ESE 531: Digital Signal Processing

Lec 9: February 14th, 2019

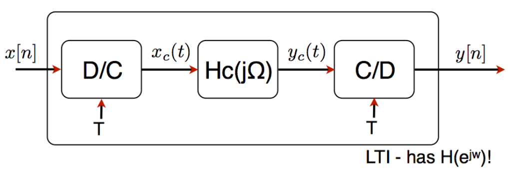
Downsampling/Upsampling and Practical Interpolation

Lecture Outline

- CT processing of DT signals
- Downsampling
- Upsampling
- Practical Interpolation (time permitting)

Continuous-Time Processing of Discrete-Time

- Useful to interpret DT systems with no simple interpretation in discrete time

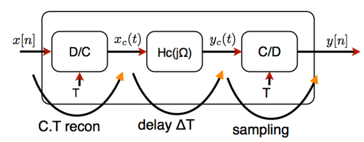


Example: Non-integer Delay

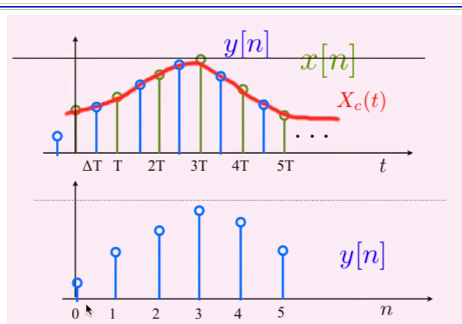
- What is the time domain operation when Δ is non-integer? I.e Δ=1/2

$$H(e^{j\omega}) = e^{-j\omega\Delta}$$

Let: $H_c(j\Omega) = e^{-j\Omega\Delta T}$ delay of ΔT in time

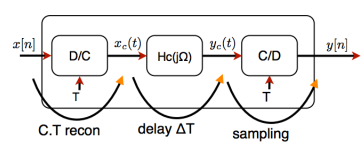


Example: Non-integer Delay



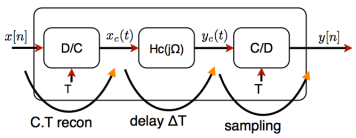
Example: Non-integer Delay

- The block diagram is for interpretation/analysis only



Example: Non-integer Delay

- The block diagram is for interpretation/analysis only



$$y[n] = y_c(nT) = x_c(nT - T\cdot\Delta) \qquad x_c(t) = \sum_k x[k]\text{sinc}\left(\frac{t-kT}{T}\right)$$

$$x_c(nT - T\cdot\Delta) = \sum_k x[k]\text{sinc}\left(\frac{nT - T\cdot\Delta - kT}{T}\right) = \sum_k x[k]\text{sinc}\left(n - \Delta - k\right)$$

Example: Non-integer Delay

- Delay system has an impulse response of a sinc with a continuous time delay

$$y[n] = \sum_k x[k]\text{sinc}\left(n - \Delta - k\right)$$
$$= x[n] * \text{sinc}\left(n - \Delta\right)$$

$$\Rightarrow h[n] = \text{sinc}(n - \Delta)$$

Example: Non-integer Delay

- What is the time domain operation when Δ is non-integer? I.e $\Delta = 1/2$

$$H(e^{j\omega}) = e^{-j\omega\Delta}$$

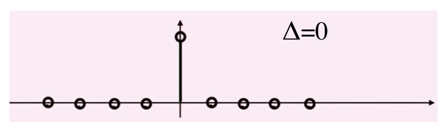
$$\delta[n] \leftrightarrow 1$$
$$\delta[n - n_d] \leftrightarrow e^{-j\omega n_d}$$

