Additional Examples of Chapter 13:
Multirate Digital Signal Processing Fundamentals

Example E13.1: Develop an expression for the output y[n] as a function of the input x[n] for the multirate structure of Figure E13.1.

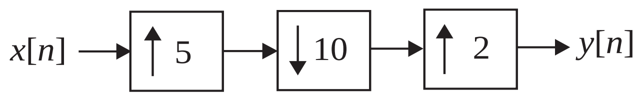
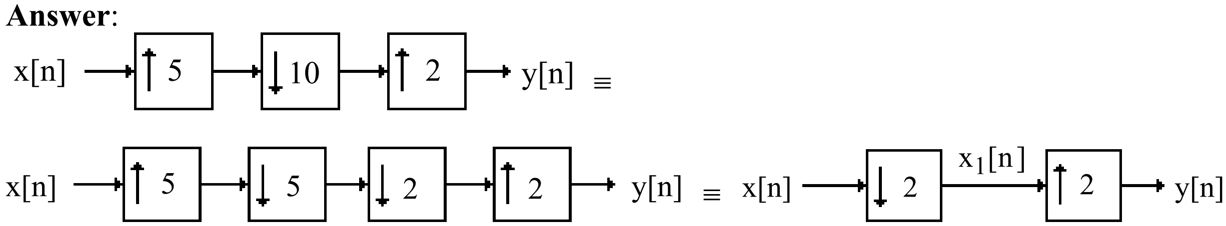


Figure E13.1

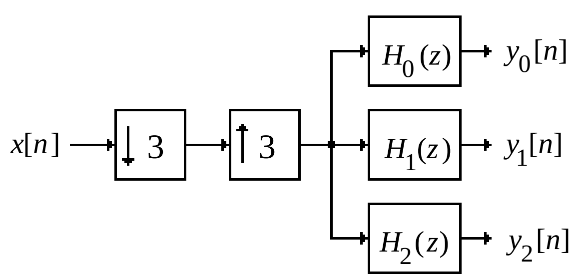
Answer:



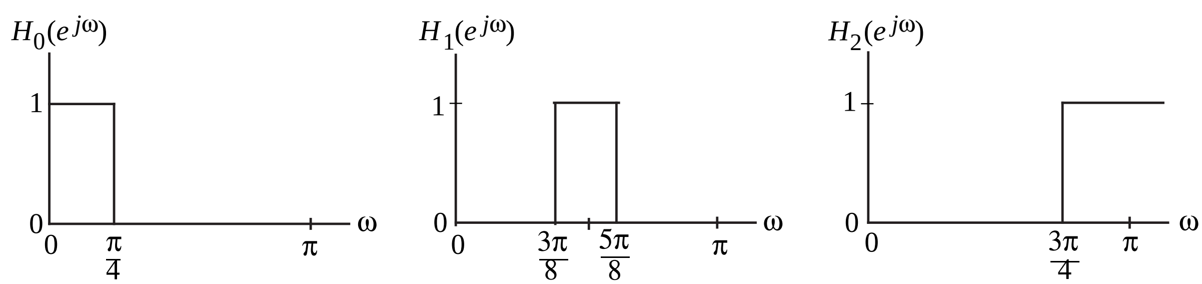
Hence, $x_1[n] = x[2n]$ and $y[n] = \begin{cases} x_1[n/2], & \text{for } n = 2r, \\ 0, & \text{otherwise} \end{cases} = \begin{cases} x[n], & \text{for } n = 2r, \\ 0, & \text{otherwise} \end{cases}$ Therefore,

$$y[n] = \begin{cases} x[n], & \text{for } n = 2r, \\ 0, & \text{otherwise.} \end{cases}$$

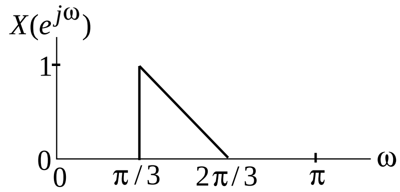
Example E13.2: Consider the multirate structure of Figure E13.2(a) where $H_0(z)$, $H_1(z)$, and $H_2(z)$ are, respectively, ideal zero-phase real-coefficient lowpass, bandpass, and highpass filters with frequency responses as indicated in Figure E13.2(b). If the input is a real sequence with a discrete-time Fourier transform as shown in Figure E13.2(c), sketch the discrete-time Fourier transforms of the outputs $y_0[n]$, $y_1[n]$, and $y_2[n]$.



