

Digital Signal Processing (DSP)

LECTURE 09

Infinite Impulse Response Filters (IIR)

- Discrete Fourier Transform (DFT) Analysis Representations of DFT Synthesis Representations DFT Basis Functions and Coefficients of DFT
- Applications of DFT

Spectrum Analysis System Frequency Response Analysis Modulation

- Properties of DFT
- Introduction to Fast Fourier Transform (FFT)

Discrete Fourier Transform (DFT)

Discrete Fourier Transform (DFT)

- Among the families of Fourier Transforms, DFT is the only member which can be implemented on a computer.
- DFT provides a mean whereby a **discrete-time periodic signal** can be decomposed into its equivalent sinusoidal signals represented in frequency domain.
- DFT is based on the claim that any continuous periodic signal could be represented as the sum of properly chosen sine and cosine waves of different frequencies and amplitudes.

Discrete Fourier Transform (DFT)

- **DFT Decomposition example:** A 16 point signal Fig. (1) is decomposed into 16 cosine waves in Fig. (2) and 16 sine waves Fig. (3).
- The decomposed sinusoidal waves each have different frequencies and amplitudes.
- The amplitude values are the DFT coefficients computed during the transform.



Figure 1. A 16 point signal

Discrete Fourier Transform (DFT)

Sine Waves



14

14

Figure 3. A 16 point cosine waves

Analysis Representation of DFT

• **DFT Analysis equations:** for a finite sample discrete-time signal x(n) with N sample points, the transform equation in can be derived from DTFT Eq. (5.1)

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \qquad Eq. 5.1$$

• The sequence x(n) is assumed to be **one period** of an infinite periodic signals with **sample rate N** and **frequency** k, then DFT is given by Eq. (5.2)

$$X(k) = \sum_{k=0}^{N-1} x(n) e^{-j2\pi kn/N} \qquad Eq. 5.2$$

Synthesis Representation of DFT

• **DFT Synthesis Equation:** the synthesis equations transform the signal from its frequency domain representation X(k) to its original time domain form x(n)

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \qquad Eq. 5.3$$

• An alternative representation of the synthesis equation in Eq. (5.3) is (5.4) using Euler's relation.

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cos\left(\frac{2\pi kn}{N}\right) + \frac{1}{N} j \sum_{k=0}^{N-1} X(k) \sin(\frac{2\pi kn}{N}) \quad Eq. 5.4$$

• The synthesis equations means that any arbitrary signal can be represented by sinusoidal waves of different frequencies and amplitudes.

DFT coefficients and Basis Functions

• **DFT coefficients:** the coefficients $(a_k, and b_k)$ of the DFT are the normalized values of the analysis equations X(k)

$$x(n) = \sum_{k=0}^{N-1} a_k \cos\left(\frac{2\pi kn}{N}\right) + \sum_{k=0}^{N-1} b_k \sin(\frac{2\pi kn}{N})$$

• **DFT basis functions:** The basis unit functions or vectors of DFT are the sines and cosines waves of unity amplitudes at different frequencies

$$\sum_{k=0}^{N-1} \cos\left(\frac{2\pi kn}{N}\right) and \sum_{k=0}^{N-1} \sin\left(\frac{2\pi kn}{N}\right)$$

• DTF basis functions are orthogonal vectors

• Problem Statement 1

Calculate the four-point DFT of the aperiodic sequence x(n) of length N = 4, which is defined as follows:

$$x(n) = \begin{cases} 2, & n = 0\\ 3, & n = 1\\ -1, & n = 2\\ 1, & n = 3 \end{cases}$$

Solution

Using the analysis equation of DFT in Eq. (5.2) for $0 \le k \le 3$

$$X(k) = \sum_{k=0}^{3} x(n) e^{-j2\pi kn/4}$$

- For k = 0 $X(0) = x(0)e^{-j2\pi * 0 * \frac{0}{4}} + x(1)e^{-j2\pi * 0 * \frac{1}{4}} + x(2)e^{-j2\pi * 0 * \frac{2}{4}} + x(3)e^{-j2\pi * 0 * 3/4}$ $X(0) = 2e^{-j2\pi * 0 * \frac{0}{4}} + 3e^{-j2\pi * 0 * \frac{1}{4}} - e^{-j2\pi * 0 * \frac{2}{4}} + e^{-j2\pi * 0 * 3/4}$ X(0) = 5• For k = 1
- For k = 1

$$X(1) = 2e^{-j2\pi * 1 * \frac{0}{4}} + 3e^{-j2\pi * 1 * \frac{1}{4}} - e^{-j2\pi * 1 * \frac{2}{4}} + e^{-j2\pi * 1 * 3/4}$$

