# EEE6440
# Advanced Signal Processing (ASP)

- Multirate Digital Signal Processing and Applications:
    - Introduction
    - Decimation & Interpolation
    - Sampling rate conversion by a rational factor
    - Decimation and Interpolation filters
    - Multistage implementation of sampling rate conversion
    - Applications

- MATLAB: Commands: decimate, interp
- Reference Books: Proakis & Manolakis (Ch.10)

**Dr Charith Abhayaratne**

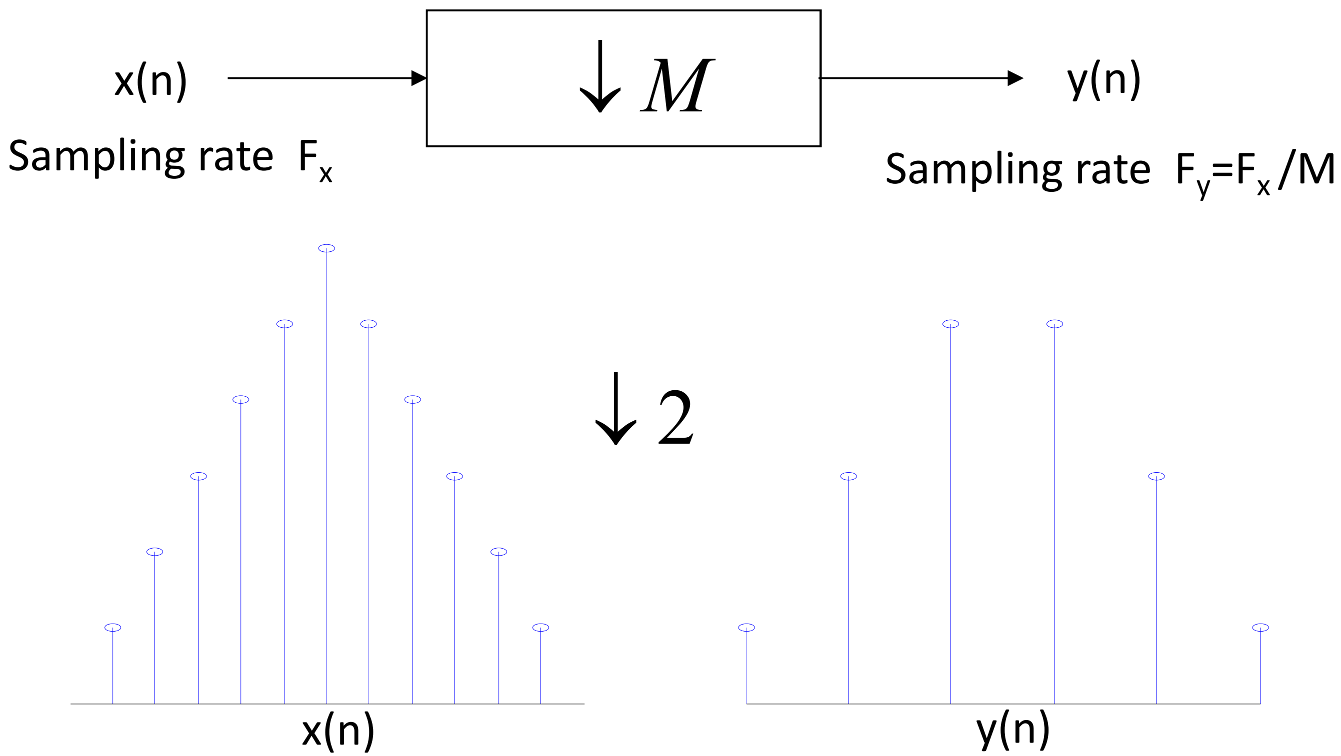
Tel: 25893
Email: c.abhayaratne@sheffield.ac.uk
Office: F176

# 1. Introduction

- The increasing need in modern digital systems to process data at more than one sampling rate led to the development of multi-rate DSP systems.
- Example: In digital audio, three different sampling rates are used: 192 kHz in studio recording, 32 kHz in broadcasting, 44.1 kHz in CD and 48kHz in Digital Audio Tape (DAT).
- These systems find applications in
  - Digital audio systems
  - Speech and image coding
  - Radar and sonar systems
  - High resolution analogue-to-digital converters
  - High quality data acquisition and storage systems
  - Efficient Implementation of digital filters

# 2. Decimation and Interpolation

- Decimation or down-sampling by an integer factor M




- The output sequence is related to the input by

$$y(n) = x(Mn)$$

- Y(z), the z-transform of y(n), is

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n) z^{-n} = \sum_{n=-\infty}^{\infty} x(nM) z^{-n}$$

- Let $\widetilde{x}(m) = c(m)x(m)$, where

$$c(m) = \frac{1}{M} \sum_{k=0}^{M-1} e^{j2\pi km/M} = \begin{cases} 1, & m = 0, \pm M, \pm 2M, \ldots \\ 0 & \text{otherwise} \end{cases}$$

- Therefore,

$$\widetilde{x}(m) = \begin{cases} x(m), & m = 0, \pm M, \pm 2M, \ldots \\ 0 & \text{otherwise} \end{cases}$$

$$Y(z) = \sum_{n=-\infty}^{\infty} \widetilde{x}(nM) z^{-n} = \sum_{m=-\infty}^{\infty} \widetilde{x}(m) z^{-m/M}$$

- Substituting for $\tilde{x}(m) = c(m)x(m)$, we get

$$Y(z) = \frac{1}{M}\sum_{k=0}^{M-1}\sum_{m=-\infty}^{\infty} x(m)e^{j2\pi km/M} z^{-m/M}$$

