ECE 8440 - Unit 3

Multirate Signal Processing (See section 4.7)

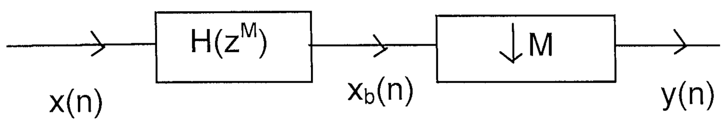
The frequency response of a stable system having system function H(z) is

$$H(e^{j\omega}) = H(z)\Big|_{z=e^{j\omega}}$$

Likewise, the frequency response of the system having system function $H_1(z) = H(z^M)$ is

$$H_1(z)\Big|_{z=e^{j\omega}} = H(z^M)\Big|_{z=e^{j\omega}} = H(e^{j\omega M})$$

Therefore, in the system shown below,



$$X_b(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega M})$$

In the same diagram, y(n) is a downsampled version of $x_b(n)$, where the downsampling ratio is M. Therefore, in the frequency domain we can write:

$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X_b\left(e^{j\left(\frac{\omega-2\pi i}{M}\right)}\right)$$

Note that $X_b\left(e^{j\left(\frac{\omega-2\pi i}{M}\right)}\right) = X(e^{\frac{j(\omega-2\pi i)}{M}})H(e^{j(\omega-2\pi i)})$

Therefore $Y(e^{j\omega})$ can be expressed as

$$Y(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X(e^{\frac{j(\omega-2\pi i)}{M}})H(e^{j(\omega-2\pi i)}) \qquad \text{(equation 4.100)}$$

Because $H(e^{j\omega})$ is periodic with period = 2π, the above expression for $Y(e^{j\omega})$ can be written as

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

If we define $X_a(e^{j\omega})$ as

$$X_a(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X(e^{\frac{j(\omega-2\pi i)}{M}})$$

Then

$$Y(e^{j\omega}) = H\left(e^{j\omega}\right)X_a(e^{j\omega})$$

Based on our previous developments, we know that the $\mathbf{x_a(n)}$ is a down-sampled version of **x(n)**, with a downsampling ratio of M. Therefore, the following two systems are equivalent:

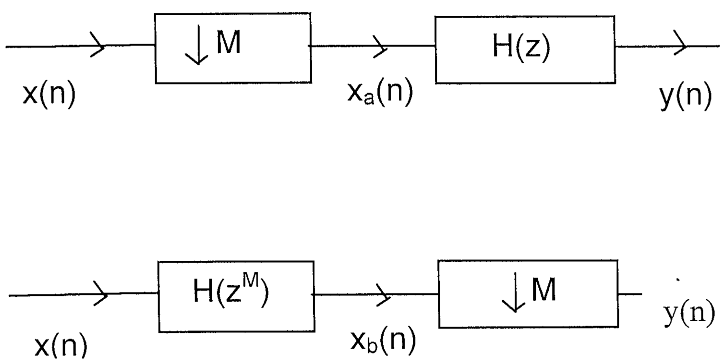


Figure 4.31 Two equivalent systems based on downsampling identities.

Therefore, the order of down-sampling and filtering in the first figure can be interchanged, if the filter is adjusted accordingly, e.g., the filtering step is $H(z^M)$ in the second figure and H(z) in the first figure.

Likewise, the following configurations, which involve filtering and up-sampling, are equivalent:

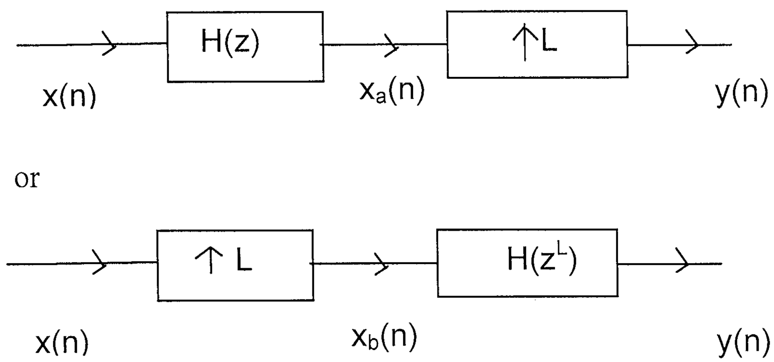


Figure 4.32 Two equivalent systems based on upsampling identities.

