ESE 531: Digital Signal Processing

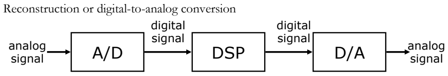
Lec 25: April 28, 2020
Review

Course Content

- Introduction
- Discrete Time Signals & Systems
- Discrete Time Fourier Transform
- Z-Transform
- Inverse Z-Transform
- Sampling of Continuous Time Signals
- Frequency Domain of Discrete Time Series
- Downsampling/Upsampling
- Data Converters, Sigma Delta Modulation
- Oversampling, Noise Shaping
- Frequency Response of LTI Systems
- Basic Structures for IIR and FIR Systems
- Design of IIR and FIR Filters
- Filter Banks
- Adaptive Filters
- Computation of the Discrete Fourier Transform
- Fast Fourier Transform
- Spectral Analysis
- Wavelet Transform
- Compressive Sampling

Digital Signal Processing

- Represent signals by a sequence of numbers
  - Sampling and quantization (or analog-to-digital conversion)
- Perform processing on these numbers with a digital processor
  - Digital signal processing
- Reconstruct analog signal from processed numbers
  - Reconstruction or digital-to-analog conversion



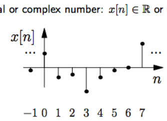
- Analog input → analog output
  - Eg. Digital recording music
- Analog input → digital output
  - Eg. Touch tone phone dialing, speech to text
- Digital input → analog output
  - Eg. Text to speech
- Digital input → digital output
  - Eg. Compression of a file on computer

Discrete Time Signals and Systems

Signals are Functions

DEFINITION: A **signal** is a function that maps an independent variable to a dependent variable.

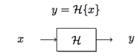
- Signal $x[n]$: each value of n produces the value $x[n]$
- In this course, we will focus on **discrete-time** signals:
  - Independent variable is an **integer**: $n \in \mathbb{Z}$ (will refer to as time)
  - Dependent variable is a real or complex number: $x[n] \in \mathbb{R}$ or $\mathbb{C}$



Discrete Time Systems

DEFINITION: A discrete-time **system** $\mathcal{H}$ is a transformation (a rule or formula) that maps a discrete-time input signal x into a discrete-time output signal y

$$y = \mathcal{H}\{x\}$$



- Systems manipulate the information in signals
- Examples:
  - A speech recognition system converts acoustic waves of speech into text
  - A radar system transforms the received radar pulse to estimate the position and velocity of targets
  - A functional magnetic resonance imaging (fMRI) system transforms measurements of electron spin into voxel-by-voxel estimates of brain activity
  - A 30 day moving average smooths out the day-to-day variability in a stock price

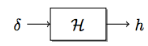
System Properties

- Causality
  - y[n] only depends on x[m] for m<=n
- Linearity
  - Scaled sum of arbitrary inputs results in output that is a scaled sum of corresponding outputs
    - $Ax_1[n]+Bx_2[n] \rightarrow Ay_1[n]+By_2[n]$
- Memoryless
  - y[n] depends only on x[n]
- Time Invariance
  - Shifted input results in shifted output
    - $x[n-q] \rightarrow y[n-q]$
- BIBO Stability
  - A bounded input results in a bounded output (ie. max signal value exists for output if max)

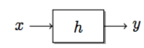
LTI Systems

DEFINITION: A system $\mathcal{H}$ is **linear time-invariant** (LTI) if it is both linear and time-invariant

- LTI system can be completely characterized by its impulse response



- Then the output for an arbitrary input is a sum of weighted, delay impulse responses



$$y[n] = \sum_{m=-\infty}^{\infty} h[n-m]\,x[m]$$

$$y[n] = x[n]*h[n]$$

Discrete Time Fourier Transform

Penn

DTFT Definition

$$X(e^{j\omega}) = \sum_{k=-\infty}^{\infty} x[k]e^{-j\omega k}$$

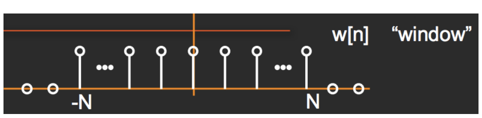
$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n}\,d\omega$$

Alternate

$$X(f) = \sum_{k=-\infty}^{\infty} x[k]e^{-j2\pi fk}$$

$$x[n] = \int_{-0.5}^{0.5} X(f)e^{j2\pi fn}\,df$$

Example: Window DTFT



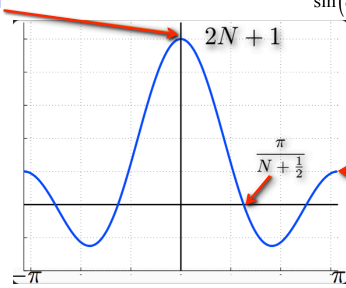
$$W(e^{j\omega}) = \sum_{k=-\infty}^{\infty} w[k]e^{-j\omega k}$$

$$= \sum_{k=-N}^{N} e^{-j\omega k}$$

Example: Window DTFT

$$W(e^{j\omega}) = \frac{\sin\left((N+1/2)\omega\right)}{\sin\left(\omega/2\right)}$$

Also, Σx[n]



=1 why?

