

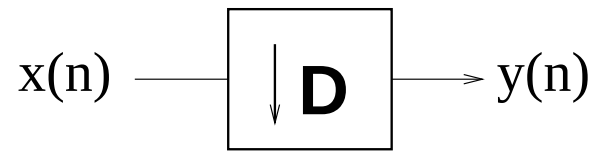
1-D Rate Conversion

- Decimation
 - Reduce the sampling rate of a discrete-time signal.
 - Low sampling rate reduces storage and computation requirements.
- Interpolation
 - Increase the sampling rate of a discrete-time signal.
 - Higher sampling rate preserves fidelity.

1-D Periodic Subsampling

- Time domain subsampling of $x(n)$ with period D

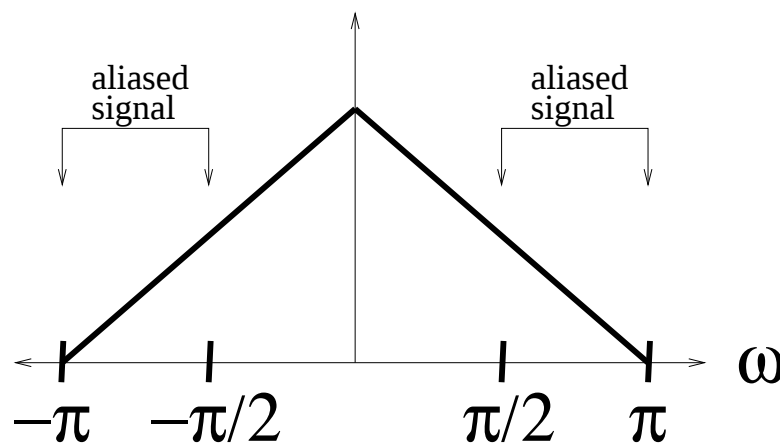
$$y(n) = x(Dn)$$



- Frequency domain representation

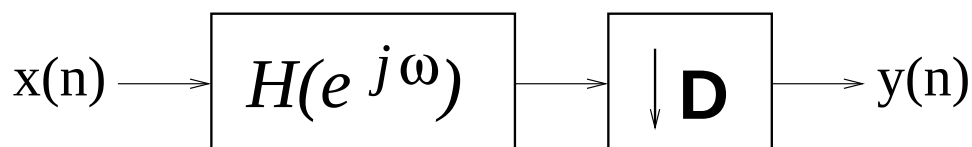
$$Y(e^{j\omega}) = \frac{1}{D} \sum_{k=-\infty}^{\infty} X(e^{j(\omega - 2\pi k)/D})$$

- Problem: Frequencies above π/D will alias.
- Example when $D = 2$



- Solution: Filter out frequencies below π/D .

Decimation System



- Apply the filter $H(e^{j\omega})$ to remove high frequencies
 - For $|\omega| < \pi$

$$H(e^{j\omega}) = \text{rect}(D\omega/\pi)$$

- For all ω

$$\begin{aligned} H(e^{j\omega}) &= \sum_{k=-\infty}^{\infty} \text{rect}\left(D\frac{\omega - k2\pi}{\pi}\right) \\ &= \text{prect}_{\pi/D}(\omega) \end{aligned}$$

