show that $X_{\text{even}}(z)$ is the z-transform of the downsampled vector $(\downarrow 2)\boldsymbol{x}$. What vector is transformed to $X_{\text{odd}}(z)$?

16. Write $\boldsymbol{H}_p(z)$ in terms of $H_0(z)$ for these filters of length $N + 1$.

(a) $H_1(z) = H_0(-z)$ (b) $H_1(z) = z^{-N}H_0(z^{-1})$ (c) $H_1(z) = z^{-N}H_0(-z^{-1})$.

17. Find the analysis filters $H_0(z)$ and $H_1(z)$ for the following polyphase matrices:

a. $\boldsymbol{H}_p(z) = \begin{bmatrix} 1+2z^{-1}-z^{-2} & 2-z^{-1} \\ z^{-3} & 1+2z^{-1}+z^{-2} \end{bmatrix}$

b. $\boldsymbol{H}_p(z) = \begin{bmatrix} 1+2z^{-2} & 1+z^{-1} \\ 1-z^{-1} & 2+z^{-1} \end{bmatrix} \begin{bmatrix} 2 & 1+z^{-2} \\ 1-z^{-1} & -z^{-3} \end{bmatrix}$

c. $\boldsymbol{H}_p(z) = \begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix}$

d. $\boldsymbol{H}_p(z) = \begin{bmatrix} \frac{1}{2+z^{-1}} & 1 \\ z^{-3} & \frac{1+z^{-1}}{3-z^{-1}} \end{bmatrix}$

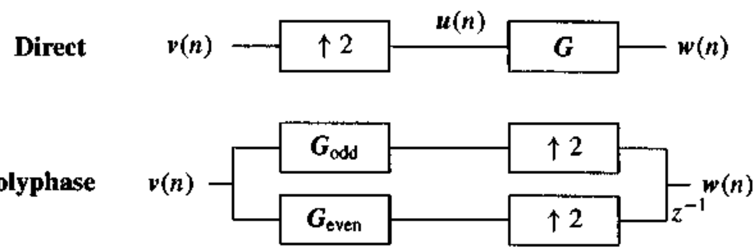
e. $\boldsymbol{H}_p(z) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \frac{1+2z^{-1}}{2+z^{-1}} & 0 \\ 0 & \frac{1-2z^{-1}}{2-z^{-1}} \end{bmatrix}$

18. For each system (a-e) in Problem 17, find the modulation matrix $\boldsymbol{H}_m(z)$.

4.4 Polyphase for Upsampling and Reconstruction

The reader can anticipate what is coming. The polyphase form above was for the analysis bank, with downsampling. The same ideas apply to the synthesis bank, with upsampling. The expanders $(\uparrow 2)$ will again move outside the filters in the polyphase implementation. This time "outside" means *after* the synthesis filters.

We start with one filter called $\boldsymbol{G}$, *and move the upsampler*:



The polyphase form is more efficient, because the half-length filters $G_0 = G_{\text{even}}$ and $G_1 = G_{\text{odd}}$ receive input at the *slow rate* (low bandwidth). *The direct form unnecessarily doubles the rate by* $(\uparrow 2)$ before the filter.

It is always important to verify formulas in the z-domain. In the direct form, upsampling produces $U(z) = V(z^2)$. The only powers to enter $U(z)$ are *even* powers, because upsampling puts zeros in the odd components. Then the filter produces $W(z) = G(z)V(z^2)$. In this multiplication, even powers in G yield even powers in W. The odd phase of G multiplies $V(z^2)$ to give the odd phase of W. That is why polyphase works in the synthesis filters. Even times even is separated from odd times even:

$$W(z) = [1 \quad z^{-1}]\begin{bmatrix} W_{\text{even}}(z) \\ W_{\text{odd}}(z) \end{bmatrix} = [1 \quad z^{-1}]\begin{bmatrix} G_{\text{even}}(z^2) \\ G_{\text{odd}}(z^2) \end{bmatrix} V(z^2).$$

Suppose the filter G is a simple delay. Then its Type 1 polyphase components are $G_{\text{even}} = 0$ and $G_{\text{odd}} = \text{identity}$. The output from this polyphase equation is correct—a delay of $\uparrow 2v$.

Example 4.7. Suppose G has four coefficients $g(0), g(1), g(2), g(3)$. In the time domain, G is a constant-diagonal matrix. When upsampling acts before the filter to produce $G(\uparrow 2)$, it *removes columns* of G. (Just as downsampling removed rows of C.) The time-domain matrix in direct and inefficient form is $G(\uparrow 2)$:

$$G\,(\uparrow 2) = \begin{bmatrix} g(0) & & & & \\ g(1) & g(0) & & & \\ g(2) & g(1) & g(0) & & \\ g(3) & g(2) & g(1) & g(0) & \\ & \cdot & \cdot & \cdot & \cdot \end{bmatrix}\begin{bmatrix} 1 & & \\ 0 & & \\ & 1 & \\ & 0 & \\ & & \cdot \end{bmatrix} = \begin{bmatrix} g(0) & & \\ g(1) & & \\ g(2) & g(0) & \\ g(3) & g(1) & \\ & \cdot & \cdot \end{bmatrix} \qquad (4.49)$$

Now comes the polyphase idea. Each row is either all even or all odd. The even rows (*not columns as for C*) combine into G_{even}. The odd rows combine into G_{odd}. Each of those is a constant-diagonal matrix (a time-invariant filter!). When they act separately and simultaneously, the process is more efficient:

$$(\uparrow 2)\begin{bmatrix} G_{\text{even}} \\ G_{\text{odd}} \end{bmatrix} = \begin{bmatrix} g(0) & & \\ g(2) & g(0) & \\ & \cdot & \cdot \\ g(1) & & \\ g(3) & g(1) & \\ & \cdot & \cdot \end{bmatrix}$$

In this polyphase form, upsampling comes *after* the two filters. The odd phase is delayed before the even and odd outputs are combined:

$$G(\uparrow 2) \text{ becomes } [1 \quad \text{delay}](\uparrow 2)\begin{bmatrix} G_{\text{even}} \\ G_{\text{odd}} \end{bmatrix}. \tag{4.50}$$

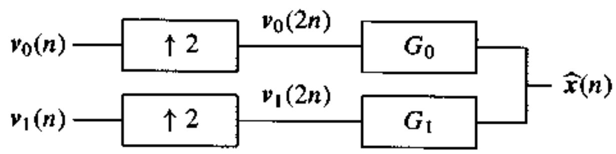
This is the polyphase form in the time domain. Its transform is the polyphase form in the z-domain. We are using the second Noble Identity *on each phase separately*:

$$\textbf{direct} \quad (\uparrow 2)G_{\text{even}}(z^2) \;=\; G_{\text{even}}(z)(\uparrow 2) \quad \textbf{polyphase}$$

