Signal Processing and Linear Systems1

Lecture 13: Discrete-Time Fourier Transform

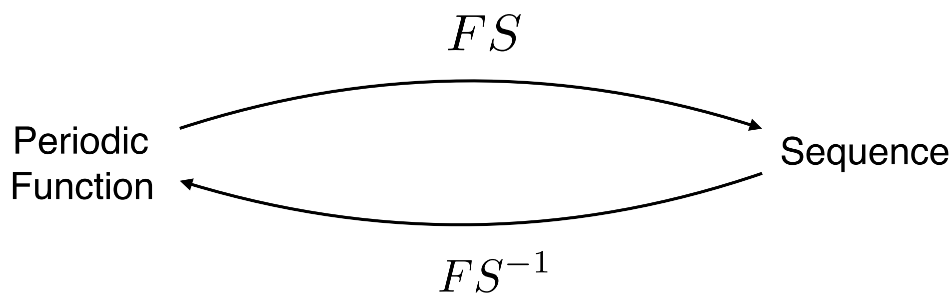
Nicholas Dwork
www.stanford.edu/~ndwork

Spectrums of Functions

	Periodic	Non-Periodic
Continuous	Fourier Series	Fourier Transform
Discrete	Discrete Fourier Transform	Discrete-Time Fourier Transform

We have already seen that we can represent a periodic function whose domain is all real numbers with a sequence.

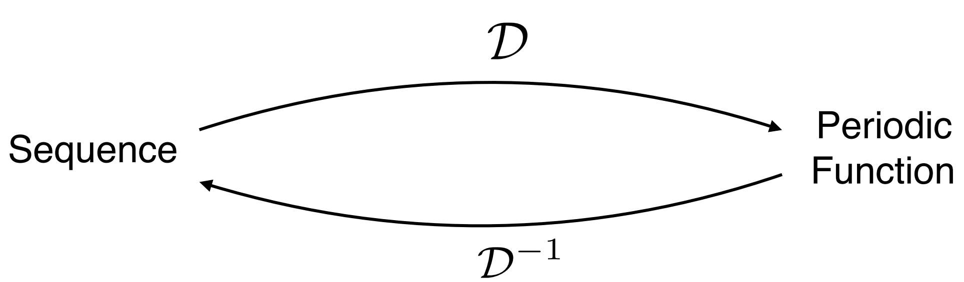
This was called the Fourier Series.



The sequence is the amount each frequency contributes to the periodic function. It is the function's spectrum.

We can look at this in reverse and realize that we can represent any sequence with a periodic function.

This is called the Discrete-Time Fourier Transform.



Here, the periodic function is the spectrum of the sequence.

Fourier Series

Forward Transform (Analysis Equations):

$$F_n = \frac{1}{P} \int_0^P f(x) \, \exp\left(-i \, 2\pi \frac{nx}{P}\right) dx$$

Inverse Transform (Synthesis Equations):

$$f(x) = \sum_{n=-\infty}^{\infty} F_n \, \exp\left(i \, 2\pi \frac{nx}{P}\right)$$

Discrete Time Fourier Transform

Forward Transform (Analysis Equations):

$$F(k) = \sum_{n=-\infty}^{\infty} f[n] \, \exp\left(-i \, 2\pi \, nk\right)$$

Inverse Transform (Synthesis Equations):

$$f[n] = \int_0^1 F(k) \, \exp\left(i \, 2\pi \, nk\right) \, dk$$

A Sampled Function

Consider a function f and its samples $\{f(n\Delta) : n \in \mathbb{Z}\}$.

We again construct a weighted delta function: $f_s = \sum_{n=-\infty}^{\infty} f(n\Delta)\, \delta(x - n\Delta)$.

The spectrum of f_s is $\mathcal{F}\{f_s\} = \mathcal{F}\{f\, \text{III}_\Delta\} = F * \frac{1}{\Delta}\text{III}_{1/\Delta}$.

These are just replicas of the original spectrum separated by $1/\Delta$.

This is the spectrum identified by the DTFT.

Discrete Convolution

$$(f * g)[n] = \sum_{m=\infty}^{\infty} f[m]\, g[n-m]$$

Convolution Theorem

$$\mathcal{D}\left\{f * g\right\} = \mathcal{D}\{f\} \odot \mathcal{D}\{g\}$$

$$\mathcal{D}\left\{f * g\right\}[n] = \mathcal{D}\{f\}[n] \cdot \mathcal{D}\{g\}[n]$$

Many More Theorems

We basically get all the theorems that we had for the Fourier Transform, and the Discrete Fourier Transform.