X(1) = 3 - 2j

• For k = 2

$$X(2) = 2e^{-j2\pi * 2 * \frac{0}{4}} + 3e^{-j2\pi * 2 * \frac{1}{4}} - e^{-j2\pi * 2 * \frac{2}{4}} + e^{-j2\pi * 2 * 3/4}$$

$$X(2) = -3$$

• For k = 3

$$X(3) = 2e^{-j2\pi * 3 * \frac{0}{4}} + 3e^{-j2\pi * 3 * \frac{1}{4}} - e^{-j2\pi * 3 * \frac{2}{4}} + e^{-j2\pi * 3 * 3/4}$$

X(3) = 3 + j2

• Hence DFT X(k) of x(n) is

$$X(k) = \begin{cases} 5, & k = 0\\ 3 - 2j, & k = 1\\ -3, & k = 2\\ 3 + j2, & k = 3 \end{cases}$$

• Problem Statement 2

Calculate the inverse DFT of the which is defined as follows:

$$X(k) = \begin{cases} 5, & k = 0\\ 3 - 2j, & k = 1\\ -3, & k = 2\\ 3 + j2, & k = 3 \end{cases}$$

Solution

Using the synthesis equation of DFT in Eq. (5.3) for $0 \le k \le 3$

$$x(n) = \sum_{k=0}^{3} X(k) e^{j2\pi kn/4}$$

• For *k*=0

$$\begin{aligned} x(0) &= 5e^{-j2\pi * 0 * \frac{0}{4}} + (3 - 2j)e^{-j2\pi * 0 * \frac{1}{4}} \\ &+ (-3)e^{-j2\pi * 0 * \frac{2}{4}} + (3 + 2j)e^{-j2\pi * 0 * 3/4} \\ x(0) &= 2 \end{aligned}$$

• For *k*=1

$$\begin{aligned} x(1) &= 5e^{-j2\pi * 1*\frac{0}{4}} + (3-2j)e^{-j2\pi * 1*\frac{1}{4}} \\ &+ (-3)e^{-j2\pi * 1*\frac{2}{4}} + (3+2j)e^{-j2\pi * 1*3/4} \\ x(1) &= 3 \end{aligned}$$

• For *k*=2

$$x(2) = 5e^{-j2\pi * 0 * \frac{0}{4}} + (3 - 2j)e^{-j2\pi * 0 * \frac{1}{4}}$$
$$+ (-3)e^{-j2\pi * 0 * \frac{2}{4}} + (3 + 2j)e^{-j2\pi * 0 * 3/4}$$
$$x(2) = -1$$

• For *k*=3

$$\begin{aligned} x(0) &= 5e^{-j2\pi * 3*\frac{0}{4}} + (3-2j)e^{-j2\pi * 3*\frac{1}{4}} \\ &+ (-3)e^{-j2\pi * 3*\frac{2}{4}} + (3+2j)e^{-j2\pi * 3*3/4} \\ x(0) &= 1 \end{aligned}$$

• Hence the inverse of the DFT signal X(k) synthesizes the exact original signal x(n) and proves the DFT pairs

$$x(n) \stackrel{DFT}{\iff} = X(k)$$

$$\begin{cases} 2, & n = 0 \\ 3, & n = 1 \\ -1, & n = 2 \\ 1, & n = 3 \end{cases} \begin{cases} 5, & k = 0 \\ 3 - 2j, & k = 1 \\ -3, & k = 2 \\ 3 + j2, & k = 3 \end{cases}$$

DFT Applications

- **DFT** is one of the most important tools in Digital Signal Processing, and is applied in a wide range of applications:
- 1. Spectrum Analysis
- 2. Performing Convolution
- 3. Frequency Response of LTI system
- 4. Signal Modulation

Spectrum Analysis using DTF

- **Spectrum analysis:** This is a direct examination of information encoded in the frequency, phase, and amplitude of the component sinusoids
- Time domain representation of a signal recorded from underwater sensor (Fig. 4) and DFT transform of the signal detailing the various frequencies contents and amplitudes in the original signal.



Figure 4. Spectrum analysis

Spectrum Analysis using DTF

- **The** frequency spectrum analysis of the signal is used to analyze and remove any undesirable signals or noises from the original signal via **windowing** or **frequency domain filtering**
- **Signal Windowing:** is use to improve the spectral characteristics of a sampled signal where a special signals called window functions (Fig. 5) are multiplied with the original signal in the time-domain before DFT transform.