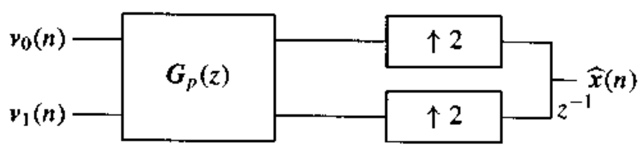
Synthesis Bank: Direct and Polyphase

Now put *two filters* into a synthesis filter bank. The filters will be G_0 and G_1. Remember that synthesis has two input vectors (at half rate) and one output vector (at full rate). The direct form and the polyphase form are (with our present conventions) as follows:

Direct



Polyphase



This is a 2-channel synthesis bank. In the z-domain, the *polyphase matrix* $\boldsymbol{G}_p(z)$ multiplies the inputs $V_0(z)$ and $V_1(z)$. The indices 0 and 1 are now channel numbers. The extension of equation (4.41) to two signals coming through two filters G_0 and G_1 will produce $\boldsymbol{G}_p(z)$:

$$\widehat{X}(z) = [1 \quad z^{-1}]\begin{bmatrix} G_{0,\,\text{even}}(z^2) & G_{1,\,\text{even}}(z^2) \\ G_{0,\,\text{odd}}(z^2) & G_{1,\,\text{odd}}(z^2) \end{bmatrix}\begin{bmatrix} V_0(z^2) \\ V_1(z^2) \end{bmatrix} \tag{4.51}$$

Notice! In the analysis polyphase matrix $\boldsymbol{H}_p(z)$, the lower left entry was H_{10}. Now G_{01} is at the lower left. Where we had the even part of $\boldsymbol{H}_1$, we have the odd part of $\boldsymbol{G}_0$. The channel index always comes before the phase index, and this forces a transpose in the polyphase synthesis matrix. Other authors agree with this convention — a necessary evil.

In the M by M case, H_{ij} is the jth polyphase component of the filter in the ith channel. So is G_{ij}. But the entry in row i, column j of the synthesis matrix is G_{ji}.

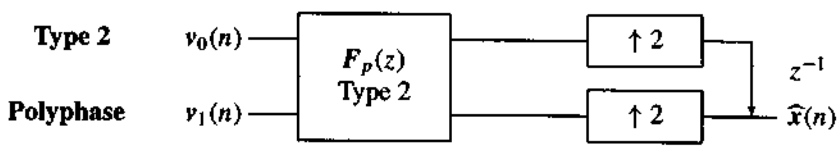
Useful convention for synthesis: Type 2 polyphase. Vaidyanathan delays channel 0 instead of channel 1, at the end of the synthesis bank. Therefore he reverses the two polyphase components. This produces the Type 2 polyphase decomposition, where 1 *means even and* 0 *means odd.* We write F rather than G to keep this type separate:

$$\textbf{Type 2 is} \quad F(z) = F_1(z^2) + z^{-1}F_0(z^2). \tag{4.52}$$

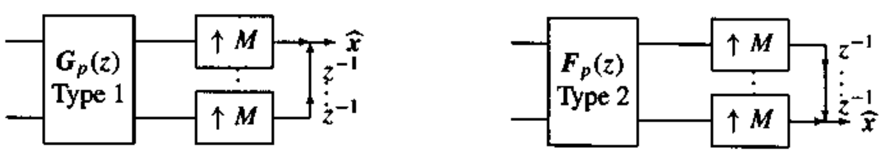
In the Type 2 polyphase matrix for two filters, the delay z^{-1} goes with 0:

$$\widehat{X}(z) = [z^{-1} \quad 1] \begin{bmatrix} F_{0,0}(z^2) & F_{1,0}(z^2) \\ F_{0,1}(z^2) & F_{1,1}(z^2) \end{bmatrix} \begin{bmatrix} V_0(z^2) \\ V_1(z^2) \end{bmatrix}. \tag{4.53}$$

We just reversed $[1 \quad z^{-1}]$ into $[z^{-1} \quad 1]$, and *reversed the rows of the matrix*. The product is exactly the same. The only difference in the block form is the new position of the delay, now in the upper channel:



We will follow this arrangement: Type 2 for synthesis and Type 1 for analysis. The main point is that upsampling follows filtering. The phases of the filters are shorter than the filters themselves. Those subfilters operate simultaneously to produce separate phases of the outputs. Then upsampling with delay assembles $\widehat{x}$. We draw an M-channel synthesis bank in both Type 1 and Type 2 polyphase form:



The M polyphase components of a single filter $\boldsymbol{H}$ are

$$\textbf{Type 1:} \quad H(z) = G_0(z^M) + z^{-1}G_1(z^M) + \cdots + z^{-(M-1)}G_{M-1}(z^M)$$

$$\textbf{Type 2:} \quad H(z) = F_{M-1}(z^M) + z^{-1}F_{M-2}(z^M) + \cdots + z^{-(M-1)}F_0(z^M)$$

When the bank has M filters, each with M phases, we have an M by M matrix. This is the polyphase matrix containing M^2 time-invariant filters.

Perfect Reconstruction

One reason for introducing the polyphase form is efficiency. The analysis bank and synthesis bank are faster, when $(\downarrow 2)$ and $(\uparrow 2)$ are moved outside. The even and odd subfilters that appear inside are about 50% shorter. So they can be executed more quickly.

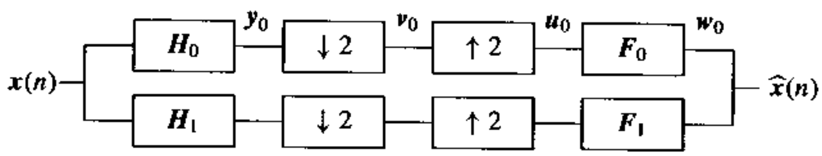
The other reason for polyphase is to simplify the theory. Underlying the whole book is the goal of perfect reconstruction. We have to *connect* the synthesis bank to the analysis bank. One must be the inverse of the other, if the signal $\widehat{x}$ is to agree with the input x. When pure filters connect to pure filters, the products are pure filters — and the z-transforms tell all.

Since the analysis bank is represented by a matrix $H_p(z)$, and the synthesis bank is represented by $F_p(z)$, we hope very much that *inverse filter banks* are associated with *inverse matrices*:

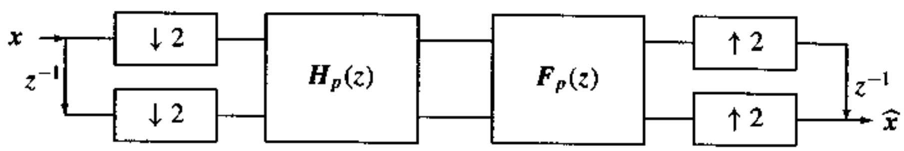
Perfect reconstruction should mean that $F_p(z) = H_p^{-1}(z)$.

