FOURIER TRANSFORM

- Introduction
- Continuous-time Fourier transform (CTFT)
- Discrete-time Fourier transform (DTFT)
- Example
- Discrete Fourier transform (DFT)
- Frequency analysis of signals using the DFT (example)
- Frequency analysis of discrete-time signals (example)
- Parseval's theorem for the DFT
- Convolution of two finite-duration signals using the DFT
- Fast Fourier transform (FFT)
- Frequency ranges of some biological signals

- (Properties of the DFT)
- (Symmetry properties of the DFT)
- (The overlap-save method for convolution)

Introduction

- The discrete Fourier transform (DFT) is an efficient method for computing the discrete-time convolution of two signals
- The DFT is a tool for filter design
- The DFT is an efficient method for measuring spectra of discrete-time signals
- The *interpretation* of the DFT of a signal can be difficult because the DFT only provides a complete representation of *finite-duration* signals

Continuous-time Fourier transform (CTFT)

- Fourier transform provides a representation of arbitrary signals as a sum of complex exponentials
- Fourier transform pair for continuous signals:

$$x(t) = \int_{-\infty}^{\infty} X(F)\, e^{j2\pi Ft}\, dF$$

$$X(F) = \int_{-\infty}^{\infty} x(t)\, e^{-j2\pi Ft}\, dt$$

$$x(t) \longleftrightarrow X(F)$$

- Time and frequency show duality
- The frequency response *H(F)* of an LTI system with *unit-sample response* (*impulse response*) *h(t)* is:

$$H(F) \triangleq \int_{-\infty}^{\infty} h(t)\, e^{-j2\pi Ft}\, dt$$

Continuous-time Fourier transform (CTFT)

- The response of an LTI system *y(t)* with frequency response *H(F)* to an arbitrary input *x(t)*:

$$y(t) = \int_{-\infty}^{\infty} H(F)\, X(F)\, e^{j2\pi F t}\, dF$$

- The Fourier transform of the convolution *x(t) * h(t)* is the product of Fourier transforms *X(F) H(F)* of *x(t)* and *h(t):*

$$x(t) * h(t) \longleftrightarrow X(F)\, H(F)$$

Discrete-time Fourier transform (DTFT)

- The discrete-time Fourier transform (DTFT) of *x*[*n*]:

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

- The *X(f)* is *periodic*. The signal *x*[*n*] can be expressed as a function of *X(f)*:

$$x[n] = \int_{-\frac{1}{2}}^{\frac{1}{2}} X(f)\, e^{j2\pi f n}\, df$$

- Fourier transform pair for discrete-time signal:

$$x[n] \longleftrightarrow X(f)$$

- The time domain is discrete, while the frequency domain is continuous and periodic with the period of *1*

Discrete-time Fourier transform (DTFT)

- If we define:

$$Y(f) \;=\; H(f)\; X(f)$$

- The output of a system *y[n]* with frequency response *H(f)* to the input *x*[*n*] is the ”sum” of the input exponentials, each one being weighted by the frequency response:

$$y[n] \;=\; \int_{-\frac{1}{2}}^{\frac{1}{2}} H(f)\; X(f)\; e^{j2\pi f n}\; df$$

- This means that the Fourier transform of the convolution *x*[*n*] ∗ *h*[*n*] is the product of the Fourier transforms (convolution theorem):

$$x[n] \;*\; h[n] \;\longleftrightarrow\; X(f)\; H(f)$$

Example

- The Fourier transform *W(f)* of the symmetric rectangular pulse *w*[*n*]:

$$w[n] = \Pi_N[n] \triangleq \begin{cases} 1 & \text{if } -N \leq n \leq N \\ 0 & \text{otherwise} \end{cases}$$