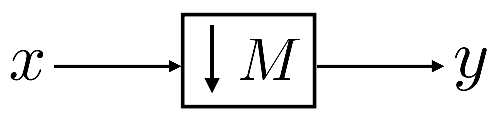
You will prove several of these theorems in your homework.

Downsampling

Consider a sequence $x = (\ldots, x[-2], x[-1], x[0], x[1], x[2], \ldots).$

We can construct a new sequence from this sequence where we keep every *M* th sample of the original sequence.

$$y[n] = x[M\, n]$$



Downsampling stretches the spectrum of the original signal by a factor of *M*.

$$y[n] = x[M\, n] \qquad \mathcal{D}\{y\}(f) = \frac{1}{M} \sum_{v=0}^{M-1} \mathcal{D}\{x\} \left(\frac{f - v}{M} \right).$$

Proof:

$$\mathcal{D}\{y\}(f) = \sum_{n=-\infty}^{\infty} y[n] \, \exp(-i\, 2\pi\, n f) \;=\; \sum_{n=-\infty}^{\infty} x[Mn] \, \exp(-i\, 2\pi\, n f)$$

$$= \sum_{u=-\infty}^{\infty} s[u]\, x[u] \, \exp\left(-i\, 2\pi\, \frac{u}{M} f\right)$$

$$s[u] = \begin{cases} 1 & \text{if } M \text{ divides } u \\ 0 & \text{otherwise} \end{cases} = \frac{1}{M} \sum_{v=0}^{M-1} e^{i 2\pi u v / M}$$

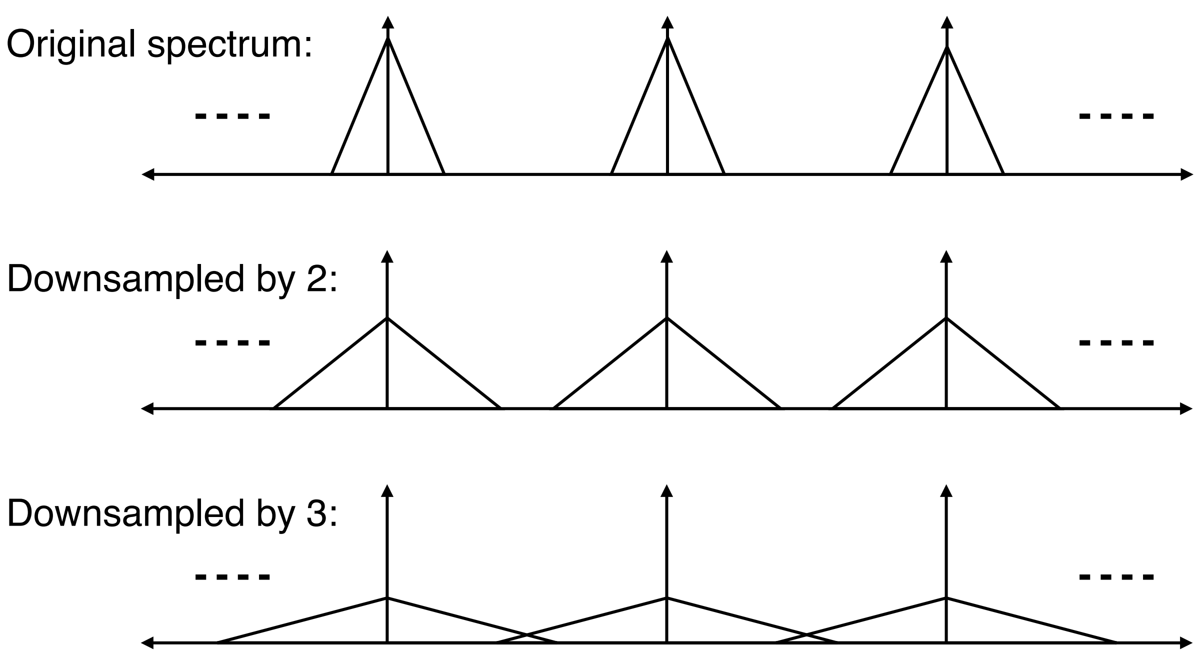
$$= \sum_{u=-\infty}^{\infty} \left(\frac{1}{M} \sum_{v=0}^{M-1} e^{i 2\pi u v / M} \right) x[u] \, \exp\left(-i\, 2\pi\, \frac{u}{M} f\right)$$

$$\mathcal{D}\{y\}(f) = \sum_{u=-\infty}^{\infty} \left(\frac{1}{M} \sum_{v=0}^{M-1} e^{i2\pi uv/M} \right) x[u] \exp\left(-i\,2\pi\,\frac{u}{M} f\right)$$

$$= \frac{1}{M} \sum_{v=0}^{M-1} \sum_{u=-\infty}^{\infty} x[u] \exp\left(-i2\pi \frac{u}{M}(f-v)\right).$$