We note that delays are allowed and expected. The output signal may be $\widehat{x}(n) = x(n-l)$. Then the system delay is l. In the z-domain this is $F_pH_p = z^{-(\ell-1)/2}I$.

The direct connection of analysis bank to synthesis bank has the decimators and expanders inside. This produces the standard QMF bank:



But the standard order is inefficient. The polyphase order is much better, with ($\downarrow$ 2) and ($\uparrow$ 2) moved to the outside. We draw the polyphase form with Type 2 leading to $F_p(z)$. The final delay is moved from channel 2 in the Type 1 form (row 2 of G_p) to channel 1 in the Type 2 form (row 1 of F_p):



Polyphase is simpler and better because H_p and F_p are now side by side. Those are matrices coming from pure time-invariant filters — the even and odd phases of two analysis filters and two synthesis filters. In between would come compression or transmission. With nothing happening between, F_pH_p disappears into I.

Example 4.8. The simplest analysis bank has $H_0(z) = 1$ and $H_1(z) = z^{-1}$. Their polyphase components are $H_{00} = 1$ and $H_{01} = 0$, then $H_{10} = 0$ and $H_{11} = 1$. The polyphase matrix is $H_p = I$.

The corresponding synthesis bank has $G_0(z) = z^{-1}$ and $G_1(z) = 1$ in the two channels. The Type 1 polyphase components of z^{-1} are $G_{00} = 0$ and $G_{01} = 1$, in the first column (not row!) of G_p. Then $G_{10} = 1$ and $G_{11} = 0$ go into the second column. These enter the synthesis matrix G_p, and in this example the transposing has no effect:

$$G_p = \begin{bmatrix} G_{00} & G_{10} \\ G_{01} & G_{11} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{4.54}$$

When we account for the delays in the lower channel, the final output is $\widehat{x}(n) = x(n-1)$. The QMF bank is a simple delay chain, with perfect reconstruction.

Now use Type 2 for synthesis. This reverses the rows of G_p to give F_p. Thus $F_p = I$. Starting from the two synthesis filters, z^{-1} has Type 2 components $F_{00} = 1$ and $F_{01} = 0$. The second channel yields $F_{10} = 0$ and $F_{11} = 1$ in the second column (not row!) of F_p:

$$F_p = \begin{bmatrix} F_{00} & F_{10} \\ F_{01} & F_{11} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.55}$$

You see the advantage of the second form, which is the Type 2 *form* $F_p(z)$. The perfect reconstruction from this delay chain is reflected in

$$F_p(z) H_p(z) = I. \tag{4.56}$$

That is the important equation in this section. The example itself is a terrible QMF bank. It has delays but no genuine filters. However the conclusion remains correct when $H_p(z)$ and $F_p(z)$ are polynomials in z^{-1}, from FIR filters. That is the whole point—that useful matrices can be inverses of each other and *both can be polynomials.*

The direct and polyphase forms of a QMF bank are externally equivalent. The observer of x and $\widehat{x}$ does not notice a difference. But the efficiency is improved and the theory is simplified. The theory of perfect reconstruction is a perfect matrix equation:

Theorem 4.5 *QMF banks give perfect reconstruction when F_p and H_p are inverses:*

$$F_p(z) H_p(z) = I \quad or \quad z^{-L} I.$$

$H_p(z)$ is Type 1, for analysis, and $F_p(z)$ is Type 2 transposed, for synthesis.

Example 4.9. In the average-difference filter bank all filters have *two taps*. Their even and odd phases have only one tap. Therefore the polyphase matrices are constant. But they are not diagonal, as they were in the simple delay chain. The analysis polyphase matrix is

$$H_p(z) = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

The synthesis matrices are transposed as always. F_p has "even" in the lower row:

$$G_p(z) = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad \text{and} \quad F_p(z) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad F_p H_p = I.$$

Problem Set 4.4

1. Express $X(z) = X_0(z^2) + z^{-1} X_1(z^2)$ as a sum $x = (\uparrow 2)x_0 + (\_\_)$ in the time domain. What are the coefficients $x_0(n)$ and $x_1(n)$ in the polyphase decomposition (Type 1) in terms of the original $x(n)$?

The next three exercises are the synthesis bank equivalents of the time-domain analysis matrices H_d, H_p, H_b in Section 4.3.

2. With filter coefficients $c(0), \ldots, c(3)$ in G_0 and $d(0), \ldots, d(3)$ in G_1, write down the infinite time-domain matrix G_d for the synthesis bank. The analysis bank had the lowpass $(\downarrow 2)C$ above the highpass $(\downarrow 2)D$. The synthesis bank will have $G_0(\uparrow 2)$ and $G_1(\uparrow 2)$ side by side. Each of these parts looks like equation (4.49).

3. Rearrange the rows of the direct matrix G_d in the previous exercise to give the polyphase matrix G_p in the time domain. This should be a 2 by 2 matrix with time-invariant filters C_{even}, C_{odd}, D_{even}, D_{odd} as the blocks ($C = G_0$ and $D = G_1$).

4. Rearrange the columns of the direct matrix G_d to produce an infinite matrix G_b of 2 by 2 blocks.

5. Draw a delay chain with M channels. Then $H_0 = 1, H_1 = z^{-1}, \ldots, H_{M-1} = z^{-(M-1)}$ leads to what matrix $\boldsymbol{H}_p(z)$? Choose the synthesis delays to reconstruct $\widehat{x}(n) = x(n-(M-1))$. Create the Type 1 matrix $\boldsymbol{G}_p(z)$ and the Type 2 matrix $\boldsymbol{F}_p(z)$. Check $\boldsymbol{F}_p\boldsymbol{H}_p = \boldsymbol{I}$.

6. Establish the relation between $\boldsymbol{G}_m(z)$ (modulation) and $\boldsymbol{G}_p(z)$ (polyphase Type 1).

7. If $\boldsymbol{G}_m(z)\boldsymbol{H}_m(z) = \begin{bmatrix} 2z^{-l} & 0 \\ 0 & -2z^{-l} \end{bmatrix}$ show that $\boldsymbol{G}_p(z)\boldsymbol{H}_p(z) = \begin{bmatrix} 0 & z^{-L} \\ z^{-L} & 0 \end{bmatrix}$, with $l = 2L+1$. The matrices $\boldsymbol{G}_m$ and $\boldsymbol{G}_p$ include a transpose. Then $\boldsymbol{F}_p$ includes a row exchange for Type 2, and $\boldsymbol{F}_p(z)\boldsymbol{H}_p(z) = z^{-L}\boldsymbol{I}$.