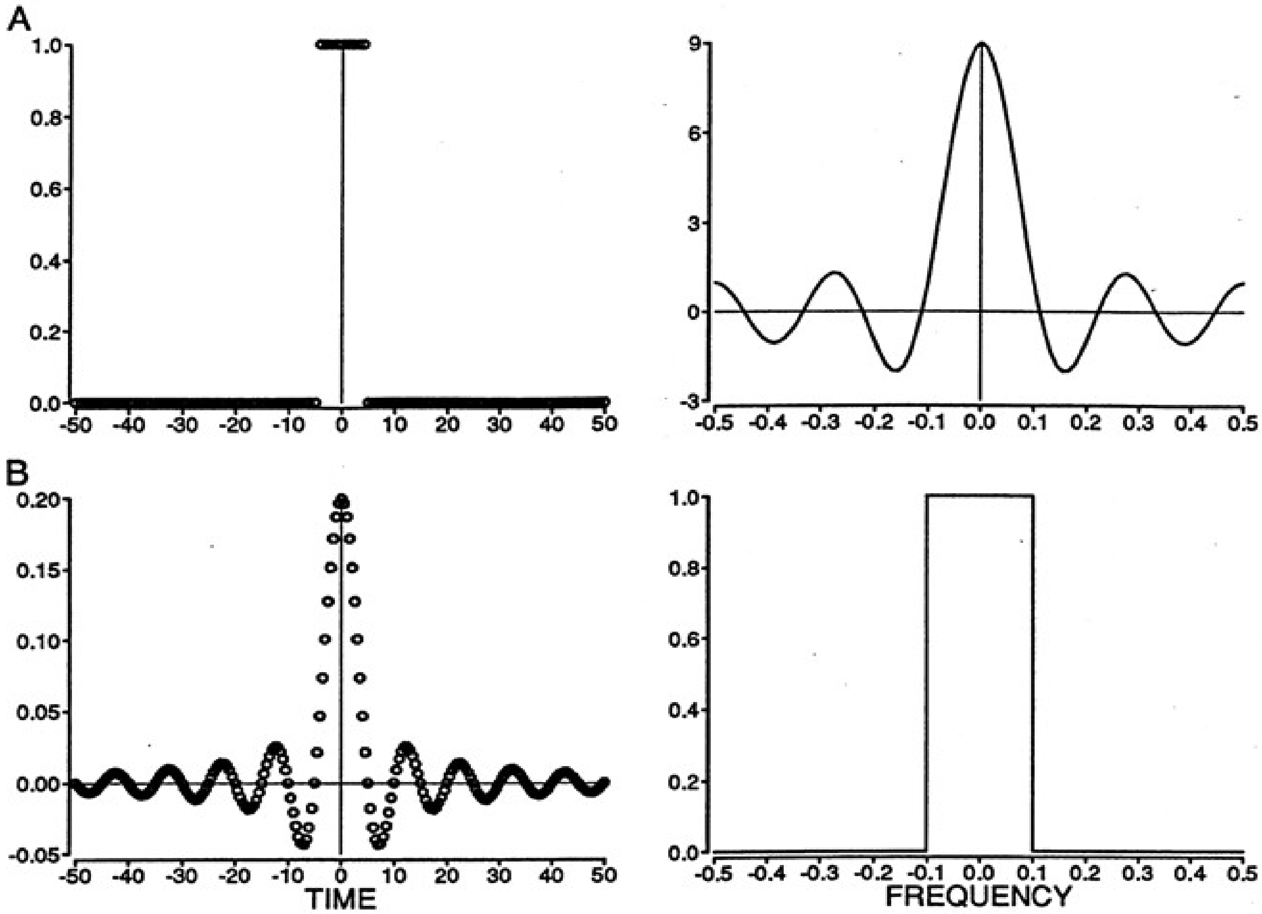
$$W(f) = \sum_{n=-N}^{N} e^{-j2\pi fn} = \frac{\sin \pi(2N+1)f}{\sin \pi f}$$