- Impulse response

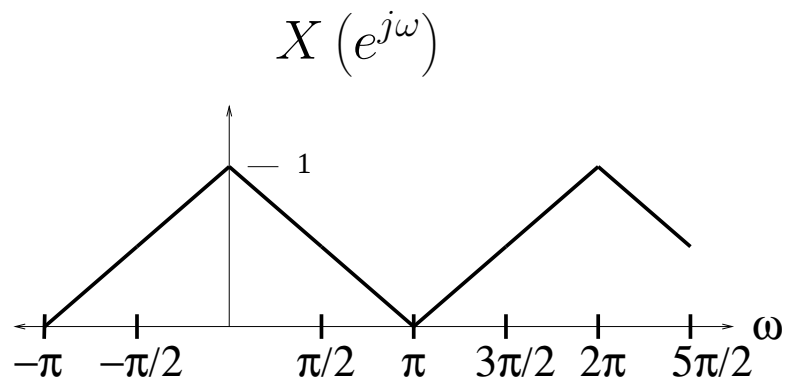
$$h(n) = \frac{1}{D} \text{sinc}(n/D)$$

- Frequency domain representation

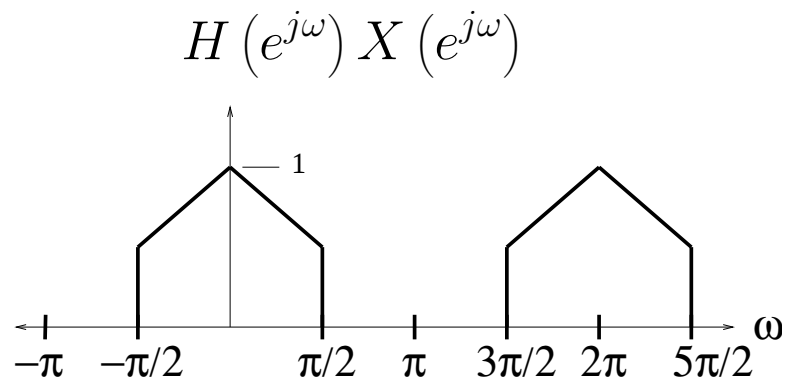
$$Y(e^{j\omega}) = \frac{1}{D} \sum_{k=-\infty}^{\infty} H(e^{j(\omega - 2\pi k)/D}) X(e^{j(\omega - 2\pi k)/D})$$

Graphical View of Decimation

- Spectral content of signal

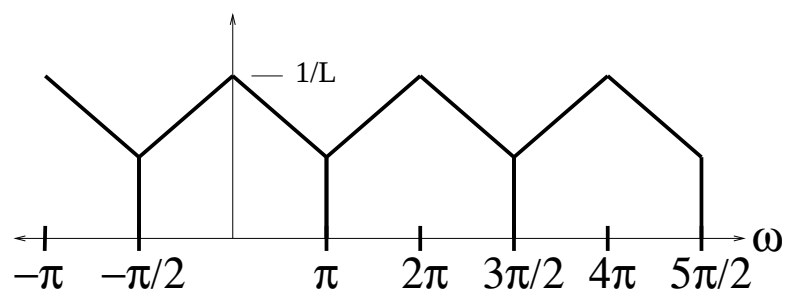


- Spectral content of filtered signal



- Spectral content of decimated signal

$$Y(e^{j\omega}) = \frac{1}{D} \sum_{k=-\infty}^{\infty} H(e^{j(\omega-2\pi k)/D}) X(e^{j(\omega-2\pi k)/D})$$



Decimation for Images

- Extension to decimation of images is direct
- Apply 2-D Filter

$$f(i, j) = h(i, j) * x(i, j)$$

- Subsample result

$$y(i, j) = f(Di, Dj)$$

- Ideal choice of filter is

$$h(m, n) = \frac{1}{D^2} \text{sinc}(m/D) \text{sinc}(n/D)$$

- Problems:
 - Filter has infinite extent.
 - Filter is not strictly positive.

Alternative Filters for Image Decimation

- Direct subsampling

$$h(m, n) = \delta(m, n)$$

- Advantages/Disadvantages:

- * Low computation
 - * Excessive aliasing

- Block averaging

$$\begin{aligned} h(m, n) = & \delta(m, n) + \delta(m + 1, n) \\ & + \delta(m, n + 1) + \delta(m + 1, n + 1) \end{aligned}$$

- Advantages/Disadvantages:

- * Low computation
 - * Some aliasing

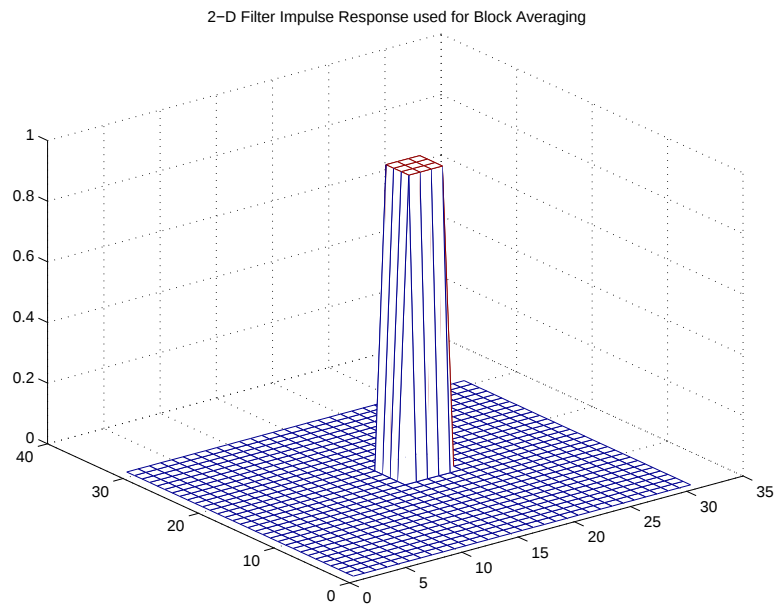
- Sinc function

$$h(m, n) = \frac{1}{D^2} \text{sinc}(m/D, n/D)$$

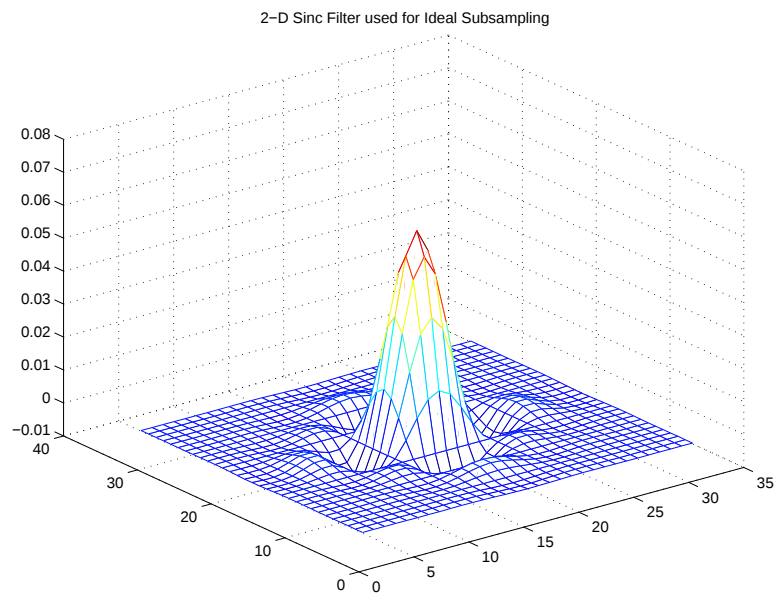
- Advantages/Disadvantages:

- * Optimal, if signal is band limited...
 - * High computation

Decimation Filters



Block averaging filter



Sinc filter

Original Image



- Full resolution

Image Decimation by 4 using Subsampling



- Severe aliasing

Image Decimation by 8 using Subsampling



- More severe aliasing

Image Decimation by 4 using Block Averaging



- Sharp, but with some aliasing

Image Decimation by 4 using Sinc Filter



- Theoretically optimal, but not necessarily the best visual quality

1-D Up-sampling

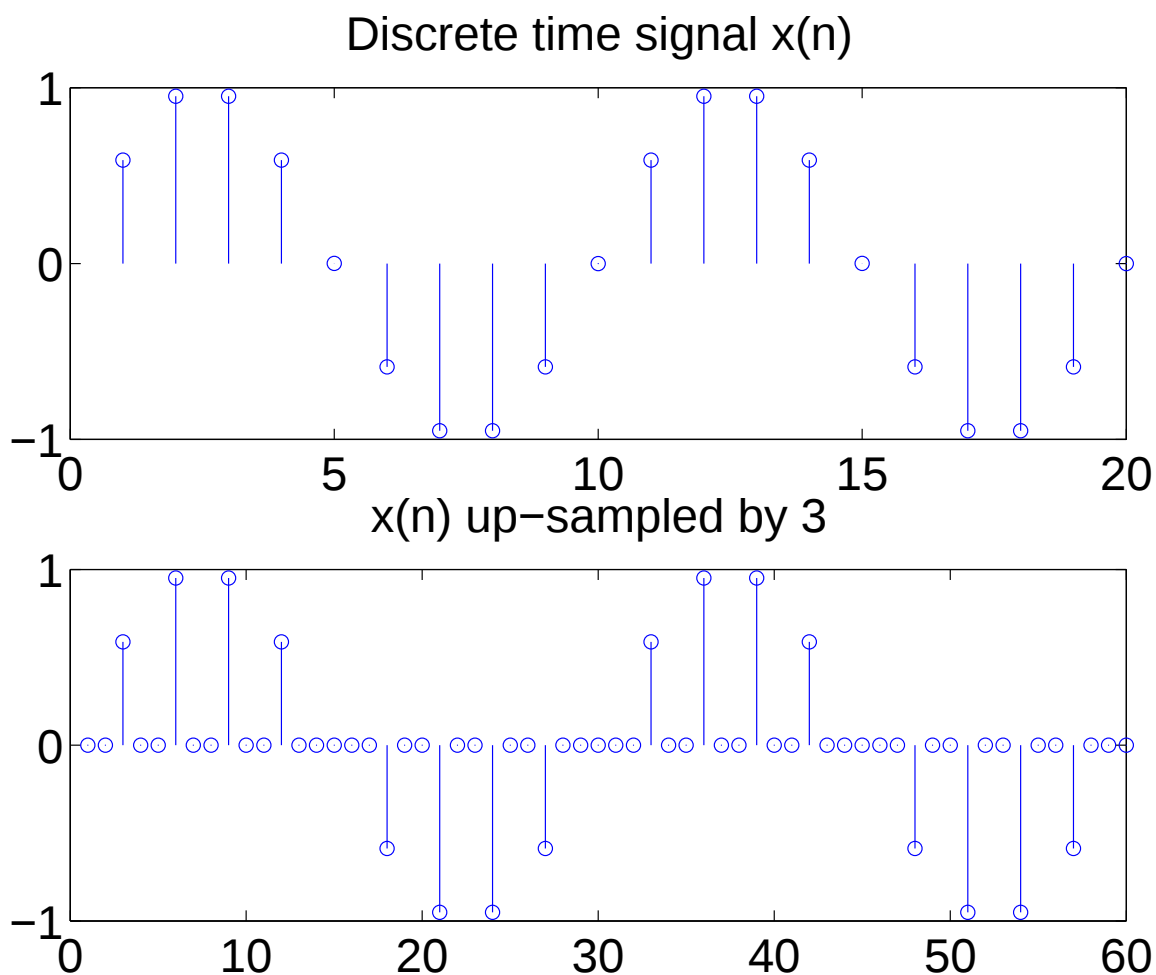
- Up-sampling by L

$$y(n) = \begin{cases} x(n/L) & \text{if } n = KL \text{ for some } K \\ 0 & \text{otherwise} \end{cases}$$