Plot for N=2

LTI System Frequency Response

- Fourier Transform of impulse response

$x[n]=e^{j\omega n}$ → LTI System → $y[n]=H(e^{j\omega n})e^{j\omega n}$

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega k}$$

z-Transform

- The z-transform generalizes the Discrete-Time Fourier Transform (DTFT) for analyzing infinite-length signals and systems
- Very useful for designing and analyzing signal processing systems
- Properties are very similar to the DTFT with a few caveats

$$H(z) = \sum_{n=-\infty}^{\infty} h[n]z^{-n}$$

Region of Convergence (ROC)

DEFINITION

Given a time signal $x[n]$, the **region of convergence** (ROC) of its z-transform $X(z)$ is the set of $z \in \mathbb{C}$ such that $X(z)$ converges, that is, the set of $z \in \mathbb{C}$ such that $x[n]\,z^{-n}$ is absolutely summable

$$\sum_{n=-\infty}^{\infty} |x[n]\,z^{-n}| < \infty$$

Inverse z-Transform

- Ways to avoid it:
  - Inspection (known transforms)
  - Properties of the z-transform
  - Partial fraction expansion

$$X(z) = \frac{b_0}{a_0}\frac{\prod_{k=1}^{M}(1-c_k z^{-1})}{\prod_{k=1}^{N}(1-d_k z^{-1})} = \sum_{k=1}^{N}\frac{A_k}{1-d_k z^{-1}}$$

  - Power series expansion

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$
$$= \cdots + x[-2]z^{2} + x[-1]z + x[0] + x[1]z^{-1} + x[2]z^{-2} + \cdots$$

Difference Equation to z-Transform

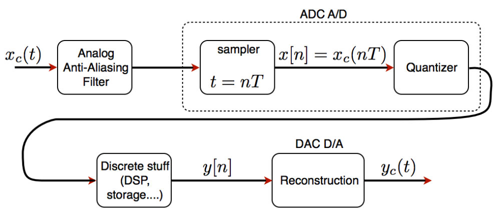
$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{m=0}^{M} b_m x[n-m]$$

$$H(z) = \frac{\sum_{m=0}^{M}(b_k)z^{-k}}{\sum_{k=0}^{N}(a_k)z^{-k}}$$

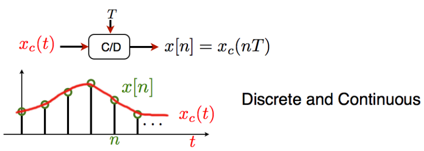
- Difference equations of this form behave as causal LTI systems
  - when the input is zero prior to n=0
  - Initial rest equations are imposed prior to the time when input becomes nonzero
    - i.e y[-N]=y[-N+1]=…=y[-1]=0

Sampling and Reconstruction

DSP System

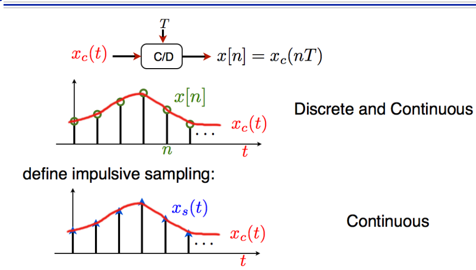


Ideal Sampling Model



- Ideal continuous-to-discrete time (C/D) converter
  - T is the sampling period
  - f_s=1/T is the sampling frequency
  - Ω_s=2π/T

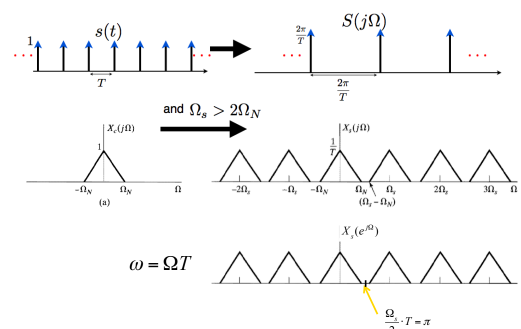
Ideal Sampling Model



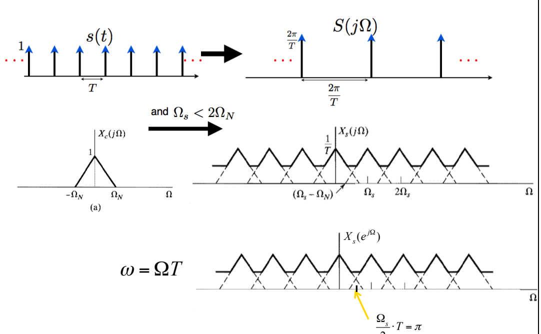
$$x_s(t) = \cdots + x_c(0)\delta(t) + x_c(T)\delta(t-T) + \cdots$$

