CHAPTER 28

Block Filtering

We discussed earlier in Sec. 19.3 block convolution methods of the overlap-add and overlap-save type. These methods are useful when at least one of the sequences that are being convolved has a long duration. The methods are based on segmenting the long sequence into smaller blocks of data and transforming the original convolution problem into the computation of several smaller linear convolutions by means of the DFT. In this chapter, and motivated by the theory we developed in the adjacent chapters on filter banks, we revisit the convolution problem and derive efficient structures for implementing long FIR filters. In this way, by filtering signals through these efficient FIR structures we will effectively be computing the convolution of the input signal with the long impulse response sequence of the FIR filter in an efficient manner. The structures of this chapter will be derived by showing how the FIR filters can be implemented in a manner that processes the input data on a *block-by-block* basis rather than on a sample-by-sample basis. While our focus will be largely on DFT-based structures, we will also comment on other types of transformations such as the discrete cosine transform (DCT) and the discrete Hartley transform (DHT). We will also make connections with the filter bank techniques from Chapter 26.

28.1 BLOCK PROCESSING

To motivate the block processing framework, we start from a causal FIR filter, $H(z)$, of length N, assumed long, say,

$$\boxed{H(z) \;=\; h(0) + h(1)z^{-1} + h(2)z^{-2} + \ldots + h(N-1)z^{-(N-1)}} \qquad (28.1)$$

The causal input sequence to the filter is denoted by $x(n)$ and the resulting causal output sequence is denoted by $y(n)$. Figure 28.1 shows a direct tapped-delay-line implementation of $H(z)$. We shall refer to the transfer function, $H(z)$, as the *fullband* filter. In this implementation, the samples of $y(n)$ are obtained, one sample at a time, through the linear convolution operation

$$\begin{aligned} y(n) &= \sum_{k=0}^{n} h(k)x(n-k) \\ &= h(0)x(n) + h(1)x(n-1) + \ldots + h(N-1)x(n-N+1) \qquad (28.2) \end{aligned}$$

Each time a new sample $x(n)$ arrives at the input of the fullband realization, a new sample $y(n)$ is generated by means of the above input-output relation. We therefore have a realization whose operation is based on the principle of one-sample-in and one-sample-out at each iteration.

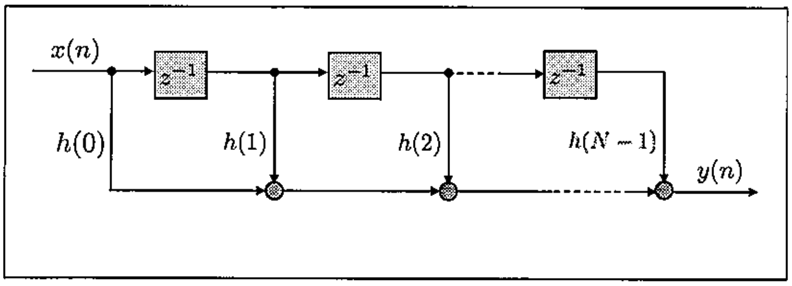

FIGURE 28.1 Tapped-delay-line implementation of the fullband FIR filter, $H(z)$. In this implementation, one output sample $y(n)$ is generated for each input sample $x(n)$ arriving at the input terminal.

We can describe the operation of the fullband filter in the $z-$transform domain as well. Thus, let $Y(z)$ and $X(z)$ denote the $z-$transforms of the causal input and output sequences $y(n)$ and $x(n)$, respectively,

$$Y(z) = \sum_{n=0}^{\infty} y(n)z^{-n} \quad \text{and} \quad X(z) = \sum_{n=0}^{\infty} x(n)z^{-n} \tag{28.3}$$

Due to the assumed causality of the sequences $\{y(n), x(n)\}$, the ROCs of the transforms $Y(z)$ and $X(z)$ are the outside of some circular domains, say, $|z| > r_y$ and $|z| > r_x$, respectively, for some positive scalars $\{r_y, r_x\}$. Then, the input/output filter relation (28.2) translates into

$$\boxed{Y(z) = H(z)X(z)} \tag{28.4}$$

which is depicted in Fig. 28.2. In this implementation, scalar entries $x(n)$ are fed into the filter, $H(z)$, and scalar outputs $y(n)$ are obtained as a result.

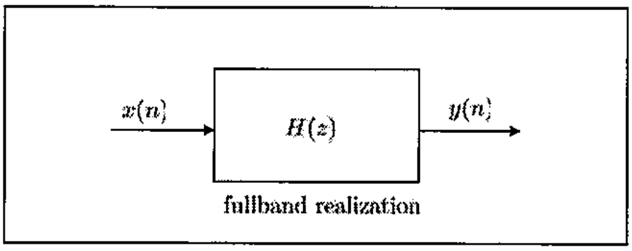

FIGURE 28.2 Fullband filtering by $H(z)$: the input sequence is processed on a sample-by-sample basis.

Transfer Matrix Representation

In this chapter, we shall explain how the same FIR filtering operation can be implemented by operating on a *block* of input data in order to generate a *block* of output data. That is, and in contrast to (28.2), the realizations that will be described in the sequel will operate on data on a block-by-block basis rather than on a sample-by-sample basis. In this way, the block implementation will operate on the principle of one-block-in and one-block-out at each iteration. The block implementation we are seeking can be motivated in different ways. One convenient is to continue to rely on the $z-$transform notation.

Let us introduce blocks of data in the form of column vectors of size M each as follows:

$$x_{M,n} \triangleq \begin{bmatrix} x(nM) \\ x(nM-1) \\ \vdots \\ x((n-1)M+1) \end{bmatrix}, \quad y_{M,n} \triangleq \begin{bmatrix} y(nM) \\ y(nM-1) \\ \vdots \\ y((n-1)M+1) \end{bmatrix} \tag{28.5}$$

where the integer $n = 0, 1, 2, \ldots$ in the vectors $x_{M,n}$ and $y_{M,n}$ is used to denote the *block index.* The data in each block vector, say $x_{M,n}$, include the sample $x(nM)$ and all prior $M - 1$ samples for a total of M samples. For example, assuming the block size is $M = 3$, the block vectors $\{x_{M,n}, y_{M,n}\}$ will result from partitioning the streams of data $\{x(n), y(n)\}$ into successive blocks of size M each:

$$\begin{array}{c} \underbrace{0 \quad 0 \quad x(0)}_{x_{3,0}} \quad \underbrace{x(1) \quad x(2) \quad x(3)}_{x_{3,1}} \quad \underbrace{x(4) \quad x(5) \quad x(6)}_{x_{3,2}} \quad \ldots \\ \underbrace{0 \quad 0 \quad y(0)}_{y_{3,0}} \quad \underbrace{y(1) \quad y(2) \quad y(3)}_{y_{3,1}} \quad \underbrace{y(4) \quad y(5) \quad y(6)}_{y_{3,2}} \quad \ldots \end{array} \tag{28.6}$$

We assume, without loss of generality, that the filter length, N, and the block size, M, are such that the ratio N/M is an integer. Usually, both N and M are powers of 2, e.g., $N = 1024$ and $M = 32$ or some other values. Just like $H(z)$ is the transfer function that maps $X(z)$ to $Y(z)$ in the fullband realization (28.4), we can also determine a similar mapping that maps the $z-$transforms of the block sequences $\{x_{M,n}, y_{M,n}\}$ to each other. But first we need to explain how to define these $z-$transforms.

For that purpose, we note that we can regard the succession of vectors $\{x_{M,n},\ n \geq 0\}$ as samples of a vector-valued sequence (i.e., a discrete-time signal whose sample values happen to be vectors of size M each). The sample at block time $n = 0$ is $x_{M,0}$, the sample at block time $n = 1$ is $x_{M,1}$, and so forth. Accordingly, we define the $z-$transform of the vector-valued sequence $\{x_{M,n}\}$ as follows:

$$\mathcal{X}(z) \triangleq \sum_{n=0}^{\infty} x_{M,n} z^{-n} \qquad (M \times 1) \tag{28.7}$$

In this case, the $z-$transform will be a vector-valued function of z of size $M \times 1$. Note that we are using the calligraphic letter, $\mathcal{X}(z)$, to refer to *vector* functions of z. We can similarly introduce the vector $z-$transform of the block sequence $\{y_{M,n},\ n \geq 0\}$:

$$\mathcal{Y}(z) \triangleq \sum_{n=0}^{\infty} y_{M,n} z^{-n} \qquad (M \times 1) \tag{28.8}$$

We now verify that there is a mapping from $\mathcal{X}(z)$ to $\mathcal{Y}(z)$ similar to the transfer function transformation (28.4) in the fullband case.

Indeed, let $\{E_k(z)\}$ denote the $M-$th order polyphase components of $H(z)$, say,

$$H(z) = \sum_{k=0}^{M-1} z^{-k} E_k(z^M) \tag{28.9}$$

where the $E_k(z)$ are $\frac{N}{M}$−long FIR filters each defined as follows:

$$\left\{\begin{array}{lcl} E_o(z) & = & h(0) + h(M)z^{-1} + h(2M)z^{-2} + \cdots \\ E_1(z) & = & h(1) + h(M+1)z^{-1} + h(2M+1)z^{-2} + \cdots \\ E_2(z) & = & h(2) + h(M+2)z^{-1} + h(2M+2)z^{-2} + \cdots \\ & \vdots & \\ E_{M-1}(z) & = & h(M-1) + h(2M-1)z^{-1} + h(3M-1)z^{-2} + \cdots \end{array}\right. \tag{28.10}$$

That is, the first M coefficients of $H(z)$ are the leading coefficients of the $\{E_k(z)\}$; the next M coefficients of $H(z)$ are the second coefficients of the $\{E_k(z)\}$, and so on. For example, assume $M = 3$ and $N = 12$. Then, $H(z)$ will have 3 polyphase components that are given by

$$\left\{\begin{array}{lcl} E_o(z) & = & h(0) + h(3)z^{-1} + h(6)z^{-2} + h(9)z^{-3} \\ E_1(z) & = & h(1) + h(4)z^{-1} + h(7)z^{-2} + h(10)z^{-3} \\ E_2(z) & = & h(2) + h(5)z^{-1} + h(8)z^{-2} + h(11)z^{-3} \end{array}\right. \tag{28.11}$$

We collect the polyphase components $\{E_k(z)\}$ into an $M \times M$ matrix $\mathcal{H}(z)$ as follows, e.g., for $M = 3$,

$$\mathcal{H}(z) = \left[\begin{array}{ccc} E_o(z) & E_1(z) & E_2(z) \\ z^{-1}E_2(z) & E_o(z) & E_1(z) \\ z^{-1}E_1(z) & z^{-1}E_2(z) & E_o(z) \end{array}\right] \tag{28.12}$$

The first row of $\mathcal{H}(z)$ contains all the polyphase components $\{E_k(z)\}$, while the entries of the subsequent rows are constructed in a manner that results in $\mathcal{H}(z)$ having what is called a pseudo-circulant structure. This structure is defined as follows. First, we say that a matrix has a Toeplitz structure when the entries along each of its diagonals are identical. For example, the following 4×4 matrix is a symmetric Toeplitz matrix

$$T = \left[\begin{array}{cccc} c_o & c_1 & c_2 & c_3 \\ c_1 & c_o & c_1 & c_2 \\ c_2 & c_1 & c_o & c_1 \\ c_3 & c_2 & c_1 & c_o \end{array}\right] \quad \text{(Toeplitz)} \tag{28.13}$$

Observe that the entries along the main diagonal are all equal to c_o, while the entries in the upper diagonal are all equal to c_1, and so on. Now a circulant matrix is a Toeplitz matrix with the additional property that the successive rows of the circulant matrix are obtained by circularly shifting its top row to the right by one position at a time. For example, the following 4×4 matrix is circulant:

$$C = \left[\begin{array}{cccc} c_o & c_1 & c_2 & c_3 \\ c_3 & c_o & c_1 & c_2 \\ c_2 & c_3 & c_o & c_1 \\ c_1 & c_3 & c_2 & c_o \end{array}\right] \quad \text{(circulant matrix)} \tag{28.14}$$

Observe that if we shift the top row circularly to the right, then its rightmost entry, c_3, will appear as the leading entry in the second row. Likewise for the subsequent rows of C. When the entries of the circulant matrix are functions of z (or z^{-1}) rather than numbers, we say instead that we have a circulant matrix *function*, e.g.,

$$\mathcal{C}(z) = \left[\begin{array}{cccc} c_o(z) & c_1(z) & c_2(z) & c_3(z) \\ c_3(z) & c_o(z) & c_1(z) & c_2(z) \\ c_2(z) & c_3(z) & c_o(z) & c_1(z) \\ c_1(z) & c_3(z) & c_2(z) & c_o(z) \end{array}\right] \quad \text{(circulant matrix function)} \tag{28.15}$$

Now, a pseudo-circulant matrix function is essentially a circulant matrix function with the exception that all entries below the main diagonal are further multiplied by z^{-1}, as we see in (28.12).

With the functions $\{\mathcal{Y}(z), \mathcal{X}(z), \mathcal{H}(z)\}$ defined in this manner, some straightforward algebra will show that, starting from expression (28.4), we can relate the vector z–transforms of the input and output block sequences $\{x_{M,n}, y_{M,n}\}$ as follows (see Probs. 28.7 and 28.8):

$$\boxed{\mathcal{Y}(z) = \mathcal{H}(z)\mathcal{X}(z)} \tag{28.16}$$

which is the desired relation. This result is depicted in Fig. 28.3. In this implementation, block entries $x_{M,n}$, of size $M \times 1$, are fed into the $M \times M$ matrix filter, $\mathcal{H}(z)$, and block outputs $y_{M,n}$, also of size $M \times 1$ are obtained as a result. Comparing the implementations of Figs. 28.2 and 28.3, we observe that the former relies on the transfer function $H(z)$ while the latter relies on the matrix transfer function $\mathcal{H}(z)$.

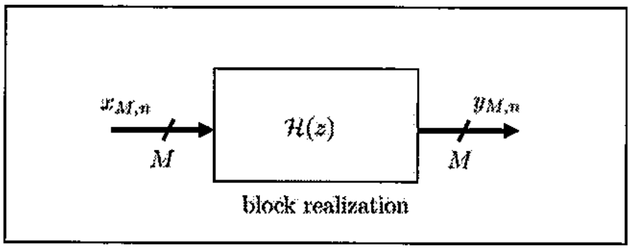

FIGURE 28.3 Block processing: the samples of the input sequence are processed on a block-by-block basis. The letters M next to the arrows are meant to input that the input sample is a block of size $M \times 1$ and the output sample is also a block of size $M \times 1$.

Example 28.1 (Matrix transfer function)

We illustrate the above construction with the following example. Consider the FIR filter

$$H(z) = 1 + \frac{1}{2}z^{-1} + \frac{1}{3}z^{-2} + \frac{1}{4}z^{-3} \tag{28.17}$$

which corresponds to $N = 4$. We select $M = 2$. The second-order polyphase components of $H(z)$ are

$$E_o(z) = 1 + \frac{1}{3}z^{-1}, \qquad E_1(z) = \frac{1}{2} + \frac{1}{4}z^{-1} \tag{28.18}$$

That is,

$$H(z) = E_o(z^2) + z^{-1}E_1(z^2) \tag{28.19}$$

Using (28.12), the corresponding 2×2 transfer matrix is given by

$$\mathcal{H}(z) = \begin{bmatrix} 1 + \frac{1}{3}z^{-1} & \frac{1}{2} + \frac{1}{4}z^{-1} \\ z^{-1}\left(\frac{1}{2} + \frac{1}{4}z^{-1}\right) & 1 + \frac{1}{3}z^{-1} \end{bmatrix} \tag{28.20}$$

According to (28.5), the entries of the vectors $x_{2,n}$ and $y_{2,n}$ in this case are given by

$$x_{2,n} = \begin{bmatrix} x(2n) \\ x(2n-1) \end{bmatrix}, \quad y_{2,n} = \begin{bmatrix} y(2n) \\ y(2n-1) \end{bmatrix} \tag{28.21}$$

Figure 28.4 shows the resulting block realization for the FIR filter $H(z)$ by relying on the 2×2 matrix function $\mathcal{H}(z)$. In this example, even and odd-indexed samples of $x(n)$ are fed into the realization on the left side and similar even and odd-indexed output samples are extracted from the right side.

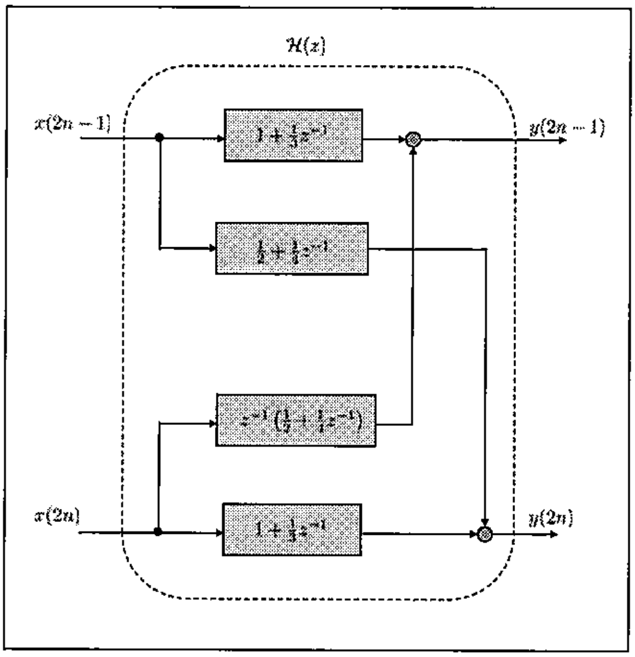

FIGURE 28.4 A 2×2 block realization for the FIR filter $H(z)$ defined by (28.17).

◇

Computational Cost

Although we have arrived at an equivalent block implementation for $H(z)$ in terms of the transfer matrix $\mathcal{H}(z)$, this second implementation is inefficient and more needs to be done in order to transform it into a truly efficient block filtering scheme. Specifically, note that the polyphase components, $E_k(z)$, in (28.12) are polynomials in z^{-1}, and each one of them has in general degree $\frac{N}{M}-1$. It follows that $\mathcal{H}(z)$ is a matrix polynomial function with highest degree equal to $\frac{N}{M}$. This means that we can express $\mathcal{H}(z)$ in the form (compare with (28.1) in the fullband case):

$$\mathcal{H}(z) = H_o + H_1 z^{-1} + H_2 z^{-2} + \ldots + H_{N/M} z^{-N/M} \tag{28.22}$$

for some $M \times M$ coefficient matrices $\{H_k\}$. For instance, for the case of Example 28.1 we have

$$\begin{aligned}\mathcal{H}(z) &= \begin{bmatrix} 1+\frac{1}{3}z^{-1} & \frac{1}{2}+\frac{1}{4}z^{-1} \\ z^{-1}\left(\frac{1}{2}+\frac{1}{4}z^{-1}\right) & 1+\frac{1}{3}z^{-1}\end{bmatrix} \\ &= \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 1\end{bmatrix} + \begin{bmatrix} \frac{1}{3} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3}\end{bmatrix} z^{-1} + \begin{bmatrix} 0 & 0 \\ \frac{1}{4} & 0\end{bmatrix} z^{-2}\end{aligned} \tag{28.23}$$

Then, we could envision a block tapped-delay-line structure for implementing $\mathcal{H}(z)$ with block coefficients $\{H_k\}$ and block input and output vectors, $\{x_{M,n}, y_{M,n}\}$ — see Fig. 28.5. However, such structure would be inefficient for two reasons. First, the $\frac{N}{M}+1$ block coefficients $\{H_k\}$ in (28.22) amount to a total of $M(N+M)$ scalar coefficients, which is at least M times larger than the original number of N coefficients $\{h(n)\}$ that we started with for the fullband filter, $H(z)$, in Fig. 28.1. Second, the transfer matrix $\mathcal{H}(z)$ has a special circulant structured, as can be seen from (28.12). We should be able to exploit this structure to our advantage. Our purpose is therefore to show how to devise a truly efficient structure for implementing $\mathcal{H}(z)$.

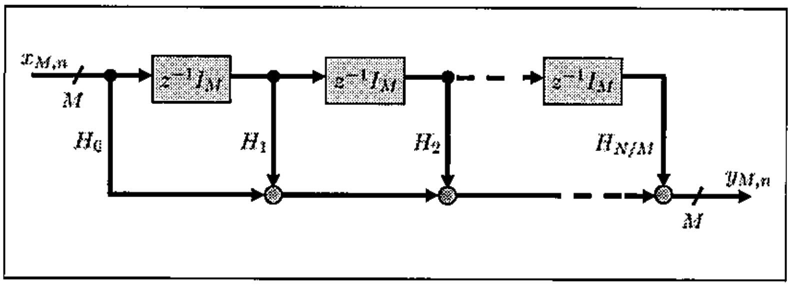

FIGURE 28.5 A block tapped-delay line implementation for $\mathcal{H}(z)$, where the delay elements are multiplied by the identity matrix, I_M, of size $M \times M$, since the data blocks passing through the implementation are now of size $M \times 1$.