or the alternative form

$$y(n) = \sum_{-\infty}^{\infty} x(m)\delta(n - mL)$$

- Example



Up-sampling in the Frequency Domain

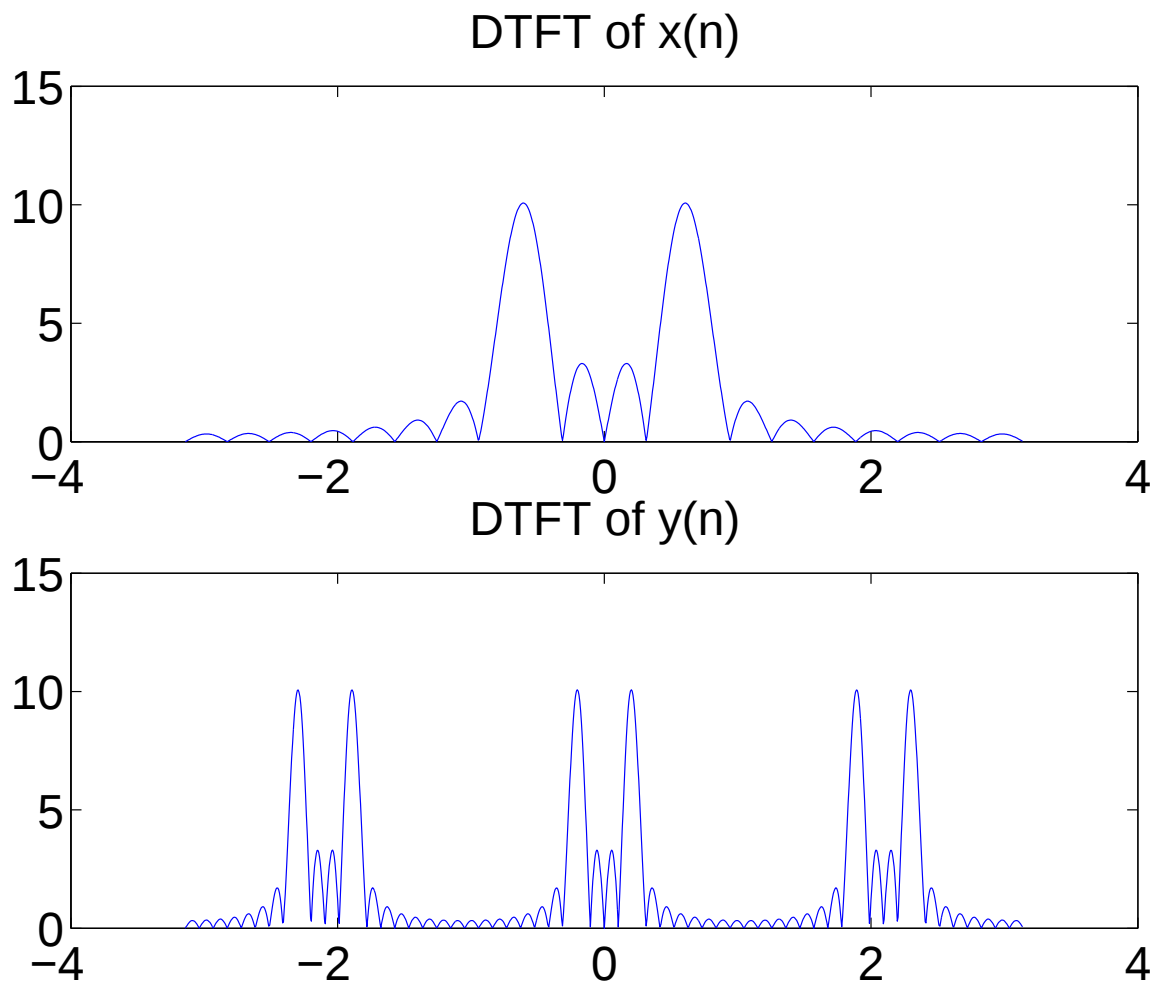
- Up-sampling by L

$$y(n) = \sum_{m=-\infty}^{\infty} x(m)\delta(n - mL)$$

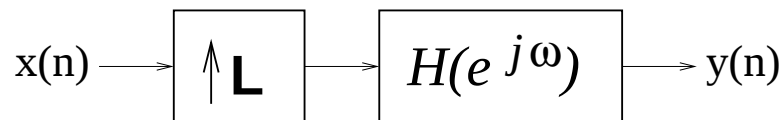
