INDIAN INSTITUTE OF SCIENCE
E9-252: MATHEMATICAL METHODS AND TECHNIQUES IN SIGNAL PROCESSING
HOME WORK #2 - SOLUTIONS, FALL 2014

INSTRUCTOR: SHAYAN G. SRINIVASA
TEACHING ASSISTANTS: ANKUR RAINA, CHAITANYA KUMAR MATCHA

Problem 1. (Frequency domain analysis) From Figure 1,

$$\begin{aligned} X_1(z) &= \frac{1}{M}\sum_{i=0}^{M-1} X\left(z^{\frac{1}{M}} e^{j\frac{2\pi i}{M}}\right) \\ \implies Y_1(z) &= X_1\left(z^L\right) \\ &= \frac{1}{M}\sum_{i=0}^{M-1} X\left(z^{\frac{L}{M}} e^{j\frac{2\pi i}{M}}\right). \end{aligned} \tag{1}$$

Similarly,

$$\begin{aligned} X_2(z) &= X\left(z^L\right) \\ Y_2(z) &= \frac{1}{M}\sum_{i=0}^{M-1} X_2\left(z^{\frac{1}{M}} e^{j\frac{2\pi i}{M}}\right) \\ &= \frac{1}{M}\sum_{i=0}^{M-1} X\left(\left(z^{\frac{1}{M}} e^{j\frac{2\pi i}{M}}\right)^L\right) \\ &= \frac{1}{M}\sum_{i=0}^{M-1} X\left(z^{\frac{L}{M}} e^{j\frac{2\pi iL}{M}}\right). \end{aligned} \tag{2}$$

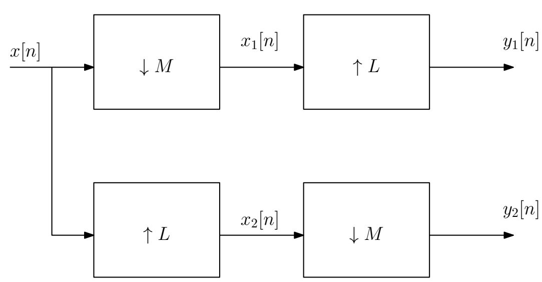


FIGURE 1. Comparing the outputs by changing the order of decimator and upsampler.

To prove that $Y_1(z) = Y_2(z)\,\forall X(z)$, it is necessary and sufficient to satisfy the following condition:

$$\begin{aligned} \left\{X\left(z^{\frac{L}{M}} e^{j\frac{2\pi iL}{M}}\right) \mid i = 0,1,\cdots,M-1\right\} &= \left\{X\left(z^{\frac{L}{M}} e^{j\frac{2\pi i}{M}}\right) \mid i = 0,1,\cdots,M-1\right\}\,\forall X(z) \\ i.e., \left\{e^{j\frac{2\pi iL}{M}} \mid i = 0,1,\cdots,M-1\right\} &= \left\{e^{j\frac{2\pi i}{M}} \mid i = 0,1,\cdots,M-1\right\}. \end{aligned}$$

Since $e^{j2\pi k} = 1 \forall k \in \mathbb{Z}$, we have $e^{j\frac{2\pi iL}{M}} = e^{j\frac{2\pi(iL \bmod M)}{M}}$. Hence, the equivalent condition is

$$\{(iL) \bmod M \mid i = 0,1,\cdots,M-1\} = \{0,1,\cdots,M-1\}. \tag{3}$$

Let $0 \le i_1 \le M-1$ and $0 \le i_2 \le M-1$ such that $i_1 \ne i_2$. Without loss of generality, consider $i_1 < i_2$. Using the following identity on modulo operation

$$(a-b) \bmod M = (a \bmod M - b \bmod M) \bmod M,$$

we have,

$$((i_1 L) \bmod M - (i_2 L) \bmod M) \bmod M = ((i_1 - i_2) L) \bmod M. \quad (4)$$

Case L and M are relatively prime:
Since $0 < i_1 - i_2 < M$, and $\gcd(L, M) = 1$, $((i_1 - i_2) L) \bmod M \neq 0$. Therefore from (4),

$$((i_1 L) \bmod M - (i_2 L) \bmod M) \bmod M \neq 0,$$
$$\implies (i_1 L) \bmod M \neq (i_2 L) \bmod M.$$

We have proved that $i_1 \neq i_2 \implies (i_1 L) \bmod M \neq (i_2 L) \bmod M \, \forall i_1, i_2 \in \{0, 1, 2 \cdots, M-1\}$. Therefore, when $\gcd(L, M) = 1$, equation (3) holds true.