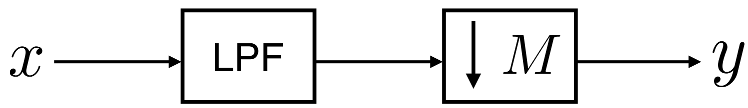
Recall: $\mathcal{D}\{x\}(f) = \sum_{n=-\infty}^{\infty} x[n] \exp(-i2\pi n f)$.

Therefore $\mathcal{D}\{y\}(f) = \frac{1}{M} \sum_{v=0}^{M-1} \mathcal{D}\{x\}\left(\frac{f-v}{M}\right)$. ■



Decimation

Low-Pass filter prior to downsampling to avoid aliasing.

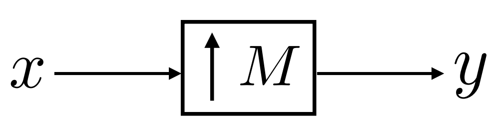


Upsampling

Consider a sequence $x = (\ldots, x[-2], x[-1], x[0], x[1], x[2], \ldots).$

We can construct a new sequence from this sequence where we insert *M* zeros in between every sample.

$$y[n] = \begin{cases} x[n/M] & \text{if } M \text{ divides } n \\ 0 & \text{otherwise} \end{cases} \qquad y[n] = \sum_{m=-\infty}^{\infty} x[m]\,\delta[n - m\,M]$$



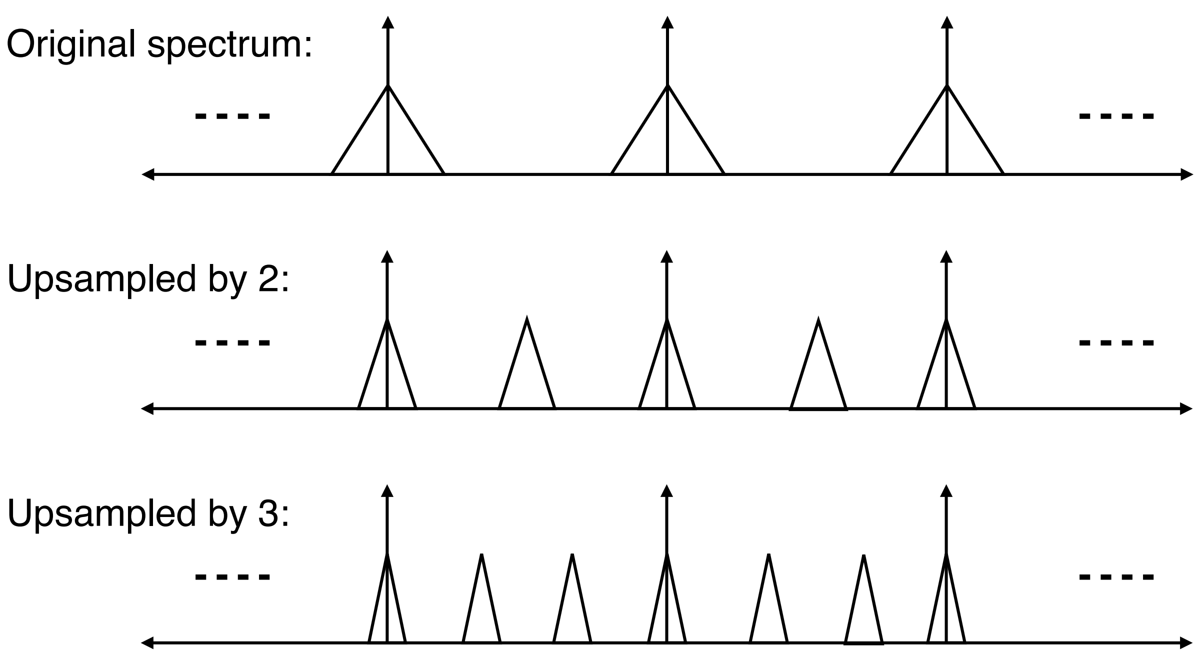
Upsampling shrinks the spectrum of the original signal by a factor of *M*.