$$x_s(t) = x_c \sum_{n=-\infty}^{\infty} \delta(t - nT)$$

Frequency Domain Analysis

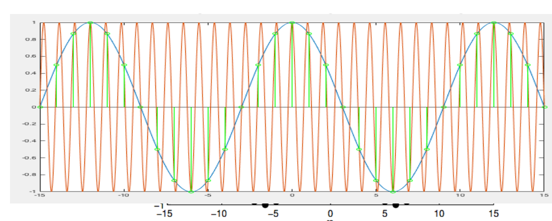


Frequency Domain Analysis



Aliasing Example

- $x_1[n] = \cos\left(\frac{\pi}{6}n\right)$

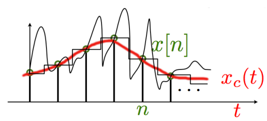


Reconstruction of Bandlimited Signals

- Nyquist Sampling Theorem: Suppose $x_c(t)$ is bandlimited. I.e.

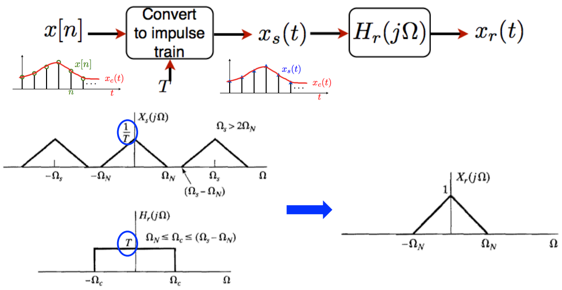
$$X_c(j\Omega) = 0 \quad \forall \quad |\Omega| \geq \Omega_N$$

- If $\Omega_s \geq 2\Omega_N$, then $x_c(t)$ can be uniquely determined from its samples $x[n]=x_c(nT)$
- Bandlimitedness is the key to uniqueness



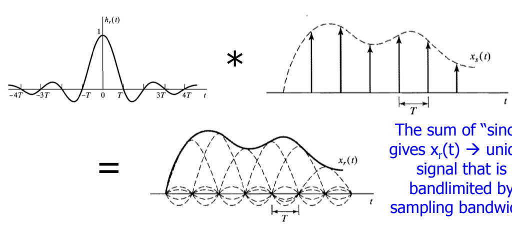
Mulitiple signals go through the samples, but only one is bandlimited within our sampling band

Reconstruction in Frequency Domain



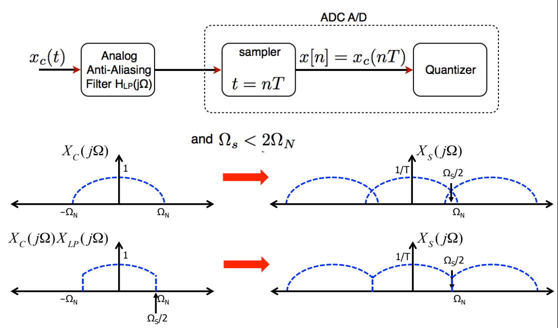
Reconstruction in Time Domain

$$x_r(t) = x_s(t) * h_r(t) = \left(\sum_n x[n]\delta(t-nT)\right) * h_r(t) = \sum_n x[n]h_r(t-nT)$$



The sum of "sincs" gives $x_r(t)$ → unique signal that is bandlimited by sampling bandwidth