Case M divides L:
Let $L = P \times M$, $P > 1$. Therefore, it is possible to chose $i_1 = i_2 + M$. Under this condition,

$$((i_1 - i_2) L) \bmod M = (ML) \bmod M = 0.$$

Therefore,

$$((i_1 L) \bmod M - (i_2 L) \bmod M) \bmod M = 0$$
$$\implies (i_1 L) \bmod M = (i_2 L) \bmod M.$$

We have shown that for some choice of $i_1 \neq i_2$, $(i_1 L) \bmod M = (i_2 L) \bmod M$. Hence, the values $\{(iL) \bmod M\}_{i=0}^{M-1}$ are not distinct. Therefore, when M divides L, equation (3) does not hold true.

Case $\gcd(M, L) = G > 1$:
Let $M = G \times P_M$ and $L = G \times P_L$. We can chose $i_1 = i_2 + G$. Under this condition, $e^{j2\pi \frac{iL}{M}} = e^{j2\pi \frac{iP_L}{P_M}}$. Therefore, $\left\{ e^{j2\pi \frac{iP_L}{P_M}} \mid i = 0, 1, \cdots, M-1 \right\}$ has P_M distinct values. Therefore, equation (3) does not hold true under this condition.

Hence, the equation (3) holds true iff L and M are relatively prime. This proves that M fold decimator and L fold upsampler blocks can be interchanged iff L and M are relatively prime.

(Time domain analysis) From the definitions of decimator and upsampler,

$$\begin{aligned}
x_1[n] &= x[Mn]. \\
y_1[n] &= \begin{cases} x_1\left[\frac{n}{L}\right], & n \text{ is a multiple of } L \\ 0 & \text{otherwise,} \end{cases} \\
y_1[n] &= \begin{cases} x\left[M\frac{n}{L}\right], & n \text{ is a multiple of } L \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \quad (5)$$

Similarly,

$$\begin{aligned}
x_2[n] &= \begin{cases} x_1\left[\frac{n}{L}\right], & n \text{ is a multiple of } L \\ 0 & \text{otherwise.} \end{cases} \\
y_1[n] &= x_1[Mn], \\
y_1[n] &= \begin{cases} x\left[\frac{Mn}{L}\right], & Mn \text{ is a multiple of } L \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \quad (6)$$

From equations (5) and (6), the outputs are same iff n is a multiple of L when ever Mn is a multiple of L.

Case $\gcd(L, M) = 1$: Trivial in this case that L divides $Mn \iff L$ divides n.

Case $\gcd(L, M) = P \neq 1$: Let $L = P \times Q$. In this case L divides Mn when ever Q divides n. Hence L divides $Mn \not\Rightarrow L$ divides n.

Therefore, the outputs are same iff L and M are relatively prime.

Problem 2. We use the identities in Figure 2 to simplify the given transformations.