A Useful Factorization

As a prelude to our arguments, we start by noting that the transfer matrix $\mathcal{H}(z)$ in (28.12) can be factored as

$$\boxed{\mathcal{H}(z) = \mathcal{E}(z)\mathcal{Q}(z)} \qquad (28.24)$$

where $\mathcal{E}(z)$ is an $M \times (2M-1)$ matrix function with Toeplitz structure, e.g., for $M = 3$,

$$\mathcal{E}(z) = \begin{bmatrix} E_o(z) & E_1(z) & E_2(z) & 0 & 0 \\ 0 & E_o(z) & E_1(z) & E_2(z) & 0 \\ 0 & 0 & E_o(z) & E_1(z) & E_2(z) \end{bmatrix}; \qquad M \times (2M-1) \qquad (28.25)$$

and $\mathcal{Q}(z)$ is a $(2M-1) \times M$ matrix with a leading $M \times M$ identity block and a lower block with unit delays, say, for $M = 3$ again,

$$\mathcal{Q}(z) = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline z^{-1} & 0 & 0 \\ 0 & z^{-1} & 0 \end{array}\right]; \qquad (2M-1) \times M \qquad (28.26)$$

Again, for the case of Example 28.1 we have

$$\begin{aligned} \mathcal{H}(z) &= \begin{bmatrix} 1+\frac{1}{3}z^{-1} & \frac{1}{2}+\frac{1}{4}z^{-1} \\ z^{-1}\left(\frac{1}{2}+\frac{1}{4}z^{-1}\right) & 1+\frac{1}{3}z^{-1} \end{bmatrix} \\ &= \begin{bmatrix} 1+\frac{1}{3}z^{-1} & \frac{1}{2}+\frac{1}{4}z^{-1} & 0 \\ 0 & 1+\frac{1}{3}z^{-1} & \frac{1}{2}+\frac{1}{4}z^{-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ z^{-1} & 0 \end{bmatrix} \qquad (28.27) \end{aligned}$$

28.2 OVERLAP-SAVE DFT-BASED BLOCK FILTERING

The first class of efficient block filtering structures that we shall derive is based on the use of the discrete Fourier transform (DFT) operation. Later, in subsequent sections, we shall derive alternative structures that are based on other transform operations, such as the discrete cosine transform (DCT) and the discrete Hartley transform (DHT).

To begin with, since it is usually desirable to work with sequences whose lengths are powers of 2 when dealing with the DFT, it is therefore convenient to redefine the matrices $\mathcal{E}(z)$ and $\mathcal{Q}(z)$ in (28.25) and (28.26) as follows:

$$\bar{\mathcal{E}}(z) = \begin{bmatrix} E_o(z) & E_1(z) & E_2(z) & 0 & 0 & 0 \\ 0 & E_o(z) & E_1(z) & E_2(z) & 0 & 0 \\ 0 & 0 & E_o(z) & E_1(z) & E_2(z) & 0 \end{bmatrix} ; \qquad (M \times 2M) \qquad (28.28)$$

and

$$\bar{\mathcal{Q}}(z) = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline z^{-1} & 0 & 0 \\ 0 & z^{-1} & 0 \\ 0 & 0 & z^{-1} \end{array}\right] ; \qquad (2M \times M) \qquad (28.29)$$

with an additional zero column added to $\mathcal{E}(z)$ and an additional row added to $\mathcal{Q}(z)$. Of course, the product $\bar{\mathcal{E}}(z)\bar{\mathcal{Q}}(z)$ remains equal to $\mathcal{H}(z)$. Now, however, $\bar{\mathcal{E}}(z)$ is $M \times 2M$ and $\bar{\mathcal{Q}}(z)$ is $2M \times M$, and the factor $2M$ will be a power of 2 whenever M is, which is generally the case.

Determination of Subfilters

With the matrices $\{\bar{\mathcal{E}}(z), \bar{\mathcal{Q}}(z)\}$ so defined, we start by embedding the rectangular matrix $\bar{\mathcal{E}}(z)$ into a $2M \times 2M$ circulant matrix function $\mathcal{C}(z)$, say, for $M = 3$,

$$\mathcal{C}(z) = \left[\begin{array}{cccccc} E_o(z) & E_1(z) & E_2(z) & 0 & 0 & 0 \\ 0 & E_o(z) & E_1(z) & E_2(z) & 0 & 0 \\ 0 & 0 & E_o(z) & E_1(z) & E_2(z) & 0 \\ \hline 0 & 0 & 0 & E_o(z) & E_1(z) & E_2(z) \\ E_2(z) & 0 & 0 & 0 & E_o(z) & E_1(z) \\ E_1(z) & E_2(z) & 0 & 0 & 0 & E_o(z) \end{array}\right] \qquad (28.30)$$

where $\mathcal{E}(z)$ forms the top part of $\mathcal{C}(z)$. Therefore, $\bar{\mathcal{E}}(z)$ can be recovered from the top M rows of $\mathcal{C}(z)$ via

$$\boxed{\bar{\mathcal{E}}(z) = [I_M \;\; 0_{M\times M}]\, \mathcal{C}(z)} \qquad (28.31)$$

where I_M is the $M \times M$ identity matrix and $0_{M\times M}$ is the $M \times M$ null matrix. The main reason for embedding $\bar{\mathcal{E}}(z)$ into $\mathcal{C}(z)$ is the following useful result. Let

$$\boxed{[F]_{km} \;\triangleq\; e^{-\frac{j2\pi mk}{2M}} \qquad k, m = 0, 1, \ldots, 2M-1} \qquad (28.32)$$

denote the entries of the DFT matrix of size $2M \times 2M$. Then, it is a well-known result that any circulant matrix, such as $\mathcal{C}(z)$ above, can be diagonalized by the DFT matrix —

see Probs. 28.15 and 28.19. Specifically, it holds that $\mathcal{C}(z)$ can be written as

$$\boxed{\mathcal{C}(z) = F^*\mathcal{L}(z)F} \tag{28.33}$$

for some $2M \times 2M$ diagonal matrix function, $\mathcal{L}(z)$, with entries:

$$\mathcal{L}(z) = \text{diag}\{L_o(z), L_1(z), \ldots, L_{2M-1}(z)\} \tag{28.34}$$

and where each term $L_k(z)$ represents an FIR transfer function with $\frac{N}{M}$ coefficients. We refer to the $\{L_k(z)\}$ as the *subfilters* that correspond to the original FIR transfer function $H(z)$. Observe that there are $2M$ subfilters, where M is the size of the block vectors. Therefore, starting with $H(z)$, we now have a procedure that allows us to determine its subfilters. The significance of these filters is explained in the sequel. First, we explain how to determine them directly from the polyphase filters — see (28.38).

Using the fact that the DFT matrix is symmetric, i.e., $F = F^{\mathsf{T}}$ (where the superscript $^{\mathsf{T}}$ denotes matrix transposition), it is easy to verify by transposing (28.33) that

$$\mathcal{C}^{\mathsf{T}}(z) \;=\; F^{\mathsf{T}}\mathcal{L}^{\mathsf{T}}(z)\,(F^*)^{\mathsf{T}} \;=\; F\mathcal{L}^{\mathsf{T}}(z)F^* \tag{28.35}$$

Now recall that the first row of F is the vector whose entries are all equal to one; similarly for the first column of F^*. Therefore, multiplying both sides of the above equality by the first basis vector, $\text{col}\{1, 0, \ldots, 0\}$, from the right we readily conclude that the entries on the first column of $\mathcal{C}^{\mathsf{T}}(z)$ are given by the following relation in terms of the subfilters, $L_k(z)$, e.g., for $M = 3$:

$$\begin{bmatrix} E_o(z) \\ E_1(z) \\ E_2(z) \\ 0 \\ 0 \\ 0 \end{bmatrix} = F \begin{bmatrix} L_o(z) \\ L_1(z) \\ L_2(z) \\ L_3(z) \\ L_4(z) \\ L_5(z) \end{bmatrix} \tag{28.36}$$

This important relation tells us how to map the M polyphase components $\{E_k(z)\}$ of $H(z)$ into the $2M$ subfilters $\{L_k(z)\}$ and vice-versa. For instance, using the fact that

$$F^*F = 2M \cdot I_{2M} \tag{28.37}$$

and multiplying (28.36) from the left by F^* we find that, for general M,

$$\boxed{\underbrace{\begin{bmatrix} L_o(z) \\ L_1(z) \\ L_2(z) \\ \vdots \\ L_{2M-1}(z) \end{bmatrix}}_{2M\times 1} = \frac{1}{2M} \cdot F^* \cdot \underbrace{\begin{bmatrix} E_o(z) \\ \vdots \\ E_{M-1}(z) \\ \hline 0_{M\times 1} \end{bmatrix}}_{2M\times 1}} \tag{28.38}$$

which tells us how to evaluate the $L_k(z)$ from the $E_k(z)$. It should be clear that although any diagonal matrix $\mathcal{L}(z)$ in (28.33) will result in a circulant matrix when transformed by F and F^*, it does not hold, however, that any such $\mathcal{L}(z)$ will result in a circulant matrix that has the special form (28.30) with the corresponding zero pattern. This fact is also obvious from the transformation (28.36), which shows that for the resulting circulant matrix to be of the desired form, then the $\{L_k(z)\}$ need to be such that the last M entries of the transformed vector in (28.36) are zero.

Example 28.2 (Computing subfilters)

Returning to the transfer function $H(z)$ from Example 28.1, we have $M = 2$ with polyphase components:

$$E_o(z) = 1 + \frac{1}{3}z^{-1}, \quad E_1(z) = \frac{1}{2} + \frac{1}{4}z^{-1} \tag{28.39}$$

The 4 × 4 DFT matrix is given by

$$F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \tag{28.40}$$

so that using (28.38) we have

$$\begin{bmatrix} L_o(z) \\ L_1(z) \\ L_2(z) \\ L_3(z) \end{bmatrix} = \frac{1}{4}F^* \begin{bmatrix} 1 + \frac{1}{3}z^{-1} \\ \frac{1}{2} + \frac{1}{4}z^{-1} \\ 0 \\ 0 \end{bmatrix} \tag{28.41}$$

and, consequently, the corresponding subfilters are given by

$$L_o(z) = \frac{3}{8} + \frac{7}{48}z^{-1} \tag{28.42a}$$

$$L_1(z) = \left(\frac{1}{4} + j\frac{1}{8}\right) + \left(\frac{1}{12} + j\frac{1}{16}\right)z^{-1} \tag{28.42b}$$

$$L_2(z) = \frac{1}{8} + \frac{1}{48}z^{-1} \tag{28.42c}$$

$$L_3(z) = \left(\frac{1}{4} - j\frac{1}{8}\right) + \left(\frac{1}{12} - j\frac{1}{16}\right)z^{-1} \tag{28.42d}$$

Observe that although $H(z)$ and $E_k(z)$ have real coefficients, the subfilters, $L_k(z)$, generally have complex-valued coefficients.

◇

Block Representation

Combining (28.31) and (28.33) we can express the factorization

$$\mathcal{H}(z) = \bar{\mathcal{E}}(z)\bar{\mathcal{Q}}(z) \tag{28.43}$$

in the more detailed form:

$$\boxed{\mathcal{H}(z) = \underbrace{[I_M \quad 0_{M\times M}]\mathcal{C}(z)}_{\bar{\mathcal{E}}(z)}\,\bar{\mathcal{Q}}(z) = [I_M \quad 0_{M\times M}]\underbrace{F^*\mathcal{L}(z)F}_{\mathcal{C}(z)}\,\bar{\mathcal{Q}}(z)} \tag{28.44}$$

This important result shows that the transfer matrix function $\mathcal{H}(z)$ from Fig. 28.3 can be obtained as the top M rows of the transfer matrix function $F^*\mathcal{L}(z)F\bar{\mathcal{Q}}(z)$. In this way, the mapping from $x_{M,n}$ to $y_{M,n}$, which we depicted generically in Fig. 28.3, can now be implemented in the more efficient manner shown in Fig. 28.6 in terms of the quantities $\{F, L_k(z), \bar{Q}(z)\}$. In this implementation, the $M \times 1$ block vector, $x_{M,n}$ is first transformed by the matrix $\bar{Q}(z)$ into a $2M \times 1$ vector. This vector is subsequently transformed by F and the resulting entries are then fed into the subfilters. The outputs of the subfilters are transformed by F^* and the last M outputs of F^* are discarded, while the top M entries are retained and they correspond to the desired block vector, $y_{M,n}$.

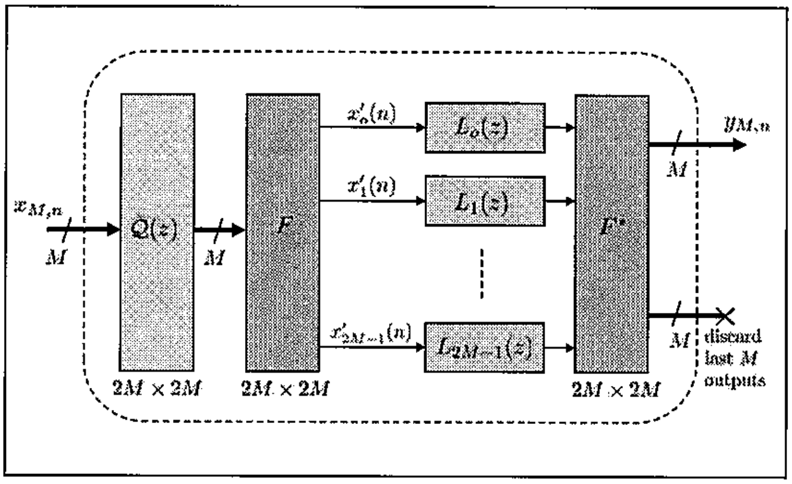

FIGURE 28.6 Equivalent implementation of the mapping shown earlier in Fig. 28.3 for the filter $H(z)$ in terms of $2M \times 2M$ DFT operations and a bank of $2M$ subfilters, $L_k(z)$.

Filter Bank Implementation

We can rework the block diagram shown in Fig. 28.6 into a more explicit filter bank implementation, along the lines studied in Secs. 27.3 and 27.4 when we discussed subband processing and oversampled filter banks. To explain the connection with subband processing, we first examine the operation of the various steps involved in Fig. 28.6 in the time-domain.

Downsampling. To begin with, if we refer to expression (28.29) for the $2M \times 2M$ matrix function $\bar{\mathcal{Q}}(z)$, which we rewrite as follows:

$$\bar{\mathcal{Q}}(z) \;=\; \left[\begin{array}{c} I_M \\ \hline z^{-1}I_M \end{array}\right] \tag{28.45}$$

we conclude that

$$\bar{\mathcal{Q}}(z)\mathcal{X}(z) \;=\; \left[\begin{array}{c} \mathcal{X}(z) \\ \hline z^{-1}\mathcal{X}(z) \end{array}\right] \tag{28.46}$$

and, hence, feeding the block sequence $x_{M,n}$ into $\bar{\mathcal{Q}}(z)$ leads to an output sequence whose samples consist of vectors of size $2M \times 1$ with two sub-vectors stacked on top of each other, namely,

$$x_{M,n} \;\xrightarrow{\bar{\mathcal{Q}}(z)}\; \left[\begin{array}{c} x_{M,n} \\ \hline x_{M,n-1} \end{array}\right] \tag{28.47}$$

The top part is $x_{M,n}$ and the bottom part is the vector from the previous block, $n-1$. For example, for block size $M = 3$ and for $n = 2$, we get

$$\underbrace{\left[\begin{array}{c} x(6) \\ x(5) \\ x(4) \end{array}\right]}_{x_{3,2}} \;\xrightarrow{\bar{\mathcal{Q}}(z)}\; \left[\begin{array}{c} x(6) \\ x(5) \\ x(4) \\ \hline x(3) \\ x(2) \\ x(1) \end{array}\right] \;=\; \left[\begin{array}{c} x_{3,2} \\ \hline x_{3,1} \end{array}\right] \tag{28.48}$$

It is now clear that the transformation by $\bar{\mathcal{Q}}(z)$ can be accomplished by performing a serial-to-parallel (S/P) conversion on the samples of the original sequence, $x(n)$. Observe in particular how the succession of samples $\{x(1), x(2), \ldots, x(6)\}$ have been parallelized and collected into the vector shown in (28.48). One useful way to accomplish the S/P conversion is by means of a chain of downsamplers of order M, as was explained earlier in Example 26.1; recall that every such downsampler keeps the samples of its input signal that occur at multiples of M and discards all other samples. Accordingly, using the downsampler representation, we can construct the data vector (28.47) from the input sequence $x(n)$ by employing the structure shown in Fig. 28.7 using a total of $2M$ downsamplers; after every M input samples, a $2M \times 1$ vector is formed consisting of $\{x_{M,n}, x_{M,n-1}\}$ as in (28.47).

Filtering. The $2M \times 1$ vector (28.47) at the output of $\bar{\mathcal{Q}}(z)$ in Fig. 28.6 is subsequently processed by the DFT matrix F, resulting in a $2M \times 1$ transformed vector, whose entries we shall denote by

$$\left[\begin{array}{c} x'_o(n) \\ x'_1(n) \\ \vdots \\ x'_{2M-1}(n) \end{array}\right] \;\stackrel{\Delta}{=}\; F\left[\begin{array}{c} x_{M,n} \\ \hline x_{M,n-1} \end{array}\right] \tag{28.49}$$

i.e., we are using primes to denote DFT-transformed data. Each sequence $x'_k(n)$ is then fed into the corresponding subfilter, $L_k(z)$. The vector collecting the outputs of the subfilters

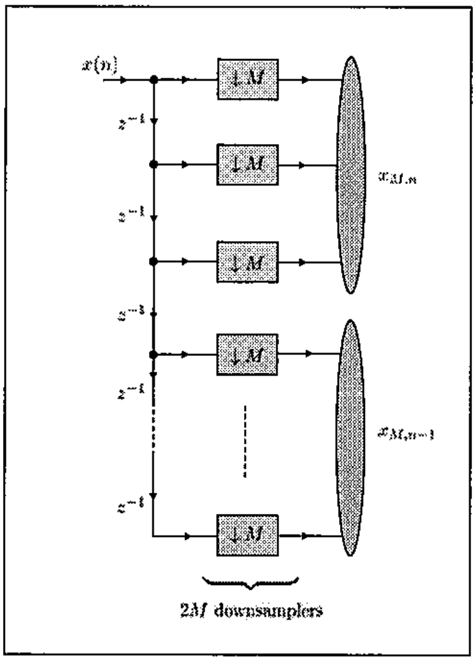

FIGURE 28.7 Formation of the block vectors $x_{M,n}$ and $x_{M,n-1}$ by means of a delay line with $2M$ downsamplers. This downsampler-based structure is equivalent to processing by $\bar{\Omega}(z)$ in Fig. 28.6. The elliptical curves in the above figure are only used for illustration purposes to indicate which entries are aggregated to form the block vectors $x_{M,n}$ and $x_{M,n-1}$.

is further processed by F^*. The top M outputs of F^* from the desired block vector $y_{M,n}$, and the bottom M entries are discarded.

Upsampling. In a manner similar to constructing $x_{M,n}$ from the $\{x(n)\}$ by means of downsamplers, we can also perform a reverse operation and recover the output sequence $y(n)$ from the vector $y_{M,n}$ by means of parallel-to-serial (P/S) conversion using upsamplers of order M, as was already explained in Example 26.3. Recall that the output of every such upsampler is obtained by inserting $M-1$ zeros between successive samples at its input. Using the upsampler representation, we can recover the sequence $y(n)$ from the entries of the block vector $y_{M,n}$ as shown in Fig. 28.8 using a total of M upsamplers; the structure in the figure performs parallel-to-serial data conversion and introduces a delay of size $M-1$ with the signal generated at its output being equal to $y(n-M+1)$. The following example illustrates the operation of this structure and the reason behind the delay.

Filter bank. Using the structures of the serial-to-parallel and parallel-to-serial converters in terms of chains of downsamplers and upsamplers, we conclude that the original structure of Fig. 28.6 can be redrawn more explicitly in the equivalent form shown in Fig. 28.9, with the outputs of the subfilters now denoted by $\{y_k'(n),\ k=0,1\ldots,2M-1\}$. We therefore find that starting from a fullband FIR filter, $H(z)$, it can be equivalently implemented in terms of a bank of $2M$ FIR subfilters, $\{L_k(z)\}$, of length $\frac{N}{M}$ each. More importantly, this new implementation relies on a total of $2M \times (N/M) = 2N$ coefficients of the filters $\{L_k(z)\}$. It may appear, at first sight, that an implementation that relies on $2N$ coefficients would be costlier than an implementation that relies on the N coefficients of the original fullband implementation of Fig. 28.1. However, this is not the case because the filters $L_k(z)$ are

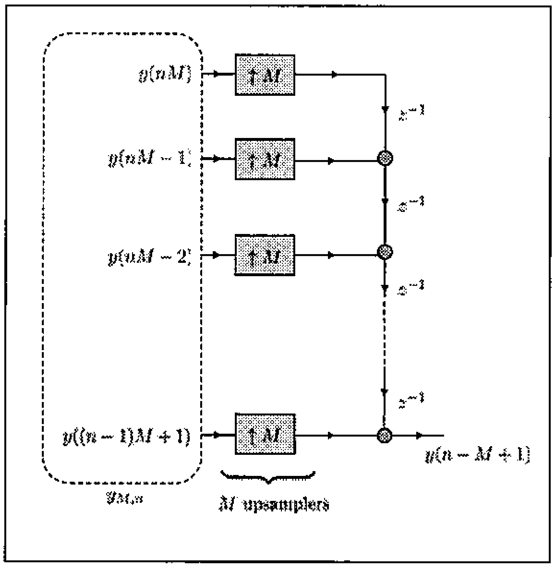

FIGURE 28.8 Reconstruction of the sequence $y(n)$ from the entries of $y_{M,n}$ by means of M upsamplers. Observe that the output sequence is $y(n-M+1)$ with a delay of $M-1$ samples relative to $y(n)$.

operating at a sampling rate that is M times slower than $H(z)$ due to the downsampling operation on the far left side of Fig. 28.9. This difference in the rate of operation leads to significant savings in computation, as we explain further ahead. Here, it is useful to note that although the block implementation of Fig. 28.9 is more efficient than the traditional tapped-delay-line implementation of $H(z)$, as in Fig. 28.1, it nevertheless introduces a delay of $M-1$ samples in the signal path. In Prob. 28.23 we show one way to modify the block implementation of Fig. 28.9 in order to address the delay problem. Basically, a direct convolution path of length M is added to Fig. 28.9.