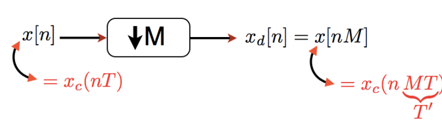
Anti-Aliasing Filter



Rate Re-Sampling

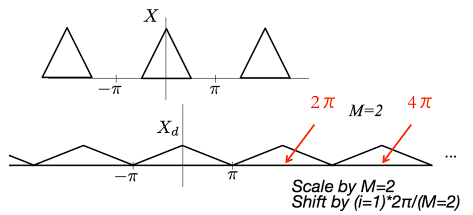
Downsampling

- Definition: Reducing the sampling rate by an integer number



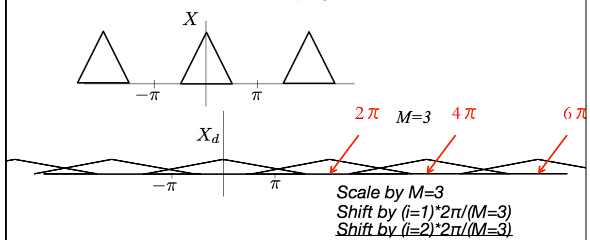
Example: M=2

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

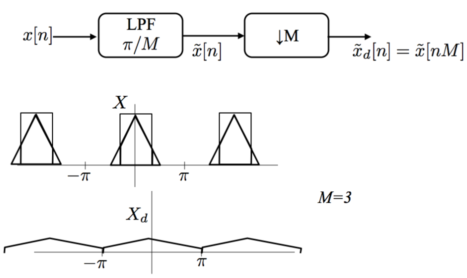


Example: M=3

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



Example: M=3 w/ Anti-aliasing



Upsampling

- Definition: Increasing the sampling rate by an integer number

$x[n] = x_c(nT)$

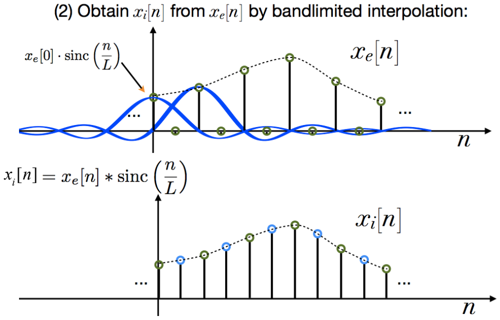
$x_i[n] = x_c(nT')$ where $T' = \frac{T}{L}$ $\quad L$ integer

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

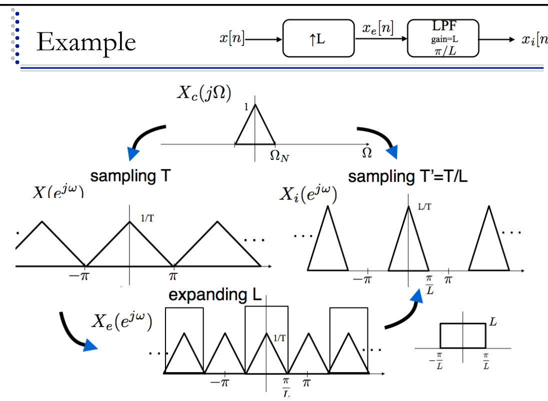
(1) Generate: $x_e[n] = \begin{cases} x[n/L] & n = 0, \ \pm L, \ \pm 2L, \cdots \\ 0 & \text{otherwise} \end{cases}$

Upsampling

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

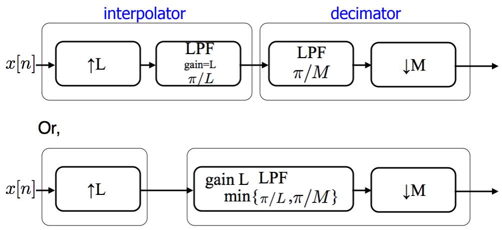


Example



Non-integer Sampling

- T'=TM/L
  - Upsample by L, then downsample by M



Interchanging Operations

not LTI!

"expander"

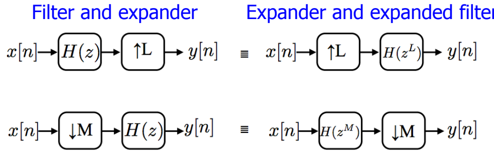
Upsampling
-expanding in time
-compressing in frequency

"compressor"

Downsampling
-compressing in time
-expanding in frequency

Interchanging Operations - Summary

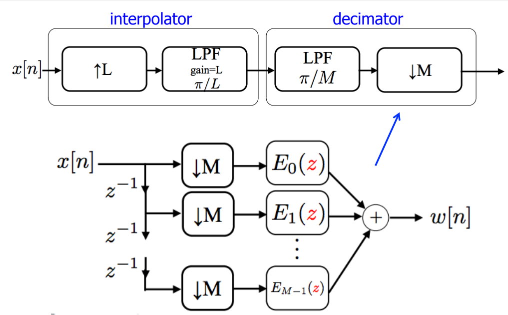
Filter and expander | Expander and expanded filter*



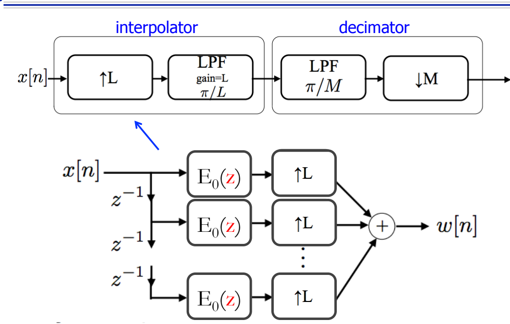
Compressor and filter | Expanded filter* and compressor