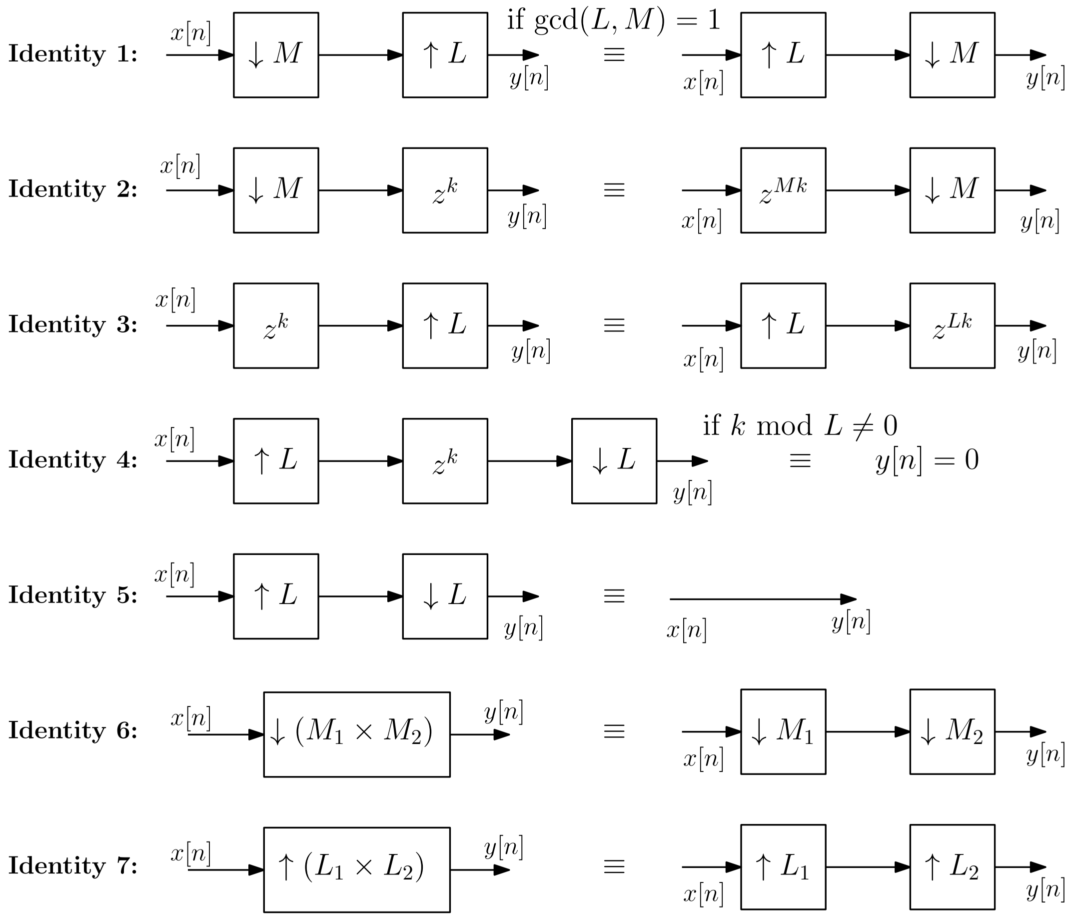
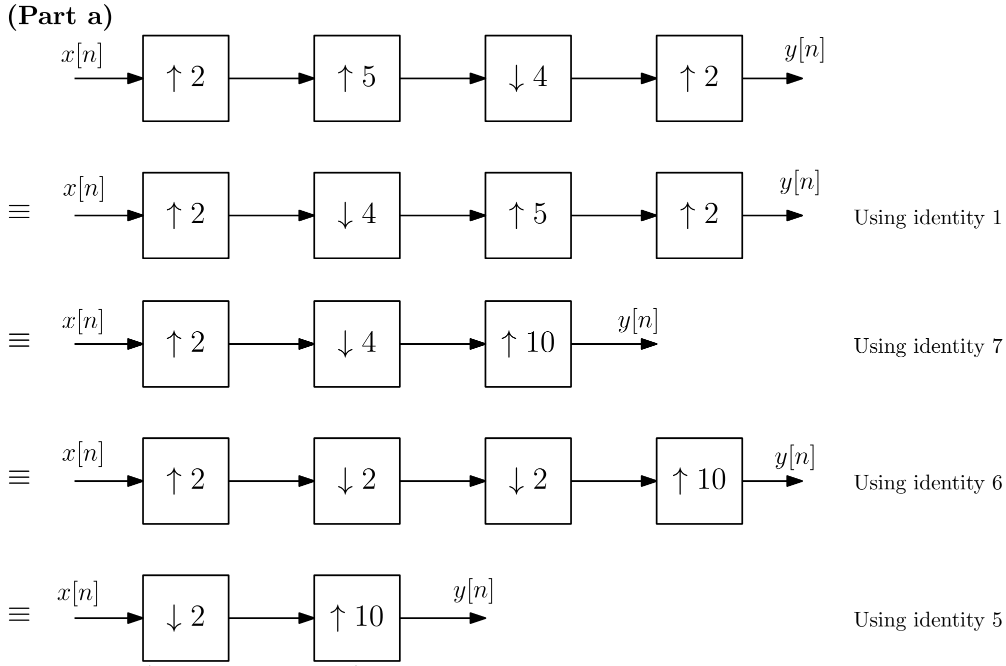


FIGURE 2. Identities related to decimation, upsampling and delay operations.