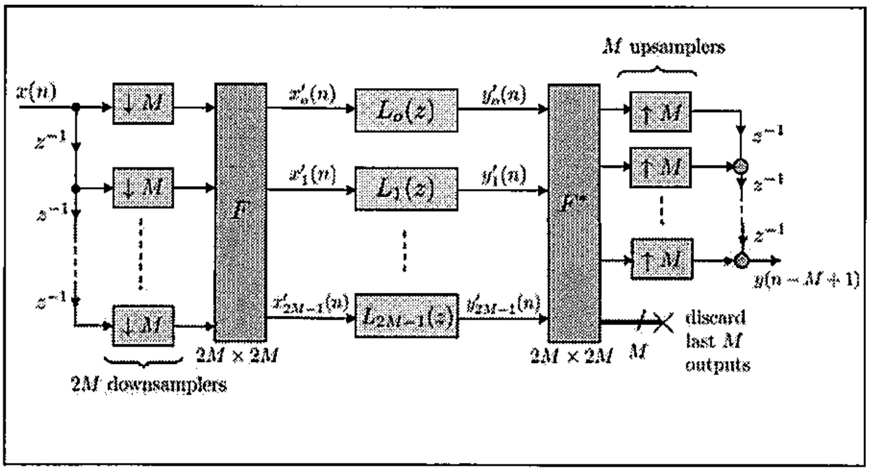

FIGURE 28.9 Overlap-save DFT-based realization corresponding to the block filtering implementation of Fig. 28.6. A delay of $(M-1)$ samples is introduced in the signal path.

We shall refer to the implementation of Fig. 28.9 as an *overlap-save* block-convolution structure. The term overlap-save is used here to indicate that two successive input blocks $\{x_{M,n}, x_{M,n-1}\}$ are aggregated prior to the DFT operation and then one of these blocks is saved and used in the subsequent step. Specifically, at block instants n and $n+1$, the aggregated vectors feeding into the first DFT operation will be

$$\underbrace{\left[\begin{array}{c} x_{M,n} \\ \hline x_{M,n-1} \end{array}\right]}_{\triangleq\, x_{2M,n}} \quad \text{and} \quad \underbrace{\left[\begin{array}{c} x_{M,n+1} \\ \hline x_{M,n} \end{array}\right]}_{\triangleq\, x_{2M,n+1}} \tag{28.50}$$

which shows that the block vector $x_{M,n}$ is shared over these two consecutive block instants. In the above expression, we are denoting the aggregated vectors of size $2M \times 1$ by $x_{2M,n}$ and $x_{2M,n+1}$ for ease of reference. There is an alternative *overlap-add* structure for block filtering, which we describe further ahead. In that case, the successive input blocks to the DFT operation will not share a common sub-block. In summary, we arrive at the following efficient block filtering implementation for an FIR filter $H(z)$ of length N (refer to Figs. 28.6 or 28.9). In the listing, we are using the notation col$\{a, b\}$ to denote a column vector that is formed by stacking its entries a and b on top of each other.

```
Overlap-save DFT-based implementation of a filter $H(z)$ of length $N$
Choose a block size $M$, where $N$ and $M$ are powers of 2 and $N/M$ is an integer.
Determine the $M$ polyphase components $E_k(z)$ of $H(z)$ of order $N/M$ from (28.10).
Use (28.38) or (28.52) to determine the $2M$ subfilters $\{L_k(z)\}$ of size $N/M$ each.
Set $x_{M,-1} = 0$.
for $n \geq 0$ (block index):
    Construct the block vector $x_{M,n}$ as in (28.5) and set $x_{2M,n} = \text{col}\{x_{M,n}, x_{M,n-1}\}$.
    Perform the DFT transformation: $x'_{2M,n} = Fx_{2M,n} = \text{col}\{x'_k(n)\}$.
    Filter $\{x'_k(n)\}$ through $\{L_k(z)\}$ to generate $y'_k(n)$. Let $y'_{2M,n} = \text{col}\{y'_k(n)\}$.
    Perform the DFT transformation $y_{M,n} = \begin{bmatrix} I_M & 0_{M\times M} \end{bmatrix} F^* y'_{2M,n}$.
end for
```

Before proceeding, it is worth comparing the structure of Fig. 28.9 with the oversampled DFT analysis and synthesis filter banks studied earlier in Figs. 27.35 and 27.36. It is observed that the DFT-based realization of Fig. 28.9 relies on the use of an oversampled filter bank. Moreover, the signals $\{x'_k(n)\}$ are subjected to processing, which is performed by the subfilters $\{L_k(z)\}$. The result of this step is subsequently fed into the synthesis filter bank. Compared with Figs. 27.35 and 27.36 where F^* is used for analysis and F for synthesis, the roles of F and F^* are reversed in Fig. 28.9; this is due to the factorization used in (28.33) where F^* multiplies $L(z)$ from the left and F multiplies it from the right — see Prob. 28.17.

Example 28.3 (Overlap-save DFT-based realization)

Returning to Example 28.2, Fig. 28.10 shows the corresponding overlap-save filter bank realization with four subfilters $L_o(z)$, $L_1(z)$, $L_2(z)$, and $L_3(z)$ and using $M = 2$.

◇

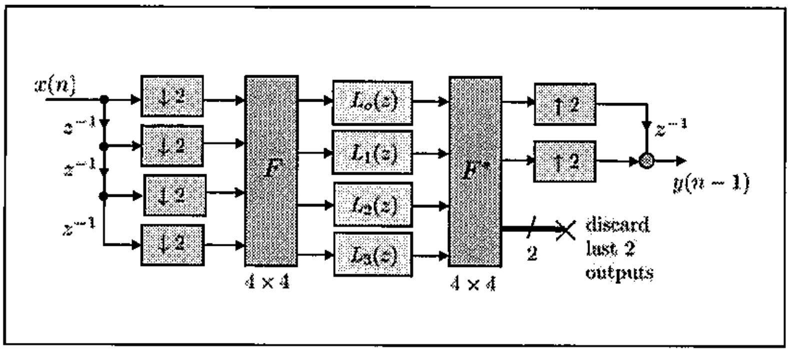

FIGURE 28.10 Filter bank realization for the transfer function $H(z)$ from Example 28.2.

Computational Complexity

Let us now compare the computational complexity of the direct tapped-delay line implementation of $H(z)$ in Fig. 28.1 with the DFT-based block implementation of Fig. 28.9. To get an idea of how these structures compare to each other, it is sufficient to examine the required number of multiplications per input sample. In the following arguments we assume that a DFT of size K requires approximately $\frac{K}{2}\log_2(K)$ complex multiplications when evaluated by means of the Fast Fourier transform (FFT).

In the tapped-delay-line implementation of Fig. 28.1 for $H(z)$, we need to compute an inner product of order N, which translates into N complex multiplications per input sample. In the DFT-based implementation of Fig. 28.9, on the other hand, the following steps are necessary for each block of data of size M:

(1) A DFT transform of size $2M$ applied to the aggregate vector formed from the subvectors $x_{M,n}$ and $x_{M,n-1}$. This step requires $M\log_2(2M)$ complex multiplications.

(2) Filtering by $2M$ filters $\{L_k(z)\}$ of order N/M each. Each filtering operation requires an inner product of size N/M and, therefore, N/M complex multiplications. In total, this step requires $2N$ complex multiplications.

(3) A second DFT transform of size $2M$ to generate the output signals. This step requires $M\log_2(2M)$ complex multiplications.

Steps 1–3 add up to $2N{+}2M\log_2 2M$ complex multiplications for each block of input data of size M. If we normalize by the size of the block, we conclude that the cost associated with the implementation of Fig. 28.9 is approximately

$$\boxed{\frac{2N}{M} + 2\log_2(2M) \quad \text{complex multiplications per input sample}} \tag{28.51}$$

The main conclusion is that while the direct tapped-delay-line implementation method requires $O(N)$ operations per sample, the DFT-based block implementation method of Fig. 28.9 requires $O(2N/M)$ operations per sample; the reduction in complexity is determined by the block size M. However, larger values for M result in longer delays in the signal path.

In addition, there is a one-time overhead cost that is associated with the evaluation of the subfilters $\{L_k(z)\}$, which need to be determined from $H(z)$. Using (28.38), we find that, for example, for $M = 3$ and $N = 12$:

$$\underbrace{\begin{bmatrix} l_{00} & l_{01} & l_{02} & l_{03} \\ l_{10} & l_{11} & l_{12} & l_{13} \\ l_{20} & l_{21} & l_{22} & l_{23} \\ l_{30} & l_{31} & l_{32} & l_{33} \\ l_{40} & l_{41} & l_{42} & l_{43} \\ l_{50} & l_{51} & l_{52} & l_{53} \end{bmatrix}}_{2M \times N/M} = \frac{1}{6}F^* \underbrace{\begin{bmatrix} h(0) & h(3) & h(6) & h(9) \\ h(1) & h(4) & h(7) & h(10) \\ h(2) & h(5) & h(8) & h(11) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{2M \times N/M} \qquad (28.52)$$

where the $\{l_{km}\}$ denote the coefficients of the filters $\{L_k(z)\}$. The above step therefore requires computing N/M DFT's of size $2M$ each, which amounts to a total cost of $N\log_2(2M)$ complex multiplications. If we normalize by the size of the block, we get an overhead cost on the order of $\frac{N}{M}\log_2(2M)$ complex multiplications per input sample.

Example 28.4 (Cost comparison)

We plot in Fig. 28.11 two curves that show how the ratio

$$\frac{\text{block implementation cost per sample}}{\text{tapped-delay-line cost per sample}} = \frac{\frac{2N}{M} + 2\log_2(2M)}{N} \qquad (28.53)$$

varies as a function of N and M. In the top plot of the figure we fix the fullband length at $N = 1024$ taps and vary the block size M in powers of 2 between 2 and 1024. In the bottom plot we fix the block length at $M = 32$ and vary the fullband filter length N in powers of 2 between 32 and 1024. We see in both cases that ratio is below one for the range of values that are being considered for M and N.

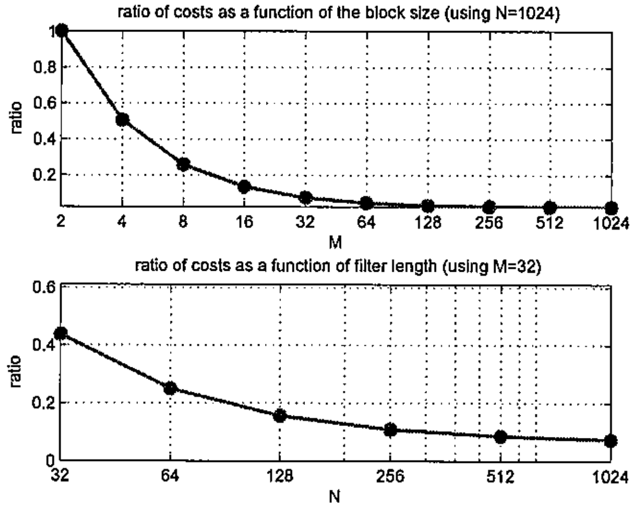

FIGURE 28.11 Plots of the ratio (28.53) for fixed N and varying M (top curve) and fixed M and varying N (bottom curve).

◇

28.3 OVERLAP-ADD DFT-BASED BLOCK FILTERING

In the implementation of Fig. 28.9, successive input blocks overlap and share a sub-vector with M entries prior to the DFT operation, as indicated by (28.50). In this section, we derive an alternative structure for block filtering where the successive input blocks do not share entries. The overlap-add structure can be motivated as follows.

Rather than factor $\mathcal{H}(z)$ as in (28.24) with $\{\bar{\mathcal{E}}(z), \bar{\mathcal{Q}}(z)\}$ given by (28.28)–(28.29), we instead factor it as

$$\mathcal{H}(z) = \bar{\mathcal{Q}}_1(z)\bar{\mathcal{E}}_1(z) \tag{28.54}$$

where now, e.g., for $M = 3$,

$$\bar{\mathcal{E}}_1(z) = \begin{bmatrix} 0 & 0 & 0 \\ E_2(z) & 0 & 0 \\ E_1(z) & E_2(z) & 0 \\ E_o(z) & E_1(z) & E_2(z) \\ 0 & E_o(z) & E_1(z) \\ 0 & 0 & E_o(z) \end{bmatrix}, \quad \bar{\mathcal{Q}}_1(z) = \begin{bmatrix} z^{-1} & 0 & 0 & 1 & 0 & 0 \\ 0 & z^{-1} & 0 & 0 & 1 & 0 \\ 0 & 0 & z^{-1} & 0 & 0 & 1 \end{bmatrix} \tag{28.55}$$

If we refer again to the case of Example 28.1 we have

$$\begin{aligned} \mathcal{H}(z) &= \begin{bmatrix} 1+\frac{1}{3}z^{-1} & \frac{1}{2}+\frac{1}{4}z^{-1} \\ z^{-1}\left(\frac{1}{2}+\frac{1}{4}z^{-1}\right) & 1+\frac{1}{3}z^{-1} \end{bmatrix} \\ &= \begin{bmatrix} z^{-1} & 0 & 1 & 0 \\ 0 & z^{-1} & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ \frac{1}{2}+\frac{1}{4}z^{-1} & 0 \\ 1+\frac{1}{3}z^{-1} & \frac{1}{2}+\frac{1}{4}z^{-1} \\ 0 & 1+\frac{1}{3}z^{-1} \end{bmatrix} \end{aligned} \tag{28.56}$$

Now, following the same line of reasoning we used in Sec. 28.2, we can embed $\bar{\mathcal{E}}_1(z)$ into the same circulant matrix function $\mathcal{C}(z)$ defined by (28.30). Then, $\bar{\mathcal{E}}_1(z)$ can be recovered from the last M columns of $\mathcal{C}(z)$ as follows:

$$\bar{\mathcal{E}}_1(z) = \mathcal{C}(z)\begin{bmatrix} 0_{M\times M} \\ I_M \end{bmatrix} \tag{28.57}$$

in which case expression (28.44) becomes

$$\boxed{\mathcal{H}(z) = \bar{\mathcal{Q}}_1(z)\underbrace{\mathcal{C}(z)\begin{bmatrix} 0_{M\times M} \\ I_M \end{bmatrix}}_{\bar{\mathcal{E}}_1(z)} = \bar{\mathcal{Q}}_1(z)\underbrace{F^*\mathcal{L}(z)F}_{\mathcal{C}(z)}\begin{bmatrix} 0_{M\times M} \\ I_M \end{bmatrix}} \tag{28.58}$$

This result shows that the mapping from $x_{M,n}$ to $y_{M,n}$ in Fig. 28.3 can be equivalently implemented as shown in Fig. 28.12 — compare with Fig. 28.6.

We can rework the structure of Fig. 28.12 into an alternative form as follows. First, we note that the delays that appear after the F^* operation can be moved prior to the subfilters $\{L_k(z)\}$. To see how this step can be accomplished, we first express $\bar{\mathcal{Q}}_1(z)$ as

$$\bar{\mathcal{Q}}_1(z) = \begin{bmatrix} 0_{M\times M} & I_M \end{bmatrix} + z^{-1}\begin{bmatrix} I_M & 0_{M\times M} \end{bmatrix} \tag{28.59}$$

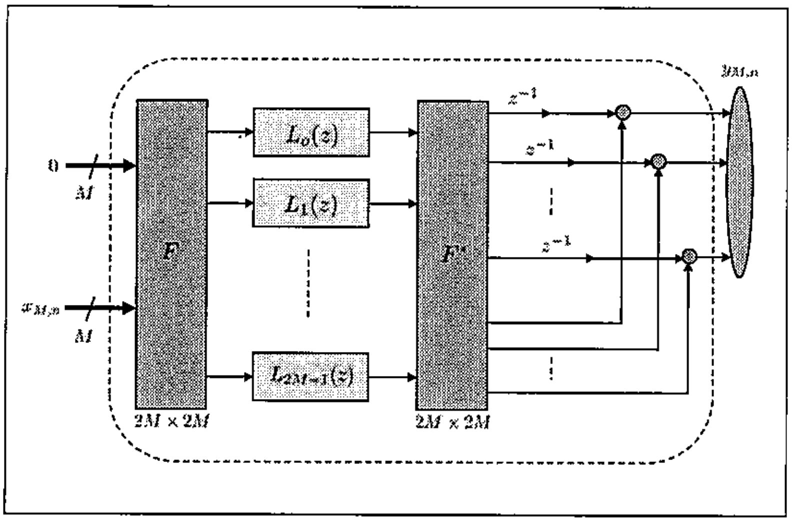

FIGURE 28.12 An overlap-add DFT-based implementation of the block mapping of Fig. 28.3; compare with the overlap-save DFT-based implementation of Fig. 28.9. The elliptical curve on the far right in the above figure is only used for illustration purposes to indicate which entries are aggregated to form the block vector $y_{M,n}$.

so that expression (28.58) for $\mathcal{H}(z)$ becomes

$$\begin{aligned}\mathcal{H}(z) &= \begin{bmatrix} 0_{M\times M} & I_M \end{bmatrix} F^*\mathcal{L}(z)F \begin{bmatrix} 0_{M\times M} \\ I_M \end{bmatrix} + \\ & \quad z^{-1}\begin{bmatrix} I_M & 0_{M\times M} \end{bmatrix} F^*\mathcal{L}(z)F \begin{bmatrix} 0_{M\times M} \\ I_M \end{bmatrix} \end{aligned} \qquad (28.60)$$

Now it can be verified that the DFT matrix satisfies the property (see Prob. 28.21):

$$F\begin{bmatrix} I_M \\ 0_{M\times M} \end{bmatrix} = JF\begin{bmatrix} 0_{M\times M} \\ I_M \end{bmatrix} \qquad (28.61)$$

where

$$J = \text{diag}\{1,-1,1,-1,\ldots,1,-1\} \qquad (28.62)$$

is the $2M \times 2M$ diagonal matrix with alternating $\pm 1's$; here, the notation diag$\{a,b\}$ denotes a diagonal matrix with entries a and b. It follows from (28.61) by transposition that

$$\begin{bmatrix} I_M & 0_{M\times M} \end{bmatrix} F^* = \begin{bmatrix} 0_{M\times M} & I_M \end{bmatrix} F^*J \qquad (28.63)$$

Substituting (28.63) into the second term in (28.60) we find that $\mathcal{H}(z)$ can be rewritten more compactly as

$$\boxed{\mathcal{H}(z) = \begin{bmatrix} 0_{M\times M} & I_M \end{bmatrix} F^*\mathcal{L}(z)(I_{2M}+z^{-1}J)F \begin{bmatrix} 0_{M\times M} \\ I_M \end{bmatrix}} \qquad (28.64)$$

This result shows that the implementation of Fig. 28.12 can be redrawn as shown in Fig. 28.13 where the signals $\{s'_k(n)\}$ are generated via

$$s'_k(n) = x'_k(n) + (-1)^k x'_k(n-1), \quad k = 0,1,\ldots,2M-1 \qquad (28.65)$$

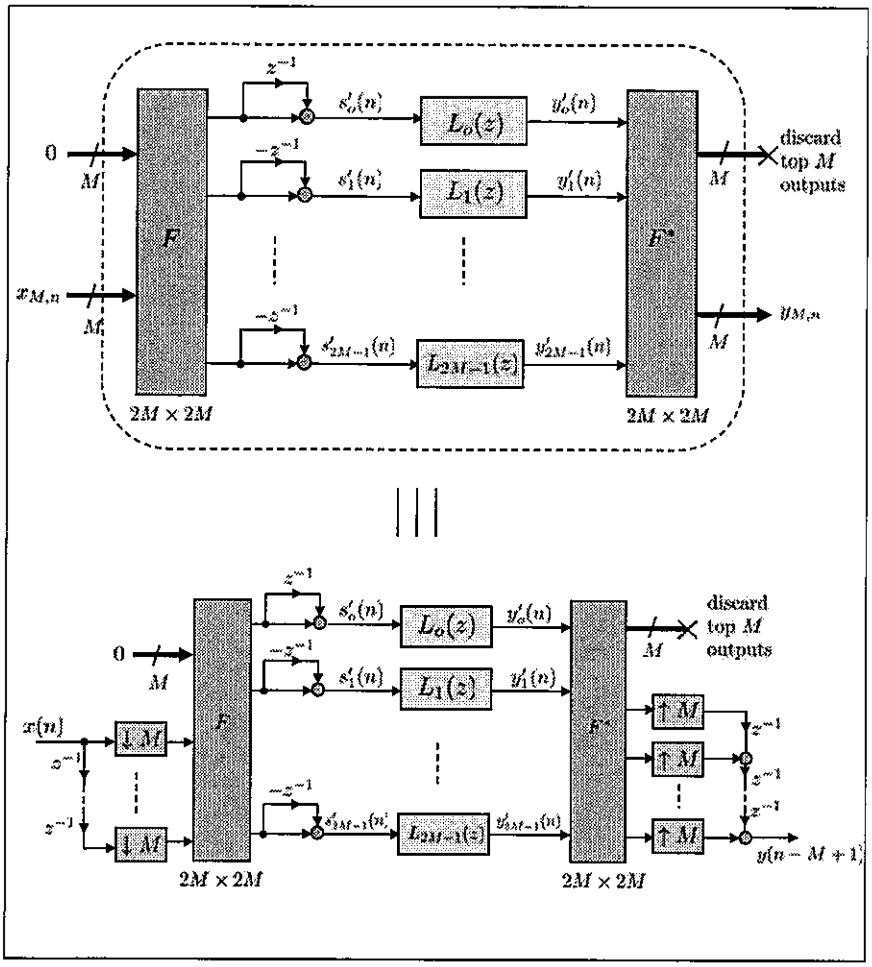

FIGURE 28.13 Overlap-add DFT-based block implementation (top) and an equivalent implementation in terms of upsamplers and downsamplers (bottom). The delays at the output of Fig. 28.12 are now moved prior to the subfilters.

Overlap-add DFT-based implementation of a filter $H(z)$ of length N

```
Choose a block size $M$, where $N$ and $M$ are powers of 2 and $N/M$ is an integer.
Determine the $M$ polyphase components $E_k(z)$ of $H(z)$ of order $N/M$ from (28.10).
Use (28.38) or (28.52) to determine the $2M$ subfilters $\{L_k(z)\}$ of size $N/M$ each.
Set $x_{M,-1} = 0$.
for $n \geq 0$ (block index):
    Construct the block vector $x_{M,n}$ as in (28.5) and set $x_{2M,n} = \text{col}\{0_{M\times 1}, x_{M,n}\}$.
    Perform the DFT transformation: $x'_{2M,n} = Fx_{2M,n}$
    Set $s'_{2M,n} = x'_{2M,n} + Jx'_{2M,n-1} = \text{col}\{s'_k(n)\}$
    Filter $\{s'_k(n)\}$ through $\{L_k(z)\}$ to generate $y'_k(n)$. Let $y'_{2M,n} = \text{col}\{y'_k(n)\}$.
    Perform the DFT transformation $y_{M,n} = \begin{bmatrix} 0_{M\times M} & I_M \end{bmatrix} F^* y'_{2M,n}$.
end for
```

Example 28.5 (Overlap-add DFT-based filter bank realization)

Returning to Example 28.2, Fig. 28.14 shows the corresponding overlap-add filter bank realization with four subfilters $L_o(z), L_1(z), L_2(z)$, and $L_3(z)$ and using $M = 2$.