*Expanded filter = expanded impulse response, compressed freq response

Polyphase Implementation of Decimator

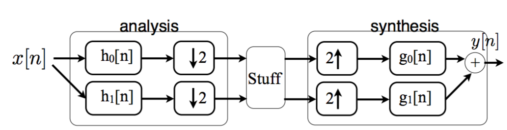


Polyphase Implementation of Interpolation



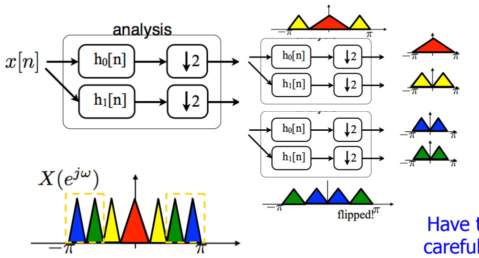
Multi-Rate Filter Banks

- Use filter banks to operate on a signal differently in different frequency bands
  - To save computation, reduce the rate after filtering
- $h_0[n]$ is low-pass, $h_1[n]$ is high-pass
  - Often $h_1[n]=e^{j\pi n}h_0[n]$ ← shift freq resp by π



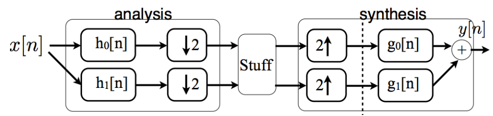
Multi-Rate Filter Banks

- Assume h_0, h_1 are ideal low/high pass with $\omega_C = \pi/2$



Have to be careful with order!

Perfect Reconstruction non-Ideal Filters

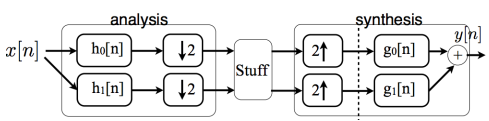


$$Y(e^{j\omega}) = \frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j\omega}) + G_1(e^{j\omega})H_1(e^{j\omega})\right] X(e^{j\omega})$$
$$+\frac{1}{2}\left[G_0(e^{j\omega})H_0(e^{j(\omega-\pi)}) + G_1(e^{j\omega})H_1(e^{j(\omega-\pi)})\right] X(e^{j(\omega-\pi)})$$

need to cancel!

aliasing

Quadrature Mirror Filters



Quadrature mirror filters

$$H_1(e^{j\omega}) = H_0(e^{j(\omega-\pi)})$$
$$G_0(e^{j\omega}) = 2H_0(e^{j\omega})$$
$$G_1(e^{j\omega}) = -2H_1(e^{j\omega})$$

Frequency Response of Systems

Penn

Frequency Response of LTI System

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

- We can define a magnitude response

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

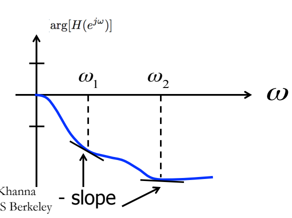
- And a phase response

$$\angle Y(e^{j\omega}) = \angle H(e^{j\omega}) + \angle X(e^{j\omega})$$

Group Delay

- General phase response at a given frequency can be characterized with group delay, which is related to phase

$$\text{grd}[H(e^{j\omega})] = -\frac{d}{d\omega}\{\arg[H(e^{j\omega})]\}$$



LTI System

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]$$

Example: $y[n] = x[n] + 0.1y[n-1]$

Stable and causal if all poles inside unit circle

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \ldots + a_N z^{-N}} = \frac{b_0 \prod_{k=1}^{M}(1 - c_k z^{-1})}{a_0 \prod_{k=1}^{N}(1 - d_k z^{-1})}$$

- Transfer function is not unique without ROC
  - If diff. eq represents LTI and causal system, ROC is region outside all singularities
  - If diff. eq represents LTI and stable system, ROC includes unit circle in z-plane

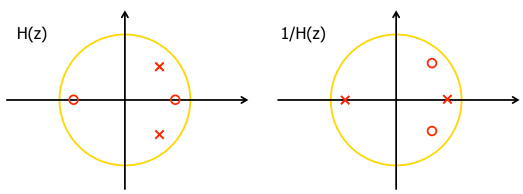
General All-Pass Filter

- d_k=real pole, e_k=complex poles paired w/ conjugate, e_k^*

$$H_{ap}(z) = A \prod_{k=1}^{M_r} \frac{z^{-1} - d_k}{1 - d_k z^{-1}} \prod_{k=1}^{M_c} \frac{(z^{-1} - e_k^*)(z^{-1} - e_k)}{(1 - e_k z^{-1})(1 - e_k^* z^{-1})}$$