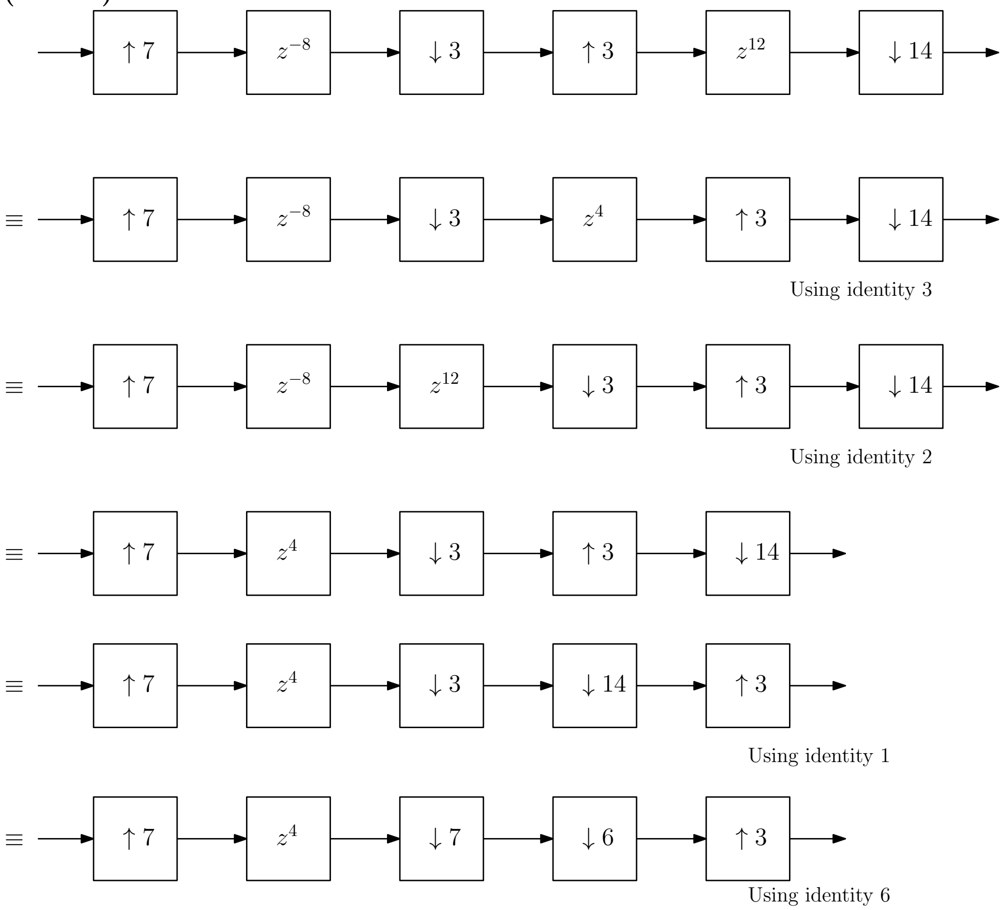
(Part a)



$X_1(z) = \frac{1}{2}\left(X(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}})\right)$

Therefore, $Y(z) = X_1\left(z^{10}\right) = \frac{1}{2}\left(X\left(z^5\right) + X\left(-z^5\right)\right).$

(Part b)



$\equiv \quad y[n] = 0$ Using identity 4

Therefore, $Y(z) = 0$.

Problem 3. Figure 3 indicates the stopband and passband frequencies for the lowpass filter when the signal is bandlimited to ω_B. We have

$$\omega_s = \frac{2\pi - \omega_B}{L} = 0.2742,$$
$$\omega_p = \frac{\omega_B}{L} = 0.04.$$

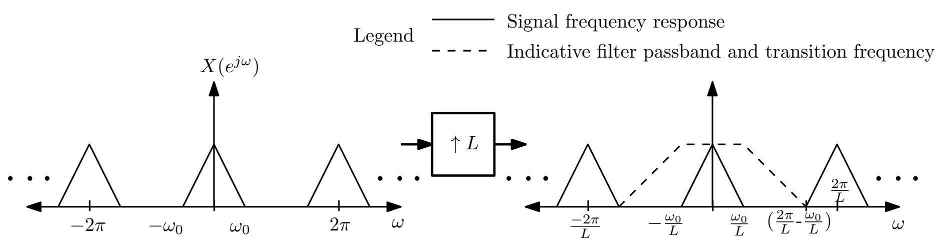


FIGURE 3. The frequency response at the output of upsampler is shown. The stopband and passband frequencies for the filter are also indicated.