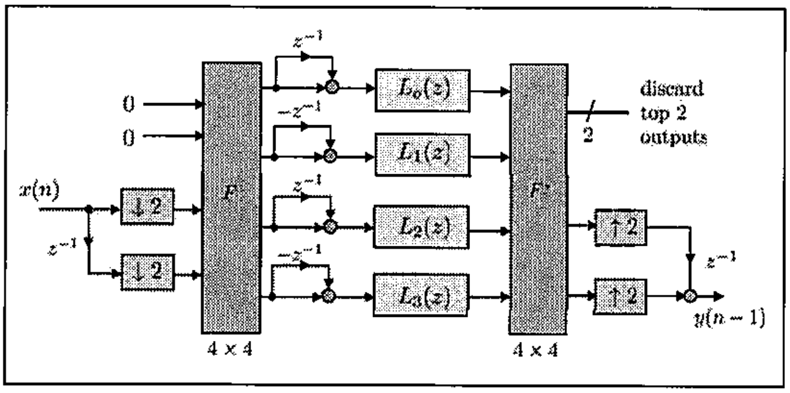

FIGURE 28.14 Filter bank realization for the transfer function $H(z)$ from Example 28.2.

◇

28.4 DCT-BASED BLOCK FILTERING

In the previous sections we derived block filter implementations that were based on the DFT operation. In this section, we show how to derive block implementations that are based instead on the discrete-cosine transform (DCT). There are several variations of the DCT transform and we encountered a couple of them earlier in Sec. 18.3. In this section, we are going to employ the following matrix transformation, say, for a transform of size $K \times K$:

$$\boxed{[C]_{km} \triangleq \alpha(k)\cos\left(\frac{\pi k(2m+1)}{2K}\right), \quad k,m = 0,1,\ldots,K-1} \tag{28.66}$$

where $\alpha(0) = 1/\sqrt{K}$ and $\alpha(k) = \sqrt{2/K}$, $k \neq 0$. Here k indicates the row index and m the column index. For example, the resulting matrix of size 4×4 is given by

$$C = \frac{1}{\sqrt{2}}\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos\left(\frac{\pi}{8}\right) & \cos\left(\frac{3\pi}{8}\right) & \cos\left(\frac{5\pi}{8}\right) & \cos\left(\frac{7\pi}{8}\right) \\ \cos\left(\frac{\pi}{4}\right) & \cos\left(\frac{3\pi}{4}\right) & \cos\left(\frac{5\pi}{4}\right) & \cos\left(\frac{7\pi}{4}\right) \\ \cos\left(\frac{3\pi}{8}\right) & \cos\left(\frac{9\pi}{8}\right) & \cos\left(\frac{15\pi}{8}\right) & \cos\left(\frac{21\pi}{8}\right) \end{bmatrix} \tag{28.67}$$

Example 28.6 (Relation to DCT-III)

It is straightforward to verify that the matrix (28.67) is directly related the DCT-II transformation defined earlier by expression (18.190). Indeed, if we let $C_{\rm II}$ denote the $K \times K$ matrix that maps the vector of samples $\{x(0), x(1), \ldots, x(K-1)\}$ into their DCT-II coefficients $\{X_{\rm DCT,II}(k),\ k = 0, 1, \ldots, K-1\}$, then we know from (18.190) that this matrix is given by, e.g., for $K = 4$ (where the variable K here corresponds to the variable N in expression (18.190)):

$$C_{\rm II} \;=\; C = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \cos\left(\frac{\pi}{8}\right) & \cos\left(\frac{3\pi}{8}\right) & \cos\left(\frac{5\pi}{8}\right) & \cos\left(\frac{7\pi}{8}\right) \\ \cos\left(\frac{\pi}{4}\right) & \cos\left(\frac{3\pi}{4}\right) & \cos\left(\frac{5\pi}{4}\right) & \cos\left(\frac{7\pi}{4}\right) \\ \cos\left(\frac{3\pi}{8}\right) & \cos\left(\frac{9\pi}{8}\right) & \cos\left(\frac{15\pi}{8}\right) & \cos\left(\frac{21\pi}{8}\right) \end{bmatrix} \tag{28.68}$$

Comparing with (28.67) we observe that

$$C = D\,C_{\rm II} \tag{28.69}$$

where, for a general K, the diagonal scaling matrix D is $K \times K$ and defined by

$$D = \frac{1}{\sqrt{K}} \cdot \text{diag}\left\{1,\ \sqrt{2},\ \sqrt{2}, \ldots, \sqrt{2}\right\} \tag{28.70}$$

Likewise, let $C_{\rm II}^{-1}$ denote the $K \times K$ matrix that maps the DCT-II coefficients $\{X_{\rm DCT,II}(k),\ k = 0, 1, \ldots, K-1\}$ back to the samples $\{x(0), x(1), \ldots, x(K-1)\}$. Then, we know from the inversion expression (18.198) that, for $K = 4$,

$$C_{\rm II}^{-1} \;=\; \frac{1}{2}\begin{bmatrix} 1/2 & \cos\left(\frac{\pi}{8}\right) & \cos\left(\frac{\pi}{4}\right) & \cos\left(\frac{3\pi}{8}\right) \\ 1/2 & \cos\left(\frac{3\pi}{8}\right) & \cos\left(\frac{3\pi}{4}\right) & \cos\left(\frac{9\pi}{8}\right) \\ 1/2 & \cos\left(\frac{5\pi}{8}\right) & \cos\left(\frac{5\pi}{4}\right) & \cos\left(\frac{15\pi}{8}\right) \\ 1/2 & \cos\left(\frac{7\pi}{8}\right) & \cos\left(\frac{7\pi}{4}\right) & \cos\left(\frac{21\pi}{8}\right) \end{bmatrix} \tag{28.71}$$

so that comparing with (28.68) we conclude that

$$C_{\rm II}^{-1} = C_{\rm II}^{\sf T}\, D^2 \tag{28.72}$$

Consequently, the manner by which the matrix C has been defined in (28.67) ensures that it is an orthogonal matrix (while $C_{\rm II}$ is not) since

$$C^{\sf T}C = C_{\rm II}^{\sf T}D \cdot DC_{\rm II} \;=\; C_{\rm II}^{\sf T}D^2C_{\rm II} = C_{\rm II}^{-1}C_{\rm II} = I_K \tag{28.73}$$

That is,

$$\boxed{CC^{\sf T} = C^{\sf T}C = I_K} \tag{28.74}$$

◇

Returning to (28.67), just like the DFT matrix diagonalizes circulant matrices, as was the case with (28.33), it is also known that the DCT matrix C diagonalizes any $K \times K$ matrix function $\mathcal{A}(z)$ that can be expressed in the following form:

$$\mathcal{A}(z) = \mathcal{T}(z) + \mathcal{G}(z) + \mathcal{B}(z) \tag{28.75}$$

where $\mathcal{T}(z)$ is a *symmetric* Toeplitz matrix, $\mathcal{G}(z)$ is a Hankel matrix related to $\mathcal{T}(z)$, and $\mathcal{B}(z)$ is a "border" matrix also related to $\mathcal{T}(z)$. A Hankel matrix is one with constant entries across its anti-diagonals. For example,

$$G = \begin{bmatrix} g_o & g_1 & g_2 & g_3 \\ g_1 & g_2 & g_3 & g_4 \\ g_2 & g_3 & g_4 & g_5 \\ g_3 & g_4 & g_5 & g_6 \end{bmatrix} \tag{28.76}$$

is a 4×4 Hankel matrix; note, for example, that the entries across the main anti-diagonal are all equal to g_3. Now, for $K = 4$, the matrices $\{\mathcal{T}(z), \mathcal{B}(z), \mathcal{G}(z)\}$ in (28.75) need to have the following structures:

$$\mathcal{T}(z) \;=\; \begin{bmatrix} t_0(z) & t_1(z) & t_2(z) & t_3(z) \\ t_1(z) & t_0(z) & t_1(z) & t_2(z) \\ t_2(z) & t_1(z) & t_0(z) & t_1(z) \\ t_3(z) & t_2(z) & t_1(z) & t_0(z) \end{bmatrix} \tag{28.77a}$$

$$\mathcal{G}(z) \;=\; \begin{bmatrix} t_0(z) & t_1(z) & t_2(z) & t_3(z) \\ t_1(z) & t_2(z) & t_3(z) & 0 \\ t_2(z) & t_3(z) & 0 & -t_3(z) \\ t_3(z) & 0 & -t_3(z) & -t_2(z) \end{bmatrix} \tag{28.77b}$$

$$\mathcal{B}(z) \;=\; \begin{bmatrix} -\frac{t_0(z)}{\sqrt{2}-2} & t_1(z) & t_2(z) & t_3(z) \\ t_1(z) & & & \\ t_2(z) & & & \\ t_3(z) & & & \end{bmatrix}(\sqrt{2}-2) \tag{28.77c}$$

Observe how the entries of the Hankel matrix, $\mathcal{G}(z)$, and the border matrix, $\mathcal{B}(z)$, are defined in terms of the entries of the Toeplitz matrix, $\mathcal{T}(z)$. When the three matrices $\{\mathcal{T}(z), \mathcal{G}(z), \mathcal{B}(z)\}$ are constructed in this manner, then it will hold that the resulting matrix sum $\mathcal{A}(z)$ is diagonalizable by the DCT matrix, C, defined by (28.67). We exploit this fact in the sequel to derive another efficient implementation for block filtering that is based on the DCT transformation.

Following the same arguments we employed in the previous two sections while deriving DFT-based block implementations, we start by embedding the $M \times (2M-1)$ matrix $\mathcal{E}(z)$ from (28.25) into a new matrix $\mathcal{A}(z)$ as follows. Assume, for illustration purposes, that $M = 2$ so that

$$\mathcal{E}(z) = \begin{bmatrix} E_o(z) & E_1(z) & 0 \\ 0 & E_o(z) & E_1(z) \end{bmatrix} \tag{28.78}$$

We first embed $\mathcal{E}(z)$ into a symmetric Toeplitz matrix $\mathcal{T}(z)$ as

$$\mathcal{T}(z) \;\triangleq\; \left[\begin{array}{cc|ccc|} 0 & E_o(z) & E_1(z) & 0 & 0 \\ \cline{3-5} E_o(z) & 0 & E_o(z) & E_1(z) & 0 \\ E_1(z) & E_o(z) & 0 & E_o(z) & E_1(z) \\ \cline{3-5} 0 & E_1(z) & E_o(z) & 0 & E_o(z) \\ 0 & 0 & E_1(z) & E_o(z) & 0 \end{array}\right] \tag{28.79}$$

where the framed entries correspond to $\mathcal{E}(z)$. The size of $\mathcal{T}(z)$ is $K \times K$, where K is chosen as follows

$$\boxed{K = \begin{cases} (7M-4)/2, & \text{if } M \text{ is even} \\ (7M-3)/2, & \text{if } M \text{ is odd} \end{cases}} \tag{28.80}$$

For example, for the case $M = 2$ under illustration, we get $K = 5$, which is the size of the matrix $\mathcal{T}(z)$ constructed in (28.79). More generally, for an arbitrary block size M, the first row of $\mathcal{T}(z)$ will have the form

$$\begin{bmatrix} 0_{1\times M-1} & E_o(z)\ldots & E_{M-1}(z) & 0_{1\times\alpha} \end{bmatrix} \tag{28.81}$$

with $M-1$ leading zeros and α trailing zeros where

$$\alpha = \begin{cases} (3M-2)/2, & \text{if } M \text{ is even} \\ (3M-1)/2, & \text{if } M \text{ is odd} \end{cases} \tag{28.82}$$

Using (28.79) and the definition (28.75), the corresponding matrix $\mathcal{A}(z)$ will be given by:

$$\mathcal{A}(z) = \left[\begin{array}{ccccc} 0 & \sqrt{2}E_o(z) & \sqrt{2}E_1(z) & 0 & 0 \\ \cline{3-5} \sqrt{2}E_o(z) & E_1(z) & \multicolumn{1}{|c}{E_o(z)} & E_1(z) & \multicolumn{1}{c|}{0} \\ \sqrt{2}E_1(z) & E_o(z) & \multicolumn{1}{|c}{0} & E_o(z) & \multicolumn{1}{c|}{E_1(z)} \\ \cline{3-5} 0 & E_1(z) & E_o(z) & 0 & E_o(z) \\ 0 & 0 & E_1(z) & E_o(z) & 0 \end{array}\right] \tag{28.83}$$

where the framed entries again correspond to $\mathcal{E}(z)$. More generally, for arbitrary block size M, the matrix function $\mathcal{E}(z)$ can be recovered from $\mathcal{A}(z)$ via the relation:

$$\boxed{\mathcal{E}(z) = \left[\begin{array}{ccc} 0_{M\times\eta} & I_M & 0_{M\times 2M-2} \end{array}\right] \mathcal{A}(z) \left[\begin{array}{c} 0_{\alpha\times(2M-1)} \\ I_{2M-1} \end{array}\right]} \tag{28.84}$$

where

$$\eta = \begin{cases} M/2, & \text{if } M \text{ is even} \\ (M+1)/2, & \text{if } M \text{ is odd} \end{cases} \tag{28.85}$$

As mentioned before, the matrix $\mathcal{A}(z)$ can be diagonalized by the DCT matrix, C. Specifically, it holds that

$$\mathcal{A}(z) = C\mathcal{L}(z)C^{\mathsf{T}} \tag{28.86}$$

where $\mathcal{L}(z) = \text{diag}\{L_k(z)\}$ has K entries; we continue to refer to the $\{L_k(z)\}$ as subfiltes except that now they are computed by means of a different transformation and, therefore, will be different from the subfilters obtained under the earlier DFT transformation (28.33). Indeed, using the fact that the DCT matrix, C, is orthogonal and in a manner similar to (28.36), we can verify that the K subfilters can be evaluated from the M polyphase components as follows (see Prob. 28.22):

$$\left[\begin{array}{c} L_o(z) \\ L_1(z) \\ \vdots \\ L_{K-1}(z) \end{array}\right] = \sqrt{2K}\, C^{\mathsf{T}} \left[\begin{array}{c} 0_{(M-1)\times 1} \\ E_o(z) \\ \vdots \\ E_{M-1}(z) \\ 0_{\alpha\times 1} \end{array}\right] \tag{28.87}$$

This expression is the analogue of (28.38) in the DCT case. We can now return to the decomposition (28.24) and write it as

$$\begin{aligned} \mathcal{H}(z) &= \underbrace{\left[\begin{array}{ccc} 0_{M\times\eta} & I_M & 0_{M\times 2M-2} \end{array}\right] \mathcal{A}(z) \left[\begin{array}{c} 0_{\alpha\times(2M-1)} \\ I_{2M-1} \end{array}\right]}_{\mathcal{E}(z)} \mathcal{Q}(z) \\ &= \left[\begin{array}{ccc} 0_{M\times\eta} & I_M & 0_{M\times 2M-2} \end{array}\right] \underbrace{C\mathcal{L}(z)C^{\mathsf{T}}}_{\mathcal{A}(z)} \left[\begin{array}{c} 0_{\alpha\times(2M-1)} \\ I_{2M-1} \end{array}\right] \mathcal{Q}(z) \\ &= \left[\begin{array}{ccc} 0_{M\times\eta} & I_M & 0_{M\times 2M-2} \end{array}\right] C\mathcal{L}(z)C^{\mathsf{T}} \left[\begin{array}{cc} 0_{\alpha\times(2M-1)} & 0 \\ I_{2M-1} & 0 \end{array}\right] \bar{\mathcal{Q}}(z) \end{aligned} \tag{28.88}$$

where $\{\mathcal{E}(z), \mathcal{Q}(z), \bar{\mathcal{Q}}(z)\}$ are from (28.25)–(28.29). This result shows that the mapping from $x_{M,n}$ to $y_{M,n}$ in Fig. 28.3 can be implemented as shown in Fig. 28.15 — compare with Fig. 28.9.

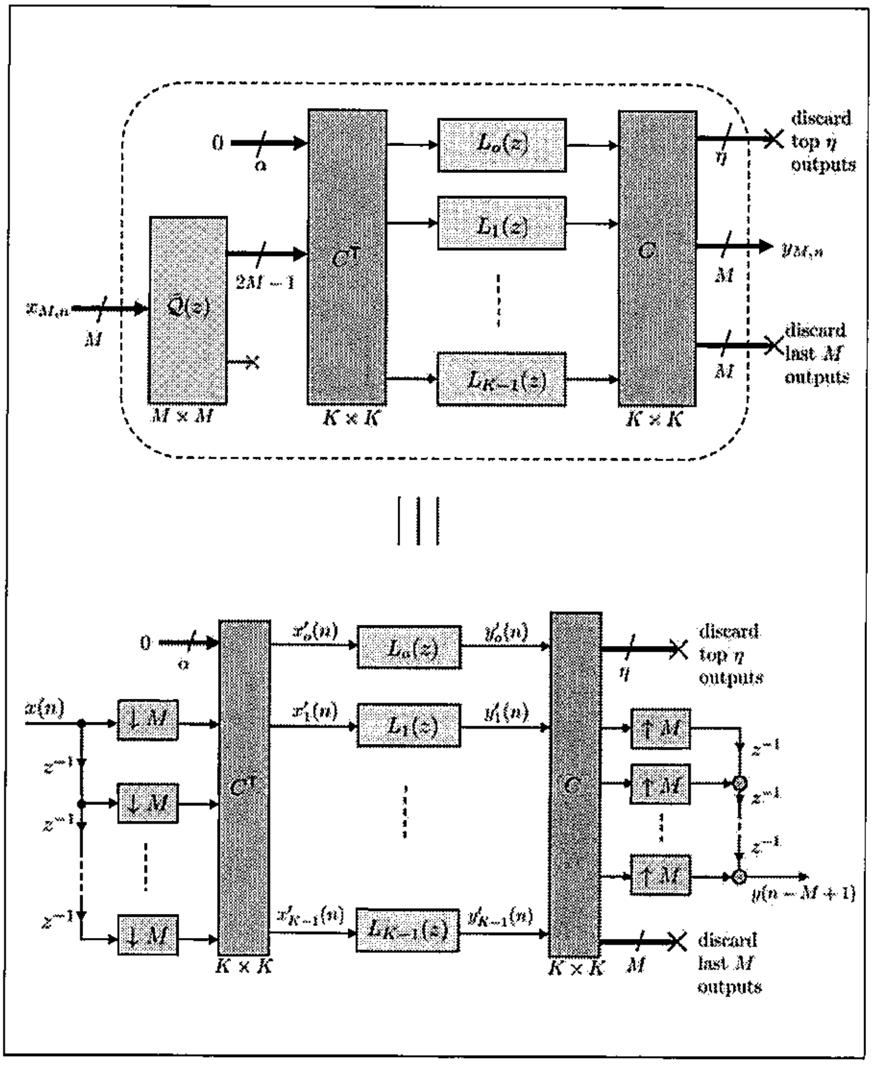

FIGURE 28.15 Equivalent implementation of the mapping shown earlier in Fig. 28.3 for the filter $H(z)$ in terms of the DCT operation and a bank of K subfilters, $L_k(z)$ (top figure) — the value of K is defined by (28.80) in terms of the block size. An equivalent realization in terms of downsamplers and upsamplers is shown in the bottom figure.

DCT-based implementation of a filter $H(z)$ of length N

```
Choose a block size $M$, where $N$ and $M$ are powers of 2 and $N/M$ is an integer.
Set $\{K, \alpha, \eta\}$ according to (28.80), (28.82), and (28.85).
Determine the $M$ polyphase components $E_k(z)$ of $H(z)$ of order $N/M$ from (28.10).
Use (28.87) to determine the $K$ subfilters $\{L_k(z)\}$ of size $N/M$ each.
Set $x_{M,-1} = 0$.
for $n \geq 0$ (block index):
    Construct the block vector $x_{M,n}$ as in (28.5) and set $x_{2M,n} = \text{col}\{x_{M,n}, x_{M,n-1}\}$.
    Perform the DCT transformation: $C^{\mathsf{T}} \begin{bmatrix} 0_{\alpha\times(2M-1)} & 0 \\ I_{2M-1} & 0 \end{bmatrix} x_{2M,n} \triangleq \text{col}\{x'_k(n)\}$
    Filter $\{x'_k(n)\}$ through $\{L_k(z)\}$ to generate $y'_k(n)$. Let $y'_{K,n} = \text{col}\{y'_k(n)\}$.
    Perform the DCT transformation $y_{M,n} = \begin{bmatrix} 0_{M\times\eta} & I_M & 0_{M\times 2M-2} \end{bmatrix} C y'_{K,n}$.
end for
```

28.5 DHT-BASED BLOCK FILTERING

We can use similar arguments to those in Sec. 28.4 to derive block filter implementations that rely on the discrete Hartley transform (DHT) as opposed to the DCT or the DFT. We do so by embedding $\mathcal{E}(z)$ into a matrix function that can be diagonalized by the DHT.