8. Find $F_0(z)$ and $F_1(z)$ for synthesis from the analysis polyphase matrix:

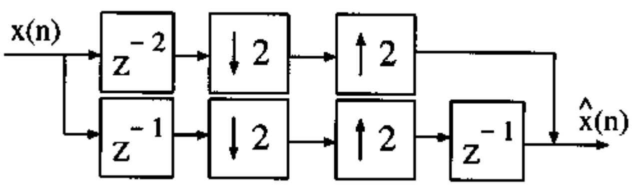
a. $\boldsymbol{H}_p(z) = \begin{bmatrix} 2-4z^{-1} & 1-z^{-1} \\ 3+z^{-1}+2z^{-2} & 1 \end{bmatrix}$

b. $\boldsymbol{H}_p(z) = \begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix}\begin{bmatrix} z^{-1} & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix}$

c. $\boldsymbol{H}_p(z) = \begin{bmatrix} \frac{1+z^{-1}}{5-z^{-1}} & 1 \\ \frac{1}{2-z^{-1}} & z^{-3} \end{bmatrix}$

d. $\boldsymbol{H}_p(z) = \begin{bmatrix} \frac{1+2z^{-1}}{2+z^{-1}} & 0 \\ 0 & \frac{1-3z^{-1}}{3-z^{-1}} \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

9. Find the polyphase matrices $\boldsymbol{H}_p(z)$ and $\boldsymbol{F}_p(z)$. Is this system PR?



4.5 Lattice Structure

A filter bank is represented by its polyphase matrix. When we know the filters, we know the matrix $\boldsymbol{H}_p(z)$. In the opposite direction, we can choose a suitable $\boldsymbol{H}_p(z)$—often as a product of simple matrices. Then the corresponding filter bank is a "lattice" of simple filters, easy to implement.

Here is a class of polyphase matrices to use as examples. They are linear in z^{-1}. One factor has the coefficients $1, 1$ and $1, -1$ from the Haar filter bank—averages and differences. Another factor has coefficients a, b, c, d that we are free to choose—as long as the matrix is invertible. Between those factors is a diagonal matrix with a delay in the second channel:

$$\boldsymbol{H}_p(z) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & \\ & z^{-1} \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{4.57}$$

$$\boldsymbol{H}_p^{-1}(z) = \frac{1}{2}\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}\begin{bmatrix} 1 & \\ & z \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{4.58}$$

First point: $\boldsymbol{H}_p^{-1}$ contains an advance (denoted by z). No problem. Good causal polyphase matrices have anticausal inverses, as this one has. The main point is that $\boldsymbol{H}_p^{-1}$ is a polynomial (FIR not IIR). Add one delay to the whole synthesis bank (multiply by z^{-1}) and it becomes causal:

$$z^{-1}H_p^{-1}(z) = \frac{1}{2}\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}\begin{bmatrix} z^{-1} & \\ & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{4.59}$$

When you multiply out the factors in $H_p(z)$, you see the possibilities:

$$H_p(z) = \begin{bmatrix} a + cz^{-1} & b + dz^{-1} \\ a - cz^{-1} & b - dz^{-1} \end{bmatrix}. \tag{4.60}$$

The top row holds the even and odd phases of the lowpass filter C:

$$C(z) = a + bz^{-1} + cz^{-2} + dz^{-3} \quad \text{has phases} \quad a + cz^{-1} \quad \text{and} \quad b + dz^{-1}.$$

The second row is the polyphase form of the highpass filter D:

$$D(z) = a + bz^{-1} - cz^{-2} - dz^{-3} \quad \text{has phases} \quad a - cz^{-1} \quad \text{and} \quad b - dz^{-1}.$$

We design these 4-tap filters by choosing a, b, c, d. They can have *linear phase or orthogonality*. Our 2 by 2 matrix can be symmetric or it can be orthogonal:

Linear phase filters (symmetric-antisymmetric). Choose $a = d$ and $b = c$.

The lowpass filter C has coefficients a, b, b, a. The highpass filter D is antisymmetric, with coefficients $a, b, -b, -a$. For the polyphase matrix, just substitute $d = a$ and $c = b$:

$$\textit{Linear phase} \qquad H_p(z) = \begin{bmatrix} a + bz^{-1} & b + az^{-1} \\ a - bz^{-1} & b - az^{-1} \end{bmatrix}. \tag{4.61}$$

How do you recognize the symmetry of C (or H_0) from its phases in the top row? One phase is the *flip* of the other phase. Then the whole filter a, b, b, a is the flip of itself, which makes it symmetric.

The synthesis filters are also linear phase. The underlying reason is that the inverse of a symmetric matrix is symmetric. Now change to orthogonal.

Orthogonal filter bank Choose $d = a$ and $c = -b$ and normalize.

The second row $[c\ d] = [-b\ a]$ becomes orthogonal to the first row $[a\ \ b]$. Both rows and both columns have length $\sqrt{a^2 + b^2}$. Each of the three factors in $H_p(z)$ is a unitary matrix, after dividing by that length. The inverses come directly from the conjugate transposes:

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix}^{-1} = \frac{1}{a^2 + b^2}\begin{bmatrix} a & -b \\ b & a \end{bmatrix}$$

$$\begin{bmatrix} 1 & \\ & z^{-1} \end{bmatrix}^{-1} = \begin{bmatrix} 1 & \\ & \overline{z^{-1}} \end{bmatrix}^T \qquad \text{if } |z| = 1.$$

We use the word *unitary* rather than *orthogonal* because z is complex. The inverse of a unitary matrix U is $\overline{U}^T$. The first matrix is real, so conjugating had no effect. The second is diagonal, so transposing had no effect. We always do both.

Notice that $|z| = 1$. The inverse of $z = e^{-i\omega}$ equals the conjugate: $z^{-1} = e^{i\omega} = \overline{z}$. Section 5.1 will introduce the word *paraunitary* for $H_p(z)$, when it is a unitary matrix on the circle $|z| = 1$. Our present example is paraunitary.