$$y[n] = \sum_{m=-\infty}^{\infty} x[m]\,\delta[n - m\,M] \qquad\qquad \mathcal{D}\{y\}(f) = \mathcal{D}\{x\}(M\,f)$$

Proof:

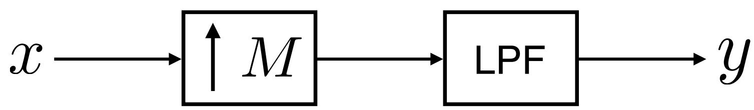
$$\begin{aligned}
\mathcal{D}\{y\}(f) &= \sum_{n=-\infty}^{\infty} y[n]\,\exp(-i\,2\pi\,nf) \\
&= \sum_{n=-\infty}^{\infty} \left(\sum_{m=-\infty}^{\infty} x[m]\,\delta[n - m\,M] \right) \exp(-i\,2\pi\,nf) \\
&= \sum_{m=-\infty}^{\infty} x[m] \sum_{n=-\infty}^{\infty} \delta[n - m\,M]\,\exp(-i\,2\pi\,nf) \\
&= \sum_{m=-\infty}^{\infty} x[m]\,\exp(-i\,2\pi\,mM\,f)
\end{aligned}$$

■

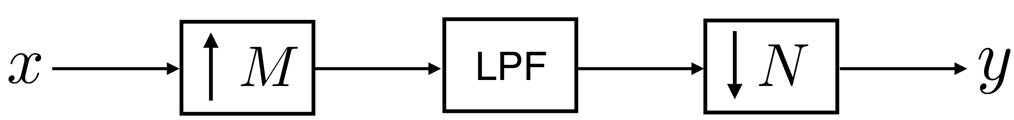


Interpolation

Low-Pass filter after upsampling to isolate center spectrum.



Rational Decimation and Interpolation



Discrete-Time System

Accepts a sequence as input and outputs a sequence.

Why do we care?

We can think of the data coming out of an A/D converter as an endless stream of data values. Systems that process this stream are discrete.

Ex: a radio station will continuously broadcast data which gets collected by an antenna and converted to a sequence by an A/D converter.

Ex: we can look at the price of a stock at each minute that the stock market is active.

Discrete-Time System Definitions

We get all the definitions you'd expect.

Memory, causal, linear, shift invariant, BIBO stable, energy, power

The impulse response of a Discrete-Time System is the response to the Kronecker delta function.

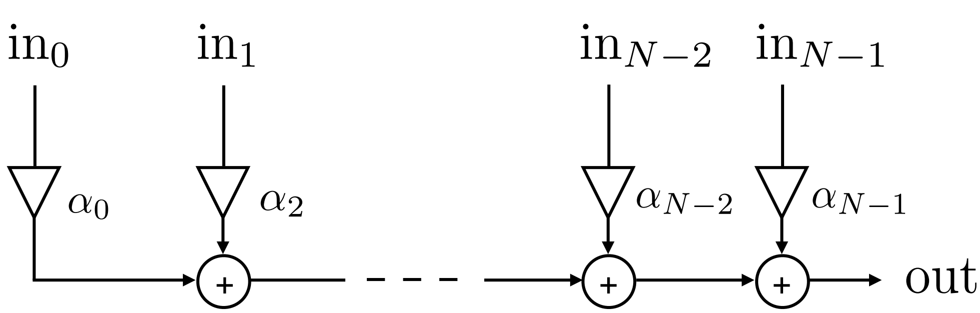
$$h[n] = S\{\delta\}[n]$$

Finite Impulse Response System

A system whose impulse responses have finite support.

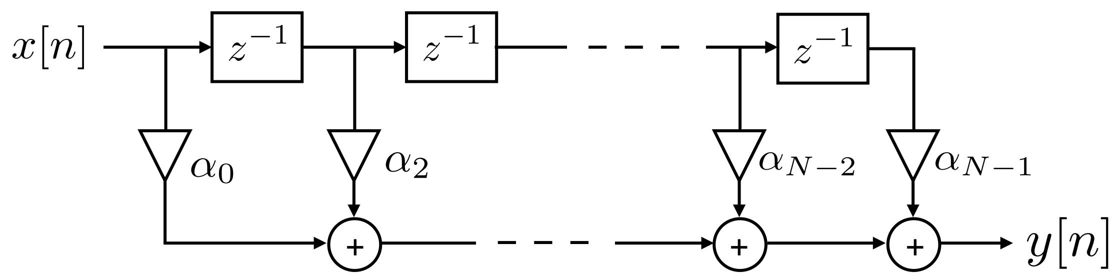
If the FIR system is causal, then the impulse responses of the system are 0 after a finite number of samples.