Show this:

From the first of the above figures,

$$Y(e^{j\omega}) = X_a(e^{j\omega L})$$

We also see that

$$X_a(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

Therefore, we can write $Y(e^{j\omega}) = H(e^{j\omega L})X(e^{j\omega L})$

From the second figure, we can also see that

$$X_b(e^{j\omega}) = X(e^{j\omega L})$$

The output the system in the second figure is therefore

$$Y(e^{j\omega}) = X_b(e^{j\omega})H(z^L)\Big|_{z=e^{j\omega}}$$
$$= X(e^{j\omega L})H(e^{j\omega L})$$

This expression for $Y(e^{j\omega})$ is the same as the expression for the $Y(e^{j\omega})$ produced in the first figure. Therefore, the operations of linear filtering and up-sampling in the first figure can be interchanged if the filter is appropriately modified; that is, if **H(z)** is replaced with **$H(z^L)$.**

<u>Polyphase Decomposition</u> (See section 4.7.3)

Consider decomposing **h(n)** into the sum of M shifted subsequences

$$h(n) = \sum_{k=0}^{M-1} h_k(n-k) \qquad \text{(equation. 4.106)}$$

where for k = 0, 1, ...,M-1

$$h_k(n) = \begin{cases} h(n+k), & \textbf{for n = integer multiple of M} \\ 0, & \textbf{for all other n.} \end{cases} \qquad \text{(equation. 4.105)}$$

For example,

$h_0(0) = h(0)$, $h_0(M) = h(M)$, $h_0(2M) = h(2M)$, etc

$h_1(0) = h(1)$, $h_1(M) = h(M+1)$, $h_1(2M) = h(2M+1)$, etc

$h_2(0) = h(2)$, $h_2(M) = h(M+2)$, $h_2(2M) = h(2M+2)$, etc

.

$h_{M-1}(0) = h(M-1)$, $h_{M-1}(M) = h(2M-1)$, $h_{M-1}(2M) = h(3M-1)$, etc

The figure below shows how the **$h_0(n)$, $h_1(n)$, . . . , $h_k(n)$** can be obtained from **h(n)** and also how these could all be combined to regenerate **h(n).**

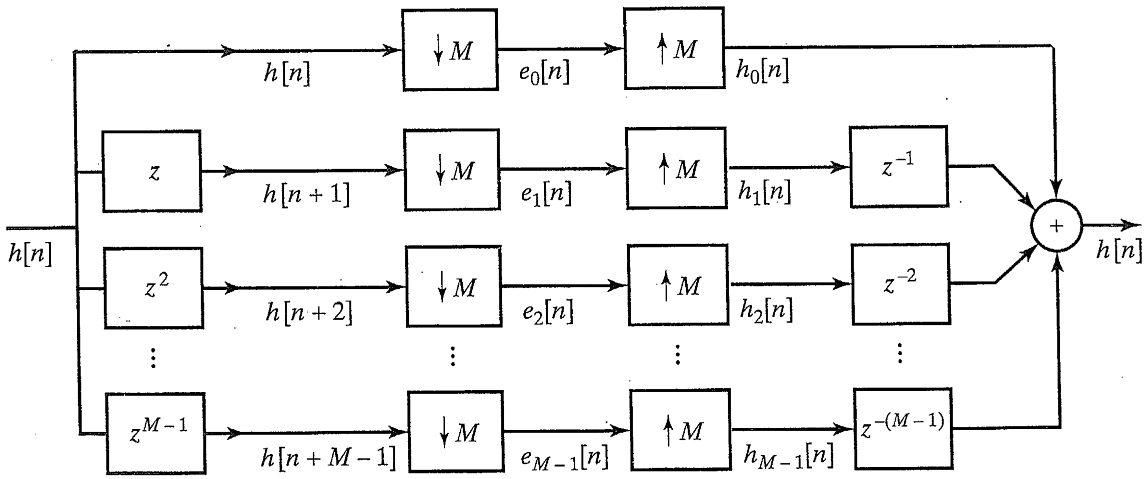


Figure 4.35 Polyphase decomposition of filter $h[n]$ using components $e_k[n]$.