$$\Rightarrow h[n] = \text{sinc}(n - \Delta)$$

Example: Non-integer Delay

- My delay system has an impulse response of a sinc with a continuous time delay

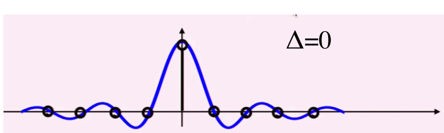
$$h[n] = \text{sinc}(n - \Delta)$$



Example: Non-integer Delay

- My delay system has an impulse response of a sinc with a continuous time delay

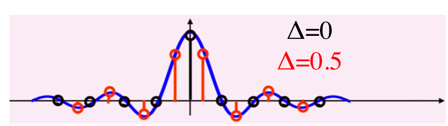
$$h[n] = \text{sinc}(n - \Delta)$$



Example: Non-integer Delay

- My delay system has an impulse response of a sinc with a continuous time delay

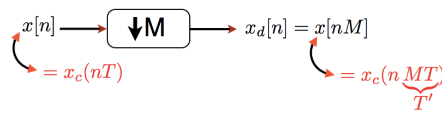
$$h[n] = \text{sinc}(n - \Delta)$$



Downsampling

Downsampling

- Definition: Reducing the sampling rate by an integer number (M>1)



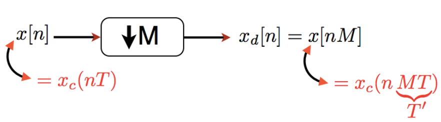
Downsampling

- Similar to C/D conversion
  - Need to worry about aliasing
  - Use anti-aliasing filter to mitigate effects

Downsampling

- Similar to C/D conversion
  - Need to worry about aliasing
  - Use anti-aliasing filter to mitigate effects
- If your discrete time signal is finely sampled (i.e oversampled) almost like a CT signal
  - Downsampling is just like sampling (C/D conversion)

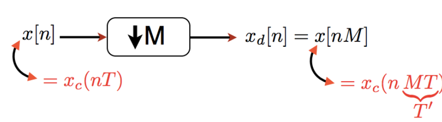
Downsampling



The DTFT:

$$X(e^{j\omega}) = \frac{1}{T}\sum_k X_c\left(j\left(\underbrace{\frac{\omega}{T}}_{\Omega} - \underbrace{\frac{2\pi}{T}}_{\Omega_s}k\right)\right)$$

Downsampling



The DTFT:

$$X(e^{j\omega}) = \frac{1}{T}\sum_k X_c\left(j\left(\underbrace{\frac{\omega}{T}}_{\Omega} - \underbrace{\frac{2\pi}{T}}_{\Omega_s}k\right)\right)$$

$$X_d(e^{j\omega}) = \frac{1}{MT}\sum_k X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi}{MT}k\right)\right)$$

Downsampling

The DTFT:

$$X(e^{j\omega}) = \frac{1}{T}\sum_k X_c\left(j\left(\underbrace{\frac{\omega}{T}}_{\Omega} - \underbrace{\frac{2\pi}{T}}_{\Omega_s}k\right)\right)$$

$$X_d(e^{j\omega}) = \frac{1}{MT}\sum_k X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi}{MT}k\right)\right)$$

- Want to relate $X_d(e^{j\omega})$ to $X(e^{j\omega})$ not $X_c(j\Omega)$
- Separate sum into two sums—fine sum and coarse sum (i.e like counting minutes within hours)

Downsampling

The DTFT:

$$X(e^{j\omega}) = \frac{1}{T}\sum_k X_c\left(j\left(\underbrace{\frac{\omega}{T}}_{\Omega} - \underbrace{\frac{2\pi}{T}}_{\Omega_s}k\right)\right)$$

$$X_d(e^{j\omega}) = \frac{1}{MT}\sum_k X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi}{MT}k\right)\right)$$

- k=rM+i
  - i = 0, 1, …, M-1
  - r = -∞, …, ∞

Downsampling

$$X_d(e^{j\omega}) = \frac{1}{MT}\sum_k X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi}{MT}k\right)\right)$$

$$= \frac{1}{M}\sum_{i=0}^{M-1}\underbrace{\frac{1}{T}\sum_{r=-\infty}^{\infty} X_c\left(j\left(\frac{\omega}{MT} - \frac{2\pi}{MT}i - \frac{2\pi}{T}r\right)\right)}_{X\left(e^{j\left(\frac{\omega}{M} - \frac{2\pi}{M}i\right)}\right)}$$