- The inverse Fourier transform to compute the impulse response *h*[*n*] of the ideal digital low-pass filter *H(f)*:

$$H(f) = \Pi_W(f) \triangleq \begin{cases} 1 & |f| \leq W \\ 0 & W < |f| \leq \frac{1}{2} \end{cases}$$

$$h[n] = \int_{-W}^{W} e^{j2\pi fn}\, df = \frac{\sin 2\pi Wn}{\pi n}$$

Example



(Bertrand Delgutte, MIT OpenCourseWare)

Discrete Fourier transform (DFT)

- To compute the DTFT requires an infinite number of operations
- A good representation of the spectrum will be achieved if computing only a finite number of *frequency samples* of the DTFT while the spacing between samples is sufficiently small. Simple results are obtained by sampling in frequency at regular intervals.
- We therefore define the *N-point discrete Fourier* transform $X[k]$ of a signal $x[n]$ of finite duration, $0 \le n \le N-1$, as samples of its transform $X(f)$ taken at intervals of $1/N$:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N} \qquad X[k] \triangleq X(k/N) \qquad \text{for } 0 \le k \le N-1$$

- Because $X(f)$ is periodic with period 1, $X[k]$ is periodic with period N, which justifies only considering the values of $X[k]$ over the interval $[0, N-1]$

Discrete Fourier transform (DFT)

- The finite-duration signal *x[n]* can be reconstructed from its DFT $X[k]$ by:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{j2\pi kn/N}$$

- Because the signal $x[n]$ is of finite duration, the definition of the DFT $X[k]$ becomes:

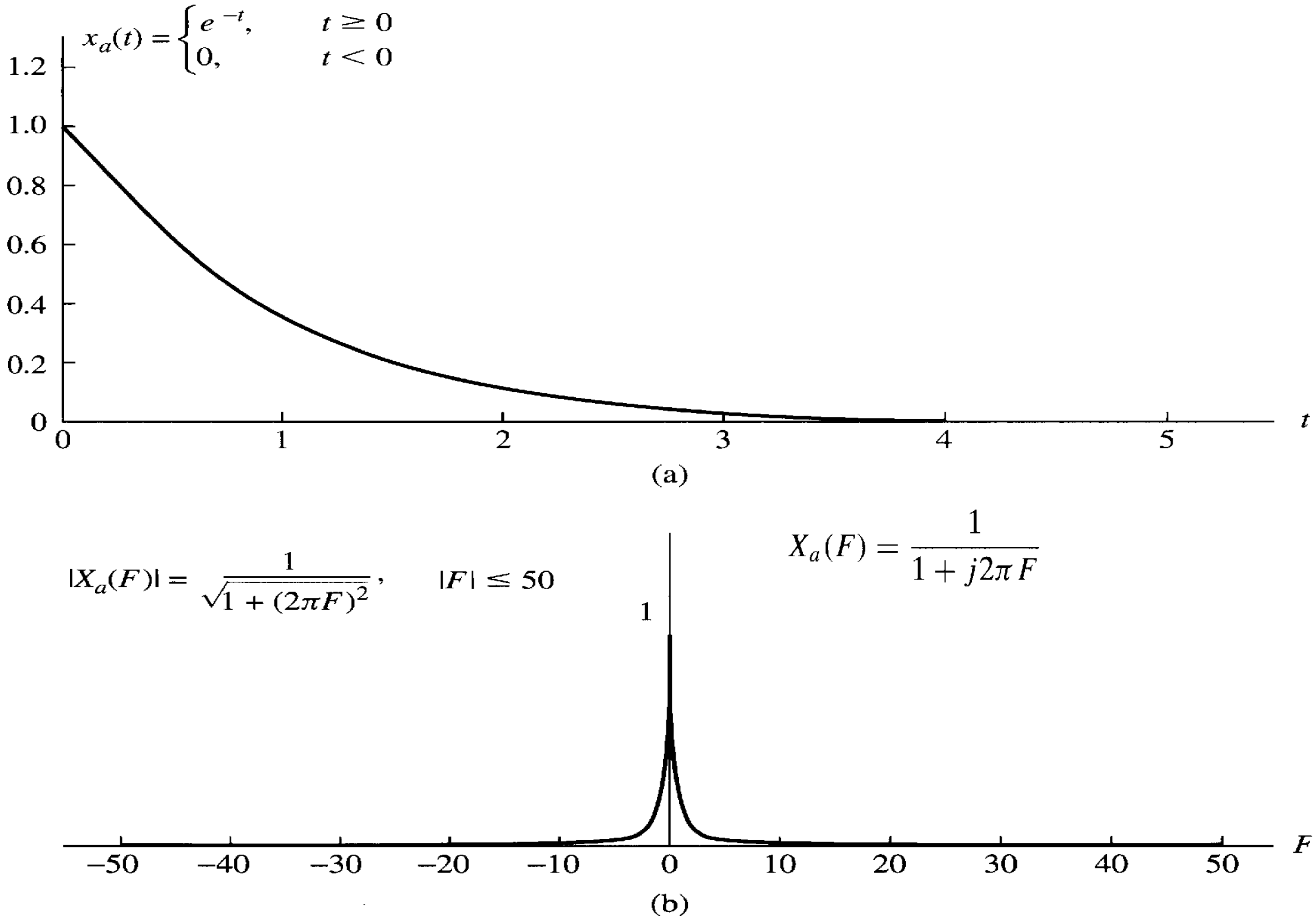
$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N}$$