The DHT matrix of dimensions $K \times K$ is defined as follows (for convenience, we will use in this section the letter H to refer to the DHT matrix, which should not be confused with the transfer function $H(z)$ of the fullband filter):

$$\boxed{[H]_{km} = \frac{1}{\sqrt{K}}\left[\cos\left(\frac{2km\pi}{K}\right) - \sin\left(\frac{2km\pi}{K}\right)\right], \quad k,m = 0,1,\ldots,K-1} \tag{28.89}$$

The matrix H so defined can be related to the *unitary* DFT matrix of size $K \times K$. Recall from (28.32) that the entries of the DFT matrix of order K are given by

$$[F]_{km} \triangleq e^{-\frac{j2\pi km}{K}} \qquad k,m = 0,1,\ldots,K-1 \tag{28.90}$$

This matrix satisfies $FF^* = K \cdot I_K$. If we scale it by $1/\sqrt{K}$ and define (recall (17.112)):

$$\overline{F} \triangleq \frac{1}{\sqrt{K}} F \tag{28.91}$$

then $\overline{F}$ becomes a unitary matrix, i.e., it now satisfies $\overline{F}\,\overline{F}^* = I_K$. The entries of $\overline{F}$ are given by

$$\left[\overline{F}\right]_{km} = \frac{1}{\sqrt{K}} e^{-\frac{j2\pi km}{K}} = \frac{1}{\sqrt{K}}\left[\cos\left(\frac{2mk\pi}{K}\right) - \sin\left(\frac{2mk\pi}{K}\right)\right] \tag{28.92}$$

so that comparing with (28.89) we conclude that

$$H = \text{Re}(\overline{F}) + \text{Im}(\overline{F}) \tag{28.93}$$

in terms of the real and imaginary parts of $\overline{F}$. It can be verified that the Hartley matrix H is symmetric and orthogonal so that

$$HH^{\mathsf{T}} = H^2 = I_K \tag{28.94}$$

Now just like the DFT matrix diagonalizes circulant matrices, as was the case with (28.33), it is also known that the DHT matrix, H, diagonalizes any $K \times K$ symmetric circulant matrix of the following form for even values of K (which is the case of interest), e.g., for $K = 6$:

$$\mathcal{A}(z) = \begin{bmatrix} a_0(z) & a_1(z) & a_2(z) & 0 & a_2(z) & a_1(z) \\ a_1(z) & a_0(z) & a_1(z) & a_2(z) & 0 & a_2(z) \\ a_2(z) & a_1(z) & a_0(z) & a_1(z) & a_2(z) & 0 \\ 0 & a_2(z) & a_1(z) & a_0(z) & a_1(z) & a_2(z) \\ a_2(z) & 0 & a_2(z) & a_1(z) & a_0(z) & a_1(z) \\ a_1(z) & a_2(z) & 0 & a_2(z) & a_1(z) & a_0(z) \end{bmatrix} \tag{28.95}$$

Proceeding similarly to the DFT and DCT cases, consider the same matrix $\mathcal{E}(z)$ from (28.78) for $M = 2$ and let us embed it into a symmetric circulant matrix $\mathcal{A}(z)$ as follows:

$$\mathcal{A}(z) \triangleq \left[\begin{array}{c|ccc|cc} 0 & E_o(z) & E_1(z) & 0 & E_1(z) & E_o(z) \\ E_o(z) & 0 & E_o(z) & E_1(z) & 0 & E_1(z) \\ \cline{2-4} E_1(z) & E_o(z) & 0 & E_o(z) & E_1(z) & 0 \\ 0 & E_1(z) & E_o(z) & 0 & E_o(z) & E_1(z) \\ E_1(z) & 0 & E_1(z) & E_o(z) & 0 & E_o(z) \\ E_o(z) & E_1(z) & 0 & E_1(z) & E_o(z) & 0 \end{array}\right] \tag{28.96}$$

where the dimensions of $\mathcal{A}(z)$ are $3M \times 3M$ or 6×6 (so that $K = 3M = 6$); since M is chosen as a power of 2, then K will be even). We can recover $\mathcal{E}(z)$ from $\mathcal{A}(z)$, for a generic block size M, via the relation

$$\boxed{\mathcal{E}(z) = [I_M \;\; 0_{M\times 2M}]\mathcal{A}(z) \begin{bmatrix} 0_{1\times 2M-1} \\ I_{2M-1} \\ 0_{M\times 2M-1} \end{bmatrix}} \tag{28.97}$$

In general, for a generic value of M, the top row of $\mathcal{A}(z)$ will have the form

$$\begin{bmatrix} 0 & E_o(z) & \ldots & E_{M-1}(z) & 0_{1\times M-1} & E_{M-1}(z) & \ldots & E_o(z) \end{bmatrix} \tag{28.98}$$

with a single leading zero and $M - 1$ zeros in the middle. The matrix $\mathcal{A}(z)$ can be diagonalized by H, namely, it holds that

$$\mathcal{A}(z) = H\mathcal{L}(z)H \tag{28.99}$$

where H is $3M \times 3M$ and $\mathcal{L}(z)$ is a $3M \times 3M$ diagonal matrix function with subfilters $\{L_k(z)\}$. In a manner similar to (28.36) and (28.87), and using the orthogonality of H, it can be verified that the required $3M$ subfilters can be recovered from the polyphase components as follows:

$$\boxed{\left[\begin{array}{c} L_o(z) \\ L_1(z) \\ \vdots \\ L_{3M-1}(z) \end{array}\right] = H\left[\begin{array}{c} 0 \\ E_o(z) \\ \vdots \\ E_{M-1}(z) \\ 0_{M-1\times 1} \\ E_{M-1}(z) \\ \vdots \\ E_o(z) \end{array}\right]} \tag{28.100}$$

Observe, in particular, that the polyphase components $\{E_k(z)\}$ are repeated twice on the right-hand side.

We can return to the decomposition (28.24) and write it as

$$\begin{aligned} \mathcal{H}(z) &= \underbrace{[I_M \;\; 0_{M\times 2M}]\mathcal{A}(z)\left[\begin{array}{c} 0_{1\times 2M-1} \\ I_{2M-1} \\ 0_{M\times 2M-1} \end{array}\right]}_{\mathcal{E}(z)} \Omega(z) \\ &= [I_M \;\; 0_{M\times 2M}]\underbrace{H\mathcal{L}(z)H}_{\mathcal{A}(z)}\left[\begin{array}{c} 0_{1\times 2M-1} \\ I_{2M-1} \\ 0_{M\times 2M-1} \end{array}\right]\Omega(z) \qquad (28.101) \\ &= [I_M \;\; 0_{M\times 2M}]H\mathcal{L}(z)H\left[\begin{array}{cc} 0_{1\times 2M-1} & 0 \\ I_{2M-1} & 0 \\ 0_{M\times 2M-1} & 0 \end{array}\right]\bar{\Omega}(z) \end{aligned}$$

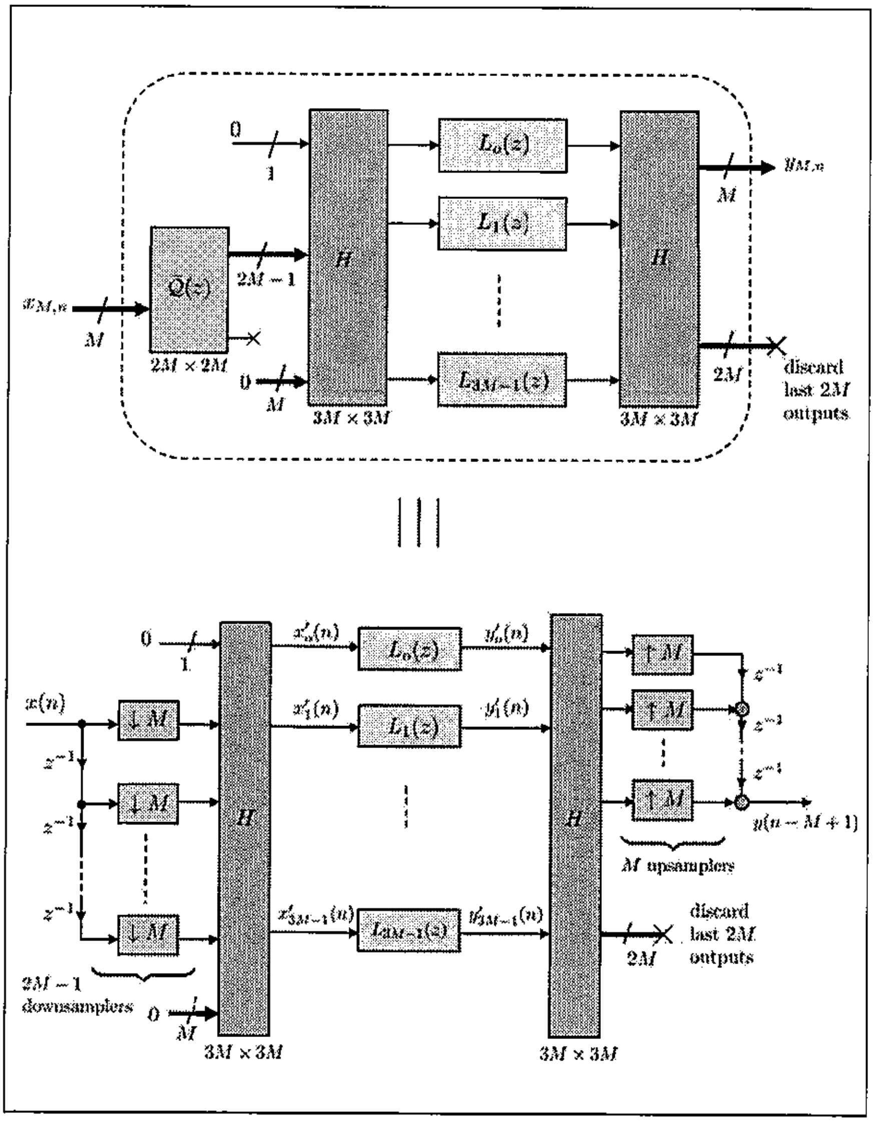

FIGURE 28.16 Equivalent implementation of the mapping shown earlier in Fig. 28.3 for the filter $H(z)$ in terms of the DHT operation H and a bank of $3M$ subfilters, $L_k(z)$ (top figure). An equivalent realization in terms of downsamplers and upsamplers (bottom figure).

where $\{\mathcal{E}(z), \mathcal{Q}(z), \bar{\mathcal{Q}}(z)\}$ are from (28.25)–(28.29). This result shows that the mapping from $x_{M,n}$ to $y_{M,n}$ in Fig. 28.3 can be alternatively implemented as shown in Fig. 28.16 — compare with Figs. 28.13, 28.9, and 28.15.

DHT-based implementation of a filter $H(z)$ of length N

Choose a block size M, where N and M are powers of 2 and N/M is an integer.
Set $K = 3M$.
Determine the M polyphase components $E_k(z)$ of $H(z)$ of order N/M from (28.10).
Use (28.100) to determine the K subfilters $\{L_k(z)\}$ of size N/M each.
Set $x_{M,-1} = 0$.
for $n \geq 0$ (block index):

Construct the block vector $x_{M,n}$ as in (28.5) and set $x_{2M,n} = \text{col}\{x_{M,n}, x_{M,n-1}\}$.

Perform the DHT transformation: $x'_{K,n} = H \begin{bmatrix} 0_{1\times 2M-1} & 0 \\ I_{2M-1} & 0 \\ 0_{M\times(2M-1)} & 0 \end{bmatrix} x_{2M,n} \triangleq \text{col}\{x'_k(n)\}$.

Filter $\{x'_k(n)\}$ through $\{L_k(z)\}$ to generate $y'_k(n)$. Let $y'_{K,n} = \text{col}\{y'_k(n)\}$.

Perform the DHT transformation $y_{M,n} = \begin{bmatrix} I_M & 0_{M\times 2M} \end{bmatrix} H y'_{K,n}$.

28.6 APPLICATION: CHANNEL INVERSION

In this section we illustrate the application of multirate techniques to the inversion of FIR channels. One of the key results will be the realization that block processing techniques enable the inversion of FIR channels by means of filters that are also FIR. This is a notable conclusion that helps simplify the implementation of channel equalizers to great extent. We motivate the main concept by considering a simple example initially.

A Simple Equalization Example

Assume data $s(n)$ are transmitted over the following first-order causal and stable IIR channel:

$$C(z) = \frac{1}{1 - \frac{1}{2}z^{-2}} \tag{28.102}$$

The sequence at the output of the channel is denoted by $y(n)$. The input-output samples are then related by the difference equation:

$$y(n) = \frac{1}{2}y(n-1) + s(n) \tag{28.103}$$

It is clear from this relation that each sample $y(n)$ contains a contribution from $s(n)$ and from the prior samples $\{s(m),\ m < n\}$ due to the channel memory, which is represented by the term $y(n-1)$. We say that intersymbol interference (ISI) occurs during data transmission over the channel. For this reason, given observations of the output samples $y(n)$, these observations will need to be subjected to some processing in order to remove the effect of ISI and recover $s(n)$. For the IIR channel described by (28.102), this step can accomplished by simply inverting the channel — see the top row in Fig. 28.17. Indeed, assume we feed the sequence $y(n)$ into the following FIR filter:

$$K(z) = \frac{1}{C(z)} = 1 - \frac{1}{2}z^{-1} \tag{28.104}$$

and let us denote the resulting sequence by $q(n)$. Then, obviously,

$$\begin{aligned} q(n) &= y(n) - \frac{1}{2}y(n-1) \\ &= s(n) \end{aligned} \tag{28.105}$$

and we achieve the objective of recovering $s(n)$. We refer to $K(z)$ as an equalizer since the transfer function from the input terminal, $s(n)$, to the output terminal, $q(n)$, is seen to be

$$C(z)K(z) = 1 \tag{28.106}$$

We thus say that $K(z)$ equalizes the channel. In this case, we are able to perfectly equalize the channel using an FIR filter. Furthermore, we are able to ensure that the overall transfer function of the channel-equalizer combination is unity. It is straightforward to recognize that this conclusion holds for any IIR channel that is minimum phase; its inverse will be minimum phase and serves as a perfect equalizer.

There are cases in which we are unable to equalize the channel perfectly but can ensure that the overall transfer function of the channel-equalizer combination is equal to some

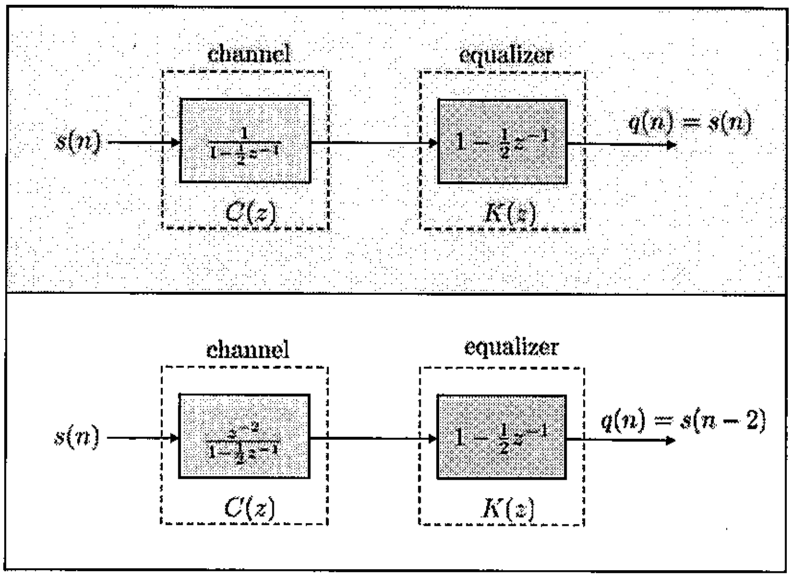

FIGURE 28.17 Equalizing a stable and causal IIR channel can be achieved by simply inverting it (as shown in the top part of the figure). The output sequence then satisfies $q(n) = s(n)$ and the transfer function of the combination channel-equalizer is equal to unity. In the bottom part of the figure, the direct inversion of the channel would result in a non-causal equalizer. If we instead employ $K(z)$, then its output will be $q(n) = s(n-2)$. In this case, we would still say that equalization occurs, albeit with a delay of two units of time.

pure delay. In these cases, we still say that perfect equalization occurs, albeit with some delay. For example, had we started from the following channel:

$$C'(z) = \frac{z^{-2}}{1-\frac{1}{2}z^{-2}} \tag{28.107}$$

and cascaded it with the same equalizer

$$K(z) = 1 - \frac{1}{2}z^{-2} \tag{28.108}$$

we would have obtained:

$$C'(z)K(z) \;=\; z^{-2} \tag{28.109}$$

In this way, when the input sequence to the channel is $s(n)$, the output sequence of the equalizer will be $q(n) = s(n-2)$. Clearly, for the channel $C'(z)$, it would not be convenient to use the equalizer that would result from direct inversion, namely,

$$K'(z) = z^2\left(1 - \frac{1}{2}z^{-2}\right) \tag{28.110}$$

since it will be a non-causal filter.

What about Noise?

In practice, the data that feeds into the equalizer is subject to noise (due to imperfections in measurements, thermal noise, and other sources of distortion). If we model this noise as additive at the output of the channel, then we are faced with the situation shown in Fig. 28.18, where the received noisy data are denoted by

$$r(n) = y(n) + v(n) \tag{28.111}$$

The design of the equalizer in this case now needs to correct for two distortion effects: (a) the ISI effect due to the channel memory and (b) the noise effect. Even when the channel is minimum phase, it is not true any longer that the "best" equalizer is the inverse of the channel. That is because the frequency response of this inverse transfer function may end up magnifying the effect of the noise when it passes through it (for example, the inverse frequency response may have peaks at frequencies when the noise is prevalent). We shall study the design of optimal (in the minimum mean-square-error sense) equalizers for such situations later in Chapter ?? when we discuss Wiener filtering — see Example 35.3. We shall further study the design of adaptive equalizers for similar situations later in Sec. ??. In the discussions in the current chapter, we shall ignore the presence of the noise component.

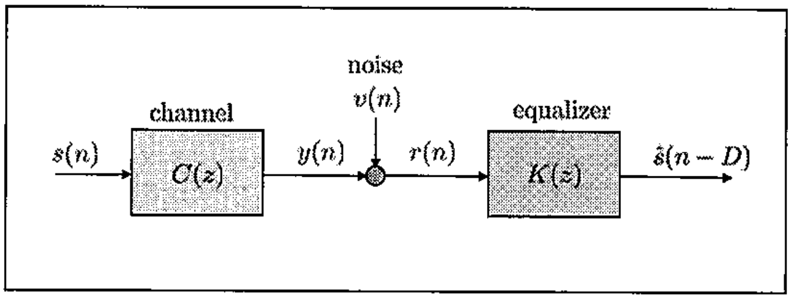

FIGURE 28.18 In practice, the received data is distorted by noise in addition to ISI. We shall study the design of optimal equalizers for such situations later in Chapters ?? and ?? on Wiener and adaptive filtering.

Equalizing an FIR Channel

Let us proceed to examine the situation in which the causal channel $C(z)$ happens to be FIR. This situation is common in practice, especially since digital and wireless communications channels are generally modeled as having finite-impulse response sequences. If we now simply invert $C(z)$ and set the equalizer to $K(z) = 1/C(z)$, at least two problematic issues come up. First, the equalizer $K(z)$ will be an IIR filter. And, second, unless all zeros of $C(z)$ are inside the unit circle, the equalizer, $K(z)$, will be an unstable filter. For these reasons, it is desirable to avoid direct inversion of the channel and to seek equalizers for $C(z)$ that are also of the FIR type. This is because FIR equalizers are always stable and they admit straightforward tapped-delay-line implementations. The challenge, however, is that it is generally impossible to perfectly equalize an FIR channel by using an FIR equalizer.

For example, starting from the FIR channel

$$C(z) = 1 - \frac{1}{2}z^{-1} \tag{28.112}$$

it is not possible to determine an FIR transfer function $K(z)$ such that

$$C(z)K(z) \;=\; z^{-d} \tag{28.113}$$

for any integer delay d. On the other hand, if we expand the inverse of the (causal and stable) channel, $C(z)$, we get:

$$\begin{aligned}\frac{1}{C(z)} &= \frac{1}{1-\frac{1}{2}z^{-1}} \\ &= \frac{1}{2}z^{-1} + \left(\frac{1}{2}\right)^2 z^{-2} + \left(\frac{1}{2}\right)^3 z^{-3} + \ldots \qquad (28.114)\end{aligned}$$

And since the factors $\left(\frac{1}{2}\right)^m$ decay exponentially fast with m, it is reasonable to assume that long enough FIR filters, $K(z)$, can still serve as reasonably good (though not perfect) equalizers for the channel.

Multirate Equalizers: DFT Filter Banks (OFDM)

It is therefore impossible to equalize FIR channels perfectly in the fullband domain by using FIR equalizers, as explained above. It turns out, however, that it is possible to *perfectly* equalize FIR channels by means of FIR equalizers if we resort instead to multirate (or block) processing implementations.