$$= \frac{1}{M}\sum_{k=0}^{M-1} X\left(e^{-j2\pi km/M} z^{1/M}\right)$$

- In frequency domain, we have

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

- Here $\omega$ is the normalised frequency w.r.t. The sampling rate $F_y$

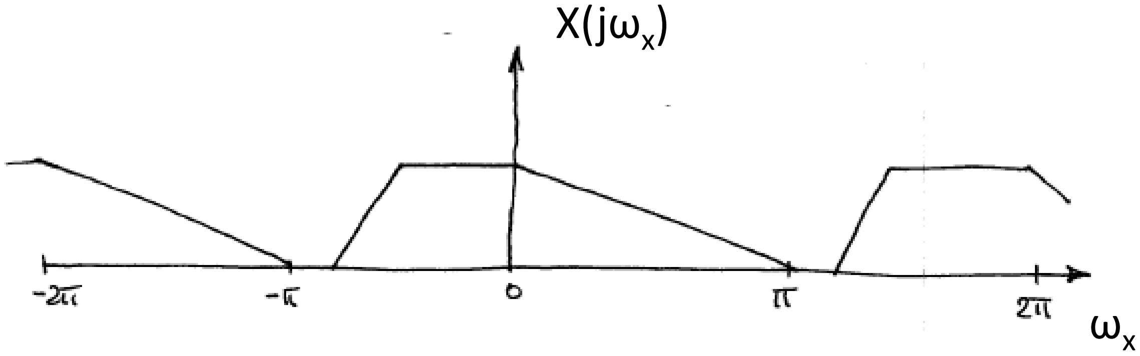
$$\omega = M\omega_x$$

- In terms of $\omega_x$ , we have

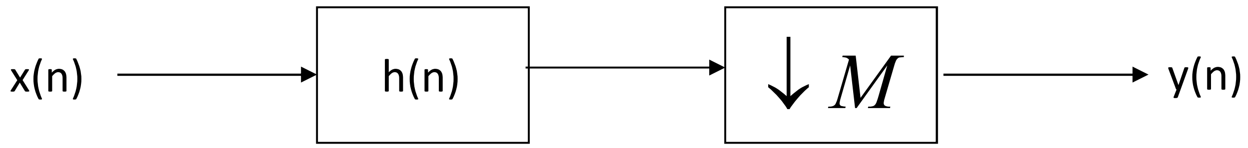
$$Y(j\omega_x) = \frac{1}{M}\sum_{k=0}^{M-1} X\left(j\left(\omega_x - 2\pi k / M\right)\right)$$

- How can we interpret $Y(j\omega_x)$?
  - It represents stretching of $X(j\omega_x)$ to $X(j\omega_x M)$
  - creating M − 1 copies of the stretched versions
  - shifting each copy by successive multiples of 2π and superimposing
  - (adding) all the shifted copies
  - dividing the result by M

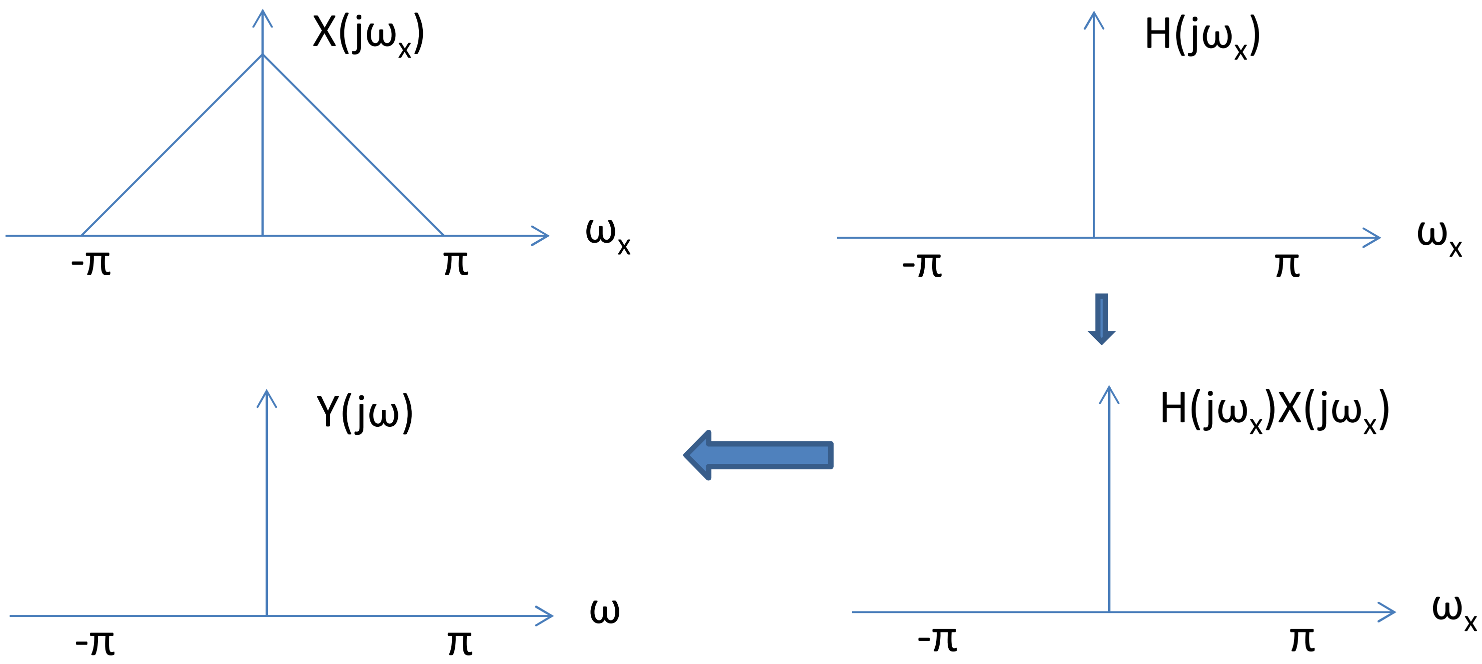
Draw $Y(j\omega)$ for the $X(j\omega_x)$ shown below. Consider M=2



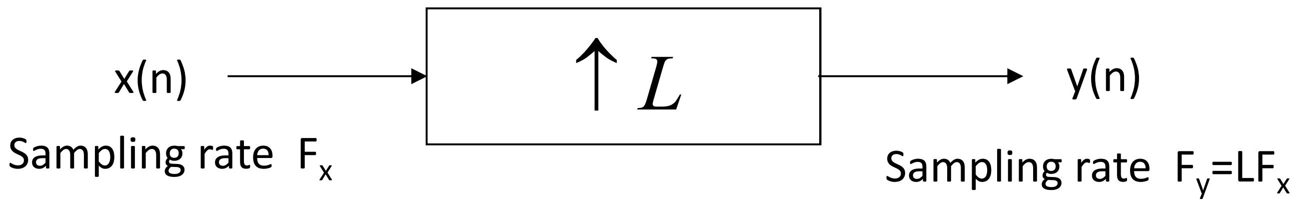
- To avoid aliasing, a digital anti-aliasing filter (h(n)) is used. Now an M-factor decimator looks as follows:

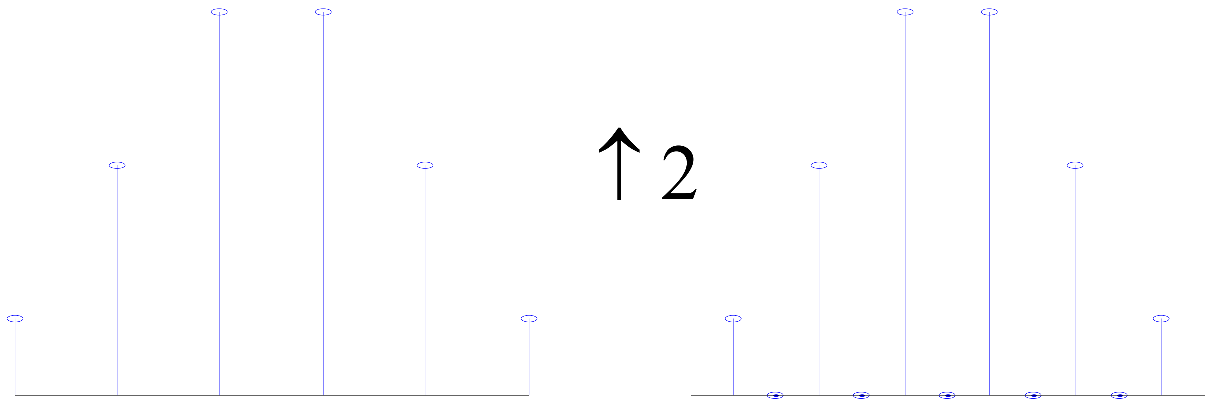



- h(n) is a low pass filter. What is its stop-band frequency?