Multiplying the three matrix factors gives $H_p(z)$, with $d = a$ and $c = -b$:

$$\textit{Paraunitary polyphase matrix} \quad H_p(z) = \tfrac{1}{2}\begin{bmatrix} a - bz^{-1} & b + az^{-1} \\ a + bz^{-1} & b - az^{-1} \end{bmatrix}. \tag{4.62}$$

The impulse response shows the difference between linear phase (earlier) and orthogonal (now). The lowpass $C(z)$ comes from the first row, where a and $-b$ give the even coefficients. The coefficients in $b + az^{-1}$ go with the odd powers z^{-1} and z^{-3}:

$$C(z) = a + bz^{-1} - bz^{-2} + az^{-3}.$$

The highpass response from the second row is

$$D(z) = a + bz^{-1} + bz^{-2} - az^{-3}.$$

We lost symmetry and gained orthogonality. One part of orthogonality is that $(a, b, -b, a)$ is perpendicular to $(a, b, b, -a)$. But this is not all. Another part is that $(a, b, -b, a, 0, 0)$ is perpendicular to $(0, 0, a, b, b, -a)$. We have to consider those *double shifts*, because they enter the time-domain matrix — two filters C and D with downsampling:

$$H = \begin{bmatrix} (\downarrow 2)C \\ (\downarrow 2)D \end{bmatrix} = \frac{1}{2}\begin{bmatrix} a & -b & b & a & & \\ & & a & -b & b & a \\ & & & \cdot & \cdot & \cdot \\ -a & b & b & a & & \\ & & -a & b & b & a \\ & & & \cdot & \cdot & \cdot \end{bmatrix}. \tag{4.63}$$

Linear phase and also orthogonal: ***Not possible for length greater than*** 2.

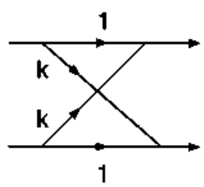
Later we prove this fact in general. Linear phase FIR pairs that are also orthogonal can have at most two nonzero coefficients in each filter. They are just variations on the average-difference pair. This is a one-step improvement over single filters, which can have only *one* nonzero coefficient. An allpass FIR filter is a delay, with one term $C(z) = cz^{-k}$.

Lattice Structures

The orthogonal bank will now be more general than this example. It will have more coefficients. But $H_p(z)$ can still be produced from simple factors. They will be *cascades* of constant matrices and diagonal matrices. The filter bank has a highly efficient *lattice structure*. We can see already the form of our a, b, c, d factor in the two important cases of linear phase and orthogonality:

Linear Phase

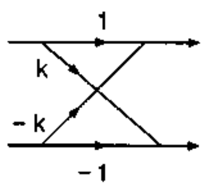
a=d, b=c



$$\begin{bmatrix} a & b \\ b & a \end{bmatrix} = a \begin{bmatrix} 1 & k \\ k & 1 \end{bmatrix} \quad \text{with } k = \frac{b}{a}$$

Orthogonal

a=d, b=−c



$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix} = a \begin{bmatrix} 1 & k \\ -k & 1 \end{bmatrix} \quad \text{with } k = \frac{b}{a}$$

Each lattice involves only *one multiplication* when implemented properly (Problem 7). There is also a single overall multiplying factor, collected from all factors in the cascade. The numbers k can be design parameters. For any k's, a cascade of linear phase filters is linear phase. A cascade of orthogonal filters (including $-k$) is orthogonal. We now study a general lattice built from simple orthogonal factors.

Note. The lattice structure gives long orthogonal filters very easily. They can be good filters. But the k's enter $\boldsymbol{H}_p(z)$ in a complicated nonlinear way. For the ***design*** step, where filter characteristics are optimized, most engineers choose simpler parameters like the coefficients $\boldsymbol{h}(n)$.

For the ***implementation*** step, the lattice has an important advantage. Often the coefficients $\boldsymbol{h}(n)$ will be "quantized"—real numbers are replaced by binary numbers (finitely many digits). In general, this destroys orthogonality. But orthogonality is not lost when implemented as a lattice of simple filters. Each orthogonal factor is determined by a real number k (or by an angle θ). When k or θ is quantized, the factor is not exactly correct—*but it is still orthogonal.* Therefore the whole filter bank remains strictly orthogonal.

We now introduce a product of ***rotation matrices*** and ***delay matrices***. The rotation matrices are constant, exactly as in our examples:

$$\boldsymbol{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = \cos\theta \begin{bmatrix} 1 & k \\ -k & 1 \end{bmatrix}. \tag{4.64}$$

This matrix gives clockwise rotation through the angle θ. When $\cos\theta$ is factored out, it leaves the number $k = \tan\theta$. We will think of the rotation matrix (orthogonal matrix) in terms of θ, but we implement it with k to save multiplications. The delay matrices are used to delay the second channel:

$$\Lambda(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \quad \text{has determinant } z^{-1}.$$

It is convenient to have an extra factor of -1 in the lower channel. The matrix $\operatorname{diag}(1, -1)$, which is also $\Lambda(-1)$, accomplishes this. Then the product of $\ell+1$ rotations separated by ℓ delays is the polyphase matrix for the whole lattice:

$$\boldsymbol{H}_p(z) = \Lambda(-1)\boldsymbol{R}_\ell\,\Lambda(z)\boldsymbol{R}_{\ell-1}\,\Lambda(z)\cdots\boldsymbol{R}_1\,\Lambda(z)\boldsymbol{R}_0. \tag{4.65}$$

The rotation R_ℓ is through an angle θ_ℓ. The rotation R_0 is through an angle θ_0. The determinant is $-z^{-\ell}$, because of the delays and the sign change from diag$(1, -1)$. Without the delays in between, the product would be a single rotation through the total angle $\sum \theta_i$. With the ℓ delays, we have something much more important. It is essentially the most general two-channel orthonormal filter bank with l delays.

$H_p(z)$ is the polyphase matrix of an orthonormal filter bank, because each factor is unitary. We show this analysis bank in its efficient form, with the downsampling operators before the filters. R_0 comes first in the structure because it is the right-hand factor in $H_p(z)$.

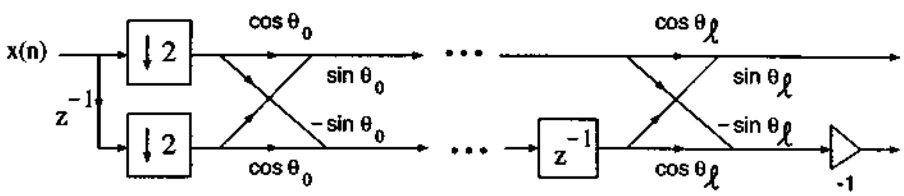


Figure 4.4: The lattice structure for $H_p(z) = \Lambda(-1) R_\ell \Lambda(z) \cdots R_1 \Lambda(z) R_0$.

The factors $\cos\theta$ that are removed give a multiplication by $\cos\theta_\ell \cdots \cos\theta_0$ at the end. Note that Haar's filter bank comes from ***one*** rotation with angle $\frac{\pi}{4}$:

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix} \quad \text{with} \quad \theta = \frac{\pi}{4}.$$

Actually this is a picture of any polyphase matrix $H_p(z)$ at the particular value $z = 1$. The matrix $H_p(1)$ corresponds to zero frequency, because $e^0 = 1$. *The response to direct current is always the Haar matrix, when the lowpass filter has a zero at $\omega = \pi$.*

Theorem 4.6 *If $H_0 = 0$ at $\omega = \pi$ (which is $z = -1$), then the polyphase matrix has*

$$H_p(1) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

For an orthonormal filter bank the angles in the lattice structure add to $\frac{\pi}{4}$.