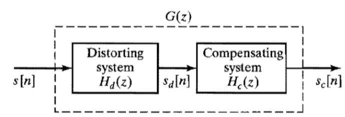
Minimum-Phase Systems

- Definition: A stable and causal system H(z) (i.e. poles inside unit circle) whose inverse 1/H(z) is also stable and causal (i.e. zeros inside unit circle)
  - All poles and zeros inside unit circle



Min-Phase Decomposition Purpose

- Have some distortion that we want to compensate for:



- If $H_d(z)$ is min phase, easy:
  - $H_c(z)=1/H_d(z)$ ← also stable and causal
- Else, decompose $H_d(z)=H_{d,min}(z)\ H_{d,ap}(z)$
  - $H_c(z)=1/H_{d,min}(z)$ → $H_d(z)H_c(z)=H_{d,ap}(z)$
    - Compensate for magnitude distortion

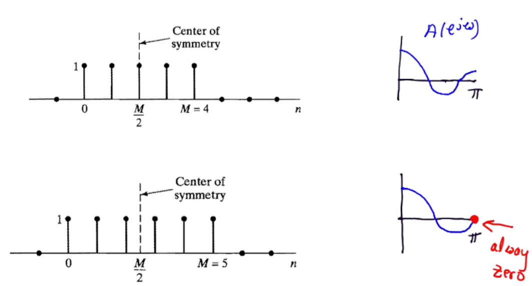
Generalized Linear Phase

- An LTI system has generalized linear phase if frequency response $H(e^{j\omega})$ can be expressed as:

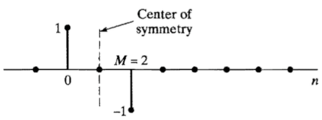
$$H(e^{j\omega}) = A(\omega)e^{-j\omega\alpha + j\beta}, \quad |\omega| < \pi$$

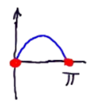
- Where $A(\omega)$ is a real function.
- What is the group delay?

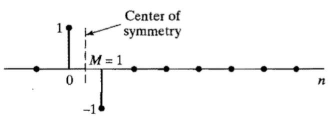
FIR GLP: Type I and II



FIR GLP: Type III and IV







Zeros of GLP System

- FIR GLP System Function

$$H(z) = \sum_{n=0}^{M} h[n]z^{-n}$$

Real system → zeros occur in conjugate-reciprocal groups of 4

$$(1 - re^{j\theta}z^{-1})(1 - re^{-j\theta}z^{-1})(1 - r^{-1}e^{j\theta}z^{-1})(1 - r^{-1}e^{-j\theta}z^{-1})$$

- If zero is on unit circle (r=1)

$$(1 - e^{j\theta}z^{-1})(1 - e^{-j\theta}z^{-1}).$$

- If zero is real and not on unit circle (θ =0)

$$(1 \pm rz^{-1})(1 \pm r^{-1}z^{-1}).$$

FIR Filter Design

Penn

FIR Design by Windowing

- Given desired frequency response, $H_d(e^{j\omega})$, find an impulse response

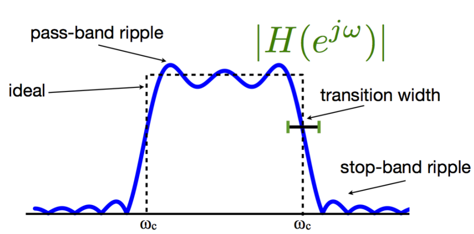
$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega$$

ideal

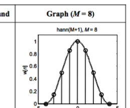
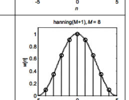
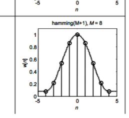
- Obtain the M^{th} order causal FIR filter by truncating/windowing it

$$h[n] = \left\{ \begin{array}{ll} h_d[n]w[n] & 0 \le n \le M \\ 0 & \text{otherwise} \end{array} \right\}$$