- Interpolation or up-sampling by an integer factor L




x(n) y(n)

- The output sequence is related to the input by

$$y(n) = \begin{cases} y(n/L), & m = 0, \pm L, \pm 2L, \ldots \\ 0 & \text{otherwise} \end{cases}$$

- Y(z), the z-transform of y(n), is

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n) z^{-n} = \sum_{n=-\infty}^{\infty} x(n/L) z^{-n}$$

$$Y(z) = \sum_{m=-\infty}^{\infty} x(m) z^{-mL} = X(z^L)$$

- The frequency domain characteristic of y(n) is

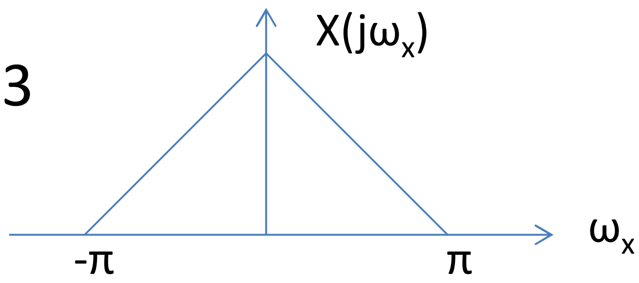
$$Y(j\omega) = X(jL\omega) = X(j\omega_x)$$

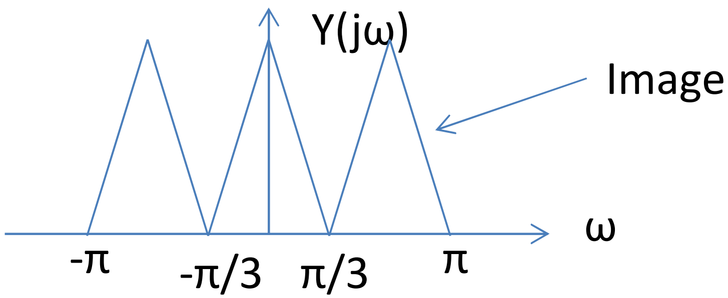
- The input and output frequencies $\omega_x$ and $\omega$ are related by

$$\omega_x = L\omega$$

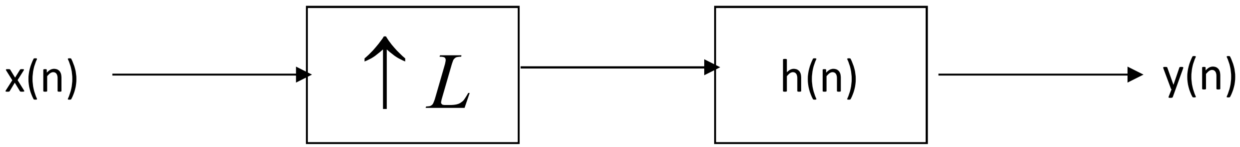
$$\omega_x = \pi \rightarrow \omega = \pi / L$$

- For example, for L=3




- To avoid imaging, a digital anti-imaging filter (h(n)) is used. Now an L-factor interpolator looks as follows:




- What type of filter is h(n)?

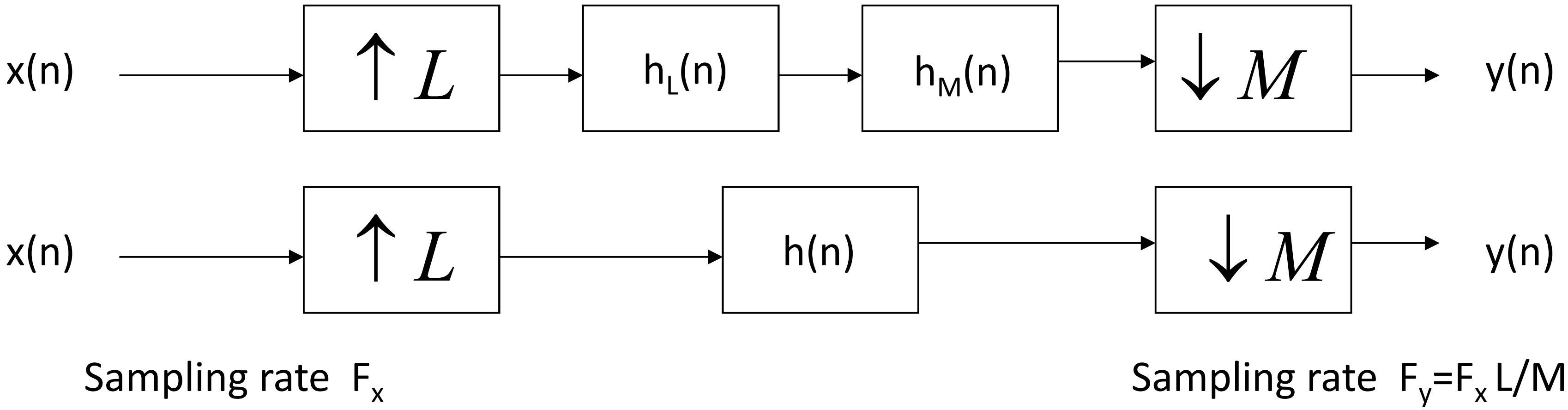
- Some multi-rate identities
- Consider R is either M or L

$$Ra \equiv aR$$
$$(x_1 + x_2)R \equiv x_1 R + x_2 R$$
$$(x * d)R \equiv xR * dR$$

- $\downarrow M \ \uparrow L \equiv \uparrow L \ \downarrow M$ iff L and M are relative prime.
  (Prove this)