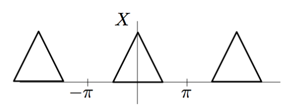
$$X(e^{j\omega}) = \frac{1}{T}\sum_k X_c\left(j\left(\underbrace{\frac{\omega}{T}} - \underbrace{\frac{2\pi}{T}}k\right)\right)$$

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

scale by 1/M — stretch by M — replicate

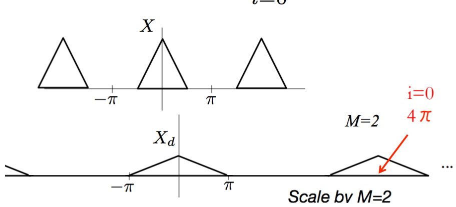
Example: M=2

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



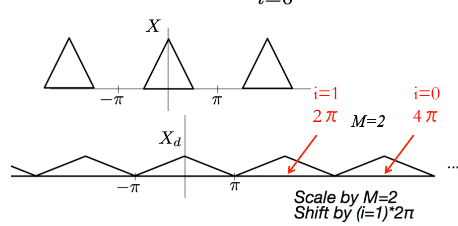
Example: M=2

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



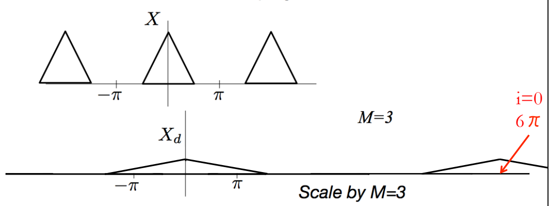
Example: M=2

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



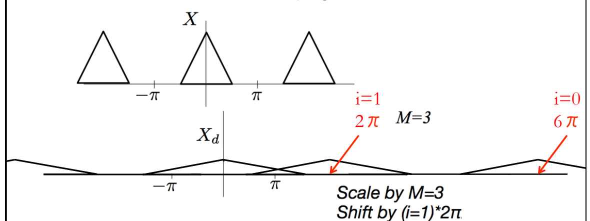
Example: M=3

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



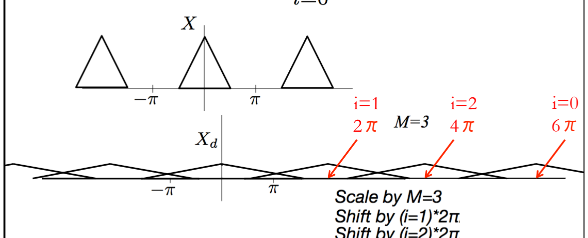
Example: M=3

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

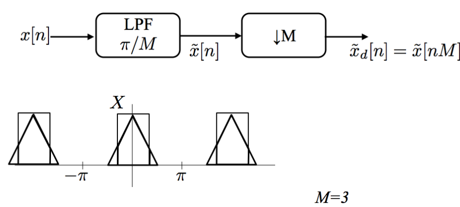


Example: M=3

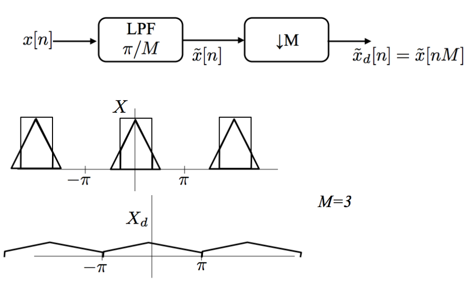
$$X_d(e^{j\omega}) = \frac{1}{M}\sum_{i=0}^{M-1} X\left(e^{j\left(\frac{w}{M}-\frac{2\pi}{M}i\right)}\right)$$



Example: M=3



Example: M=3



Upsampling

Upsampling

- Definition: Increasing the sampling rate by an integer number

$x[n] = x_c(nT)$

$x_i[n] = x_c(nT')$ where $T' = \frac{T}{L}$ $\quad L$ integer

Upsampling

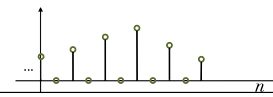
- Definition: Increasing the sampling rate by an integer number