Note that <u>for each value of k</u>, the result of down-sampling is

$$e_k(n) = h(nM + k)$$

Consider the output of the up-sampling step on the k-th path: temporarily call this output $y_k(n)$.

$$y_k(n) = e_k\left(\frac{n}{M}\right) \text{ for n = integer•M} \quad \text{or} \quad y_k(n) = h\left(\frac{nM}{M} + k\right) = h(n+k) \text{ for n = integer•M}$$

$$= 0 \text{ for other n} \qquad\qquad = 0 \text{ for other n}$$

Therefore, $y_k(n) = h_k(n)$ based on previous definition of $h_k(n)$.

(Repeated figure)

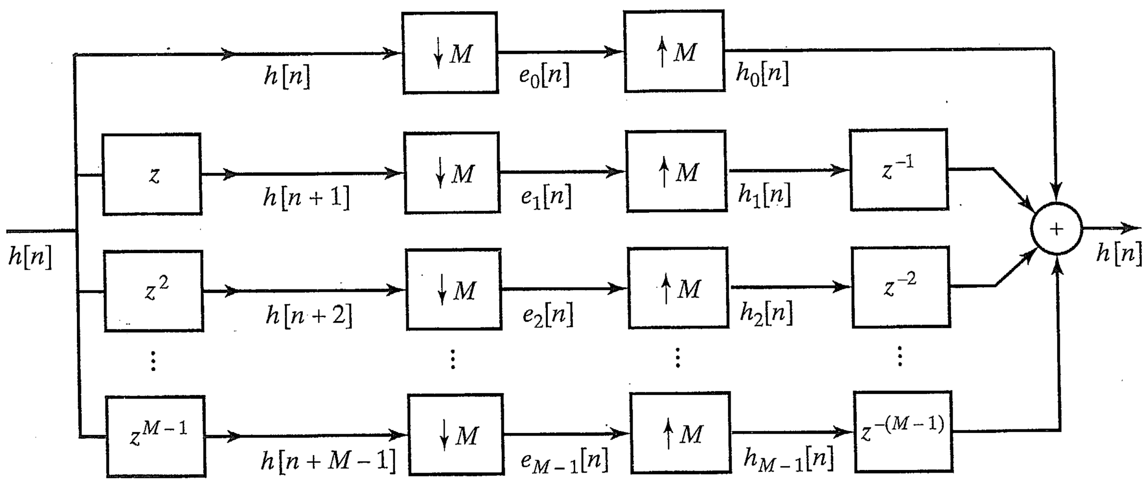


Figure 4.35 Polyphase decomposition of filter $h[n]$ using components $e_k[n]$.

$$e_k(n) = h(nM + k)$$

Example: Constructing the $\mathbf{e_k(n)}$ from **h(n)**:

$\mathbf{e_0(0) = h(0)}$	$\mathbf{e_1(0) = h(1)}$	$\mathbf{e_2(0) = h(2)}$	. . .	$\mathbf{e_{M\text{-}1}(0) = h(M\text{-}1)}$
$\mathbf{e_0(1) = h(M)}$	$\mathbf{e_1(1) = h(M+1)}$	$\mathbf{e_2(1) = h(M+2)}$	. . .	$\mathbf{e_{M\text{-}1}(1) = h(2M\text{-}1)}$
$\mathbf{e_0(2) = h(2M)}$	$\mathbf{e_1(2) = h(2M+1)}$	$\mathbf{e_2(2) = h(2M+2)}$	. . .	$\mathbf{e_{M\text{-}1}(1) = h(3M\text{-}1)}$
(etc)	**(etc)**			

Applying up-sampling to the $\mathbf{e_k(n)}$ for each k then produces $\mathbf{h_k(n)}$ for that k. (Recall that up-sampling involves inserting M-1 zeros between each input signal value.)

$\mathbf{h_0(0) = h(0)}$	$\mathbf{h_1(0) = h(1)}$	$\mathbf{h_2(0) = h(2)}$	**(etc)**
$\mathbf{h_0(1) = 0}$	$\mathbf{h_1(1) = 0}$	$\mathbf{h_2(1) = 0}$	
$\mathbf{h_0(2) = 0}$	$\mathbf{h_1(2) = 0}$	$\mathbf{h_2(2) = 0}$	
$\mathbf{h_0(M) = h(M)}$	$\mathbf{h_1(M) = h(M+1)}$	$\mathbf{h_2(M) = h(M+2)}$	
$\mathbf{h_0(M+1) = 0}$	$\mathbf{h_1(M+1) = 0}$	$\mathbf{h_2(M+1) = 0}$	
$\mathbf{h_0(M+2) = 0}$	$\mathbf{h_1(M+2) = 0}$	$\mathbf{h_2(M+2) = 0}$	
$\mathbf{h_0(2M) = h(2M)}$	$\mathbf{h_1(2M) = h(2M+1)}$	$\mathbf{h_2(2M) = h(3M+2)}$	**(etc)**