- The DFT pair for finite-duration signals:

$$x[n] \longleftrightarrow X[k]$$

- Both time and frequency domain are discrete and periodic with period N
- Computing the N-point DFT of a signal implicitly introduces a periodic signal with period N, so that all operations involving the DFT are really operations on periodic signals

Frequency analysis of signals using the DFT (example)



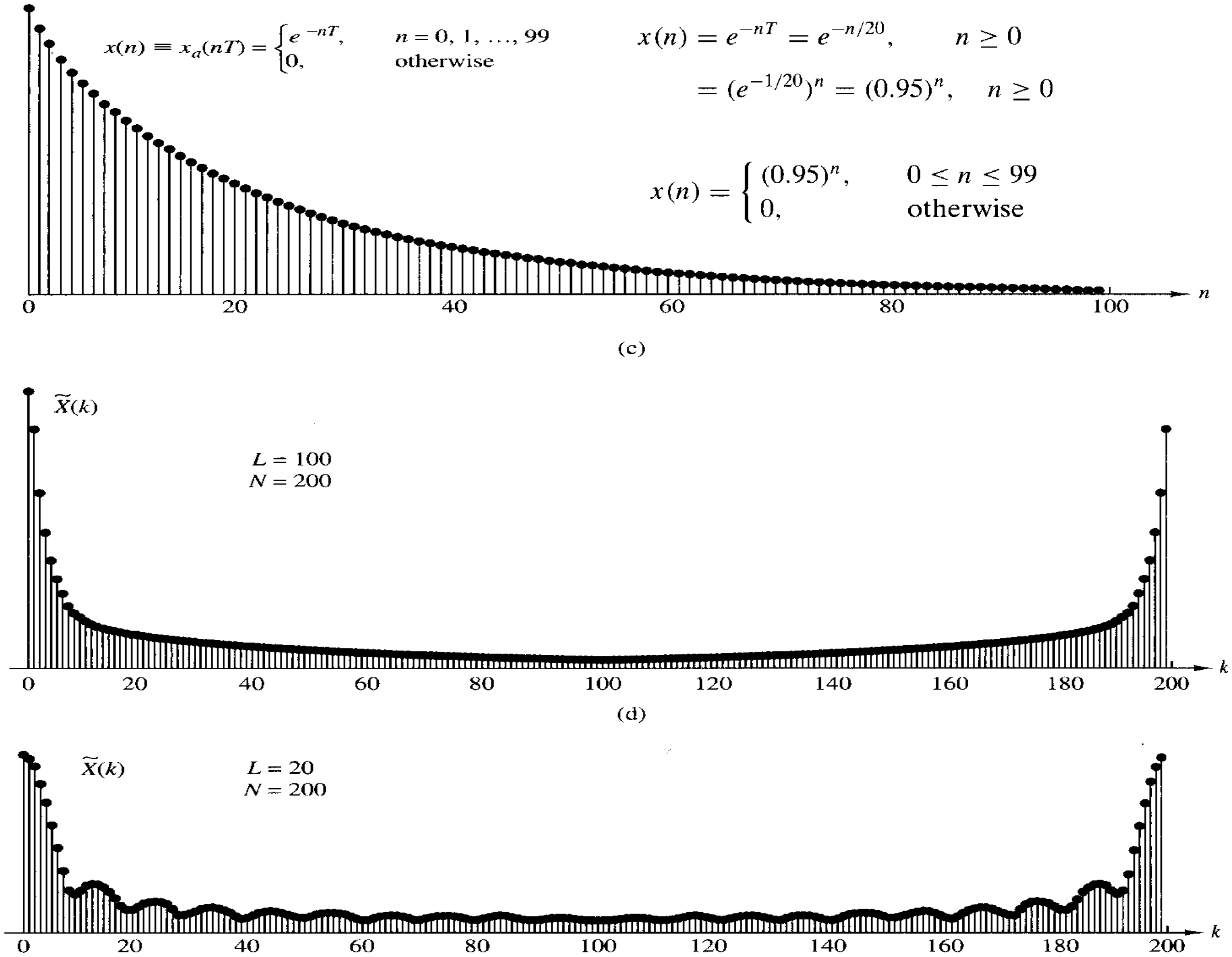
(Proakis, Manolakis)