$x[n] = x_c(nT)$

$x_i[n] = x_c(nT')$ where $T' = \frac{T}{L}$ $\quad L$ integer

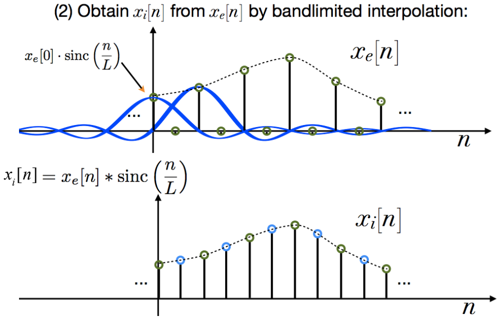
Obtain $x_i[n]$ from $x[n]$ in two steps:

(1) Generate: $x_e[n] = \begin{cases} x[n/L] & n = 0, \pm L, \pm 2L, \cdots \\ 0 & \text{otherwise} \end{cases}$



Upsampling

(2) Obtain $x_i[n]$ from $x_e[n]$ by bandlimited interpolation:



Upsampling

- Much like D/C converter
- Upsample by A LOT → almost continuous
- Intuition:
  - Recall our D/C model: x[n] → $x_s(t)$ → $x_c(t)$
  - Approximate "$x_s(t)$" by placing zeros between samples
  - Convolve with a sinc to obtain "$x_c(t)$"

Upsampling

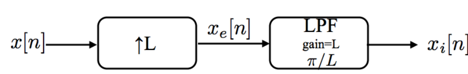
$$x_i[n] = x_e[n] * \text{sinc}(n/L)$$

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-kL]$$

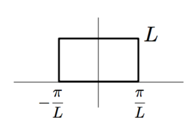
$$x_i[n] = \sum_{k=-\infty}^{\infty} x[k]\text{sinc}\left(\frac{n-kL}{L}\right)$$

Frequency Domain Interpretation

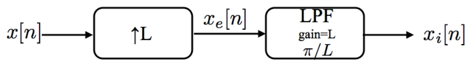
$$x_i[n] = x_e[n] * \text{sinc}(n/L)$$



$\text{sinc}(n/L)$ DTFT ⇒

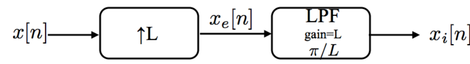


Frequency Domain Interpretation



$$X_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \underbrace{x_e[n]}_{\neq 0 \text{ only for n=mL (integer m)}} e^{-j\omega n}$$

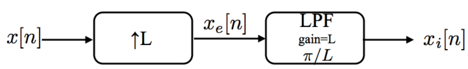
Frequency Domain Interpretation



$$X_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \underbrace{x_e[n]}_{\neq 0 \text{ only for n=mL (integer m)}} e^{-j\omega n}$$

$$= \sum_{m=-\infty}^{\infty} \underbrace{x_e[mL]} e^{-j\omega mL}$$

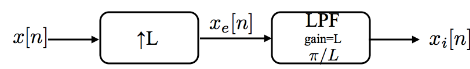
Frequency Domain Interpretation



$$X_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \underbrace{x_e[n]}_{\neq 0 \text{ only for n=mL (integer m)}} e^{-j\omega n}$$

$$= \sum_{m=-\infty}^{\infty} \underbrace{x_e[mL]}_{=x[m]} e^{-j\omega mL}$$

Frequency Domain Interpretation

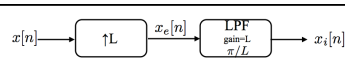


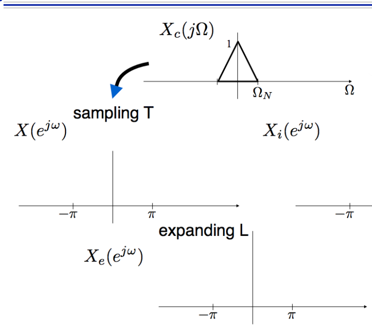
$$X_e(e^{j\omega}) = \sum_{n=-\infty}^{\infty} \underbrace{x_e[n]}_{\neq 0 \text{ only for n=mL (integer m)}} e^{-j\omega n}$$

$$= \sum_{m=-\infty}^{\infty} \underbrace{x_e[mL]}_{=x[m]} e^{-j\omega mL} = X(e^{j\omega L})$$

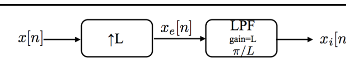
Shrink DTFT by a factor of L!