In order to reconstruct $\mathbf{h(n)}$ from the $\mathbf{h_k(n)}$**,** we can delay each $\mathbf{h_k(n)}$ by k, and then sum (as shown in figure 4.35), to get

$$h(n) = \sum_{k=0}^{M-1} h_k(n-k) \qquad \text{(equation. 4.106)}$$

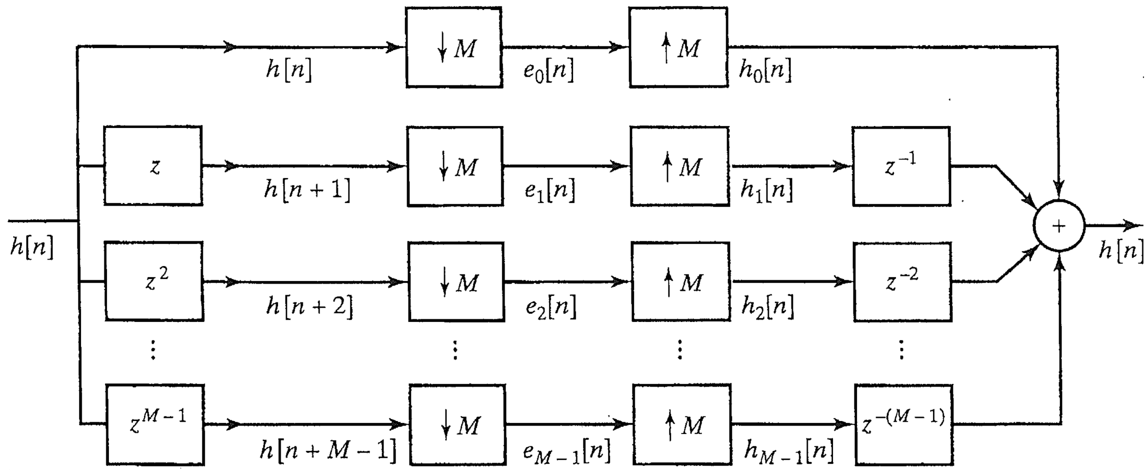
which can be expanded as

$$h(n) = h_0(n) + h_1(n-1) + h_2(n-2) + \cdots + h_{M-1}(n-M+1)$$

The following table shows the values of $h_k(n-k)$ for k = 0, 1 , 2, and M-1 for selected values of n:

n	$h_0(n)$	$h_1(n-1)$	$h_2(n-2)$		$h_{M-1}(n-M+1)$
0	h(0)	0	0	. . .	0
1	0	h(1)	0	. . .	0
2	0	0	h(2)	. . .	0
.					
.					
.					
M-1	0	0	0	. . .	h(M-1)
M	h(M)	0	0	. . .	0
M+1	0	h(M+1)	0	. . .	0
M+2	0	0	h(M+2)	. . .	0
.					
.					
2M-1	0	0	0 . . .		h(2M-1)
2M	h(2M)	0	0		0
2M+1	0	h(2M+1)	0		0
2M+2	0	0	h(2M+2)		0
.					
.					
3M-1	0	0	0 . . .		h(3M-1)
(etc)					

Now analyze the system of fig. 4.32 from a frequency domain perspectives:



(Repeated figure)

Figure 4.35 Polyphase decomposition of filter h(n) using components $e_k[n]$.

The frequency domain relation between **e_k(n)** and its up-sampled counterpart**, h_k(n),** is

$$H_k(e^{j\omega}) = E_k(e^{j\omega M})$$

In the z-domain, the previous expression can be written as

$$H_k(z) = E_k(z^M)$$

The overall contribution to **H(z)** along the k-th path in the above figure is

$$H_k(z)z^{-k} = E_k(z^M)z^{-k}$$

The overall **H(z)** is the sum of the k contributions:

$$H(z) = \sum_{k=0}^{M-1} E_k(z^M)z^{-k} \qquad \text{(equation 4.108)}$$

This represents the implementation of **H(z)** using a parallel configuration of polyphase filters.