Frequency analysis of signals using the DFT (example)

Fs = 20 smp/s

L=100 (*L*=20)

N = 200



(Proakis, Manolakis)

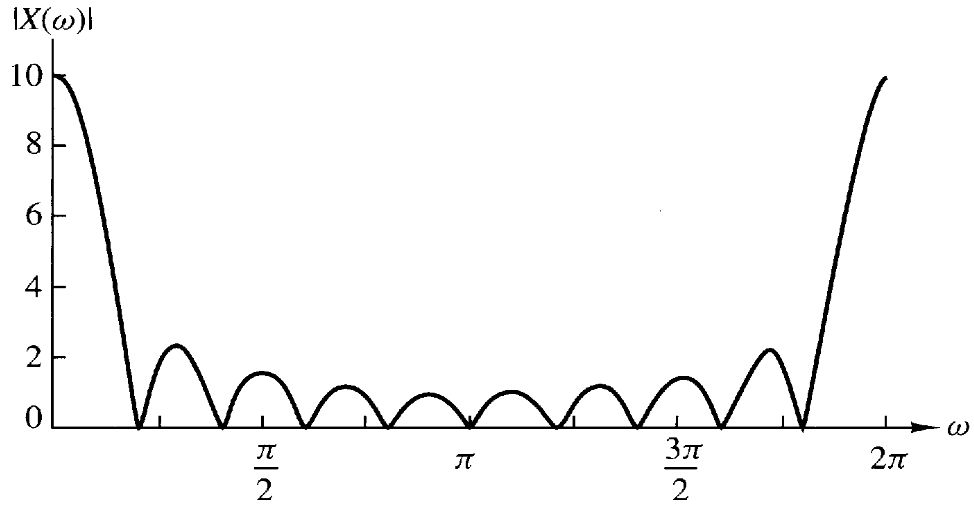
Frequency analysis of discrete-time signals (example)

- A finite-duration sequence of length L :

$$x(n) = \begin{cases} 1, & 0 \le n \le L-1 \\ 0, & \text{otherwise} \end{cases}$$

$$X(\omega) = \sum_{n=0}^{L-1} x(n) e^{-j\omega n}$$

$$= \sum_{n=0}^{L-1} e^{-j\omega n} = \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}} = \frac{\sin(\omega L/2)}{\sin(\omega/2)} e^{-j\omega(L-1)/2}$$