We illustrate some of the ideas by continuing with the simple example of a first-order FIR channel of the form (28.112), say, more generally,

$$\boxed{C(z) = c(0) + c(1)z^{-1}} \qquad (28.115)$$

for some coefficients $\{c(0), c(1)\}$; we will consider longer FIR filters in the sequel. We shall derive further ahead a general procedure for arriving at multirate structures that are able to equalize such FIR channels perfectly. For the time being, we are simply motivating the idea and showing that this is indeed possible for the first-order FIR channel. To do so, we return to the OFDM transceiver from Fig. 18.32, and to the corresponding interpretations in terms of synthesis and filter banks from Figs. 27.48 and 27.51. We assume noiseless measurements and reproduce the transmitter-receiver structure in Fig. 28.19 for the case of a two-tap channel of the form (28.115). The matrices F and F^* denote the 2×2 DFT and inverse DFT transformations, which happen to coincide in this case:

$$F = F^* = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad (28.116)$$

Moreover, the scalars λ_o and λ_1 are computed from the channel taps as follows:

$$\begin{bmatrix} \lambda_o & \\ & \lambda_1 \end{bmatrix} = \frac{1}{2}F \begin{bmatrix} c(0) & c(1) \\ c(1) & c(0) \end{bmatrix} F^* = \begin{bmatrix} c(0)+c(1) & \\ & c(0)-c(1) \end{bmatrix} \qquad (28.117)$$

so that

$$\begin{aligned}\lambda_o &= c(0) + c(1) \qquad (28.118)\\ \lambda_1 &= c(0) - c(1) \qquad (28.119)\end{aligned}$$

We already know from the discussions in Secs. 18.4 and 27.6 that, when the measurement noise is zero, the output signal is indeed $q(n) = s(n-1)$; this conclusion is obvious from relation (18.276) by setting the noise vector, v, to zero. It follows that the structure of Fig. 28.19 achieves perfect equalization and the overall mapping from the input terminal,

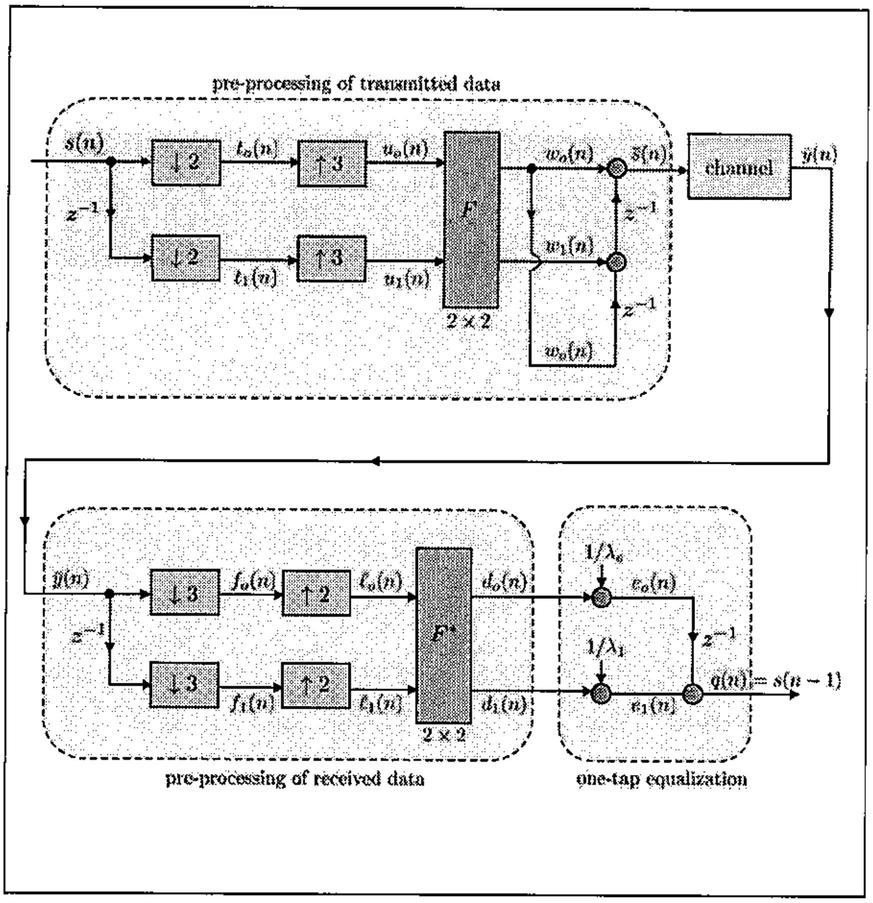

FIGURE 28.19 The data $s(n)$ is pre-processed by a multirate filter bank and transformed into $\bar{s}(n)$ before transmission over the channel. The received data $\bar{y}(n)$ is again pre-processed before being scaled by one-tap equalizers $1/\lambda_o$ and $1/\lambda_1$ to generate $s(n-1)$. In this example, an FIR filter is perfectly equalized with two one-tap equalizers (one for each branch; this is an example of equalization being performed in the subbands). In the figure we are assigning symbols to various internal signals for ease of reference.

$s(n)$, to the output terminal, $q(n)$, is the pure delay z^{-1}. In this structure, we say that the scaling coefficients $\{1/\lambda_o, 1/\lambda_1\}$ perform equalization in the subbands.

More generally, for longer FIR channels, we can use the OFDM transceiver structures of Figs. 27.48 and 27.51, which are combined together into Fig. 28.20. In the figure, $C(z)$ denotes a channel whose impulse response sequence has length M:

$$C(z) = c(0) + c(1)z^{-1} + \ldots + c(M-1)z^{-(M-1)} \tag{28.120}$$

Moreover, N is the chosen block size and $L = N + P$, where P is an integer satisfying $P \geq M - 1$. For the example discussed before in Fig. 28.17 we had $N = 2$, $M = 2$, $P = 1$, and $L = 3$. Moreover, the scalars $\{\lambda_k\}$ are defined via the transformation (written for $N = 4$ and $M = 3$, for simplicity):

$$\begin{bmatrix} \lambda_o & & & \\ & \lambda_1 & & \\ & & \lambda_2 & \\ & & & \lambda_3 \end{bmatrix} = \frac{1}{4}\cdot F \cdot \underbrace{\begin{bmatrix} c(0) & c(1) & c(2) & 0 \\ 0 & c(0) & c(1) & c(2) \\ c(2) & 0 & c(0) & c(1) \\ c(1) & c(2) & 0 & c(0) \end{bmatrix}}_{N\times N \text{ circular}} \cdot F^* \tag{28.121}$$

where F is now the 4×4 DFT matrix. The structure in Fig. 28.20 results in an overall transfer function that is equal to the pure delay z^{-P} so that the output signal is $s(n-P)$.

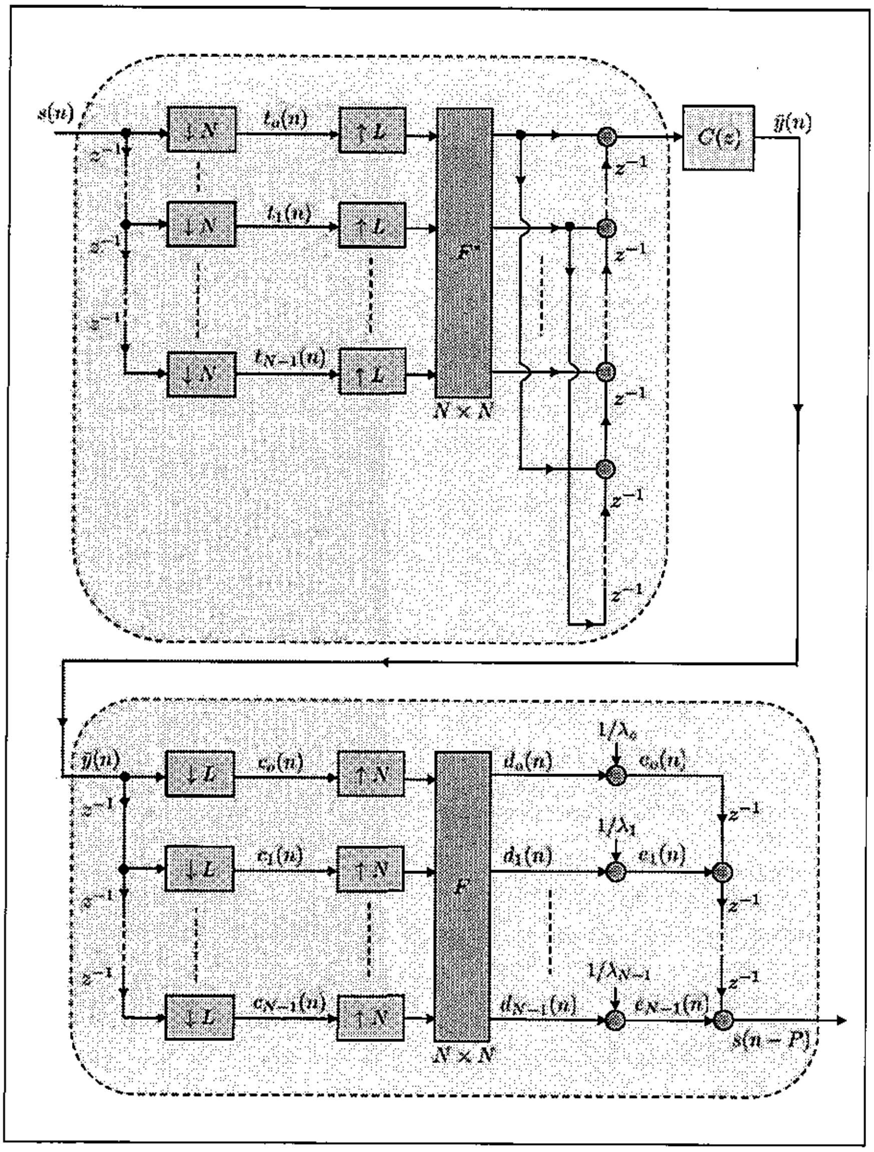

FIGURE 28.20 Perfect equalization of an FIR channel of length M. In the figure, N is the block size and $L = N + P$ where P satisfies $P \geq M - 1$. The overall transfer function is z^{-P}.

Multirate Equalizers: Transmultipexers

The perfect equalization structure of Fig. 28.20 is based on the use of efficient DFT filter banks at the transmitter and receiver sides. Equalization is achieved by scaling the signals generated by the DFT operation by the scalars $\{1/\lambda_k\}$. These scalars are related to the channel impulse response sequence, $c(n)$, through (28.121). For example, for the case $N = 2$, $M = 2$, and $L = 3$, the expressions for λ_o and λ_1 are shown in (28.118)–(28.119). Observe, in particular, that in this case the value of λ_1 can be zero, which occurs when $c(0) = c(1)$. This observation brings forth one of the deficiencies of the DFT-based solution: if the channel impulse-response sequence happens to lead to some zero scalar λ_m, then perfect equalization is not possible. One way to address this difficulty is to consider more general multirate equalization structures.

Thus, consider an FIR channel of length M as in (28.120), and refer to the filter bank structure shown in Fig. 28.21 where we assume $L \geq N + M$. As was the case with the DFT-based structure of Fig. 28.20, the input sequence $s(n)$ is first de-interleaved into blocks of size N. Subsequently, the block of data with entries $\{t_o(n), \ldots, t_{N-1}(n)\}$ is fed into a synthesis filter bank with subband filters denoted by $\{S_k(z)\}$. The outputs of these filters are combined together and the resulting sequence, $\bar{s}(n)$, is transmitted over the channel. The received data, $\bar{y}(n)$, are fed into an analysis filter bank with subband filters denoted by $\{A_k(z)\}$. The DFT-based structure of Fig. 28.20 will be shown later to be a special case of this more general structure when the subband filters $\{S_k(z)\}$ and $\{A_k(z)\}$ are chosen to be modulated versions of some prototype low-pass filters, $S_o(z)$ and $A_o(z)$, respectively. For now, we do not make this assumption and continue to consider the general case.

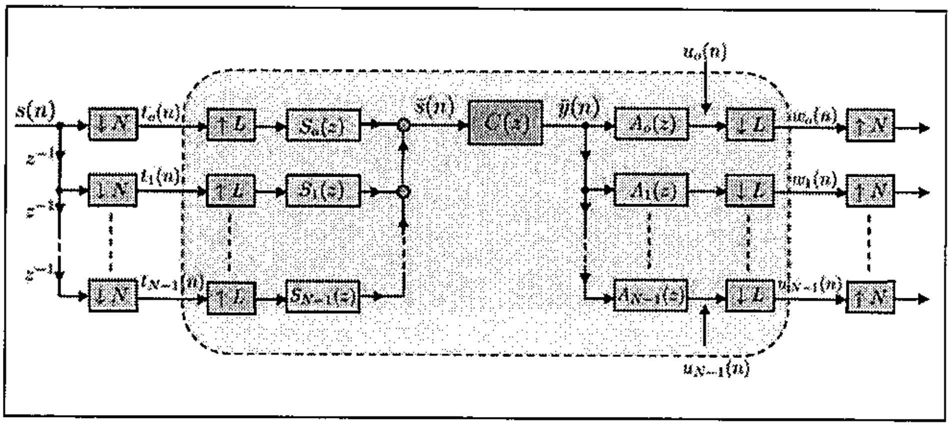

FIGURE 28.21 A transmultiplexer structure with synthesis subband filters $\{S_k(z)\}$ and analysis subband filters $\{A_k(z)\}$. The data stream $s(n)$ is de-interleaved into blocks of size N and processed by the synthesis filter bank at the transmitter before transmission over the channel. In the text, we explain how proper selection of the subband filters can achieve perfect equalization for FIR channels.

Before we establish that the structure of Fig. 28.21 is capable of perfect equalization, let us first determine the transfer matrix that maps the signals $\{t_k(n)\}$ to the signals $\{w_k(n)\}$. For perfect equalization to occur, we will need this transfer matrix to be diagonal. We shall show further ahead how to select the filter bank structure to result in a diagonal transfer matrix. Thus, let $P_{k,m}(z)$ denote the polyphase components of type-I and of order L of $S_k(z)$, and let $R_{k,m}(z)$ denote the polyphase components of type-II and of the same order

L of $A_\ell(z)$ so that:

$$\begin{aligned} S_k(z) &= \sum_{m=0}^{L-1} z^{-m} P_{k,m}(z^L), \qquad\qquad k = 0, 1, \ldots, N-1 && (28.122) \\ A_k(z) &= \sum_{m=0}^{L-1} z^{-(L-1-m)} R_{k,m}(z^L), \;\; k = 0, 1, \ldots, N-1 && (28.123) \end{aligned}$$

We can collect these definitions in vector form and write for all synthesis and analysis filters:

$$\begin{bmatrix} S_o(z) \\ S_1(z) \\ \vdots \\ S_{N-1}(z) \end{bmatrix} = \underbrace{\begin{bmatrix} P_{o,o}(z^L) & P_{o,1}(z^L) & \ldots & P_{o,L-1}(z^L) \\ P_{1,o}(z^L) & P_{1,1}(z^L) & \ldots & P_{1,L-1}(z^L) \\ \vdots & \vdots & & \vdots \\ P_{N-1,o}(z^L) & P_{N-1,1}(z^L) & \ldots & P_{N-1,L-1}(z^L) \end{bmatrix}}_{\triangleq\, \mathcal{P}(z^L)\ (N\times L)} \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(L-1)} \end{bmatrix} \tag{28.124}$$

and

$$\begin{bmatrix} A_o(z) \\ A_1(z) \\ \vdots \\ A_{N-1}(z) \end{bmatrix} = \underbrace{\begin{bmatrix} R_{o,L-1}(z^L) & \ldots & R_{o,1}(z^L) & R_{o,o}(z^L) \\ R_{1,L-1}(z^L) & \ldots & R_{1,1}(z^L) & R_{1,o}(z^L) \\ \vdots & \vdots & & \vdots \\ R_{N-1,L-1}(z^L) & \ldots & R_{N-1,1}(z^L) & R_{N-1,o}(z^L) \end{bmatrix}}_{\triangleq\, \mathcal{R}(z^L)\ (N\times L)} \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(L-1)} \end{bmatrix} \tag{28.125}$$

where we introduced the $N \times L$ matrix functions $\mathcal{P}(z^L)$ and $\mathcal{R}(z^L)$. Now, from Fig. 28.21 we have

$$\begin{aligned} \bar{S}(z) &= T_o(z^L)S_o(z) + T_1(z^L)S_1(z) + \ldots + T_{N-1}(z^L)S_{N-1}(z) \\ &= \begin{bmatrix} T_o(z^L) & T_1(z^L) & \ldots & T_{N-1}(z^L) \end{bmatrix} \cdot \mathcal{P}(z^L) \cdot \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(L-1)} \end{bmatrix} \end{aligned} \tag{28.126}$$

and

$$\begin{bmatrix} U_o(z) \\ U_1(z) \\ \vdots \\ U_{N-1}(z) \end{bmatrix} = \mathcal{R}(z^L) \cdot \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(L-1)} \end{bmatrix} \cdot \bar{Y}(z) \tag{28.127}$$

Then, the mapping from the $\{t_k(n)\}$ to the $\{w_k(n)\}$ can be represented in terms of the filter bank structure of Fig. 28.22.

To determine the transfer matrix of the overall mapping from the signals $\{t_k(n)\}$ to the signals $\{w_k(n)\}$, we examine first the portion illustrated in Fig. 28.23, which consists of

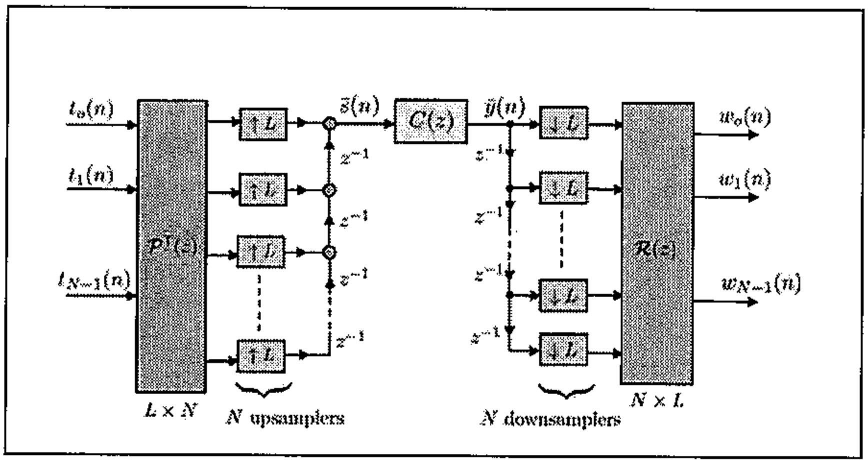

FIGURE 28.22 Filter bank representation of the mapping from the nodes $\{t_k(n)\}$ at the transmitter side to the nodes $\{w_k(n)\}$ at the receiver side in the transmultiplexer of Fig. 28.21.

the upsamplers, downsamplers, and the channel. This portion maps the signals $\{x_k(n)\}$ to the signals $\{y_k(n)\}$ and its transfer matrix can be determined as follows, where we will be employing the following notation for the L−th root of unity,

$$W_L \triangleq e^{-j\frac{2\pi}{L}} \tag{28.128}$$

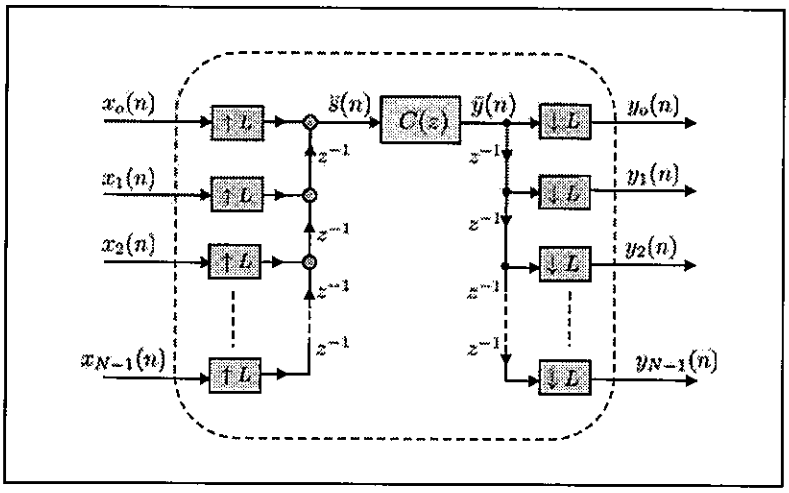

FIGURE 28.23 A portion of the filter bank of Fig. 28.22 consisting of the upsamplers, downsamplers, and the channel.

Note that

$$\bar{S}(z) = \sum_{k=0}^{L-1} z^{-k} X_k(z^L) \tag{28.129}$$

so that

$$\bar{Y}(z) \;=\; C(z) \cdot \left[X_o(z^L) + z^{-1}X_1(z^L) + \ldots + z^{-(L-1)}X_{L-1}(z^L)\right] \tag{28.130}$$

Then,

$$\begin{aligned} Y_o(z) &= \frac{1}{L}\sum_{k=0}^{L-1}\bar{Y}\left(W_L^k z^{1/L}\right) \\ &= \frac{1}{L}\sum_{k=0}^{L-1}\left[C\left(W_L^k z^{1/L}\right)\cdot\sum_{\ell=0}^{L-1}\left(W_L^k z^{1/L}\right)^{-\ell}X_\ell(z)\right] \\ &= \frac{1}{L}\sum_{\ell=0}^{L-1}\left(\sum_{k=0}^{L-1}\left(W_L^k z^{1/L}\right)^{-\ell}C\left(W_L^k z^{1/L}\right)\right)\cdot X_\ell(z) \qquad (28.131) \end{aligned}$$

For example, for the case $N = 2$ and $L = 3$ we get

$$\bar{Y}(z) \;=\; C(z)X_o(z^3) \;+\; z^{-1}C(z)X_1(z^3) \;+\; z^{-2}C(z)X_2(z^3) \qquad (28.132)$$

and

$$\begin{aligned} Y_o(z) \;=\; & \frac{1}{3}\left[C(z^{1/3}) + C(W_3z^{1/3}) + C(W_3^2z^{1/3})\right]X_o(z) \;+ \qquad (28.133) \\ & \frac{1}{3}\left[z^{-1/3}C(z^{1/3}) + \left(W_3z^{1/3}\right)^{-1}C(W_3z^{1/3}) + \left(W_3^2z^{1/3}\right)^{-1}C(W_3^2z^{1/3})\right]X_1(z) \\ & \frac{1}{3}\left[z^{-2/3}C(z^{1/3}) + \left(W_3z^{1/3}\right)^{-2}C(W_3z^{1/3}) + \left(W_3^2z^{1/3}\right)^{-2}C(W_3^2z^{1/3})\right]X_2(z) \end{aligned}$$

The terms that appear multiplying $\{X_o(z), X_1(z), X_2(z)\}$ in this example and, more generally, in (28.131), admit a useful interpretation. Specifically, recall that the sequence $\{c(n)\}$ denotes the impulse response sequence of the channel, $C(z)$. Let $C_o(z)$ denote the transfer function whose impulse response sequence is obtained by downsampling $c(n)$ by a factor L — see Fig. 28.24. Let also $C_\ell(z)$ denote the transfer function whose impulse response sequence is obtained by first delaying $c(n)$ by $z^{-\ell}$ and then downsampling the result by a factor L. The relation between $C_\ell(z)$ and $C(z)$ is given by

$$C_\ell(z) \;=\; \frac{1}{L}\sum_{k=0}^{L-1}\left(W_L^k z^{1/L}\right)^{-\ell}\cdot C\left(W_L^k z^{1/L}\right) \qquad (28.134)$$

Using the newly introduced transfer functions $\{C_\ell(z)\}$, we can rewrite expression (28.133) more compactly as

$$Y_o(z) = C_o(z)X_o(z) + C_1(z)X_1(z) + C_2(z) \qquad (28.135)$$

Likewise, expression (28.131) becomes

$$\begin{aligned} Y_o(z) &= \sum_{\ell=0}^{L-1}C_\ell(z)X_\ell(z) \\ &= C_o(z)X_o(z) + C_1(z)X_1(z) + \ldots + C_{L-1}(z)X_{L-1}(z) \qquad (28.136) \end{aligned}$$

impulse response sequence	transfer function
$c(n)$	$C(z)$
$c(n) \longrightarrow \boxed{\downarrow L} \longrightarrow c_o(n)$	$C_o(z) = \frac{1}{L}\sum_{k=0}^{L-1} C\left(W_L^k z^{1/L}\right)$
$c(n) \xrightarrow{z^{-1}} \boxed{\downarrow L} \longrightarrow c_1(n)$	$C_1(z) = \frac{1}{L}\sum_{k=0}^{L-1} \left(W_L^k z^{1/L}\right)^{-1} C\left(W_L^k z^{1/L}\right)$
$c(n) \xrightarrow{z^{-2}} \boxed{\downarrow L} \longrightarrow c_2(n)$	$C_2(z) = \frac{1}{L}\sum_{k=0}^{L-1} \left(W_L^k z^{1/L}\right)^{-2} C\left(W_L^k z^{1/L}\right)$

FIGURE 28.24 The impulse response sequence $c(n)$, along with downsampled sequences of delayed versions of it. The corresponding transfer functions are denoted by $C_k(z)$.