The stopband and passband ripple amplitudes are

$$\delta_s = 0.005,$$
$$\delta_p = 0.01.$$

Let N be the filter order of $H(z)$ and f_s indicate the sampling frequency at the output of $H(z)$. The filter does N multiplications to give one output sample. Hence, the computational complexity of the efficient implementation of the interpolation filter is Nf_s multiplications per second.

The filter order N is estimated based on the following empirical formula by Herrmann et al.[1]

$$N = \frac{D_\infty(\delta_p, \delta_s)}{(\omega_s - \omega_p)/2\pi}$$
$$D_\infty(\delta_p, \delta_s) = (\log_{10}\delta_s)\left[a_1(\log_{10}\delta_p)^2 + a_2\log_{10}\delta_p + a_3\right] + \left[a_4(\log_{10}\delta_p)^2 + a_5\log_{10}\delta_p + a_6\right]$$

where $a_1 = 5.3e-3, a_2 = 0.071, a_3 = -0.4761, a_4 = -0.0026, a_5 = -0.5941, a_6 = -0.4278$.

A simpler but less accurate empirical formula is given by Bellanger[2]:

$$N = \frac{2\log_{10}(1/\delta_s\delta_p)}{3(\omega_s - \omega_p)/2\pi}.$$

For a signal bandlimited to3 KHz, the Nyquist sampling rate is 6 KHz and 20% oversampling is at 7.2 KHz. Hence,

$$f_s = 7200 \text{ Hz}.$$

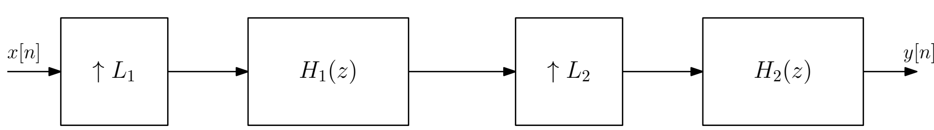


FIGURE 4. 2 stage interpolation filter.

For the 2-stage interpolator shown in Figure 4, for the filter $H_1(z)$, let the stopband and passband frequencies be ω_{s1} and ω_{p1}. Let the stopband and passband ripples be δ_{s1} and δ_{p1} respectively. Let the input to $H_1(z)$ be

[1]O. Herrmann et al., "Practical design rules for optimum low pass FIR digital filters", *Bell-sys tech.Journal*, vol 52, no.2, July 1973.

[2]M. Bellanger, "On computational complexity in digital filters,"Proc. *The Eurioeab Conference on Circuit Theory & Design*, The Haugue, The Netherlands, pp. 58-63, August 1981.

bandlimited to ω_{B1} and let f_{s1} be the sampling frequency at the output of $H_1(z)$ and N_1 be the filter order. We have

$$\begin{aligned} \omega_{B1} &= \omega_B \\ \omega_{s1} &= \frac{2\pi - \omega_{B1}}{L_1} \\ \omega_{p1} &= \frac{\omega_{B1}}{L_1} \\ f_{s1} &= f_s L_1. \end{aligned}$$

Let the $\omega_{s2}, \omega_{p2}, \delta_{s2}, \delta_{p2}, \omega_{B2}, f_{s2}, N_2$ represent the corresponding parameters for the second stage filter $H_2(z)$. Since the output of $H_1(z)$, is bandlimited to $\frac{\omega_{B1}}{L_1}$, we have

$$\begin{aligned} \omega_{B2} &= \frac{\omega_{B1}}{L_1} \\ \omega_{s2} &= \frac{2\pi - \omega_{B2}}{L_2} \\ \omega_{p2} &= \frac{\omega_{B2}}{L_1} \\ f_{s2} &= f_s L_2. \end{aligned}$$

For cascaded filters, the passband ripples get added and the stopband ripples remain the same. Hence, we choose

$$\begin{aligned} \delta_{s1} = \delta_{s2} = \delta_s = 0.005, \\ \delta_{p1} = \delta_{p2} = \frac{\delta_p}{2} = 0.005. \end{aligned}$$

Table 1, gives the parameters for various choices of L_1 and L_2. Table 2 compares the filters orders and computational complexities for various realizations of the interpolation filter. From the table, the two stage implementation with 5-fold interpolation filter followed by 4-fold interpolation filter gives the best performance. This combination is ≈ 4.1 times faster compared to the 1-stage implementation.