- Determine the N-point DFT of this sequence for $N \ge L$

(Proakis, Manolakis)

Frequency analysis of discrete-time signals (example)

$$X(k) = \frac{1 - e^{-j2\pi kL/N}}{1 - e^{-j2\pi k/N}}, \qquad k = 0, 1, \ldots, N-1$$

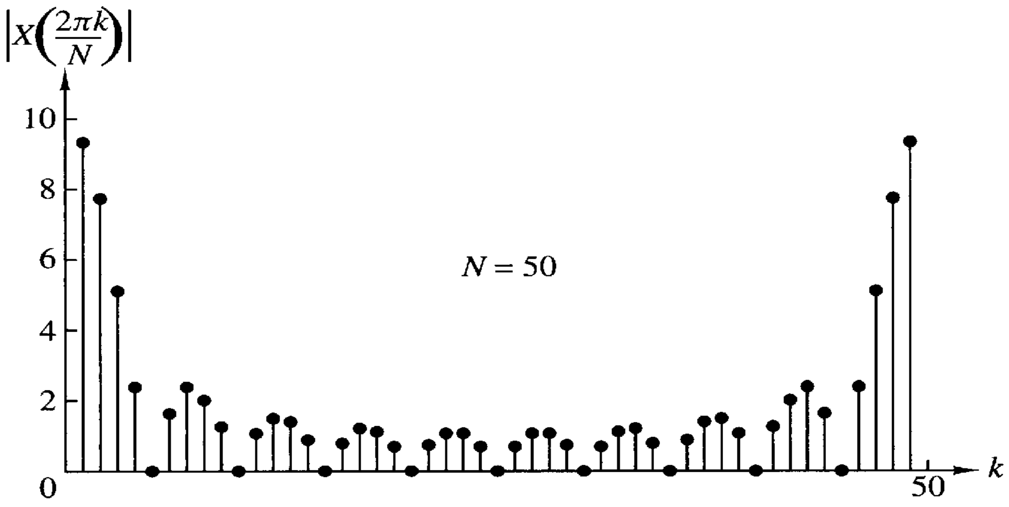
$$= \frac{\sin(\pi kL/N)}{\sin(\pi k/N)} e^{-j\pi k(L-1)/N}$$

- $X(\omega)$ evaluated at the set of N equally spaced frequencies

$$\omega_k = 2\pi k / N,$$

$$k = 0, 1, \ldots, N-1$$

$$L = 10, \; N = 50$$



(Proakis, Manolakis)

Frequency analysis of discrete-time signals (example)

$$X(k) = \frac{1 - e^{-j2\pi kL/N}}{1 - e^{-j2\pi k/N}}, \qquad k = 0, 1, \ldots, N-1$$

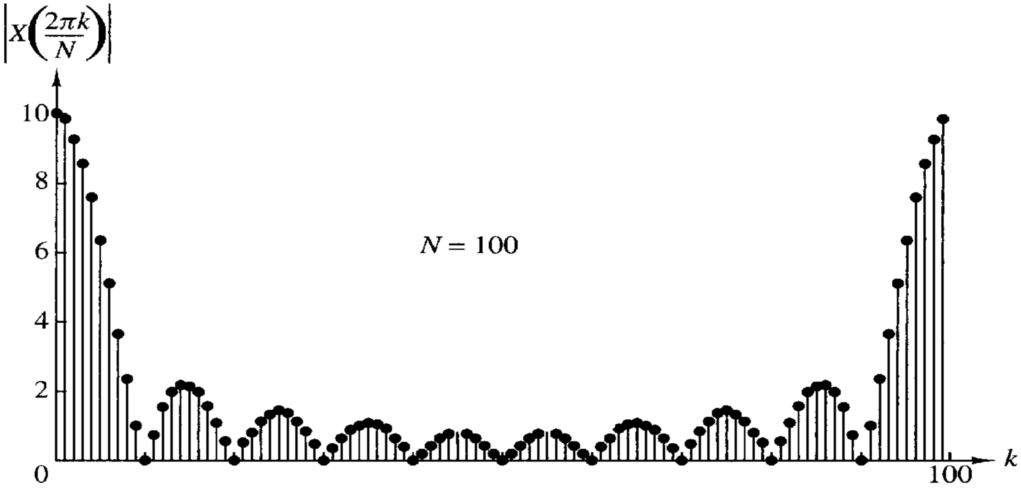
$$= \frac{\sin(\pi kL/N)}{\sin(\pi k/N)} e^{-j\pi k(L-1)/N}$$