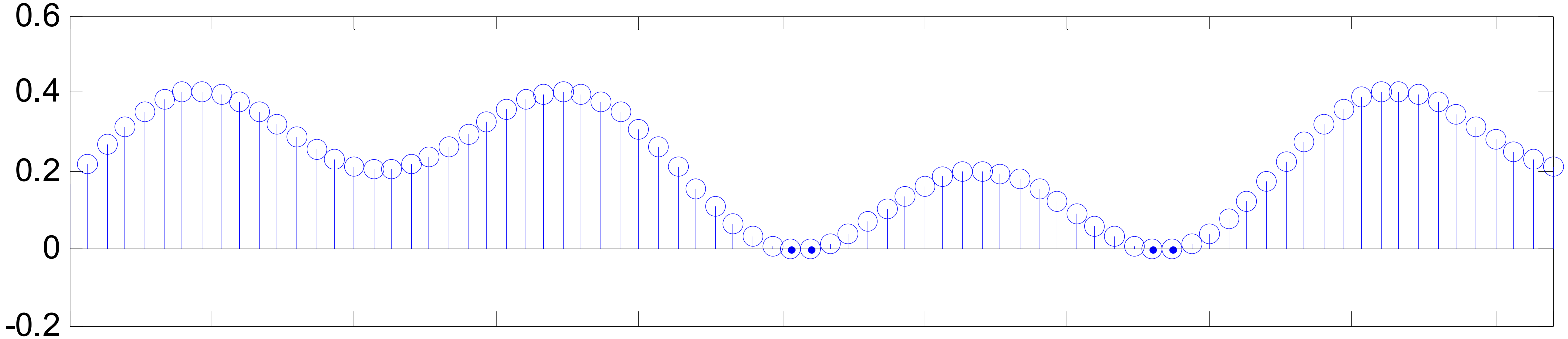
# 3. Sampling rate conversion by a rational factor L/M

- The data is interpolated by a factor L, then decimated by M

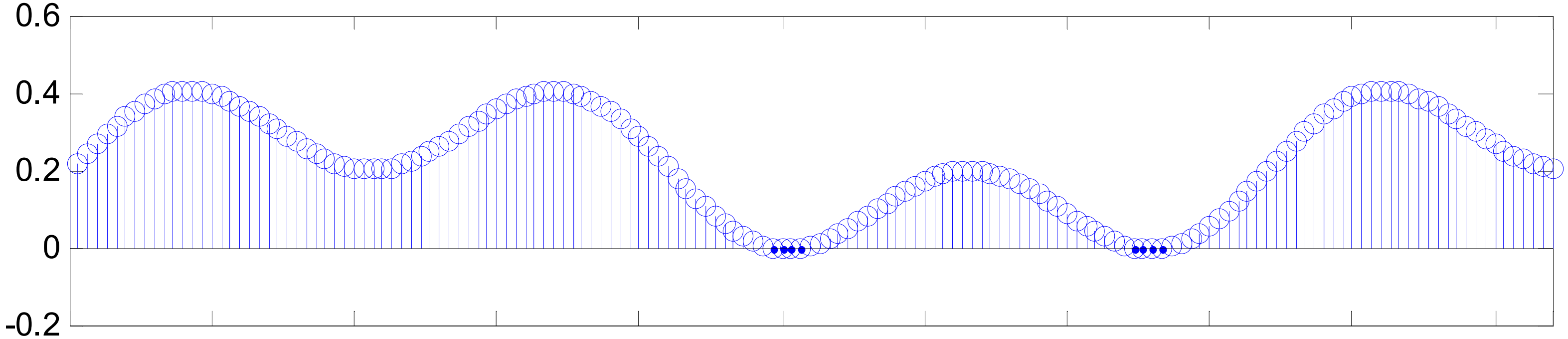



- It is necessary that the interpolation process precede the decimation process, otherwise important frequency components may be removed by the anti-aliasing filter.
- Furthermore, the anti-imaging and anti-aliasing filter can be combined to a single filter.

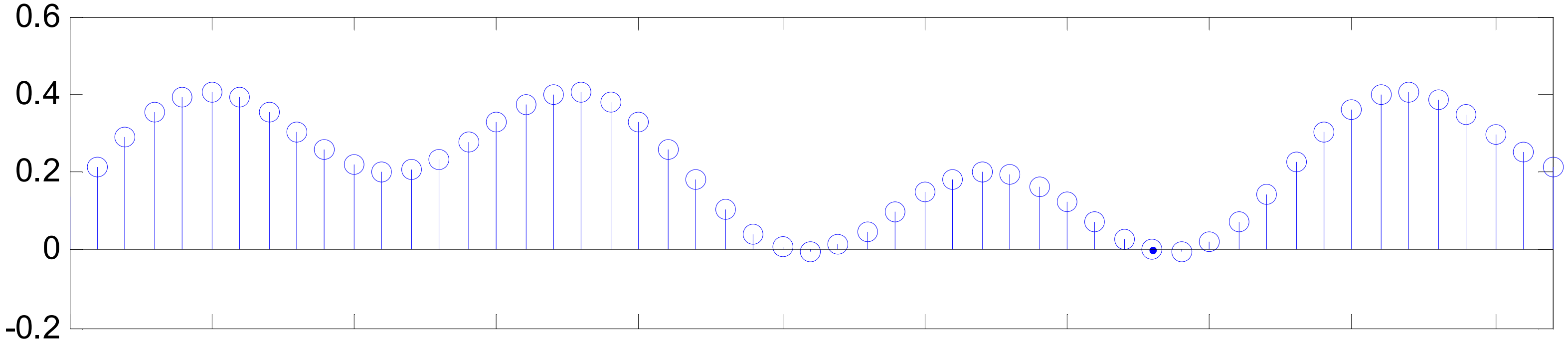
Decimation by a factor of 2/3




Original Signal    $F_x$ = 30 Hz




Interpolated Signal    F =  ?




Decimated Signal    $F_y$ = ?

# 4. Decimation and interpolation filters

- The performance of multirate systems depends on the quality of the anti-aliasing and the anti-imaging filters used.

- Either FIR or IIR filters can be used for decimation or interpolation.

- In practice, however, FIR filters are preferred since they offer significant computational savings (Think, how this is possible.)

- **For decimation:**
- Prior to down-sampling, the signal must be band limited to the range $|\omega|<\pi/M$ by a lowpass filter to avoid aliaising.
- If $\omega_p$ denotes the highest frequency that need to be preserved in the input signal, the decimation filter is

$$H(j\omega)=\begin{cases}1, & |\omega|\le\omega_p \\ 0, & \frac{\pi}{M}\le|\omega|\le\pi\end{cases}$$

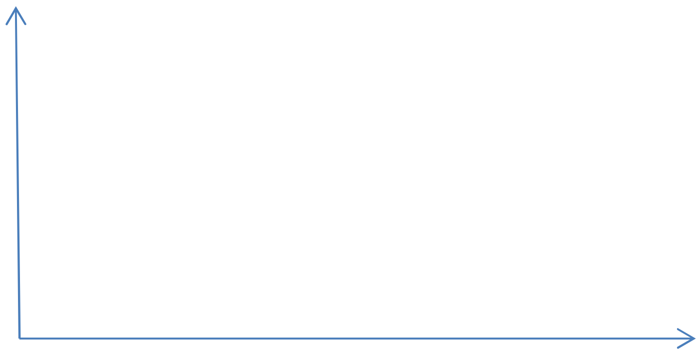
- The overall specifications of the decimation filter are
  - Passband $0 \le \omega \le \omega_p$ or $0 \le f \le f_p$
  - Stopband $\pi/M \le \omega \le \pi$ or $1/2M \le f \le 1/2$
  - Passband ripple $\delta_p$ or $A_p = 20 \log (1+ \delta_p)$
  - Stopband ripple $\delta_s$ or $A_s = -20 \log (\delta_s)$