Proof: H_0 has a zero at $\omega = \pi$ when the sum of odd-numbered coefficients equals the sum of even-numbered coefficients: $h(0) - h(1) + h(2) - \ldots = 0$. This *first sum rule* means that at $z = 1$, the even phase equals the odd phase:

$$H_{00}(1) = h(0) + h(2) + \cdots = h(1) + h(3) + \cdots = H_{01}(1). \tag{4.66}$$

Row 1 of the polyphase matrix has equal entries at $z = 1$. Row 2 comes from the highpass filter H_1. This has the opposite property $H_1 = 0$ at $\omega = 0$. The odd sum is *minus* the even sum. With the right signs, $H_p(1)$ is the Haar matrix in the Theorem.

Since this is exactly what we get in the product of rotations, when $z = 1$ and the delay matrices become I, the total rotation angle must be $\sum \theta_i = \frac{\pi}{4}$.

The 4-tap Daubechies filter in Section 5.2 has angles $\frac{\pi}{3}$ and $-\frac{\pi}{12}$. Those angles add to $\frac{\pi}{4}$. Looking back at the start of this section, we realize that Haar followed by one rotation is actually useless. That second rotation angle must be zero! The sum rule in the top row of the lowpass filter becomes $a + b = -b + a$, which forces $b = 0$. The good angles are $\frac{\pi}{3}$ and $-\frac{\pi}{12}$, not $\frac{\pi}{4}$ and 0.

The Synthesis Lattice

The synthesis bank inverts the analysis bank. To invert the lattice, reverse the order of rotations. The inverse $\Lambda(-1)\boldsymbol{R}_\ell\,\Lambda(z)\,\ldots\,\boldsymbol{R}_1\,\Lambda(z)\,\boldsymbol{R}_0$ is the product of inverses in reverse order:

$$\left(\boldsymbol{H}_p(z)\right)^{-1} = \boldsymbol{R}_0^T\Lambda^{-1}(z)\boldsymbol{R}_1^T\cdots\Lambda^{-1}(z)\boldsymbol{R}_\ell^T\Lambda(-1). \tag{4.67}$$

The inverse of each rotation is the transpose: $\boldsymbol{R}^{-1} = \boldsymbol{R}^T$ = rotation through $-\theta$. These constant matrices are orthogonal. The inverse of a delay matrix is an *advance*:

$$\Lambda^{-1}(z) = \begin{bmatrix} 1 & 0 \\ 0 & z \end{bmatrix} \quad \text{and} \quad z^{-1}\Lambda^{-1}(z) = \begin{bmatrix} z^{-1} & 0 \\ 0 & 1 \end{bmatrix}$$

Since $\boldsymbol{H}_p(z)$ has ℓ delays, its inverse has ℓ advances. Multiplying each advance by z^{-1} puts l delays into the upper channel. The synthesis half has *upsamplers last* (this is the polyphase form in Figure 4.5). An extra advantage is that we can add or remove rotations without losing

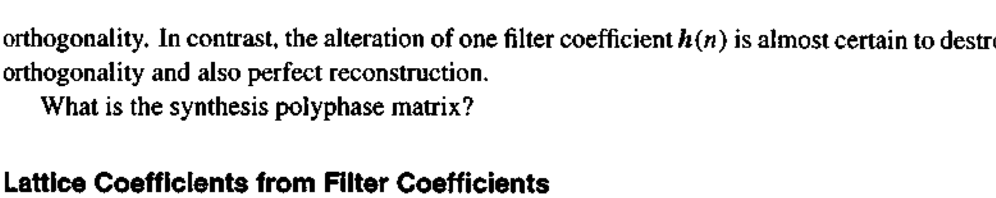


Figure 4.5: Orthogonal synthesis bank in lattice form.

orthogonality. In contrast, the alteration of one filter coefficient $h(n)$ is almost certain to destroy orthogonality and also perfect reconstruction.

What is the synthesis polyphase matrix?

Lattice Coefficients from Filter Coefficients

We now prove that any 2-channel orthonormal filter bank can be expressed in lattice form, with rotations and delays. Thus the lattice structure is "complete". Starting with $\boldsymbol{H}_p(z)$ of degree ℓ, we find a rotation-delay that reduces the degree to $\ell - 1$:

$$\boldsymbol{H}_p(z) = \begin{bmatrix} \cos\theta_\ell & -\sin\theta_\ell \\ \sin\theta_\ell & \cos\theta_\ell \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}\boldsymbol{H}_p^{(\ell-1)}(z). \tag{4.68}$$

The new matrix $\boldsymbol{H}_p^{(\ell-1)}(z)$ is still unitary for $|z| = 1$ and its determinant is $\pm z^{-(\ell-1)}$. So we reduce the degree again. After ℓ steps we reach $\boldsymbol{H}_p(0)$ whose determinant is ± 1. It is unitary for $|z| = 1$ and the only matrices with this property are constants. After properly accounting for the matrix $\Lambda(-1) = \mathrm{diag}(1, -1)$, we have the rotation angle θ_0 that completes the lattice.

Theorem 4.7 *Every lowpass-highpass orthonormal filter bank has a polyphase matrix* $\boldsymbol{H}_p(z)$ *of the lattice form* (4.65).

Proof.. The step (4.68) requires an angle θ_ℓ such that

$$\begin{bmatrix} \cos\theta_\ell & \sin\theta_\ell \\ -\sin\theta_\ell & \cos\theta_\ell \end{bmatrix} \boldsymbol{H}_p(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \boldsymbol{H}_p^{(\ell-1)}(z). \tag{4.69}$$

On both sides, the second row must have ***no constant terms***. Then we can factor z^{-1} from that row and the reduction succeeds — we get the next matrix $\boldsymbol{H}_p^{(\ell-1)}(z)$ and continue. If $\boldsymbol{H}_p(z) = \boldsymbol{h}_p(0) + \cdots + \boldsymbol{h}_p(d)z^{-d}$, the constant term on the left side of the equation comes from $\boldsymbol{h}_p(0)$. The second row on that side must give

$$\begin{bmatrix} -\sin\theta_\ell & \cos\theta_\ell \end{bmatrix} \boldsymbol{h}_p(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}. \tag{4.70}$$

This row vector that produces zero must exist if $\boldsymbol{h}_p(0)$ is singular. The whole argument rests on the fact that $\boldsymbol{h}_p^T(d)\boldsymbol{h}_p(0)$ is the ***zero matrix***. The key equation is $\overline{\boldsymbol{H}}_p^T \boldsymbol{H}_p = \boldsymbol{I}$:

$$\left[\boldsymbol{h}_p^T(0) + \cdots + \boldsymbol{h}_p^T(d)z^d\right]\left[\boldsymbol{h}_p(0) + \cdots + \boldsymbol{h}_p(d)z^{-d}\right] = \boldsymbol{I}. \tag{4.71}$$

The coefficient $\boldsymbol{h}_p^T(d)\,\boldsymbol{h}_p(0)$ of the highest order term must be zero. When two nonzero matrices multiply to give zero, $\boldsymbol{h}_p(0)$ and $\boldsymbol{h}_p(d)$ are both singular. There exists a vector $[-\sin\theta_\ell \;\; \cos\theta_\ell]$ that knocks out the constant term in the second row of (4.69). After ℓ steps we reach a final rotation $\boldsymbol{R}_0$ and the lattice is complete.