(a)



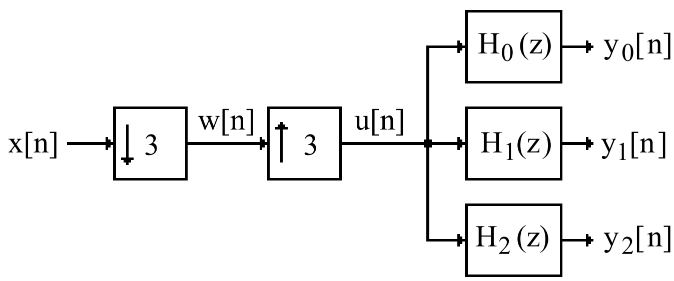
(b)

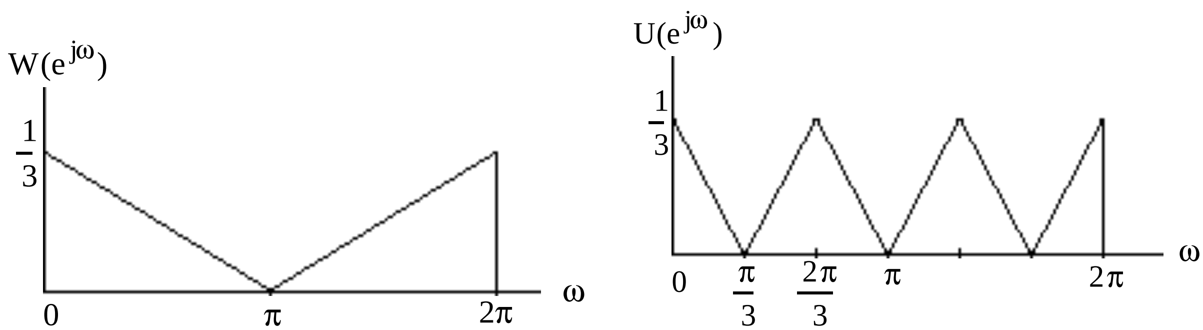


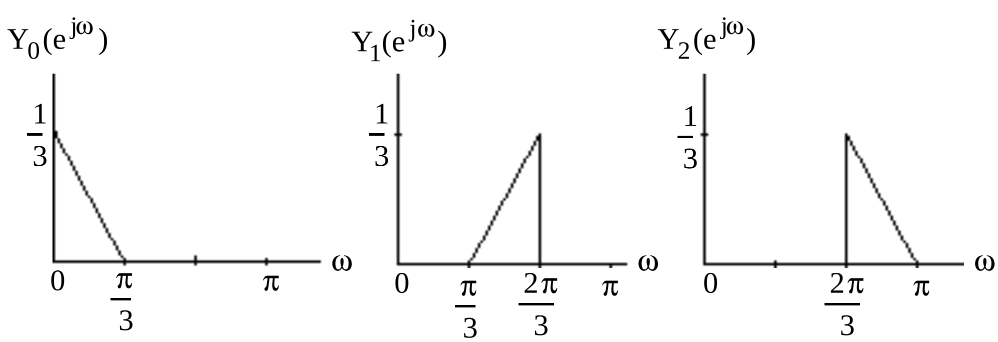
(c)

Figure E13.2

Answer:

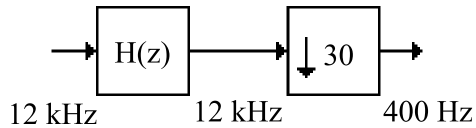




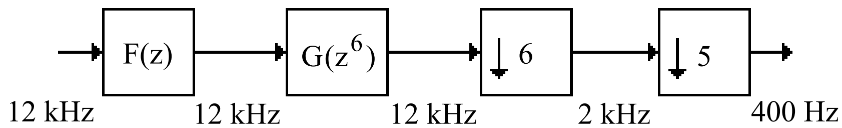


Example E13.3: Develop an alternate two-stage design of the decimator of Example 13.10 by designing the decimation filter in the form $H(z) = G(z^6)F(z)$. Compare its computational requirements with that of the design in Example 13.10.