- The length N of an equiripple FIR filter is given by

$$N \approx \frac{-10\log(\delta_p \delta_s) - 13}{14.6(\Delta f)} + 1$$

where $\Delta f$ is the transition bandwidth (in normalized frequency).

$$\Delta f = \frac{1}{2M} - f_p$$



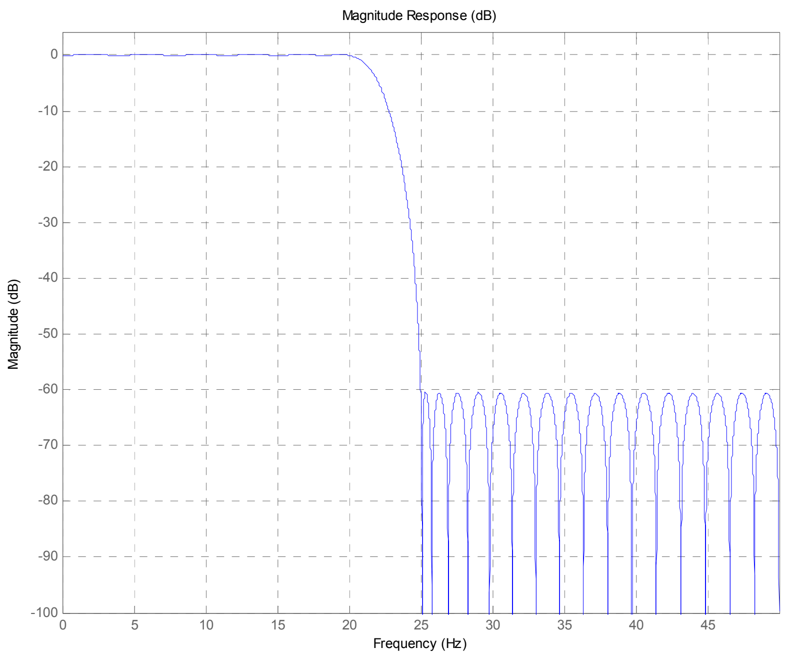
- Example 4.1: The sampling rate of a signal is to be reduced from 100Hz to 50Hz. Design a decimator which preserves frequencies up to 20 Hz. Choose passband and stopband ripples as 0.01 and 0.001, respectively .

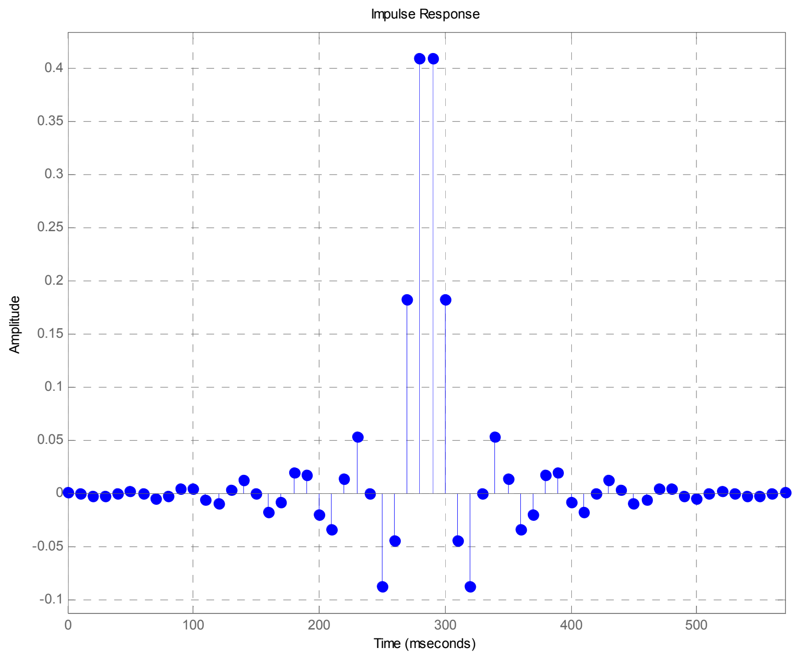
- The decimation filter should satisfy the following specifications.
  - Decimation factor: M =
  - Passband edge frequency:
  - Stopband edge frequency:
  - Passband ripple $\delta_p$ =
  - Stopband ripple $\delta_s$ =

- The length of the required equiripple FIR filter is

$$N \approx \frac{-10\log(\delta_p \delta_s) - 13}{14.6(\Delta f)} + 1 \qquad ?$$

- In MATLAB:
  - Fs=100;
  - A=20*log10(1+0.01);
  - B=-20*log10(0.001);
  - d=fdesign.lowpass('Fp,Fst,Ap,Ast',20,25,A,B,Fs);
  - Hd=design(d,'equiripple');
  - fvtool(Hd)



- **For interpolation:**
- The anti-imaging filter must remove all but the useful information by bandlimiting the interpolated data.

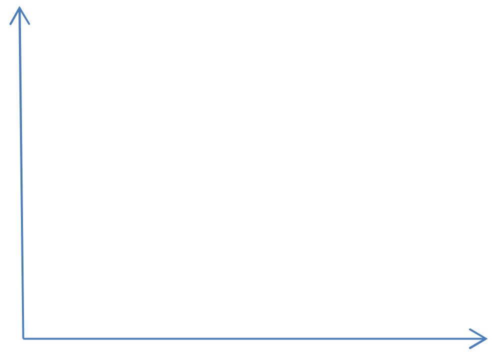
- The desired interpolation filter should have a stopband edge at $\omega_s = \pi/L$ .

- If $\omega_c$ denotes the highest frequency that need to be preserved in the input signal to be interpolated, then the passband edge frequency of the interpolation filter should be $\omega_p = \omega_c /L$ .