Lattices for Linear Phase Filters

For orthonormal filters, $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$ have the same (even) length. One is symmetric and the other is antisymmetric. For linear phase filters, those statements are not necessarily true. The great variety of linear phase PR filter banks means a great variety of lattice structures. The equal-length case copies the orthonormal case, but with factors $\boldsymbol{S} = \begin{bmatrix} a & b \\ b & a \end{bmatrix}$ instead of $\boldsymbol{R} = \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$.

Theorem 4.8 *Every linear phase perfect reconstruction filter bank with equal (even) length filters has a lattice factorization*

$$\boldsymbol{H}_p(z) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \boldsymbol{S}_L \Lambda(z) \boldsymbol{S}_{L-1} \Lambda(z) \cdots \boldsymbol{S}_1 \Lambda(z) \boldsymbol{S}_0. \tag{4.72}$$

The filter bank is a cascade of simple two-tap filters:

$$\Lambda(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix} \quad \textit{and} \quad S_i = \begin{bmatrix} a_i & b_i \\ b_i & a_i \end{bmatrix} = a_i \begin{bmatrix} 1 & k_i \\ k_i & 1 \end{bmatrix}.$$

The proof is entirely parallel to the orthogonal case (Theorem 4.7). The implementation just has a sign change. The number of lattice sections is half the length of $\boldsymbol{H}_0$ (and $\boldsymbol{H}_1$). All factors a_i are collected into a single factor $a = \prod a_i$, for efficiency. And the two multiplications by k_i are further reduced to one multiplication (and three additions) in Problem 7.

What is not parallel is that linear phase allows further possibilities: unequal length and symmetric-symmetric pairs. Here are the rules for PR with linear phase:

Even length must differ by a multiple of 4: $H_0 =$ symm and $H_1 =$ anti.

Odd length must *not* differ by a multiple of 4: $H_0 =$ symm and $H_1 =$ symm.

An example of the second type is in the "Guide to the Book", with lengths 5 and 3: $\frac{1}{8}(-1, 2, 6, 2, -1)$ in H_0 and $\frac{1}{4}(1, -2, 1)$. For symmetric filters whose lengths differ by 2, the (complete) lattice factorization is

$$H_p(z) = D\prod_{k=2}^{L} A_k(z) \text{ with } A_k(z) = \begin{bmatrix} 1+z^{-1} & 1 \\ 1+\beta_k z^{-1}+z^{-2} & 1+z^{-1} \end{bmatrix} \begin{bmatrix} \alpha_k & 0 \\ 0 & 1 \end{bmatrix} \quad (4.73)$$

D is diagonal, and a practical system can have only $\beta_2 \neq 0$. This speeds up the implementation. The FBI 9/7 system is an example that requires only two lattice sections.

The general case has a similar (but longer) factorization. The proof is also longer. Our purpose has been to establish the main point: the efficiency and the availability of the lattice structure.

Perfect Reconstruction with FIR

We are interested above all in FIR systems. Then both H_p and F_p have a finite number of terms, from the finite number of filter coefficients. The crucial observation is that with well-chosen matrices, *the product of two polynomials can be a constant or a monomial*:

$$\begin{aligned} H_p^{-1}(z)H_p(z) &= I \\ F_p(z)H_p(z) &= z^{-L}I \end{aligned} \qquad \text{are possible for matrix polynomials.}$$

In the scalar case, $1 + \beta z^{-1}$ is a polynomial but $1/(1 + \beta z^{-1})$ is not. The reciprocal of a polynomial scalar is not a polynomial. If an ordinary time-invariant filter is FIR, its inverse is IIR. The only scalar exceptions are trivial cases (delays). *The real exceptions are matrix polynomials — which are polyphase matrices of filter banks.* That is what this book is about.

The inverse of a matrix polynomial can be a matrix polynomial. We ask when.

Theorem 4.9 *An FIR analysis bank has an FIR synthesis bank that gives perfect reconstruction if and only if the determinant of $H_p(z)$ is a monomial cz^{-l} with $c \neq 0$.*

Proof. The entries of $(H_p(z))^{-1}$ are always cofactors divided by the determinant:

$$(H_p^{-1})_{ij} = \frac{(j, i) \text{ cofactor of } H_p(z)}{\text{determinant of } H_p(z)}.$$

The cofactor (the determinant without row j and column i) is certainly a polynomial. If the determinant is cz^{-l} (one term only!) then the division leaves a polynomial: the inverse is FIR. When this inverse $H_p^{-1}(z)$ is delayed by z^{-l}, it becomes causal. This is $F_p(z)$. It is the Type 2 polyphase matrix of the FIR synthesis bank.

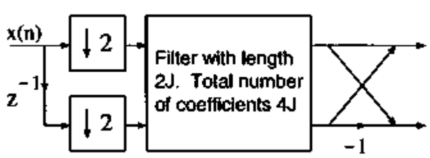
The design problem is to maintain a monomial determinant cz^{-l} while building an efficient filter bank. For an M by M matrix the cofactors of $H_p(z)$ may have high degrees. The analysis and synthesis filters generally have different lengths. The design problem is harder, but more is possible. Cosine-modulated filter banks are winners.

Efficiency of Lattice Structure

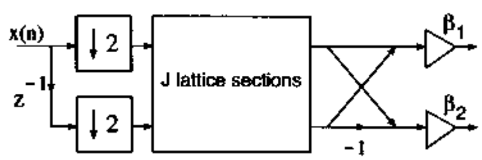
Consider a symmetric filter bank with length $N = 2J$. The number of multiplications per unit time is $4J$ using the direct form and $2J$ using polyphase. Additions are the same, with two extra at the 2-pt DFT matrix.

On the other hand, there are J lattice sections. Each section requires 2 multiplications and 2 additions. Counting β_1 and β_2 in the figure brings the total complexity to $2J$ multiplications and $2J+2$ additions. The effective complexity (at the input rate) is J multiplications and $J+1$ additions.