In a similar manner, we can find that the other output nodes $\{y_k(n)\}$ in Fig. 28.23 can be similarly described as follows:

$$\begin{aligned}
Y_1(z) &= C_1(z)X_o(z) + C_2(z)X_1(z) + \ldots + C_L(z)X_{L-1}(z) && (28.137)\\
Y_2(z) &= C_2(z)X_o(z) + C_3(z)X_1(z) + \ldots + C_{L+1}(z)X_{L-1}(z) && (28.138)\\
\vdots &= \vdots && (28.139)\\
Y_{L-1}(z) &= C_{L-1}(z)X_o(z) + C_L(z)X_1(z) + \ldots + C_{2L-2}(z)X_{L-1}(z) && (28.140)
\end{aligned}$$

In matrix form, we conclude that the mapping from the nodes $\{x_k(n)\}$ to the nodes $\{y_k(n)\}$ in Fig. 28.23 is given by

$$\begin{bmatrix} Y_o(z) \\ Y_1(z) \\ \vdots \\ Y_{L-1}(z) \end{bmatrix} = \underbrace{\begin{bmatrix} C_o(z) & C_1(z) & \ldots & C_{L-1}(z) \\ C_1(z) & \cdot & \cdot & C_L(z) \\ \vdots & \cdot & \cdot & \vdots \\ C_{L-1}(z) & C_L(z) & \ldots & C_{2L-2}(z) \end{bmatrix}}_{\triangleq\ \mathcal{C}(z)\ (L\times L)} \begin{bmatrix} X_o(z) \\ X_1(z) \\ \vdots \\ X_{L-1}(z) \end{bmatrix} \qquad (28.141)$$

where the $L \times L$ matrix $\mathcal{C}(z)$ has a Hankel structure; i.e., the entries on its anti-diagonals are equal.

Let us illustrate this construction by means of an example. Assume the channel $C(z)$ has length $M = 2$ and select $L = 3$. Then, the samples of the impulse response sequences

$\{c(n), c_o(n), c_1(n), c_2(n), c_3(n), c_4(n)\}$ are given by

$$\begin{aligned} c(n) &= \{c(0), c(1), 0, 0, 0, 0, 0, 0, \ldots\} && (28.142a) \\ c_o(n) &= \{c(0), 0, 0, 0, 0, 0, 0, 0, \ldots\} && (28.142b) \\ c_1(n) &= \{0, 0, 0, 0, 0, 0, 0, 0, \ldots\} && (28.142c) \\ c_2(n) &= \{0, c(1), 0, 0, 0, 0, 0, 0, \ldots\} && (28.142d) \\ c_3(n) &= \{0, c(0), 0, 0, 0, 0, 0, 0, \ldots\} && (28.142e) \\ c_4(n) &= \{0, 0, 0, 0, 0, 0, 0, 0, \ldots\} && (28.142f) \end{aligned}$$

and the corresponding transfer functions are

$$\begin{aligned} C(z) &= c(0) + c(1)z^{-1} && (28.143a) \\ C_o(z) &= c(0) && (28.143b) \\ C_1(z) &= 0 && (28.143c) \\ C_2(z) &= c(1)z^{-1} && (28.143d) \\ C_3(z) &= c(0)z^{-1} && (28.143e) \\ C_4(z) &= 0 && (28.143f) \end{aligned}$$

so that

$$\mathcal{C}(z) = \begin{bmatrix} c(0) & 0 & c(1)z^{-1} \\ 0 & c(1)z^{-1} & c(0)z^{-1} \\ c(1)z^{-1} & c(0)z^{-1} & 0 \end{bmatrix} \tag{28.144}$$

More generally, for longer channels (say, $M = 5$ and $L = 8$ for illustration purposes), the matrix $\mathcal{C}(z)$ takes the form:

$$\mathcal{C}(z) = z^{-1}\cdot \begin{array}{c} \left[\begin{array}{cccc|cccc} \boxed{c(0)z} & 0 & 0 & 0 & c(4) & c(3) & c(2) & c(1) \\ \hline 0 & 0 & 0 & c(4) & c(3) & c(2) & c(1) & c(0) \\ 0 & 0 & c(4) & c(3) & c(2) & c(1) & c(0) & 0 \\ 0 & c(4) & c(3) & c(2) & c(1) & c(0) & 0 & 0 \\ c(4) & c(3) & c(2) & c(1) & c(0) & 0 & 0 & 0 \\ \hline c(3) & c(2) & c(1) & c(0) & 0 & 0 & 0 & 0 \\ c(2) & c(1) & c(0) & 0 & 0 & 0 & 0 & 0 \\ c(1) & c(0) & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \\ \begin{array}{cc} L\times(L-M+1) \qquad & \qquad L\times(M-1) \end{array} \end{array} \begin{array}{l} 1\times L \\ \\ \\ (L-M+1)\times L \\ \\ \\ (M-2)\times L \\ \\ \end{array} \tag{28.145}$$

where we extracted z^{-1} and multiplied $c(0)$ by z. After extraction of z^{-1}, only the first row of $\mathcal{C}(z)$ contains an entry that depends on z. Observe further that the $L - M + 1$ rows of $\mathcal{C}(z)$ that appear between the horizontal lines have constant entries along their anti-diagonals with all entries of the first row also appearing in the last row. We shall exploit these properties to achieve perfect equalization.

Returning to Fig. 28.22 we conclude that the overall $N \times N$ transfer matrix from the nodes $\{t_k(n)\}$ to the nodes $\{w_k(n)\}$ is given by

$$\boxed{\mathcal{T}(z) = \mathcal{R}(z)\cdot\mathcal{C}(z)\cdot\mathcal{P}^{\mathsf{T}}(z)} \qquad (N\times N) \tag{28.146}$$

Perfect equalization of the channel $C(z)$ with one-tap equalization will be possible if the matrix $\mathcal{T}(z)$ turns out to be of the form:

$$\mathcal{T}(z) = z^{-d} \cdot \mathcal{D} \tag{28.147}$$

for some integer delay d and nonzero diagonal matrix D with entries denoted by $\{\lambda_k\}$. When this happens, the equalization scalars can be taken as the inverses of the $\{\lambda_k\}$. We illustrate how this can be achieved for the example (28.145) involving $M = 5$, $L = 8$. The $(k, m)-$th entry of $\mathcal{T}(z)$ denotes the transfer function from the $k-$th node, $t_k(n)$, to the $m-$the node, $w_m(n)$. It is given by

$$\begin{aligned} \mathcal{T}_{km}(z) &= \begin{bmatrix} R_{k,7}(z) & \ldots & R_{k,2}(z) & R_{k,1}(z) & R_{k,o}(z) \end{bmatrix} \cdot \mathcal{C}(z) \cdot \begin{bmatrix} P_{m,o}(z) \\ P_{m,1}(z) \\ \vdots \\ P_{m,7}(z) \end{bmatrix} \\ & k, m = 0, 1, \ldots, N-1 \end{aligned} \tag{28.148}$$

Now assume we set the first polyphase component, $R_{k,7}(z)$, and the last $M - 2$ polyphase components, $\{R_{k,2}(z), R_{k,1}(z), R_{k,o}(z)\}$, to zero:

$$R_{k,7}(z) \;=\; R_{k,2}(z) \;=\; R_{k,1}(z) \;=\; R_{k,o}(z) \;=\; 0 \tag{28.149}$$

then the expression for $T_{km}(z)$ simplifies to

$$\begin{aligned} \mathcal{T}_{km}(z) &= z^{-1} \cdot \begin{bmatrix} R_{k,6}(z) & R_{k,5}(z) & R_{k,4}(z) & R_{k,3}(z) \end{bmatrix} \cdot \\ & \left[\begin{array}{cccc|cccc} 0 & 0 & 0 & c(4) & c(3) & c(2) & c(1) & c(0) \\ 0 & 0 & c(4) & c(3) & c(2) & c(1) & c(0) & 0 \\ 0 & c(4) & c(3) & c(2) & c(1) & c(0) & 0 & 0 \\ c(4) & c(3) & c(2) & c(1) & c(0) & 0 & 0 & 0 \end{array} \right] \cdot \\ & \begin{bmatrix} P_{m,o}(z) \\ P_{m,1}(z) \\ \vdots \\ P_{m,7}(z) \end{bmatrix} \end{aligned} \tag{28.150}$$

To proceed, we select (for now) an arbitrary number ρ_m and assume the synthesis filter $S_m(z)$ has order $L - 1$ and is chosen to be of the following form:

$$S_m(z) \;=\; 1 + \rho_m z^{-1} + \rho_m^2 z^{-2} + \rho_m^3 z^{-3} + \rho_m^4 z^{-4} + \rho_m^5 z^{-5} + \rho_m^6 z^{-6} + \rho_m^7 z^{-7} \tag{28.151}$$

In other words, the coefficients of $S_m(z)$ are powers of ρ_m. Then, the $L-$th order polyphase components of $S_m(z)$ of type-I are scalars and given by:

$$\begin{aligned} P_{m,o}(z) &= 1, & P_{m,4}(z) &= \rho_m^4 & \text{(28.152a)} \\ P_{m,1}(z) &= \rho_m, & P_{m,5}(z) &= \rho_m^5 & \text{(28.152b)} \\ P_{m,2}(z) &= \rho_m^2, & P_{m,6}(z) &= \rho_m^6 & \text{(28.152c)} \\ P_{m,3}(z) &= \rho_m^3, & P_{m,7}(z) &= \rho_m^7 & \text{(28.152d)} \end{aligned}$$

Assume further that analysis filter $A_k(z)$ has order $L - M + 1$ and is selected to be of the following form:

$$A_k(z) = a_k(1)z^{-1} + a_k(2)z^{-2} + a_k(3)z^{-3} + a_k(4)z^{-4} \tag{28.153}$$

with impulse response sequence $\{a_k(n)\}$. Then, the $L-$th order polyphase components of $A_k(z)$ of type-II are scalars and given by:

$$\begin{aligned}
R_{k,7}(z) &= 0, & R_{k,3}(z) &= a_k(4) && \text{(28.154a)}\\
R_{k,6}(z) &= a_k(1), & R_{k,2}(z) &= 0 && \text{(28.154b)}\\
R_{k,5}(z) &= a_k(2), & R_{k,1}(z) &= 0 && \text{(28.154c)}\\
R_{k,4}(z) &= a_k(3), & R_{k,o}(z) &= 0 && \text{(28.154d)}
\end{aligned}$$

Substituting into (28.150) gives

$$\begin{aligned}
\mathcal{T}_{km}(z) &= z^{-1}\cdot\left[\begin{array}{cccc} a_k(1) & a_k(2) & a_k(3) & a_k(4)\end{array}\right]\cdot\\
&\quad\left[\begin{array}{cccc|cccc}
0 & 0 & 0 & c(4) & c(3) & c(2) & c(1) & c(0)\\
0 & 0 & c(4) & c(3) & c(2) & c(1) & c(0) & 0\\
0 & c(4) & c(3) & c(2) & c(1) & c(0) & 0 & 0\\
c(4) & c(3) & c(2) & c(1) & c(0) & 0 & 0 & 0
\end{array}\right]\cdot\\
&\quad\left[\begin{array}{c} 1\\ \rho_m\\ \vdots\\ \rho_m^7\end{array}\right]\\
&= z^{-1}\cdot\left[\begin{array}{cccc} a_k(1) & a_k(2) & a_k(3) & a_k(4)\end{array}\right]\cdot C(\rho_m)\cdot\left[\begin{array}{c}\rho_m^7\\ \rho_m^6\\ \rho_m^5\\ \rho_m^4\end{array}\right]\\
&= z^{-1}\cdot\rho_m^8\cdot C(\rho_m)\cdot A_k(\rho_m) \qquad\qquad \text{(28.155)}
\end{aligned}$$

where $C(z)$ denotes the channel transfer function (28.120), which for $M=5$ is given by

$$C(z) = c(0) + c(1)z^{-1} + c(2)z^{-2} + c(3)z^{-3} + c(4)z^{-4} \qquad \text{(28.156)}$$

More generally, for other values of M and L, the above construction would lead to

$$\boxed{\mathcal{T}_{km}(z) \;=\; z^{-1}\cdot\rho_m^L\cdot C(\rho_m)\cdot A_k(\rho_m)} \qquad \text{(28.157)}$$

where

$$C(z) = \sum_{k=0}^{M-1} c(k)z^{-k}, \qquad A_k(z) = \sum_{m=1}^{L-M+1} a_k(m)z^{-m} \qquad \text{(28.158)}$$

Now since an $M-$th order FIR channel $C(z)$ has a a finite number of zeros (a total of M zeros), we can always select N distinct scalars ρ_m, one for each synthesis filter $S_m(z)$, such that $C(\rho_m)\neq 0$. Thus, assume that the N scalars $\{\rho_m\}$ are selected to satisfy the following conditions, for $m=0,1,\ldots,N-1$:

$$\begin{aligned}
C(\rho_m) &\neq 0 && \text{(28.159)}\\
A_k(\rho_m) &= 0 \text{ for } k\neq m, \text{ and } A_k(\rho_k)=1,\; k=0,1,\ldots,N-1 && \text{(28.160)}
\end{aligned}$$

The second condition imposes N constraints on the coefficients of $A_k(z)$: the analysis filter should evaluate to one at the corresponding ρ_k and to zero at the remaining $\{\rho_m,\ m\neq k\}$. In other words, the remaining $\{\rho_m\}$ should be zeros for $A_k(z)$. Now recall that $A_k(z)$ has $L-M+1$ coefficients $\{a_k(m)\}$ to be selected, which correspond to $L-M+1$ degrees of freedom. Assuming $L - M + 1 \geq N$, it is possible to select the $\{\rho_m\}$ to satisfy the second condition (28.160). We therefore need the filter bank parameters to satisfy the requirement

$$\boxed{L \geq N + M - 1} \tag{28.161}$$

Let us introduce the scalars:

$$\boxed{\lambda_m \stackrel{\Delta}{=} \rho_m^L \cdot C(\rho_m)} \quad (m = 0, 1, \ldots, N-1) \tag{28.162}$$

When the scalars $\{\rho_m\}$ are selected according to (28.159)–(28.160), then it will hold that

$$\mathcal{T}_{km}(z) = z^{-1} \cdot \lambda_m \cdot \delta_{km} \tag{28.163}$$

where $\delta_{km} = 1$ when $k = m$ and is zero otherwise. It follows that the overall transfer matrix from the $\{t_k(n)\}$ to the $\{w_k(n)\}$ becomes

$$\mathcal{T}(z) \;=\; z^{-1} \cdot \begin{bmatrix} \lambda_o & & & \\ & \lambda_1 & & \\ & & \ddots & \\ & & & \lambda_{N-1} \end{bmatrix} \tag{28.164}$$

and the channel can be perfectly equalized by scaling each $w_k(n)$ by $1/\lambda_k$. This situation is illustrated in Fig. 28.25 for the case $L = 8$, $M = 5$, and $N = 3$, in which case the matrices $\mathcal{P}^{\mathsf{T}}(z)$ and $\mathcal{R}(z)$ are given by

$$\mathcal{P}^{\mathsf{T}}(z) = \begin{bmatrix} 1 & 1 & 1 \\ \rho_o & \rho_1 & \rho_2 \\ \rho_o^2 & \rho_1^2 & \rho_2^2 \\ \rho_o^3 & \rho_1^3 & \rho_2^3 \\ \rho_o^4 & \rho_1^4 & \rho_2^4 \\ \rho_o^5 & \rho_1^5 & \rho_2^5 \\ \rho_o^6 & \rho_1^6 & \rho_2^6 \\ \rho_o^7 & \rho_1^7 & \rho_2^7 \end{bmatrix}, \quad \mathcal{R}(z) = \begin{bmatrix} 0 & a_o(1) & a_o(2) & a_o(3) & a_o(4) & 0 & 0 & 0 \\ 0 & a_1(1) & a_1(2) & a_1(3) & a_1(4) & 0 & 0 & 0 \\ 0 & a_2(1) & a_2(2) & a_2(3) & a_2(4) & 0 & 0 & 0 \end{bmatrix} \tag{28.165}$$

Special Case of DFT Filter Banks

We now explain how the earlier DFT-based transmultiplexer structure can be recovered as a special case of Fig. 28.25. Thus, assume the scalars ρ_m are chosen as follows:

$$\rho_m = W_N^{-m}, \;\; m = 0, 1, \ldots, N-1 \tag{28.166}$$

For $N = 3$, we have $\rho_m = W_3^{-m}$. Since the factor W_3^{-m} is periodic of period $N = 3$, the matrix $\mathcal{P}^{\mathsf{T}}(z)$ becomes constant (independent of z) and given by:

$$\mathcal{P}^{\mathsf{T}} \;=\; \left[\begin{array}{ccc} 1 & 1 & 1 \\ 1 & W_3^{-1} & W_3^{-2} \\ 1 & W_3^{-2} & W_3^{-4} \\ \hline 1 & 1 & 1 \\ 1 & W_3^{-1} & W_3^{-2} \\ 1 & W_3^{-2} & W_3^{-4} \\ \hline 1 & 1 & 1 \\ 1 & W_3^{-1} & W_3^{-2} \end{array}\right] \tag{28.167}$$

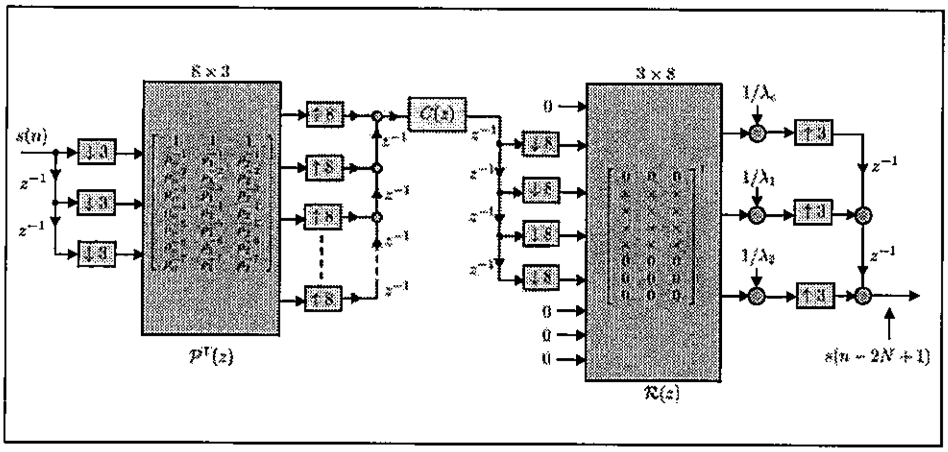

FIGURE 28.25 Perfect equalization of an FIR channel $C(z)$ of length $M = 5$ taps using $L = 8$ (upsamping factor) and $N = 3$ (downsampling factor). The scalars λ_m are defined by (28.162) and the scalars ρ_m are selected according to (28.159)–(28.160) with $L \geq N + M - 1$. The resulting overall delay from the input terminal, $s(n)$, to the rightmost output terminal is $2N - 1$.

with the first three rows repeated in the subsequent rows. Now referring to the top part of Fig. 28.26, where we are denoting the signals before and after the transformation by $\mathcal{P}^\mathsf{T}$ by $\{t_k(n), u_k(n)\}$ we can write

$$\begin{bmatrix} u_o(n) \\ u_1(n) \\ u_2(n) \\ u_3(n) \\ u_4(n) \\ u_5(n) \\ u_6(n) \\ u_7(n) \end{bmatrix} = \left[\begin{array}{ccc} 1 & 1 & 1 \\ 1 & W_3^{-1} & W_3^{-2} \\ 1 & W_3^{-2} & W_3^{-4} \\ \hline 1 & 1 & 1 \\ 1 & W_3^{-1} & W_3^{-2} \\ 1 & W_3^{-2} & W_3^{-4} \\ \hline 1 & 1 & 1 \\ 1 & W_3^{-1} & W_3^{-2} \end{array}\right] \begin{bmatrix} t_o(n) \\ t_1(n) \\ t_2(n) \end{bmatrix} \tag{28.168}$$

so that

$$\begin{bmatrix} u_o(n) \\ u_1(n) \\ u_2(n) \end{bmatrix} = F_3^* \begin{bmatrix} t_o(n) \\ t_1(n) \\ t_2(n) \end{bmatrix} \tag{28.169}$$

and

$$\begin{bmatrix} u_3(n) \\ u_4(n) \\ u_5(n) \\ u_6(n) \\ u_7(n) \end{bmatrix} = \begin{bmatrix} u_o(n) \\ u_1(n) \\ u_2(n) \\ u_o(n) \\ u_1(n) \end{bmatrix} \tag{28.170}$$

These relations imply that the top part of Fig. 28.26 is equivalent to the bottom part of the same figure, where we placed the upsamplers before the inverse DFT operation; the matrix F_3^* denotes the 3×3 inverse DFT matrix. This structure coincides with the transmitter structure shown in Fig. 28.20.