Linear Combiner



$$\mathrm{out} = \sum_{n=0}^{N-1} \alpha_n \, \mathrm{in}_n$$

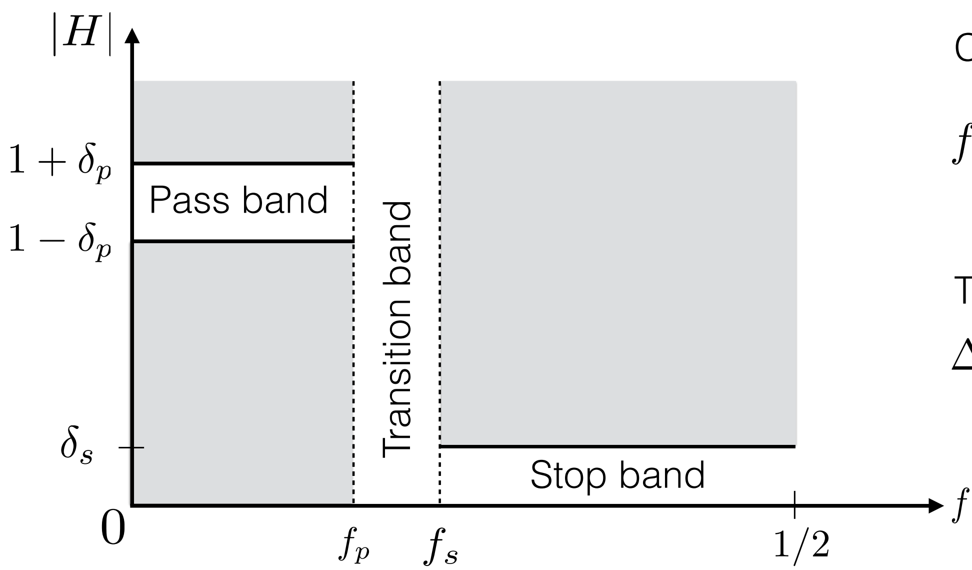
Linear FIR Filter



z^{-1} represents a delay of one sample.

$$y[m] = \sum_{n=0}^{N-1} \alpha_n \, x[m-n] \; = \sum_{n=0}^{N-1} h[n] \, x[m-n] \; = (x * h)[n]$$

Low-Pass Filter Specifications



Cutoff Frequency:

$$f_c = \frac{f_p + f_s}{2}$$

Transition Width:

$$\Delta f = f_s - f_p$$

Linear FIR Filter Design

Method #1: (Bad Method)

Design the spectrum of the ideal filter that you would like.

Perform an inverse DTFT.

Keep as many coefficients as you can.

The problem:

When we cutoff coefficients, it's like we've multiplied the impulse response by a rect function.

This is convolution with a sinc in the Frequency domain, which causes ringing.

Linear FIR Filter Design

Method #2: (Good Method)

Design the spectrum of the ideal filter that you would like.

Perform an inverse DTFT.

Window the coefficients so that they approach zero smoothly.

Keep as many coefficients as required.

Discrete Kaiser Window

$$w[n] = \begin{cases} \frac{I_0\left(\beta\sqrt{1-\left[\frac{n-\alpha}{\alpha}\right]^2}\right)}{I_0(\beta)} & 0 \le n \le M \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha = M/2$$

The parameters are the filter length $M + 1$ and shape parameter β .

Discrete-Time Signal Processing by Oppenheim and Schafer

Kaiser Window LP-FIR Filter

Kaiser developed empirical methods to determine β and M.

$$\delta = \min(\delta_p, \delta_s) \qquad A = -20\ \log_{10} \delta$$

$$\beta = \begin{cases} 0.1102(A - 8.7) & A > 50 \\ 0.5842(A-21)^{0.4} + 0.07886(A-21) & 21 \le A \le 50 \\ 0 & A < 21 \end{cases}$$

$$M = \frac{A - 8}{4.57\pi\,\Delta f} \pm 2$$

Discrete-Time Signal Processing by Oppenheim and Schafer

Kaiser Window FIR Filter

To design a band-pass or high-pass filter:

Design a low-pass filter.

Shift the filter to the appropriate band (in the frequency domain).

This corresponds to multiplication by a linear phase in the time domain.

Discrete-Time Signal Processing by Oppenheim and Schafer

Parks-McClellan and Least-Squares are two other filter design algorithms.