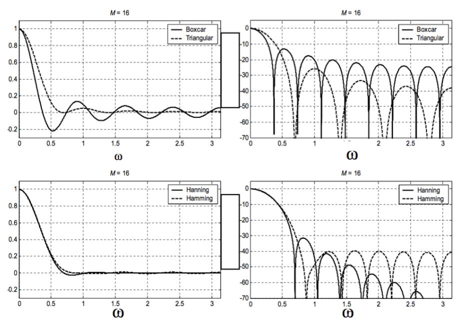
FIR Design by Windowing



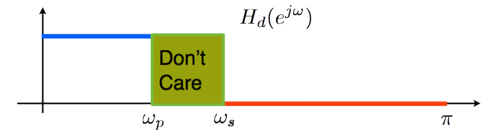
Tapered Windows

Name(s)	Definition	MATLAB Command	Graph ($M = 8$)
Hann	$w[n] = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\frac{\pi n}{M/2}\right)\right] & \lvert n\rvert \le M/2 \\ 0 & \lvert n\rvert > M/2 \end{cases}$	`hann(M+1)`	
Hanning	$w[n] = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\frac{\pi n}{M/2+1}\right)\right] & \lvert n\rvert \le M/2 \\ 0 & \lvert n\rvert > M/2 \end{cases}$	`hanning(M+1)`	
Hamming	$w[n] = \begin{cases} 0.54 + 0.46\cos\left(\frac{\pi n}{M/2}\right) & \lvert n\rvert \le M/2 \\ 0 & \lvert n\rvert > M/2 \end{cases}$	`hamming(M+1)`	

Tradeoff – Ripple vs. Transition Width



Optimality



- Least Squares:

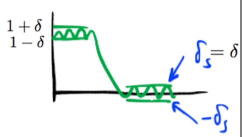
$$\text{minimize} \int_{\omega\in\text{care}} |H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

- Variation: Weighted Least Squares:

$$\text{minimize} \int_{-\pi}^{\pi} W(\omega)|H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

Min-Max Ripple Design

- Recall, $\tilde{H}(e^{j\omega})$ is symmetric and real
- Given ω_p, ω_s, M, find δ, $\tilde{h}_+$



$$\begin{array}{ll} \text{minimize} & \delta \\ \text{Subject to :} & \\ & 1-\delta \le \tilde{H}(e^{j\omega_k}) \le 1+\delta \qquad 0 \le \omega_k \le \omega_p \\ & -\delta \le \tilde{H}(e^{j\omega_k}) \le \delta \qquad \omega_s \le \omega_k \le \pi \\ & \delta > 0 \end{array}$$

- Formulation is a linear program with solution δ, $\tilde{h}_+$
- A well studied class of problems

Discrete Fourier Transform

DFT

Penn

Discrete Fourier Transform

- The DFT

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] W_N^{-kn} \quad \text{Inverse DFT, synthesis}$$

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad \text{DFT, analysis}$$

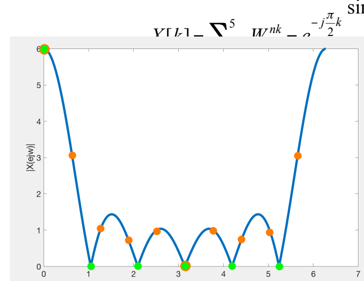
- It is understood that,

$$x[n] = 0 \quad \text{outside } 0 \le n \le N-1$$
$$X[k] = 0 \quad \text{outside } 0 \le k \le N-1$$

DFT vs DTFT

- Back to example

$$X[k] = \sum^{5} W^{nk} = e^{-j\frac{\pi}{2}k} \frac{\sin\left(\frac{3\pi}{5}k\right)}{\sin\left(\frac{\pi}{10}k\right)}$$



"6-point" DFT

"10-point" DFT

Use `fftshift` to center around dc

Circular Convolution

- For $x_1[n]$ and $x_2[n]$ with length N

$$x_1[n] \circledN x_2[n] \leftrightarrow X_1[k] \cdot X_2[k]$$

  - Very useful!! (for linear convolutions with DFT)

Linear Convolution via Circular Convolution

- Zero-pad x[n] by P-1 zeros

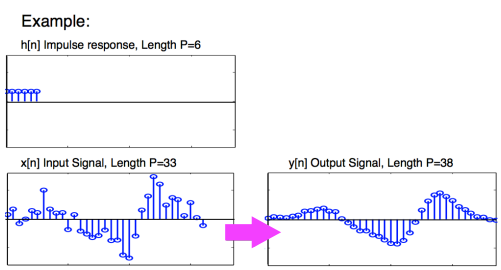
$$x_{zp}[n] = \begin{cases} x[n] & 0 \le n \le L-1 \\ 0 & L \le n \le L+P-2 \end{cases}$$

- Zero-pad h[n] by L-1 zeros

$$h_{zp}[n] = \begin{cases} h[n] & 0 \le n \le P-1 \\ 0 & P \le n \le L+P-2 \end{cases}$$

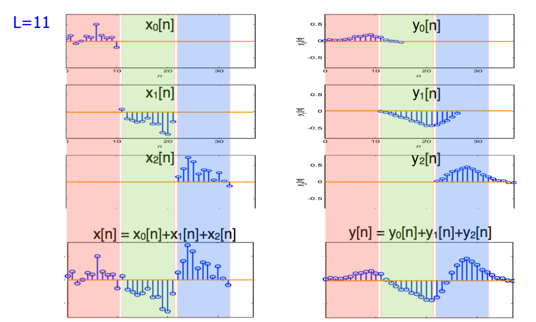
- Now, both sequences are length M=L+P-1