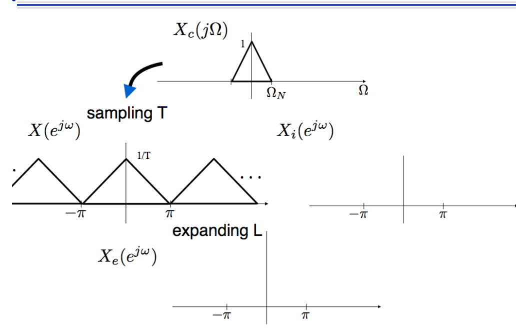
Example



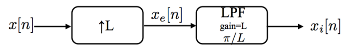


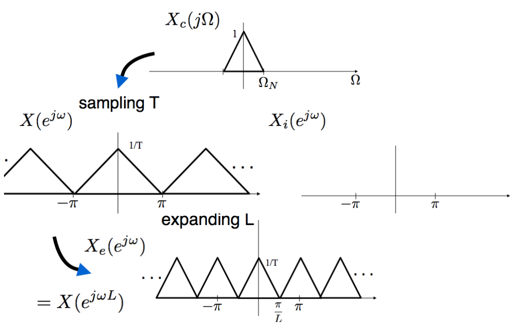
Example



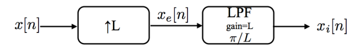


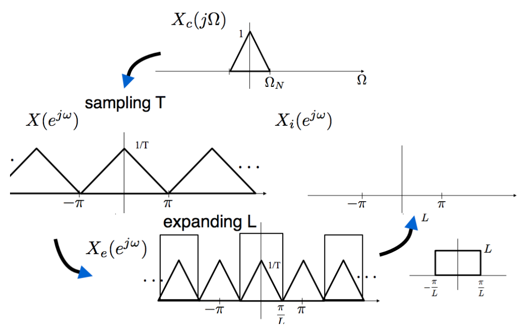
Example



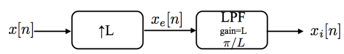


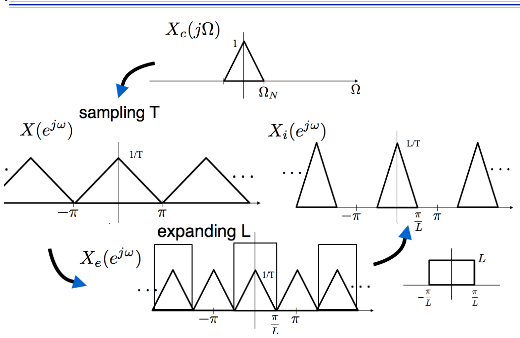
Example



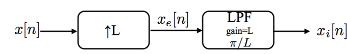


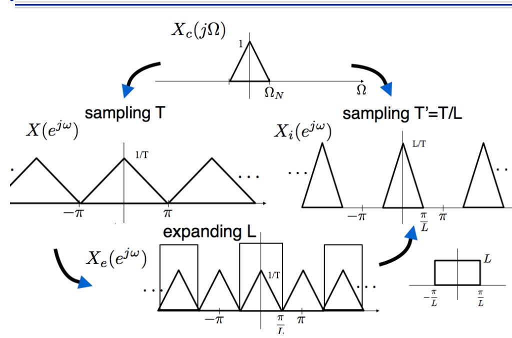
Example





Example

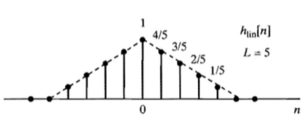




Practical Interpolation

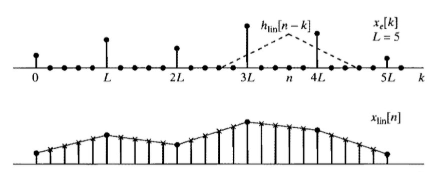
- Interpolate with simple, practical filters
  - Linear interpolation – samples between original samples fall on a straight line connecting the samples
    - Convolve with triangle instead of sinc

$$h_{\text{lin}}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise,} \end{cases}$$