(L_1, L_2)	1-stage	$(2,10)$	$(10,2)$	$(4,5)$	$(5,4)$
$\omega_{B1} = \omega_B$	0.8	0.8	0.8	0.8	0.8
$\omega_{s1} = \frac{2\pi - \omega_{B1}}{L_1}$	0.2742	2.7416	0.5483	1.3708	1.9066
$\omega_{p1} = \frac{\omega_{B1}}{L_1}$	0.04	0.4	0.08	0.2	0.16
$f_{s1} = f_s \times L_1$ (Hz)	144000	14400	72000	28800	36000
$\omega_{B2} = \omega_{B1}$	-	0.4	0.08	0.2	0.16
$\omega_{s2} = \frac{2\pi - \omega_{B2}}{L_2}$	-	0.5883	3.1016	1.2166	1.5308
$\omega_{p2} = \frac{2\pi - \omega_{B2}}{L_2}$	-	0.04	0.04	0.04	0.04
$f_{s2} = f_{s1} \times L_2$ (Hz)	-	144000	144000	144000	144000

TABLE 1. Filter design parameters for 20 fold 2 stage interpolation filters. L_1 fold interpolation filter is followed by L_2 fold interpolation filter.

	(L_1, L_2)	1-stage	$(2, 10)$	$(10, 2)$	$(4, 5)$	$(5, 4)$
Herrmann et al.	Filter order N_1	57	7	31	13	16
	complexity $C_1 = f_{s1}N_1$ of $H_1(z)$ (mult/sec)	8208000	100800	2304000	50400	576000
	Filter order N_2	-	27	5	13	10
	complexity $C_2 = f_{s2}N_2$ of $H_2(z)$	-	3888000	720000	1872000	1440000
	Overall complexity $C_1{+}C_2$(mult/sec)	8208000	3988800	3024000	2246400	2016000
Bellanger' formula	Filter order N_1	77	9	42	17	21
	complexity $C_1 = f_{s1}N_1$ of $H_1(z)$ (mult/sec)	11088000	129600	3024000	489600	756000
	Filter order N_2	-	36	7	17	13
	complexity $C_2 = f_{s2}N_2$ of $H_2(z)$	-	5184000	1008000	2448000	1872000
	Overall complexity $C_1{+}C_2$(mult/sec)	11088000	5313600	4032000	2937600	2628000

TABLE 2. Comparison of filter orders and computational complexity of 20 fold 2 stage interpolation filters. $\delta_{s1} = \delta_{s2} = 0.005$. $\delta_{p1} = \delta_{p2} = 0.005$. $D_\infty(\delta_{p1}, \delta_{s1}) = D_\infty(\delta_{p2}, \delta_{s2}) = 2.3032$. $D_\infty(\delta_p, \delta_s) = 2.135$