- The frequency response of the interpolation filter should be

$$H(j\omega) = \begin{cases} 1, & |\omega| \le \omega_p \\ 0, & \dfrac{\pi}{L} \le |\omega| \le \pi \end{cases}$$

- The overall specifications of the interpolation filter are
  - Passband edge $\omega_p = \omega_c / L$ or $f_p = f_c / L$
  - Stopband edge $\omega_s = \pi / L$ or $f_s = 1/2L$
  - Passband ripple $\delta_p$ or $A_p = 20 \log (1 + \delta_p)$
  - Stopband ripple $\delta_s$ or $A_s = -20 \log (\delta_s)$



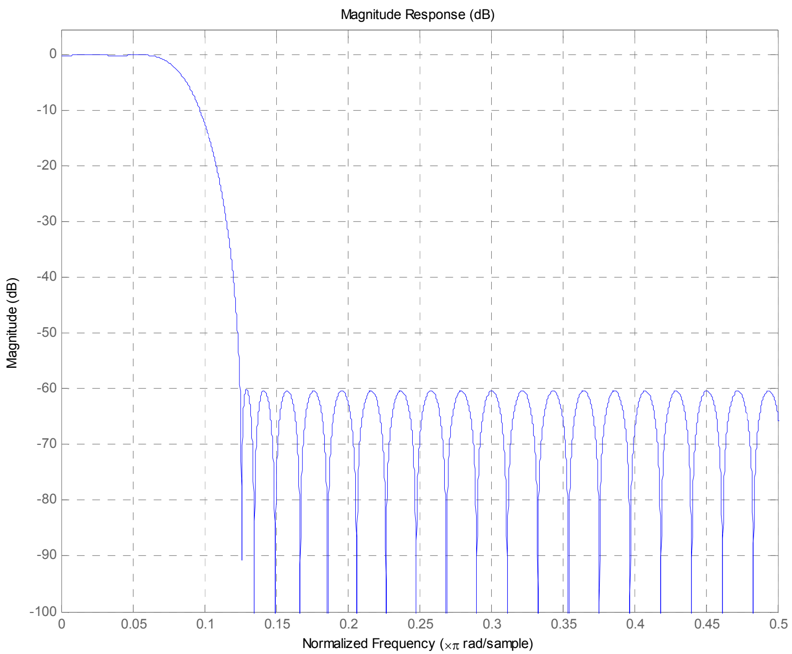
- The length N of an equiripple FIR filter is given as in slide #16.

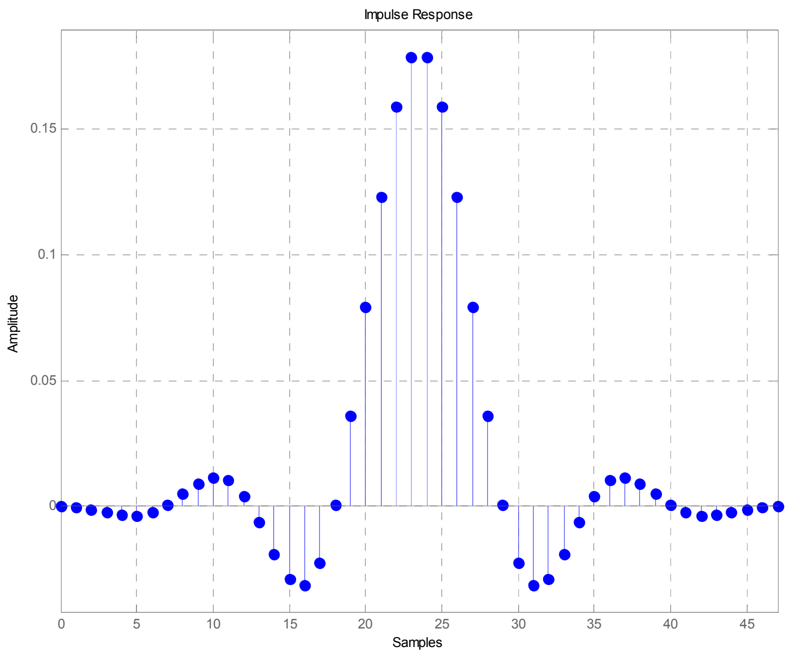
- Example 4.2: Design a 4-fold interpolator that preserves frequencies up to $\pi/2$. Use an FIR filter of 0.1dB paasband ripple and 60 dB stopband attenuation.

  - Interpolation factor: L =
  - Passband edge frequency:
  - Stopband edge frequency:
  - Passband ripple $\delta_p$ =
  - Stopband ripple $\delta_s$ =

  - The length N of the FIR equiripple filter $N \approx \dfrac{-10 \log(\delta_p \delta_s) - 13}{14.6(\Delta f)} + 1$ = 41

- In MATLAB:
  - Fs=2*pi;
  - d=fdesign.lowpass('Fp,Fst,Ap,Ast',pi/8,pi/4,0.1,60,Fs);
  - Hd=design(d,'equiripple');
  - fvtool(Hd)



# 5. Multistage implementation of sampling rate conversion

- For high sampling rate conversions, it may be practical, from an implementation view point, and computational inefficient to perform decimation or interpolation in a single stage.

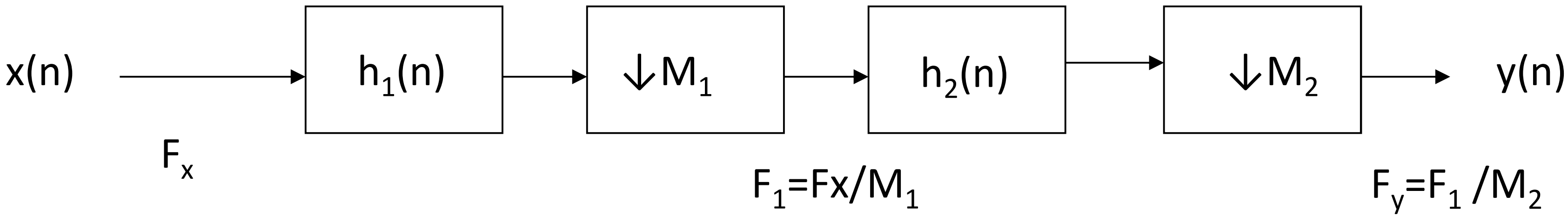
- In practice the implementation of multirate DSP systems for either L>>1 or M>>1 is done in multiple stages.

- In the case of decimation, Let's assume that M can be written as

$$M = \prod_{i=1}^{I} M_i$$

- Then decimation by M can be achieved in $I$ stages.

- For example,  M=32 can be implemented by 2 stages with $M_1$=4 and $M_2$=8 .




- $F_i$, the sampling rate after the $i^{th}$ decimation stage, is given by

$$F_i = \frac{F_{i-1}}{M_i}$$

- with $F_0 = F_x$ and $F_I = F_y$ .

- Example 5.1: Downsampling from 96 kHz to 3kHz (M=32) can be achieved in two-stages: For example, $M_1 = 16$ and $M_2 = 2$

$F_0$ =

$F_1$ =

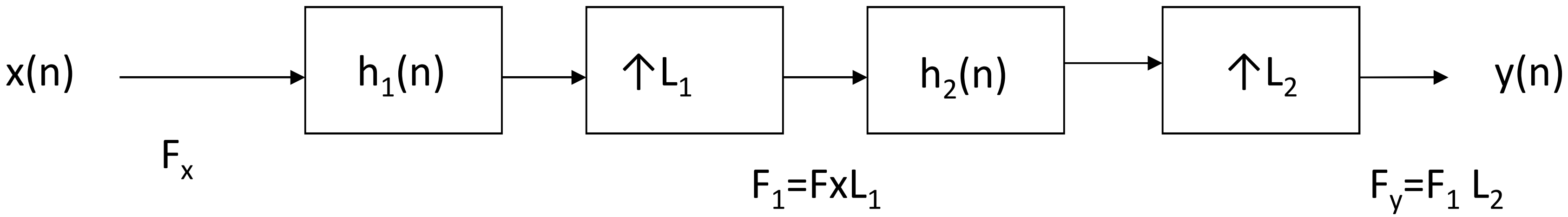
$F_2$ =

- In the case of interpolation, Let's assume that L can be written as

$$L = \prod_{i=1}^{K} L_i$$

- Then decimation by L can be achieved in $K$ stages.