- In the frequency domain

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

- Example for $L = 3$



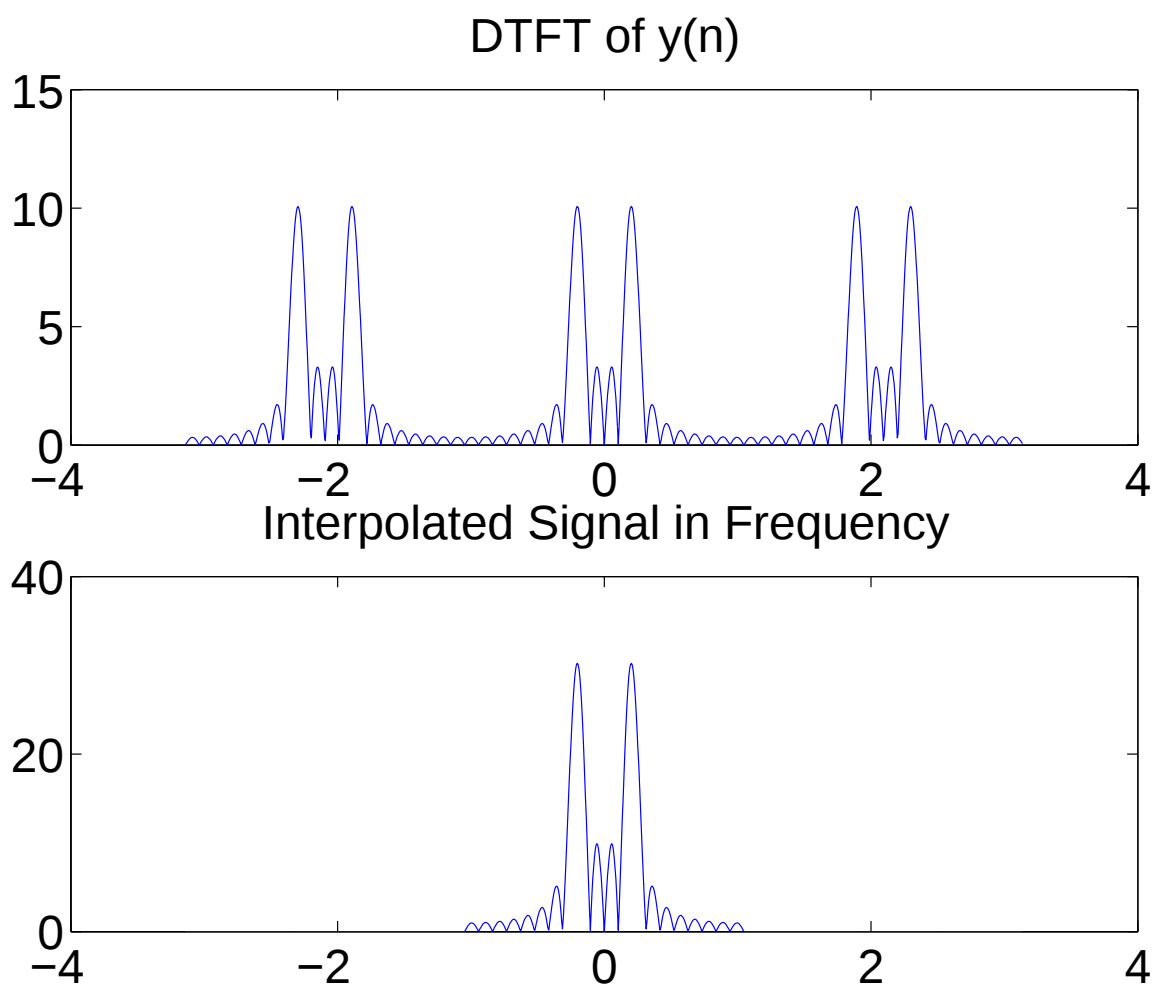
Interpolation in the Frequency Domain



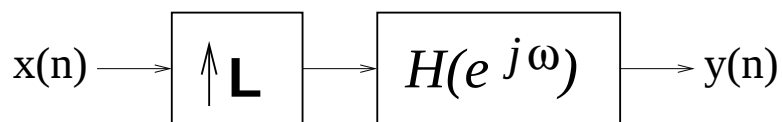
- Interpolating filter has the form