Block Convolution

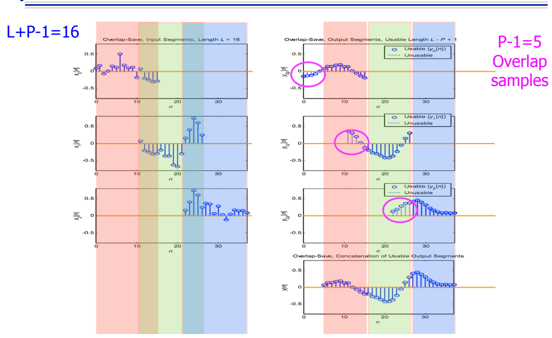


Example of Overlap-Add

L+P-1=16



Example of Overlap-Save



Circular Conv. as Linear Conv. w/ Aliasing

$$x_{3p}[n] = \begin{cases} \sum_{r=-\infty}^{\infty} x_3[n - rN], & 0 \le n \le N-1, \\ 0, & \text{otherwise}, \end{cases}$$

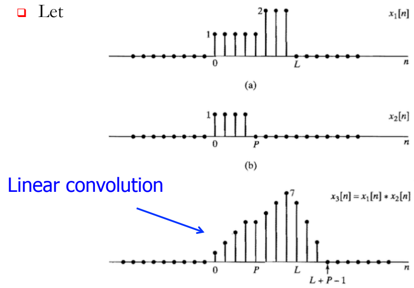
- Therefore

$$x_{3p}[n] = x_1[n] \circledN x_2[n]$$

- The N-point circular convolution is the sum of linear convolutions shifted in time by N

Example:

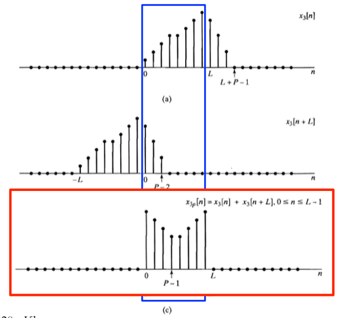
- Let



- What does the L-point circular convolution look like?

Example:

- The L-shifted linear convolutions



Fast Fourier Transform

FFT

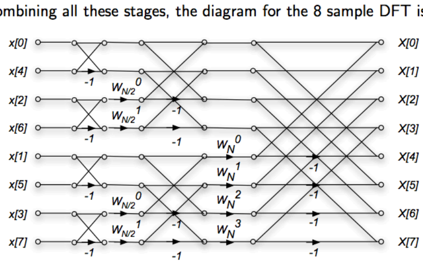
Penn

Fast Fourier Transform

- Enable computation of an N-point DFT (or DFT^{-1}) with the order of just $N \cdot \log_2 N$ complex multiplications.
- Most FFT algorithms decompose the computation of a DFT into successively smaller DFT computations.
  - Decimation-in-time algorithms
  - Decimation-in-frequency
- Historically, power-of-2 DFTs had highest efficiency
- Modern computing has led to non-power-of-2 FFTs with high efficiency
- Sparsity leads to reduce computation on order $K \cdot \log N$

Decimation-in-Time FFT

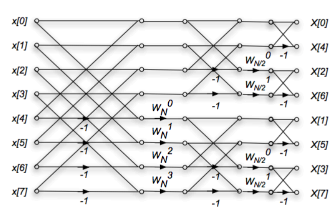
Combining all these stages, the diagram for the 8 sample DFT is:



- $3=\log_2(N)=\log_2(8)$ stages
- $4=N/2=8/2$ multiplications in each stage
  - 1st stage has trivial multiplication

Decimation-in-Frequency FFT

The diagram for and 8-point decimation-in-frequency DFT is as follows



This is just the decimation-in-time algorithm reversed!
The inputs are in normal order, and the outputs are bit reversed.

Admin

- Final Project due – Today!
  - TA advice – "The report takes time. Leave time for it."
- Last day of TA office hours – Apr 28th
  - Piazza still available
  - Review session for exam TBD
    - Poll in Piazza coming soon
- Last day of Tania office hours – May 1st
- Final Exam – May 7th

Final Exam Admin

- Final Exam – 5/7 (3pm-5pm)
  - In Canvas
    - Will have a 2 hr window to complete within a 14 hr time block (6am-8pm EDT)
  - Open course notes and textbook, but cannot communicate with each other about the exam
    - Students will have randomized and different questions
    - Reminder, it is not in your best interest to share the exam
  - Old exams posted on old course websites
  - Covers Lec 1- 20
    - Does not include lec 12 (data converters and noise shaping) or IIR Filters