- For example, L=32 can be implemented by 3 stages with $L_1$=4 and $L_2$=8 .




- $F_i$ , the sampling rate after the $i^{th}$ interpolation stage, is given by

$$F_i = L_i F_{i-1}$$

- with $F_0$=$F_x$ and $F_K = F_y$ .

- The design of a practical multistage sampling rate converter involves four steps:
  - Specify the requirements for the overall anti-aliasing or anti-imaging filters and those for individual stages;
    - What are the requirements?
  - Determine the optimum number of stages of decimation or interpolation that will yield the most efficient implementation;
  - Determine the decimation or interpolation factors for each stage;
  - Design an appropriate filter for each stage;

- Filter requirements for a multistage decimator

- For an $I$-stage decimator, the requirements for the i<sup>th</sup> filter are:

  - Passband $0 \le f \le F_p / F_{i-1}$
  - Stopband $(F_i - F_y/2)/F_{i-1} \le f \le 1/2$
  - Passband ripple $\delta_p / I$
  - Stopband ripple $\delta_s$
  - Transition bandwidth

$$\Delta f_i = \frac{\left(F_i - \frac{F_y}{2}\right) - F_p}{F_{i-1}}$$

  - Length $N_i \approx \frac{-10\log\left(\delta_p \delta_s / I\right) - 13}{14.6\left(\Delta f_i\right)} + 1$

  - The efficiency of a multistage implementation can be measured in terms of storage or the required number of multiplications/second (MPS) as follows:

$$MPS = \sum_{i=1}^{I} N_i F_i$$

- Filter requirements for a multistage interpolator

- For a $K$-stage interpolator, the requirements for the i[th] filter are:

  - Passband $0 \le f \le F_p / F_i$
  - Stopband $(F_{i-1} - F_x/2)/ F_i \le f \le 1$
  - Passband ripple $\delta_p / K$
  - Stopband ripple $\delta_s$
  - Transition bandwidth

$$\Delta f_i = \frac{\left(F_{i-1} - \frac{F_x}{2}\right) - F_p}{F_i}$$

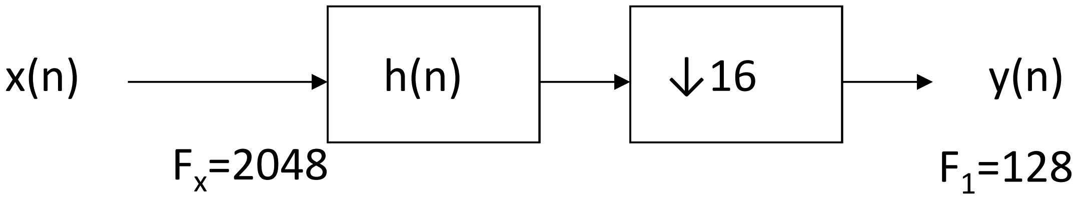
  - Length $N_i \approx \frac{-10\log(\delta_p \delta_s / K) - 13}{14.6(\Delta f_i)} + 1$

  - The efficiency of a multistage implementation $MPS = \sum_{i=1}^{K} N_i (F_i - F_{i-1})$

- Example 5.2:
    - A signal, $x(n)$, at a sampling rate of 2.048 kHz is to be decimated by a factor of 16 to yield a signal at sampling rate of 128 Hz. The signal band of interest extends from 0 to 30Hz. The overall filter must satisfy passband ripple of 0.01dB and stopband attenuation of 80dB.
        - Design the single-stage decimator and compute its efficiency
        - Design the two-stage decimator with $M_1=8$ and compute its efficiency

- Example 5.3:
    - A low pass signal (0-60Hz) sampled at a rate of 160Hz is to be interpolated by 50-fold. If $\delta_p=0.005$ and $\delta_s = 0.0001$,
        - Design the single-stage interpolator and compute its efficiency
        - Design the two-stage decimator with $L_1=2$ and compute its efficiency

- Example 5.2: the single-stage decimator




- X(n) signal contains 0-30Hz
- Passband deviation: 0.01dB -> 0.00115
- Stopband atteneuation: 80dB -> 0.0001
- We choose δp=0.00115 δs=0.0001
- Filter length given by

$$N \approx \frac{-10\log(\delta_p\ \delta_s) - 13}{14.6(\Delta f)} + 1$$

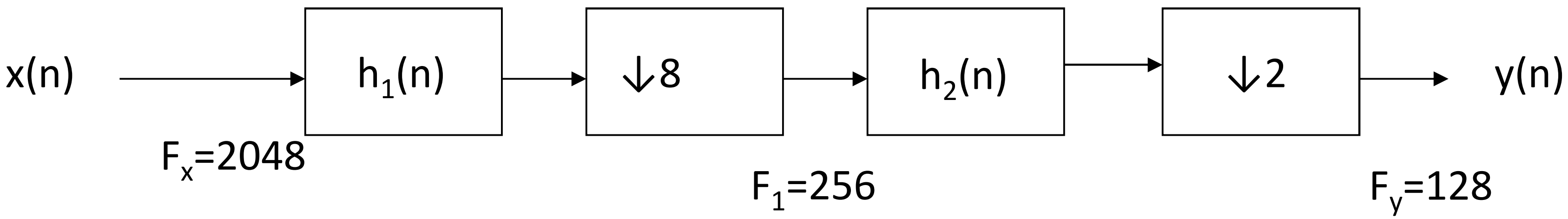
$$N \approx \frac{-10\log(0.00115 \text{ x } 0.0001) - 13}{14.6(\Delta f)} + 1$$

$$N \approx \frac{3.8625}{(\Delta f)} + 1$$

- Passband 0 - 30 Hz
- Stopband 64 -128 Hz
- Transition band 30Hz – 64Hz
- Normalised transition bandwidth (64-30)/2048 = 34/2048
- N is 3.8625 / ( 34/2048) +1 = **234**