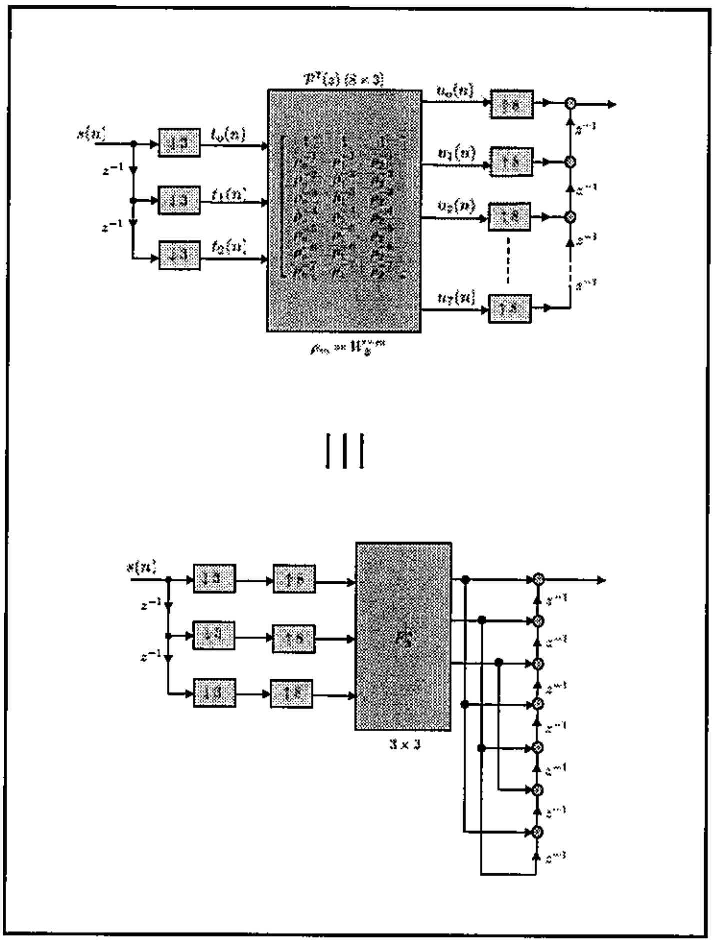

FIGURE 28.26 The transmitter structure on top reduces to the DFT structure in the bottom when ρ_m is chosen as $\rho_m = W_N^{-m}$. The bottom structure coincides with the transmitter structure introduced earlier in Fig. 28.20.

Let us further assume that the coefficients $\{a_k(n)\}$ are chosen as

$$a_k(n) = \frac{1}{N}W_N^{-k(n-1)}, \quad k = 0, 1, \ldots, N-1, \; n = 1, 2, \ldots, L-M+1 \qquad (28.171)$$

and zero elsewhere. For example, when $N = 3$, we set $a_k(n) = \frac{1}{3}W_3^{-k(n-1)}$. Again, since the factor W_3^ℓ is periodic of period $N = 3$, the matrix $\mathfrak{R}(z)$ becomes

$$\mathfrak{R} = \frac{1}{3}\left[\begin{array}{c|ccc|cccc} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & W_3^{-1} & W_3^{-2} & 0 & 0 & 0 & 0 \\ 0 & 1 & W_3^{-2} & W_3^{-4} & 0 & 0 & 0 & 0 \end{array}\right] \qquad (28.172)$$

That is, the analysis subband filters in this case are:

$$\begin{aligned}
A_o(z) &= \frac{1}{3}z^{-1}\left[1+z^{-1}+z^{-2}\right] && (28.173)\\
A_1(z) &= \frac{1}{3}z^{-1}\left[1+W_3^{-1}z^{-1}+W_3^{-2}z^{-2}\right] && (28.174)\\
A_2(z) &= \frac{1}{3}z^{-1}\left[1+W_3^{-2}z^{-1}+W_3^{-4}z^{-2}\right] && (28.175)
\end{aligned}$$

It is straightforward to verify from the fact that $F_3^*F_3 = F_3F_3^* = I$ that conditions (28.160) are satisfied, i.e.,

$$\begin{aligned}
&A_o(1)=1, \quad A_o(W_3^{-1})=0, \quad A_o(W_3^{-2})=0 && (28.176)\\
&A_1(1)=0, \quad A_1(W_3^{-1})=1, \quad A_1(W_3^{-2})=0 && (28.177)\\
&A_2(1)=0, \quad A_2(W_3^{-1})=0, \quad A_2(W_3^{-2})=1 && (28.178)
\end{aligned}$$

These relations imply that the top part of Fig. 28.27 is equivalent to the bottom part of the same figure, where we placed the upsamplers before the DFT operation. This structure coincides (CHECK????) with the receiver structure shown in Fig. 28.20.

Practice Questions:

1. Show that it is not possible to determine an FIR equalizer $K(z)$ to satisfy (28.113) no matter what the value of the integer delay d is.
2. Refer to Fig. 28.19 and show, by working with the z−transform representation of the various signals in the figure, that $Q(z) = z^{-1}S(z)$.
3. Show that the overall transfer function in Fig. 28.20 is equal to z^{-P}.
4. Establish (28.146).
5. Verify that the output signal in Fig. 28.25 is $s(n-2N+1)$ with a delay equal to $2N-1$.

◇

28.7 PROBLEMS

P 28.1 Consider an input-output mapping of the form shown in Fig. 28.2 with transfer function

$$H(z) = 1 + z^{-1} + z^{-2} + \ldots + z^{-7}$$

Define the block vectors $x_{M,n}$ and $y_{M,n}$ for $M = 4$ as in (28.5).

(a) Find the polyphase components of $H(z)$ of order 4.

(b) Find the transfer matrix $\mathcal{H}(z)$ from $x_{M,n}$ to $y_{M,n}$.

(c) Draw a direct FIR realization for $H(z)$.

(d) Draw an overlap-save DFT realization for $H(z)$ as in Fig. 28.9. Determine the subfilters $L_k(z)$.

(e) Draw an overlap-add DFT realization for $H(z)$ as in Fig. 28.13.

P 28.2 Consider an input-output mapping of the form shown in Fig. 28.2 with transfer function

$$H(z) = 1 - z^{-1} + z^{-2} - z^{-3} + \ldots - z^{-7}$$

with alternating ±1 coefficients. Define the block vectors $x_{M,n}$ and $y_{M,n}$ for $M = 8$ as in (28.5).

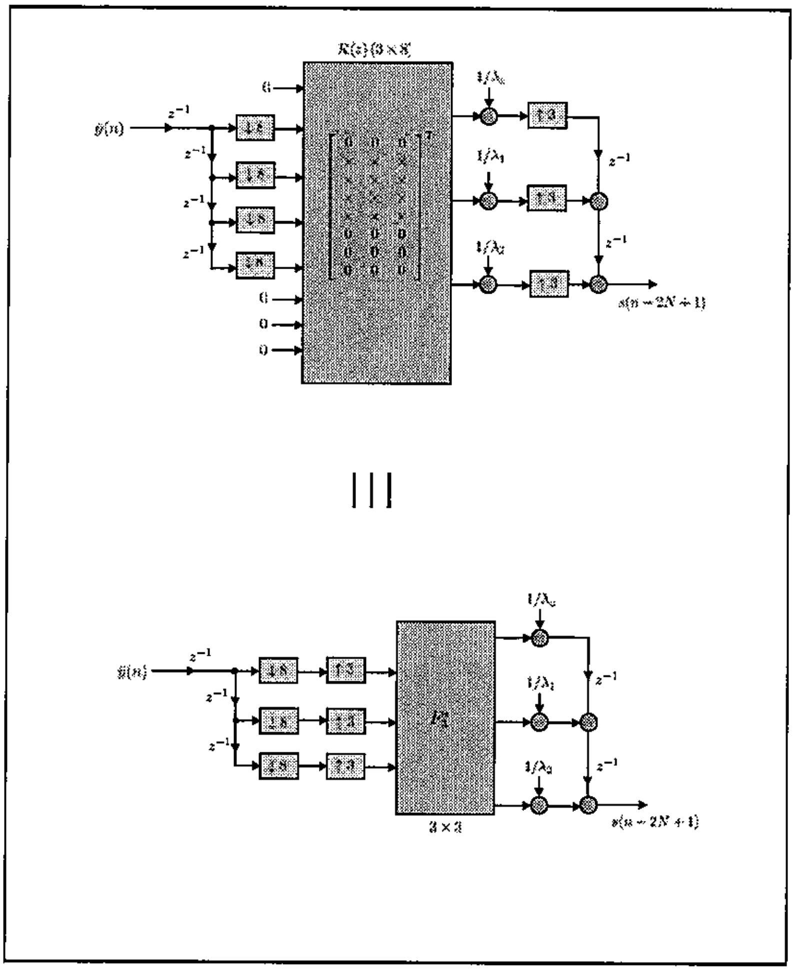

FIGURE 28.27 The receiver structure on top reduces to the DFT structure in the bottom when the coefficients $\{a_k(n)\}$ are chosen as in (28.171). The bottom structure coincides (CHECK???) with the receiver structure introduced earlier in Fig. 28.20.

(a) Find the polyphase components of $H(z)$ of order 4.

(b) Find the transfer matrix $\mathcal{H}(z)$ from $x_{M,n}$ to $y_{M,n}$.

(c) Draw a direct FIR realization for $H(z)$.

(d) Draw an overlap-save DFT realization for $H(z)$ as in Fig. 28.9. Determine the subfilters $L_k(z)$.

(e) Draw an overlap-add DFT realization for $H(z)$ as in Fig. 28.13.

P 28.3 Consider the setting of Prob. 28.1.

(a) Use the direct FIR realization to evaluate the first 4 samples of $y(n)$ when

$$x(0) = 1, \quad x(1) = 2, \quad x(2) = -1/2, \quad x(3) = 1.$$

and $x(n) = 0$ for $n < 0$.

(b) Use the overlap-save DFT realization to determine the same four samples of the output sequence.

(c) Use the overlap-add DFT realization to determine the same four samples of the output sequence.

P 28.4 Consider the setting of Prob. 28.2.

(a) Use the direct FIR realization to evaluate the first 8 samples of $y(n)$ when

$$x(0) = 2,\ x(1) = -2,\ x(2) = -1,\ x(3) = 1,\ x(4) = -1,\ x(5) = 0,\ x(6) = 0,\ x(7) = 1$$

and $x(n) = 0$ for $n < 0$.

(b) Use the overlap-save DFT realization to determine the same eight samples of the output sequence.

(c) Use the overlap-add DFT realization to determine the same eight samples of the output sequence.

P 28.5 Consider the same transfer function from Example 28.1:

$$H(z) = 1 + \frac{1}{2}z^{-1} + \frac{1}{3}z^{-2} + \frac{1}{4}z^{-3}$$

Let $M = 2$. Determine a DCT-based block implementation for $H(z)$ as in Fig. 28.15. What are the subfilters $L_k(z)$ in this case?

P 28.6 Consider the same transfer function from Example 28.1:

$$H(z) = 1 + \frac{1}{2}z^{-1} + \frac{1}{3}z^{-2} + \frac{1}{4}z^{-3}$$

Let $M = 2$. Determine a DHT-based block implementation for $H(z)$ as in Fig. 28.16. What are the subfilters $L_k(z)$ in this case?

P 28.7 Consider the FIR filter

$$H(z) = 1 + \frac{1}{2}z^{-1} + \frac{1}{3}z^{-2} + \frac{1}{4}z^{-3}$$

and define the 2×1 vectors

$$x_{2,n} = \begin{bmatrix} x(2n) \\ x(2n-1) \end{bmatrix}, \quad y_{2,n} = \begin{bmatrix} y(2n) \\ y(2n-1) \end{bmatrix}$$

in terms of the causal input and output sequences, $x(n)$ and $y(n)$, respectively. Show, from first principles, that the transfer matrix mapping $x_{2,n}$ to $y_{2,n}$ is given by

$$\mathcal{H}(z) = \begin{bmatrix} 1 + \frac{1}{3}z^{-1} & \frac{1}{2} + \frac{1}{4}z^{-1} \\ z^{-1}\left(\frac{1}{2} + \frac{1}{4}z^{-1}\right) & 1 + \frac{1}{3}z^{-1} \end{bmatrix}$$

P 28.8 Consider a causal FIR filter, $H(z)$, of length N and define the $M \times 1$ vectors

$$x_{M,n} = \begin{bmatrix} x(nM) \\ x(nM-1) \\ \vdots \\ x((n-1)M+1) \end{bmatrix}, \quad y_{M,n} = \begin{bmatrix} y(nM) \\ y(nM-1) \\ \vdots \\ y((n-1)M+1) \end{bmatrix}$$

in terms of the causal input and output sequences, $x(n)$ and $y(n)$, respectively.

(a) Repeat the argument of Prob. 28.7 to conclude that the $M \times M$ transfer matrix mapping $x_{M,n}$ to $y_{M,n}$ is the pseudo-circulant matrix $\mathcal{H}(z)$ whose first row consists of the M−th order polyphase components of $H(z)$ (see (28.12)).

(b) Define instead the column vectors

$$\bar{x}_{M,n} = \begin{bmatrix} x(nM+M-1) \\ \vdots \\ x(nM+1) \\ x(nM) \end{bmatrix}, \quad \bar{y}_{M,n} = \begin{bmatrix} y(nM+M-1) \\ \vdots \\ y(nM+1) \\ y(nM) \end{bmatrix}$$

Verify that the same transfer matrix as in part (a) still maps $\bar{x}_{M,n}$ to $\bar{y}_{M,n}$.

P 28.9 Write down the 4 × 4 DHT matrix H and verify that $HH^{\mathsf{T}} = H^2 = I$.

P 28.10 Write down the 4 × 4 DCT matrix C and verify that $CC^{\mathsf{T}} = I$.

P 28.11 What are the DFTs of the columns of the 4 × 4 DFT matrix?

P 28.12 What are the DCTs of the columns of the 4 × 4 DCT matrix?

P 28.13 What are the DHTs of the columns of the 4 × 4 DHT matrix?

P 28.14 Consider the 4 × 4 DFT, DCT, and DHT matrices F, C, and H, respectively.

(a) Find the DFTs of the columns of C.

(b) Find the DFTs of the columns of H.

(c) Find the DCTs of the columns of F.

(d) Find the DCTs of the columns of H.

(e) Find the DHTs of the columns of F.

(f) Find the DHTs of the columns of C.

P 28.15 Consider a 4 × 4 circulant matrix C and the 4 × 4 DFT matrix F, namely,

$$C = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \\ c_3 & c_0 & c_1 & c_2 \\ c_2 & c_3 & c_0 & c_1 \\ c_1 & c_2 & c_3 & c_0 \end{bmatrix}, \qquad [F]_{mk} \triangleq e^{-\frac{j2\pi mk}{4}}, \qquad m,k = 0,1,2,3$$

(a) Show that the columns of F^* are eigenvectors of C. What are the corresponding eigenvalues? Conclude that C is diagonalized by the DFT matrix.

(b) Replace C by a circulant matrix *function*, $\mathcal{C}(z)$. Use the result of part (a) to argue that $\mathcal{C}(z)$ is also diagonalized by the DFT matrix.

P 28.16 Consider an $M \times M$ constant circulant matrix C and the $M \times M$ DFT matrix F. We already know that FCF^* is diagonal, say $FCF^* = D$. Show that $F^*CF = D^*$. That is, F^*CF is also diagonal.

P 28.17 Comparing Fig. 28.9 with Figs. 27.35 and 27.36, we see that in the latter cases, F^* is used for analysis and F for synthesis. That is, the roles of F and F^* are reversed in comparison with Fig. 28.9. This change is due to the factorization used in (28.33) where F^* multiplies $L(z)$ from the left and F multiplies it from the right. Since for circulant matrices C, it holds that both FCF^* and F^*CF are diagonal (see Prob. 28.16), we could have employed an equivalent diagonalization for $C(z)$ in (28.33), say, one of the form $C(z) = F\mathcal{L}'(z)F^*$. Use this alternative factorization to motivate a new structure similar to Fig. 28.9, where F^* appears in the analysis filter bank and F appears in the synthesis filter bank.

P 28.18 Let $C = F^*DF$ denote the diagonalization of an $M \times M$ circulant matrix C by means of the DFT matrix F. Collect the entries of the diagonal matrix D into a column vector $d = \text{diag}\{D\}$.

Verify that d is proportional to the DFT of the first column of C, specifically, $d = FCe_1/M$, where e_1 is the first basis vector.

P 28.19 Use (28.33) to write $\mathcal{L}(z) = \frac{1}{4M^2} \cdot F\mathcal{C}(z)F^*$. Conclude that each $L_k(z)$ is a linear combination of the polyphase components $\{E_k(z)\}$. Conclude further that each $L_k(z)$ is a polynomial in z^{-1} with N/M coefficients.

P 28.20 Refer to the discussion in Sec. 28.2 and to the relation between the polyphase components $\{E_k(z)\}$ and the subband filters $\{L_k(z)\}$. Assume $N = M$ so that the filters $\{E_k(z), L_k(z)\}$ are all constants, namely, $E_k(z) = h(k)$ and $L_k(z) = \ell(k)$. Here, the $\{h(k), k = 0, 1, \ldots, N-1\}$ denote the impulse response coefficients of the filter $H(z)$ in (28.1). Moreover, the $\{\ell(k), k = 0, 1, \ldots, 2N-1\}$ denote the coefficients of the subband filters $\{L_k(z)\}$. Follow the argument that led to (28.36) to show that the fullband coefficients $\{h(k)\}$ and the subband coefficients $\{\ell(k)\}$ satisfy the following relations:

$$\begin{bmatrix} \ell(0) \\ \ell(1) \\ \vdots \\ \ell(2N-1) \end{bmatrix} = \frac{1}{2N}F^* \left[\begin{array}{c} h(0) \\ \vdots \\ h(N-1) \\ \hline 0_{N\times 1} \end{array}\right] = \frac{1}{2N}F \left[\begin{array}{c} h(0) \\ \hline 0_{N\times 1} \\ \hline h(N-1) \\ \vdots \\ h(1) \end{array}\right]$$

P 28.21 Use expression (28.32) to establish (28.61).

P 28.22 Refer to the definition of the $K \times K$ DCT matrix C in (28.66) and observe that its first row is $\begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}/K$.

(a) Argue that

$$C \cdot \text{col}\{1, 1, \ldots, 1\} = \text{col}\{\sqrt{K}, 0, \ldots, 0\}$$

(b) From expression (28.86), and the orthogonality of C, conclude that $\mathcal{A}(z)C = C\mathcal{L}(z)$. Multiply both sides of this equality from the right by $[1 \;\; 1 \;\; \cdots \;\; 1]^T$ and establish (28.87).

P 28.23 There is an alternative implementation that ameliorates the delay problem associated with the frequency-domain structure of Fig. 28.9. Consider the structure shown in Fig. 28.28 (for the special case $M = 3$). The input sequence, $x(n)$, is convolved directly with the first $M-1$ coefficients of $H(z)$, i.e., with the transfer function $\{h(0) + h(1)z^{-1} + \ldots + h_{M-2}z^{-(M-2)}\}$, and an intermediate sequence $\{u(n)\}$ is generated. The remaining part of $H(z)$, namely,

$$H_M(z) \stackrel{\Delta}{=} H(z) - \sum_{n=0}^{M-2} h(n)z^{-n}$$

is implemented block-wise as in Fig. 28.9 by using the subband filters that correspond to $H_M(z)$; this structure is depicted in Fig. 28.28 in terms of serial-to-parallel and parallel-to-serial conversion blocks (representing the banks of downsamplers and upsamplers from Fig. 28.9). The M-dimensional vector at the input of the parallel-to-serial converter is denoted by z_n. Assume $M = 3$.

(a) Verify that, at any particular instant n,

$$u(n) = h(0)x(n) + h(1)x(n-1), \qquad z_n = \begin{bmatrix} h(2)x(n) + h(3)x(n-1) + \ldots \\ h(2)x(n-1) + h(3)x(n-2) + \ldots \\ h(2)x(n-2) + h(3)x(n-3) + \ldots \end{bmatrix}$$

(b) Adding the output of the direct convolution path to the output of the parallel-to-serial converter, verify that

$$\begin{bmatrix} y(n+2) \\ y(n+1) \\ y(n) \end{bmatrix} = \begin{bmatrix} h(0)x(n+2) + h(1)x(n+1) + h(2)x(n) + h(3)x(n-1) + \ldots \\ h(0)x(n+1) + h(1)x(n) + h(2)x(n-1) + h(3)x(n-2) + \ldots \\ h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + h(3)x(n-3) + \ldots \end{bmatrix}$$

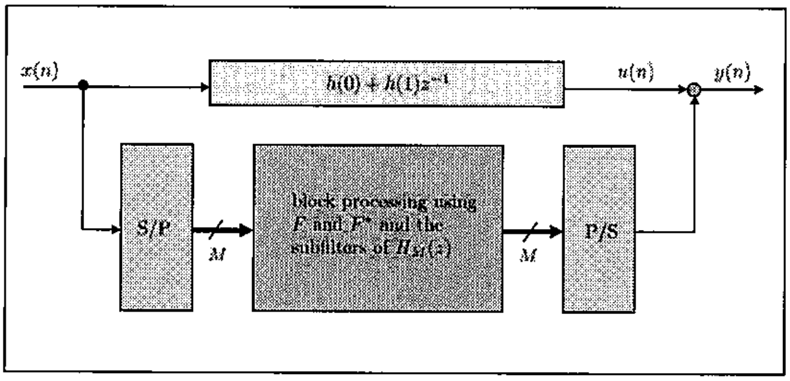
$x(n)$
$h(0)+h(1)z^{-1}$
$u(n)$
$y(n)$
S/P
M
block processing using F and F^* and the subfilters of $H_M(z)$
M
P/S

FIGURE 28.28 A delayless alternative to the block implementation of Fig. 28.9. The S/P block employs $2M$ downsamplers at the downsampling rate of M each, and the P/S block employs M upsamplers at the upsampling rate of M each. The block processing relies on $2M \times 2M$ DFT matrices $\{F, F^*\}$ and on $2M$ subband filters corresponding to $H_M(z)$ and, therefore, of order $(N - M + 1)/M$ each.

In other words, $y(n)$ is generated in response to $x(n)$ and so forth. In this way, the delay problem of Fig. 28.9 is removed.