- $X(\omega)$ evaluated at the set of N equally spaced frequencies

$$\omega_k = 2\pi k / N,$$

$$k = 0, 1, \ldots, N-1$$

$$L = 10, \; N = 100$$



(Proakis, Manolakis)

Parseval's theorem for the DFT

- Parseval's theorem:

$$\sum_{n=0}^{N-1} x[n]^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2$$

- Parseval's theorem expresses the energy in the finite duration sequence $x[n]$ in terms of the frequency components $X[k]$

Convolution of two finite-duration signals using the DFT

- The following scheme allows filtering the input $x[n]$ by the filter $h[n]$:

1. Compute the N -point DFT of $x[n]$
2. Compute the N -point DFT of $h[n]$
3. Form the product $Y[k] = X[k] . H[k]$
4. Compute the inverse N-point DFT of $Y[k]$

Fast Fourier transform (FFT)

- Computation of an N-point DFT by the straightforward method requires $N^{(2)}$ complex multiplications
- FFT methods require only of the order of $N \,.\, \log N$ complex multiplications
- For example, for $N = 4096$, an FFT requires 300 times fewer operations than a straightforward DFT

Frequency ranges of some biological signals

• Electrocardiogram	0 –	45	(100) Hz
• Electromyogram	0 –	10	(200) Hz
• Electroencephalogram	0 –	45	(100) Hz

(Properties of the DFT)

Property	Time Domain	Frequency Domain
Notation	$x(n),\ y(n)$	$X(k),\ Y(k)$
Periodicity	$x(n) = x(n+N)$	$X(k) = X(k+N)$
Linearity	$a_1x_1(n) + a_2x_2(n)$	$a_1X_1(k) + a_2X_2(k)$
Time reversal	$x(N-n)$	$X(N-k)$
Circular convolution	$x_1(n) \circledN x_2(n)$	$X_1(k)X_2(k)$
Circular correlation	$x(n) \circledN y^*(-n)$	$X(k)Y^*(k)$
Multiplication of two sequences	$x_1(n)x_2(n)$	$\frac{1}{N}X_1(k) \circledN X_2(k)$

(Symmetry properties of the DFT)

N-Point Sequence $x(n)$, $0 \le n \le N-1$	N-Point DFT
$x(n)$	$X(k)$
$x^*(n)$	$X^*(N-k)$
$x^*(N-n)$	$X^*(k)$
Real Signals	
Any real signal	$X(k) = X^*(N-k)$
$x(n)$	$X_R(k) = X_R(N-k)$
	$X_I(k) = -X_I(N-k)$

(The overlap-save method for convolution)

- Task: to convolve signal *x*[*n*] with FIR filter of which unit-sample response *h*[*n*] is of length *M* over the interval [*0, M – 1*]

1. Divide the signal $x[n]$ into *overlapping* segments $x_k[n]$, each of length N, with an overlap of $M-1$ points between segments:

$$x_k[n] \triangleq \begin{cases} x[n + k(N-M+1)] & \text{if } 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases}$$

2. Form the cyclic convolution *modulo* N

$$z_k[n] \triangleq x_k[n] \circledast_N h[n]$$

by multiplying the N-point DFTs of $x_k[n]$ and $h[n]$ and taking the inverse DFT of the result. The resulting signal has length N.