- MPS = 234 x 128 = **29,952**

- the two-stage decimator




- We choose δp=0.00115 /2 = 0.00058 δs=0.0001
- Filter length given by

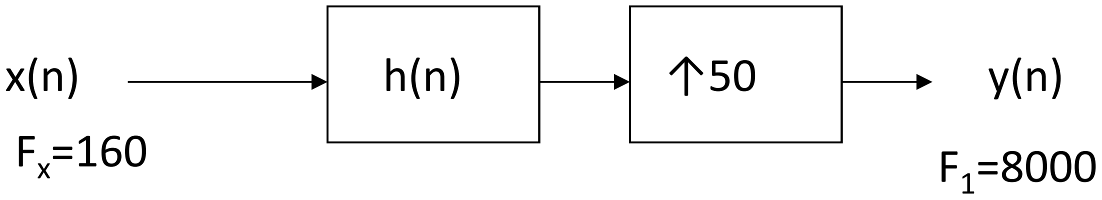
$$N \approx \frac{4.066}{(\Delta f)} + 1$$

- For $h_2$
  - Passband 0 - 30 Hz
  - Stopband 64 -128 Hz
  - Transition band 30Hz – 64Hz
  - Normalised transition bandwidth (64-30)/256 = 34/256
  - N is 4.066 / ( 34/256) +1 = **32**

- For $h_1$
  - Passband 0 - 30 Hz
  - Stopband (256-64) - 256 Hz
  - Transition band 30Hz – 192Hz
  - Normalised transition bandwidth (192-30)/2048 = 162/2048
  - N is 4.066 / ( 162/2048) +1 = **53**

- MPS = 53 x 256 + 32x64 = **15,616**

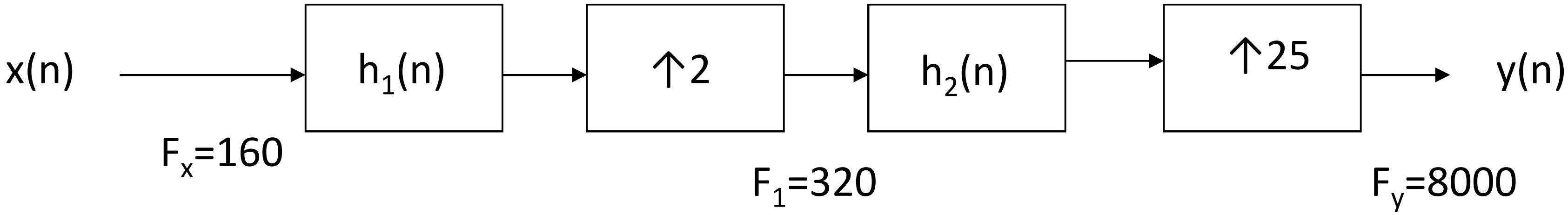
- Example 5.3: the single-stage interpolator




- X(n) signal contains 0-60Hz
- δp=0.005 δs=0.0001
- Filter length given by $N \approx \frac{-10\log(\delta_p\, \delta_s) - 13}{14.6(\Delta f)} + 1$ $N \approx \frac{3.4254}{(\Delta f)} + 1$
- Passband 0 - 60 Hz
- Stopband (160 -80) - 4000 Hz
- Transition band 60Hz – 80Hz
- Normalised transition bandwidth (80-60)/8000 = 20/8000
- N is 3.4254 / ( 20/8000) +1 = **1371**

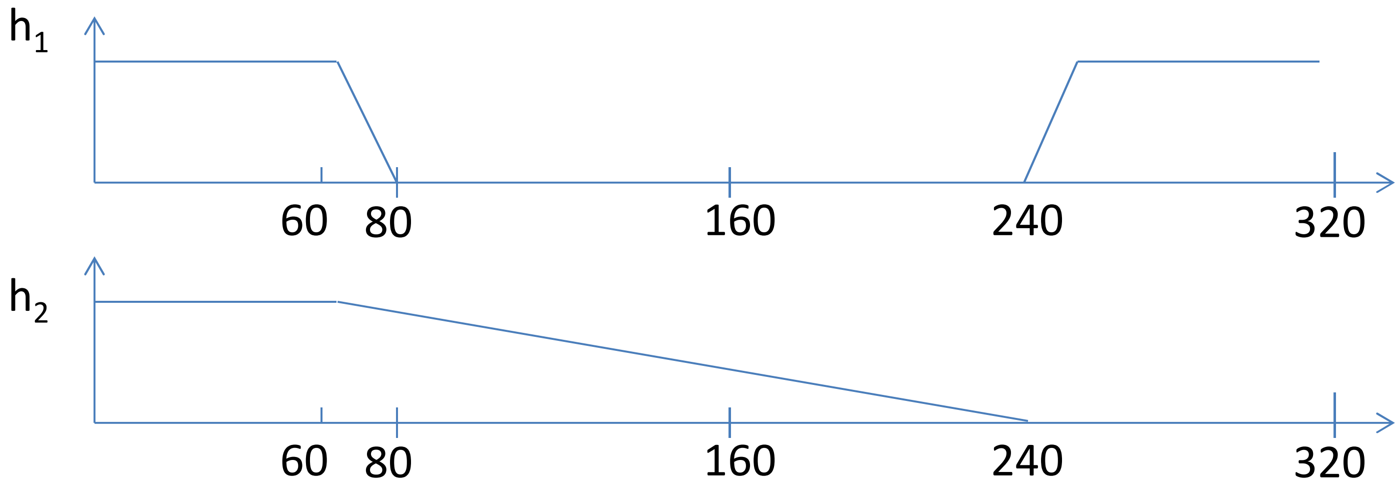
- MPS = 1371 x (8000-160) = **10.75 M (too high)**

- The multi-stage interpolator




- δp=0.005/2=0.0025　　δs=0.0001
- Filter length given by $N \approx \frac{-10\log(\delta_p\ \delta_s) - 13}{14.6(\Delta f)} + 1$　　$N \approx \frac{3.6351}{(\Delta f)} + 1$

- For $h_1$
  - Passband　0 - 60 Hz
  - Stopband　(160 -80) - 160 Hz
  - Transition band　60Hz – 80Hz
  - Normalised transition bandwidth　(80-60)/320 = 20/320
  - N is　3.6351 / ( 20/320) +1　= **60**




- For $h_2$
  - Passband 0 - 60 Hz
  - Stopband (320 -80) - 4000 Hz
  - Transition band 60Hz – 240Hz
  - Normalised transition bandwidth (240-60)/8000 = 180/8000
  - N is 3.6351 / ( 180/8000) +1 = **163**

- MPS = 60 x (320-160) + 163x(8000-320) = **1.3 M**

- Applications of Multirate DSP systems
  - Subband coding of speech
  - Hi-Fi digital audio systems
  - High quality data acquisition systems
  - Phase shifters
  - Interfacing digital systems with different sampling rates
  - Efficient implementation of narrowband filters
  - Image/wavelet coding
  - Implementation of filter banks

$$x(n) \longrightarrow \boxed{h_1(n)} \longrightarrow \boxed{\downarrow M} \longrightarrow \boxed{\uparrow M} \longrightarrow \boxed{h_2(n)} \longrightarrow y(n)$$

What can you say about y(n)?