Note that the above configuration for generating and then recombining the **$h_k(n)$** can be modified to use chains of single delay units, instead of a parallel bank of different delays, as shown below:

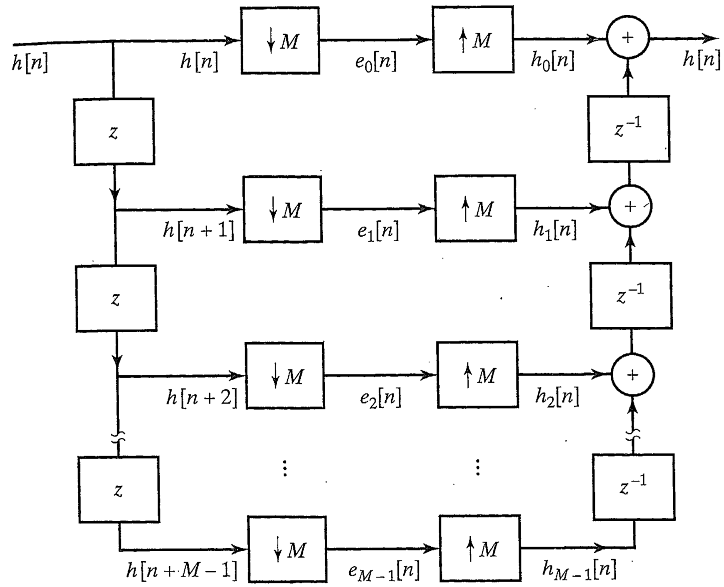


Figure 4.36 Polyphase decomposition of filter h[n] using components $e_k(n)$ with channel delays.

The overall filter having unit sample response **h(n)** can be represented compactly in the z-domain in the following form. (Refer to equation 4.108, repeated below:)

$$H(z) = \sum_{k=0}^{M-1} E_k(z^M) z^{-k}$$ (equation 4.108, repeated)

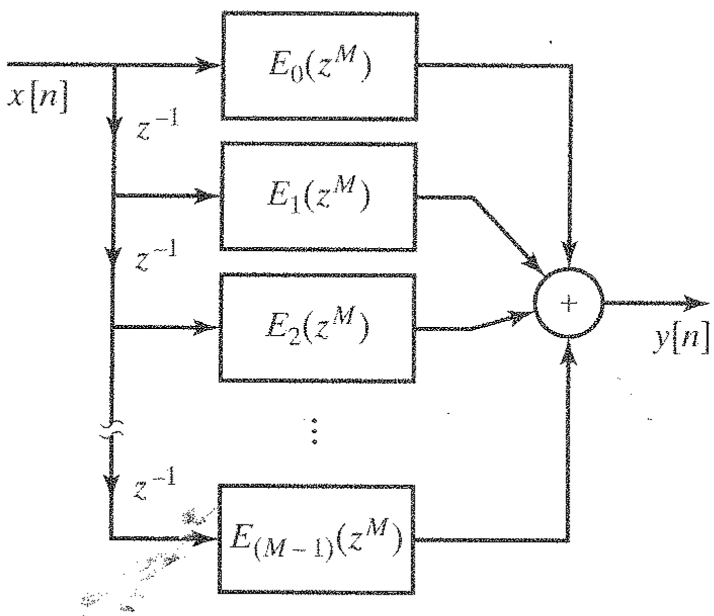


Figure 4.37 Realization structure based on polyphase decomposition of $h[n]$.

Note that for a given **h(n),** the **e_k(n)** and then **E_k(z)** can be obtained as shown in Figure 4.36.

Important example of polyphase decomposition of h(n): when we want to follow a linear filter with downsampling, as shown in the diagram below:

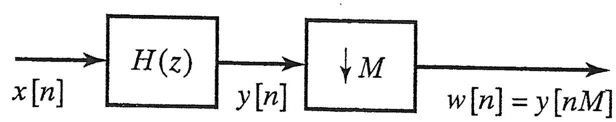


Figure 4.38 Decimation System

For polyphase decomposition, we implement **H(z)** as was shown in Figure 4.37 (on previous slide)
The overall operation of Figure 4.38 (including H(z) and ↓ M) is shown below in Figure 4.39:

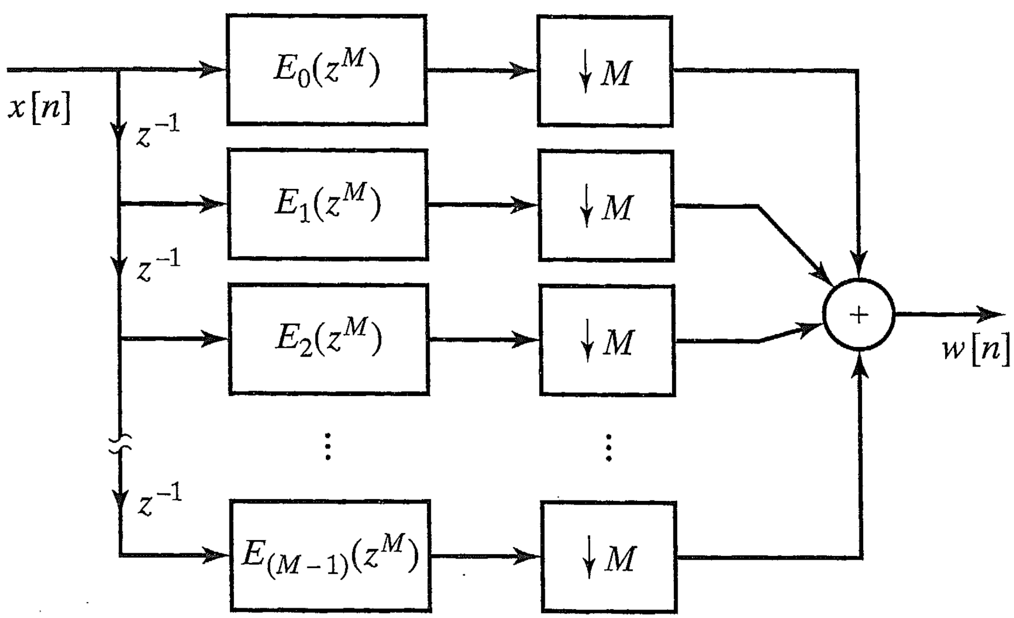


Figure 4.39 Implementation of decimation filter using polyphase decomposition.

Finally, as shown earlier in this unit, we can commute the operations of downsampling and linear filtering in the previous figure (and replace each $\mathbf{E_k(z^M)}$ **with** $\mathbf{E_k(z)}$) so the above figure can be represented as

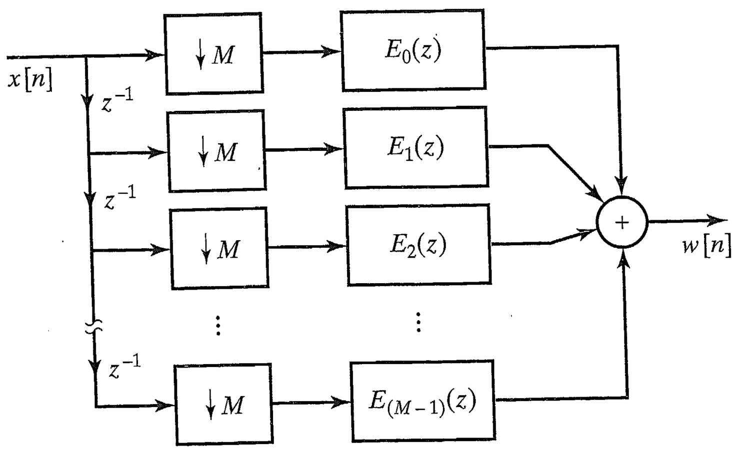


Figure 4.40 Implementation of decimation filter after applying the downsampling identity to polyphase decomposition.

<u>Efficiency of above approach:</u>

Consider the case where **H(z)** is an N-point FIR filter where the input is clocked at 1 sample per unit time. The filter output **y(n)** is also generated at this rate.

A direct Implementation of Figure 4.38 requires the following computations:

- **N multiplications (MPYs) per unit time**
- **N-1 additions (ADDs) per unit time**

For the implementation of Fig. 4.40:

For each of **M** sub-filters:

 # MPYs

(N/M MPYs per sub-filter output)

x (1/M sub-filter outputs per unit time)

= N/M^2 MPYs per unit time.