Answer: Specifications for H(z) are as follows: $F_p = 180$ Hz, $F_s = 200$ Hz, $\delta_p = 0.002$, $\delta_s = 0.001$.



We realize H(z) as $H(z) = G(z^6)F(z)$.



Therefore, specifications for G(z) are as follows:

$F_p = 1080$ Hz, $F_s = 1200$ Hz, $\delta_p = 0.001$, $\delta_s = 0.001$. Here, $\Delta f = \dfrac{120}{12000}$. Hence, from Eq. (10.3), order of G(z) is given by $N_G = \dfrac{-20\log_{10}\sqrt{0.001\times 0.001} - 13}{14.6(120/12000)} = \dfrac{47\times 12000}{14.6\times 120} = 321.92.$

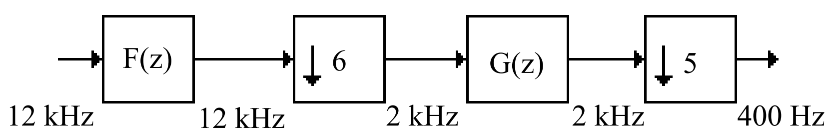
Likewise, specifications for F(z) are : $F_p = 180$ Hz, $F_s = 1800$ Hz, $\delta_p = 0.001$, $\delta_s = 0.001$. Here, $\Delta f = \dfrac{1620}{12000}$. Hence, order of F(z) is given by

$N_F = \dfrac{-20\log_{10}\sqrt{10^{-6}} - 13}{14.6(1620/12000)} = \dfrac{47\times 12000}{14.6\times 1620} = 23.846$. Thus, we choose $N_G = 322$ and $N_F = 24$.

$R_{M,G} = (322+1)\times\dfrac{2000}{5} = 129{,}200$ muliplications/second (mps), and

$R_{M,F} = (24+1)\times\dfrac{12000}{6} = 50{,}000$ mps

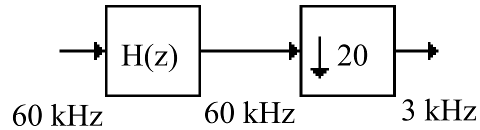
Hence, total no. of mps = 179,200. Hence the computational complexity of this particular IFIR implementation is slightly higher here than that in Example 13.10.



Example E13.4: Determine the computational complexity of a single-stage decimator designed to reduce the sampling rate from 60 kHz to 3 kHz. The decimation filter is to be designed as an equiripple FIR filter with a passband edge at 1.25 kHz, a passband ripple of 0.02, and a stopband ripple of 0.01. Use the total multiplications per second as a measure of the computational complexity.

Answer:

Additional Examples of Chapter 13:
Multirate Digital Signal Processing Fundamentals



Specifications for H(z) are: $F_p = 1250$ kHz, $F_s = 1500$ kHz, $\delta_p = 0.02$ and $\delta_p = 0.02\delta_s = 0.01$. Hence, from Eq. (10.3), order N of H(z) is given by

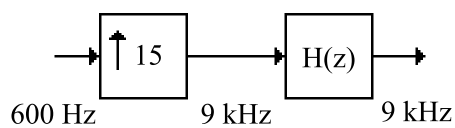
$$N = \frac{-20\log_{10}\sqrt{0.02 \times 0.01} - 13}{14.6(250/60000)} = \frac{23.989 \times 60000}{14.6 \times 250} = 394.34.$$ We thus choose N = 395.

Computational complexity is therefore $= 396 \times \dfrac{60{,}000}{20} = 1{,}188{,}000$.

Example E13.5: (a) Determine the computational complexity of a single-stage interpolator to be designed to increase the sampling rate from 600 Hz to 9 kHz. The interpolator is to be designed as an equiripple FIR filter with a passband edge at 200 Hz, a passband ripple of 0.002, and a stopband ripple of 0.004. Use Kaiser's formula given in Eq. (10.3) to estimate the order of the FIR filter. The measure of computational complexity is given by the total number of multiplications per second.

(b) Develop a two-stage design of the above interpolator and compare its computational complexity with that of the single-stage design.

Answer: (a) Specifications for H(z) are: $F_p = 200$ Hz, $F_s = 300$ Hz, $\delta_p = 0.002$ and $\delta_s = 0.004$.