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

- Example for $L = 3$



Interpolating Filter

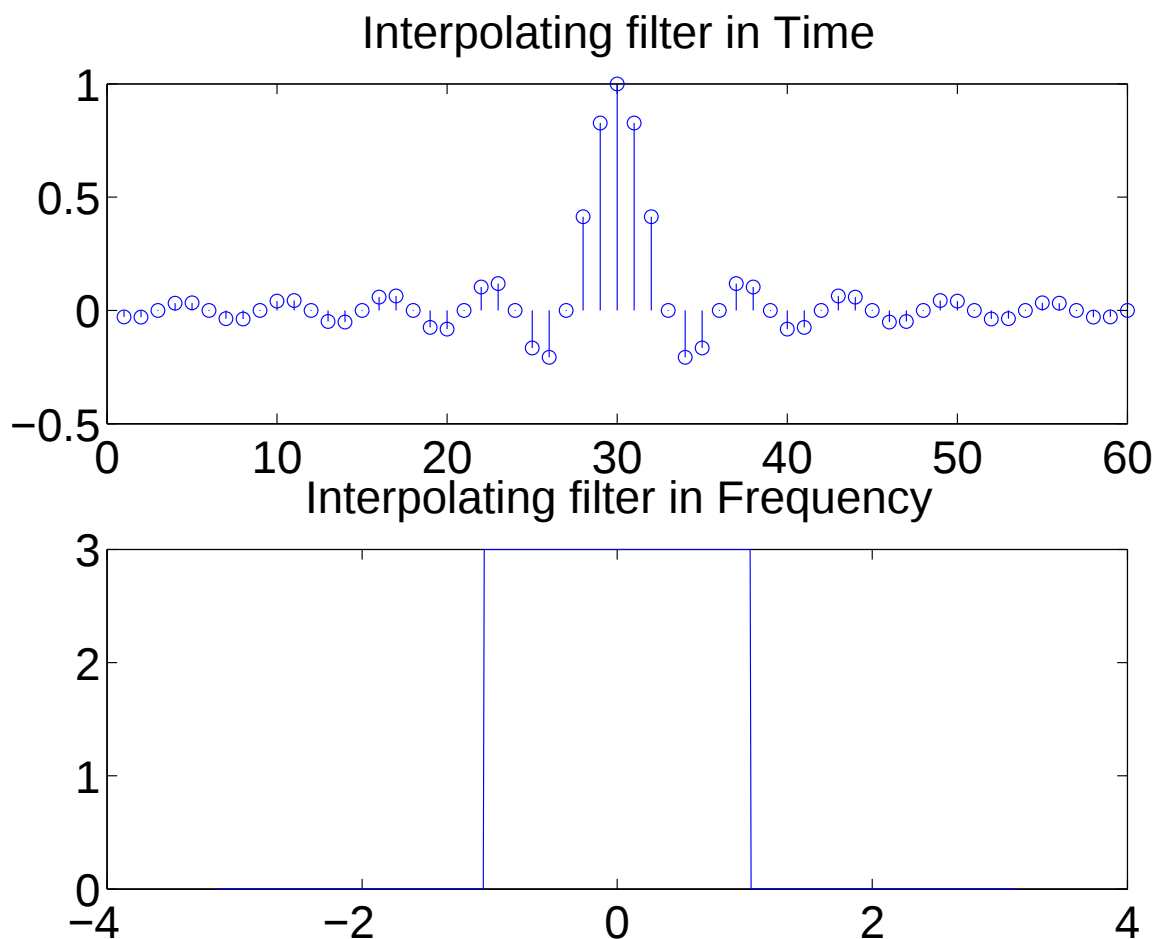


- In the frequency domain

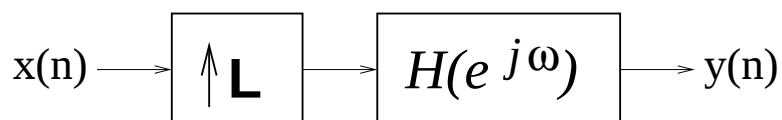
$$H(e^{j\omega}) = L \text{prect}_{2\pi/L}(\omega)$$