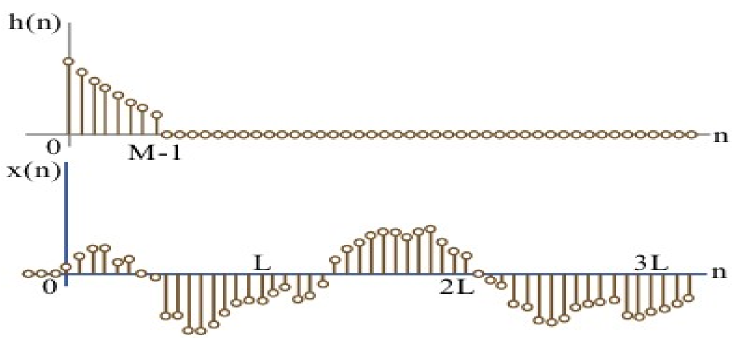
3. Form a new sequence $y_k[n]$ of length $N-M+1$ by discarding the first $M-1$ points of $z_k[n]$:

$$y_k[n] \triangleq \begin{cases} z_k[n] & \text{if } M-1 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases}$$

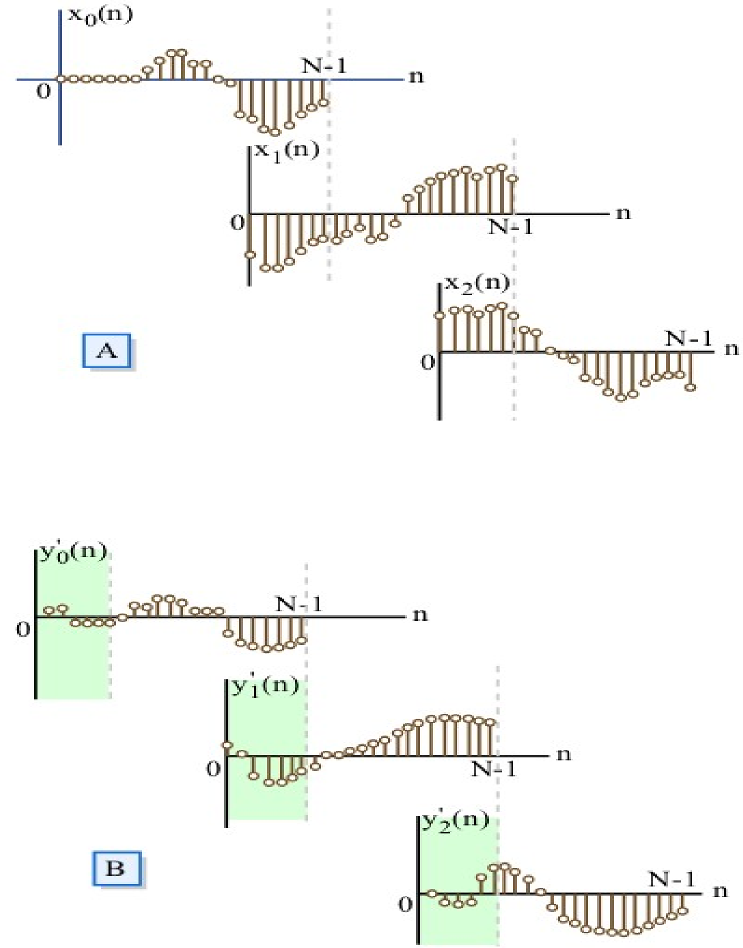
4. Form the final result $y[n]$ by joining the $y_k[n]$ with no overlap:

$$y[n] = \sum_{k=0}^{\infty} y_k[n - k(N-M+1)]$$

(The overlap-save method for convolution)



Finite-duration unit-sample response h(n) and signal x(n) to be filtered.



(a) Decomposition of x(n) into overlapping sections of length N

(b) Result of circular convolution of each section with h(n). The portions of each filtered section to be discard in forming the linear convolution are indicated.

(Bertrand Delgutte, MIT OpenCourseWare)