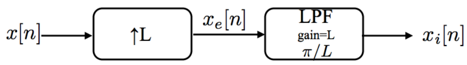
Practical Interpolation

- Interpolate with simple, practical filters
  - Linear interpolation – samples between original samples fall on a straight line connecting the samples
    - Convolve with triangle instead of sinc

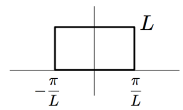


Frequency Domain Interpretation

$$x_i[n] = x_e[n] * \text{sinc}(n/L)$$

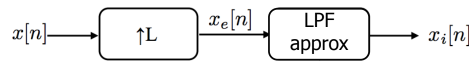


sinc(n/L) DTFT ⇒

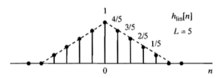


Linear Interpolation -- Frequency Domain

$$x_i[n] = x_e[n] * h_{lin}[n]$$

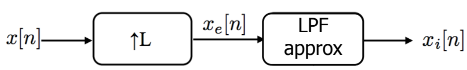


$$h_{lin}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise,} \end{cases}$$

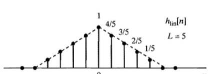


Linear Interpolation -- Frequency Domain

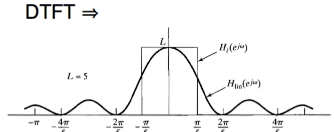
$$x_i[n] = x_e[n] * h_{lin}[n]$$



$$h_{lin}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise,} \end{cases}$$

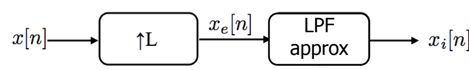


DTFT ⇒

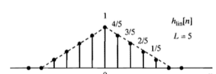


Linear Interpolation -- Frequency Domain

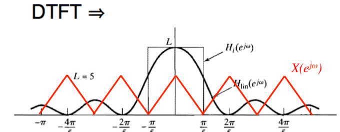
$$x_i[n] = x_e[n] * h_{lin}[n]$$



$$h_{lin}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise,} \end{cases}$$

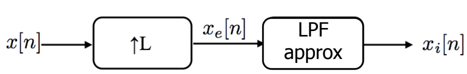


DTFT ⇒

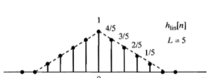


Linear Interpolation -- Frequency Domain

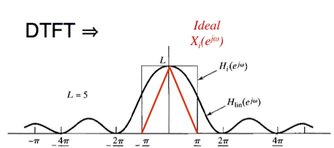
$$x_i[n] = x_e[n] * h_{lin}[n]$$



$$h_{lin}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise,} \end{cases}$$

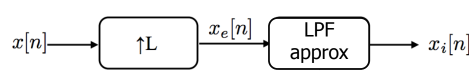


DTFT ⇒

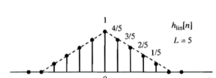


Linear Interpolation -- Frequency Domain

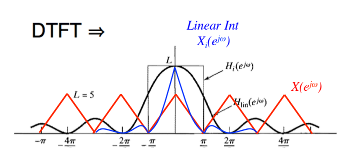
$$x_i[n] = x_e[n] * h_{lin}[n]$$



$$h_{lin}[n] = \begin{cases} 1 - |n|/L, & |n| \le L, \\ 0, & \text{otherwise,} \end{cases}$$



DTFT ⇒



Big Ideas

- CT processing of DT signals
  - Allows for interpretation of DT systems
- Downsampling
  - Like a C/D converter
- Upsampling
  - Like a D/C converter
- Practical Interpolation
  - Linear interpolation
  - Approximate sinc function with triangle

Admin

- HW 4 due Sunday