- In the time domain

$$h(n) = \text{sinc}\left(\frac{n}{L}\right)$$



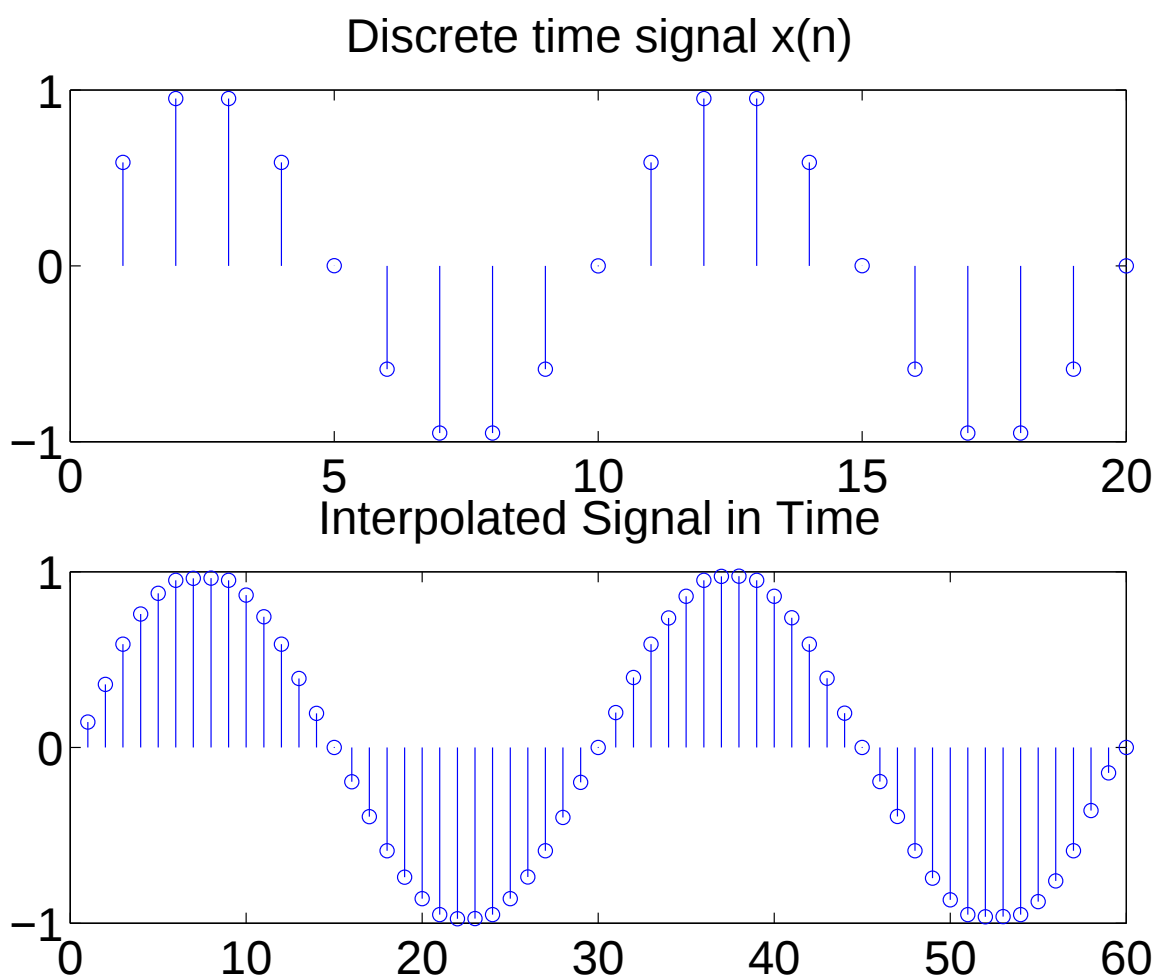
Interpolation in the Time Domain



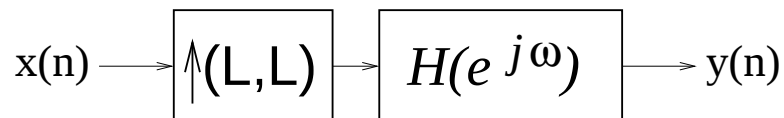
- In the frequency domain

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

- Example for $L = 3$



2-D Interpolation



- Up-sampling

$$y(m, n) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x(k, l) \delta(m - kL, n - lL)$$

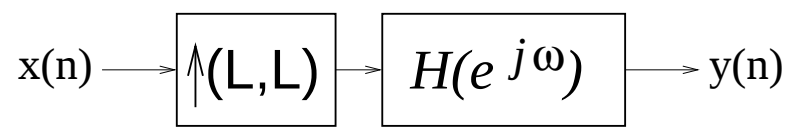
- In the frequency domain

$$H(e^{j\mu}, e^{j\nu}) = L^2 \text{prect}_{2\pi/L}(\mu) \text{prect}_{2\pi/L}(\nu)$$

$$h(m, n) = L^2 \text{sinc}(m/L, n/L)$$

- Problems: sinc function impulse response
 - Infinite support; infinite computation
 - Negative sidelobes; ringing artifacts at edges

Pixel Replication

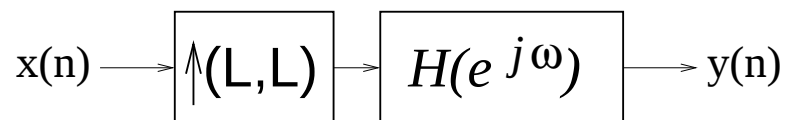


- Impulse response of filter

$$h(m, n) = \begin{cases} 1 & \text{for } 0 \leq m \leq L - 1 \text{ and } 0 \leq n \leq L - 1 \\ 0 & \text{otherwise} \end{cases}$$

- Replicates each pixel L^2 times

Bilinear Interpolation



- Impulse response of filter

$$h(m, n) = \Lambda(m/L)\Lambda(n/L)$$

- Results in linear interpolation of intermediate pixels