Problem 7 discusses an alternative that uses 1 multiplier and 3 adders per lattice section. In summary, the lattice complexity is approximately half of the polyphase complexity. The same is true for orthogonal filters.



Polyphase implementation



Lattice Structure implementation

Problem Set 4.5

1. Show that this matrix gives linear phase. Find a, b, c, d and $\boldsymbol{H}_p^{-1}(z)$:

$$\boldsymbol{H}_p(z) = \frac{1}{2}\begin{bmatrix} \cos\Theta + z^{-1}\sin\Theta & -\sin\Theta + z^{-1}\cos\Theta \\ \cos\Theta - z^{-1}\sin\Theta & -\sin\Theta - z^{-1}\cos\Theta \end{bmatrix}.$$

2. Suppose U and V are unitary matrices (constant but possibly complex). This means that $U^{-1} = \overline{U}^T$ and $V^{-1} = \overline{V}^T$. Show why their product UV is also unitary. *Notice the key point*: Inverses come in reverse order $V^{-1}U^{-1}$ and so do transposes.

3. For a 6-tap antisymmetric filter $a + bz^{-1} + cz^{-2} - cz^{-3} - bz^{-4} - az^{-5}$, show that the flip of one phase is minus the other phase. What if the number of taps is odd?

4. Redraw the lattice cascade with the downsampling operators moved to the right, after the butterfly filters. Be sure to change z^{-1} to z^{-2} (why?).

5. Find the product $\boldsymbol{H}_p(z)$ when the rotation angles are $\theta_0 = 0$, $\theta_1 = \frac{\pi}{2}$, and $\theta_2 = 0$. Check the determinant of $\boldsymbol{H}_p(z)$, remembering the $\ell = 2$ delays. This is an example in which $\boldsymbol{H}_p(z)$ contains no terms in $z^{-\ell}$, although the determinant has degree ℓ. In the notation of Theorem 4.7, $\ell = 2$ but $d = 1$.

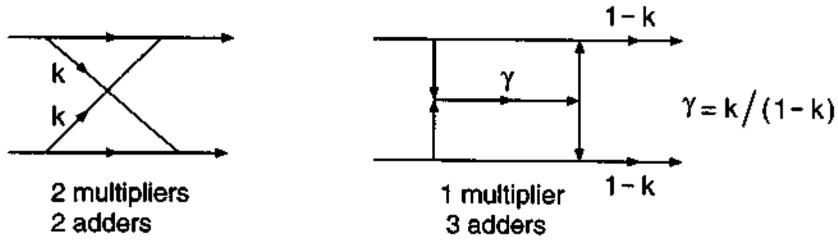
   The correct definition of degree of $\boldsymbol{H}_p(z)$ is the *Smith-McMillan degree = minimum number of delays* to realize the system. For orthonormal filter banks, this equals the degree l of the determinant.

6. Find the polyphase matrix $\boldsymbol{H}_p(z)$ for the pair $H_0(z) = 1$ and $H_1(z) = \frac{1}{2} - z^{-1} + \frac{1}{2}z^{-2}$ in Problem Set 4.3. Find the synthesis polyphase matrix $\boldsymbol{F}_p(z)$ for $F_0(z) = \frac{1}{2} + z^{-1} + \frac{1}{2}z^{-2}$ and $F_1(z) = -1$. Remember to transpose and reverse rows for $\boldsymbol{F}_p(z)$, and compute $\boldsymbol{F}_p(z)\boldsymbol{H}_p(z)$.

7. We can multiply by $\boldsymbol{S} = \begin{bmatrix} a & b \\ b & a \end{bmatrix}$ with one multiplication by $c = \frac{a+b}{a-b}$ (and one by $\frac{a-b}{2}$ that can be collected with others at the end of the cascade):

$$\boldsymbol{S} = \frac{a-b}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} c & \\ & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Write S with 3 adds using the figure. Find a "one multiplication per rotation R".



8. Find the polyphase matrix $H_p(z)$ for the analysis pair $H_0(z) = 1$ and $H_1(z) = \frac{1}{16}(-1 + 9z^{-2} - 16z^{-3} + 9z^{-4} - z^{-6})$. Find $F_p(z)$, transposed and row reversed, for the synthesis pair $F_0(z) = \frac{1}{16}(-1+9z^{-2}+16z^{-3}+9z^{-4}-z^{-6})$ and $F_1(z) = -1$. Verify that $F_p(z)H_p(z) = z^{-1}I$. Note $L = 1$ and $\ell = 3$.

9. What is the synthesis polyphase matrix F_p for the lattice structure in the last figure?

10. The rules for linear phase with PR were stated separately for even and odd lengths. Combine them into one rule for the degrees of $H_0(z)$ and $H_1(z)$: $N_0 + N_1 + 2$ *must be a multiple of* 4.

11. Change a given PR filter bank by choosing $\widehat{H}_0(z) = H_0(z)$ and $\widehat{H}_1(z) = z^{-2}H_1(z)$. What is the relation between $H_p(z)$ and $\widehat{H}_p(z)$? Is this new filter bank PR?

12. Find the analysis filters and the relation between $H_0(z)$ and $H_1(z)$:

$$H_p(z) = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & z^{-1} \end{bmatrix}\begin{bmatrix} 1 & -3 \\ -3 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & z^{-2} \end{bmatrix}\begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix}$$

13. Let $H_p(z) = \begin{bmatrix} z^{-N} & 0 \\ \beta(z) & 1 \end{bmatrix}$, where $\beta(z)$ is a polynomial. Find $H_0(z)$ and $H_1(z)$. Find $F_p(z)$ for a PR system. What are the synthesis filters?

14. Let $H_p(z) = \prod_{k=1}^{L}\begin{bmatrix} z^{-1} & 0 \\ \beta_k(z) & 1 \end{bmatrix}$. Find $F_p(z)$ and all the filters for a PR system.

15. What is the lowpass orthogonal filter with $\theta_1 = \frac{\pi}{3}, \theta_2 = -\frac{\pi}{2}$, and $\theta_3 = \frac{\pi}{4}$?

16. What are the symmetric even-length analysis filters with $k_1 = 2, k_2 = 0, k_3 = -7$, and $k_4 = 5$? What are the PR synthesis filters?

17. If we impose orthogonality on the symmetric factors S_i in (4.72), show that the filters only have two nonzero coefficients. This yields another proof of Theorem 5.3 that symmetry prevents orthogonality.

18. Show that $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & k \\ k & 1 \end{bmatrix}$ is equivalent to $\begin{bmatrix} 1+k & 0 \\ 0 & 1-k \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. This form of the last two blocks in (4.72) reduces the lattice computation time.