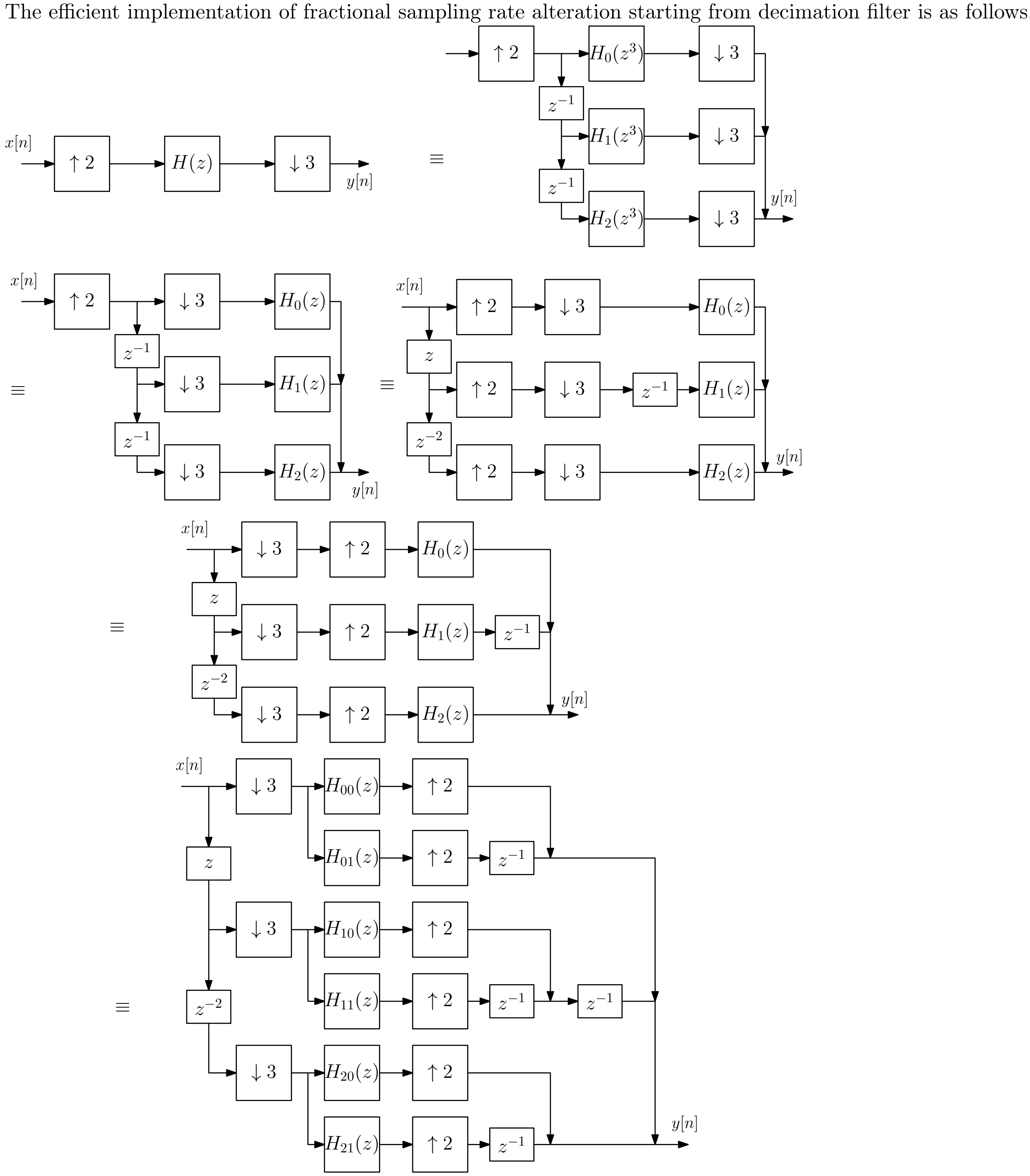
Problem 4. For the filter $H(z)$, we identify $H_0(z)$, $H_1(z)$ and $H_2(z)$ such that

$$H(z) = H_0\left(z^3\right) + z^{-1}H_1\left(z^3\right) + z^{-2}H_1\left(z^3\right).$$

We identify $H_{00}(z), H_{01}(z), H_{10}(z), H_{11}(z), H_{20}(z), H_{21}(z)$ such that

$$\begin{aligned} H_0(z) &= H_{00}(z^2) + z^{-1}H_{01}(z^2), \\ H_1(z) &= H_{10}(z^2) + z^{-1}H_{11}(z^2), \\ H_2(z) &= H_{20}(z^2) + z^{-1}H_{21}(z^2). \end{aligned}$$

The efficient implementation of fractional sampling rate alteration starting from decimation filter is as follows.



Let F be the sample rate of $x[n]$ and N be the order of the filter $H(z)$. Note that the sum of filter orders of $\{H_{ij}(z)\}_{i=0,1,2;j=0,1}$ is equal to N. Also, the filters $H_{ij}(z)$ each operates at the rate of $F/3$ samples/sec. Hence the overall computational complexity is $N\frac{F}{3}$ multiplications/second.

The efficient implementation derived in the class starting from the interpolation stage has the same number of filters operating at the rate of $F/3$ samples per second. The overall computational complexity in this case is also $N\frac{F}{3}$ multiplications/second.