 # ADDs

(N/M)-1 ADDs per subfilter output)

x (1/M sub-filter outputs per unit time)

= [(N/M)-1]/M ADDs per unit time

Total for all M sub-filters, per unit time:

M (N/M^2) = N/M MPYs (compare with N for Figure 4.38)

M[(N/M)-1]/M

+ (M-1) ADDs (to combine sub-filter outputs)

= **[(N/M)-1] + (M-1) ADDs** (compare with N-1 for Figure 4.38)

<u>If N = 40 and M = 4</u>:

	direct method	**polyphase method**
MPYs	**N = 40**	**N/M = 40/4 = 10**
ADDs	**N-1 = 39**	**[(N/M)-1] + (M-1) = (10-1) + (4-1) = 12**

<u>Polyphase Implementation of System Consisting of a Up-Sampler Followed by a Filter</u>

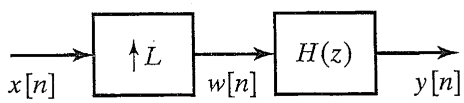


Figure 4.41 Interpolation system

If H(z) is implemented using a polyphase decomposition, the above figure becomes:

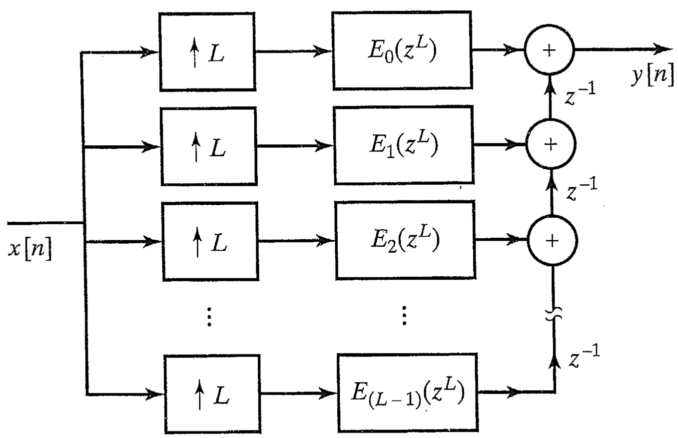


Figure 4.42 Implementation of interpolation filter using polyphase decomposition.

(Note that Figure 4.42 has implemented the delays of Figure 4.37 at the output of each sub-filter, rather than at its input.)

Then, using the equivalences of Figure 4.32,we can interchange the order of up-sampling and filtering (with each $\mathbf{E_k(z^L)}$ replaced with $\mathbf{E_k(z)}$, to express the above as

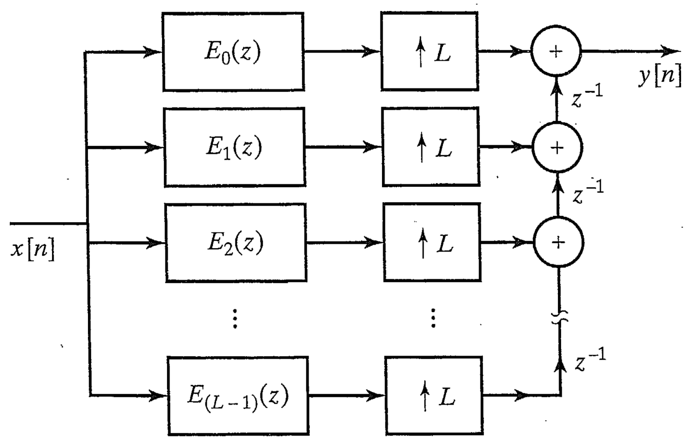


Figure 4.43 Implementation of interpolation filter after applying the upsampling identity to the polyphase decomposition.

<u>Compare the efficiency of implement of Figure 4.43 with direct implementation of Figure 4.41</u>:

Again assume that **H(z)** is an **N-point FIR filter** where the overall system input **x(n)** is clocked at 1 sample per unit time. Therefore, the filter input, **w(n)**, is clocked at a rate of **L** samples per unit time. A <u>direct implementation of Figure 4.41</u> thus requires the following computations:

- **NL MPYs per unit time**
- **(NL-1) ADDs per unit time**

In comparison, the implementation in Figure 4.43 requires the following computations for each sub-filter:

N/L MPYs per unit time

and

(N/L) -1 ADDs per unit time.

Total For all L sub-filters, per unit time:

L(N/L) = N MPYs (compare with NL)

The total number of ADDs is

L[(N/L) -1] = N - L

+(L-1) (to combine the sub-filter outputs)

= **(N - L) + (L - 1) = N -1 ADDs** (compare with NL-1)

Compare for the case of N = 40 and L = 4:

	direct method	**polyphase method**
MPYs	**NL = 40(4) = 160**	**N = 40**
ADDs	**NL-1 = 159**	**N-1 = 39**