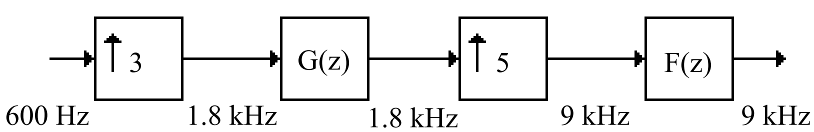


Here, $\Delta f = \dfrac{100}{9000}$. Hence, from Eq. (10.3), order N of H(z) is given by

$$N = \frac{-20\log_{10}\sqrt{0.002 \times 0.004} - 13}{14.6(100/9000)} = \frac{37.969 \times 9000}{14.6 \times 100} = 234.06.$$ We choose N = 235. Hence,

Computational complexity of H(z) $= (235+1) \times \dfrac{9000}{15} = 141{,}600$ mps.

(b) We realize $H(z) = G(z^5)F(z)$.



Specifications for G(z) are: $F_p = 5 \times 200$ Hz $= 1000$ Hz, $F_s = 5 \times 300$ Hz $= 1500$ Hz, $\delta_p = 0.002$ and $\delta_s = 0.004$. Here, $\Delta f = \dfrac{500}{9000}$. Hence, from Eq. (10.3), order N_G of G(z) is given by

$$N_G = \frac{-20\log_{10}\sqrt{0.001\times 0.004)-13}}{14.6\Delta f} = 50.5225. \text{ We choose } N_G = 51.$$

$$R_{M,G} = (52+1)\times\frac{1800}{3} = 31{,}800 \text{ mps.}$$

Specifications for F(z) are: $F_p = 200\text{ Hz} = 1000\text{ Hz}$, $F_s = 1500\text{ Hz}$, $\delta_p = 0.002$ and $\delta_s = 0.004$. Here, $\Delta f = \frac{1300}{9000}$. Hence, from Eq. (10.3), order N_F of F(z) is given by

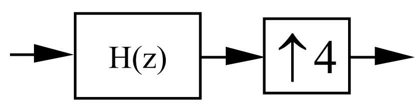
$$N_F = \frac{-20\log_{10}\sqrt{0.001\times 0.004)-13}}{14.6(1300/9000)} = 19.4317. \text{ We choose } N_F = 20.$$

$$R_{M,F} = (20+1)\times\frac{9000}{5} = 37{,}800 \text{ mps.}$$

Total computational complexity of the IFIR-based realization is therefore $R_{M,G} + R_{M,F} =$ 69,600 mps.

Example E13.6: Develop a computationally efficient realization of a factor-of-4 interpolator employing a length-16 linear-phase FIR filter.

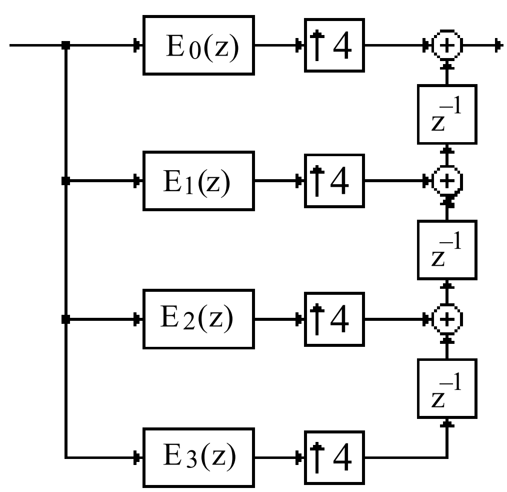
Answer: A computationally efficient realization of the factor-of-4 interpolator



is obtained by applying a 4-branch polyphase decomposition to H(z):

$$H(z) = E_0(z^4) + z^{-1}E_1(z^4) + z^{-2}E_2(z^4) + z^{-3}E_3(z^4).$$

and then moving the down-sampler through the polyphase filters resulting in



Further reduction in computational complexity is achieved by sharing common multipliers if H(z) is a linear-phase FIR filter. For example, for a length-16 Type II FIR transfer function a computationally efficient factor-of-4 interpolator structure based on the above equation is as shown below:

Additional Examples of Chapter 13: Multirate Digital Signal Processing Fundamentals