Figure 5. Hamming window

Spectrum Analysis using DTF

• Spectrum of a unwindowed signal



• Spectrum of a windowed signal using hamming window



Frequency Response Analysis

- Systems are analyzed in the time domain by using **convolution**. A similar analysis can be done in the frequency domain.
- System frequency Response H(k): A system's frequency response is the Fourier Transform (DFT) of its impulse response. The response describe how a system changes both the amplitude and phase of a signal passing through it.
- If the impulse response *h*(*n*) of a system is know then the frequency response

$$h(n) \stackrel{DFT}{\longleftrightarrow} H(k)$$

Frequency Response Analysis

- Similarly since convolution is equals to multiplication in frequency domain, system frequency response can alternatively be determined if the output of signal y(n) * is known using DFT
- Convolution in time-domain

$$x(n) * h(n) = y(n)$$

• DFT transform of convolution equation

$$X(k) * H(k) = Y(k)$$

• Hence frequency response *H*(*k*)

$$H(k) = \frac{Y(k)}{X(k)}$$

- 1. Linearity of the DFT: DFT exhibits properties of a linear transform i.e. is both *(i) homogenous and (ii) additive*
- *(i) homogeneity:* for a constant real value *c*



• *(ii) additivity:* if $x_1(n) \stackrel{DFT}{\longleftrightarrow} X_1(k)$ and $x_2(n) \stackrel{DFT}{\longleftrightarrow} X_2(k)$ then the DFT of the two signals combined is additive i.e.



2. Periodic Nature of the DFT: Unlike the other three Fourier Transforms, the DFT views both the time domain and the frequency domain as periodic.



3. Signal Compression: compression of the signal in one domain results in an expansion in the other, and vice versa.

x(n) compression $\stackrel{DFT}{\longleftrightarrow} X(k)$ expansion

 $x(n) expansion \stackrel{DFT}{\iff} X(k) compression$

4. Circular Convolution: Convolution of one signal becomes multiplication in other domains.

$$x_1(n) * x_2(n) \stackrel{DFT}{\iff} X_1(k) X_2(k)$$

$$x_1(n)x_2(n) \stackrel{DFT}{\Longleftrightarrow} X_1(k) * X_2(k)$$

• This properties can be used for **frequency modulation** and **deconvolution** operation

5. Parseval's Relation: The total energy E_x of signal in time domain x(n) is preserved when transformed to frequency domain.

$$E_x = \sum_{n=0}^{N-1} x(n)^2 = \frac{2}{N} \sum_{k=0}^{N-1} X(k)^2$$

• Parseval's theorem shows that the DFT preserves the energy of the signal within a scale factor of N

Introduction to Fast Fourier Transform (FFT)

Fast Fourier Transform (FFT): FFT are basically DFT algorithms with an efficient approach executing the both the analysis and synthesis equations of DFT.

- The computational complexity of DFT are significantly reduced by FFT algorithms
- There are several well known techniques that are used for computing the DFT using FFT.
- i. Radix-2,
- ii. Radix-4,
- iii. Split Radix,
- iv. Winograd, and
- v. Prime Factor Algorithms

Introduction to Fast Fourier Transform (FFT)

Computational Complexity of DFT: implementing DFT transform on a lengthy sequence present a real computational challenges.

- Consider the computational complexity of the direct implementation of the K-point DFT for the time-limited sequence *x*(*k*) with length *N*
- Based on the analysis equation of Eq. 5.2
- K complex multiplications and K 1 complex additions are required to compute a single DFT coefficient. Computation of all K DFT coefficients requires K^2 complex additions and K^2 complex multiplications

complex addition $\rightarrow O(K^2)$ complex multiplication $\rightarrow O(K^2)$

Introduction to Fast Fourier Transform (FFT)

Radix-2 FFT: the radix-2 algorithm is based on the following principle.

Proposition:

"For even values of K, the K-point DFT of a real-valued sequence x[k] with length $M \le K$ can be computed from the DFT coefficients of two subsequences: (i) x[2k], containing the even-valued samples of x[k], and (ii) x[2k + 1], containing the odd-valued samples of x[k]."

- **Fourier Transforms** have four fundamental categories depending on the nature of the signals.
- 1. Discrete Fourier Transform (DFT): this transformation is applied a signals x(n) that are both discrete and periodic. E.g. discrete sine waves.



2. Discrete-Time Fourier Transform (DTFT): this transformation is applied to signals x(n) that are both discrete and aperiodic. E.g. gaussian function



3. Fourier Series (FS): this transformation is applied to a signals x(t) that are both continuous-time and periodic. E.g. continuous-time sine waves.



4. Fourier Transform (FT): this transformation is applied to signals x(t) that are both continuous-time and aperiodic. E.g. Gaussian function



Problem statement 1: Find the Fourier transform of an LTI system whose input x(n) is a unit impulse $\delta(n)$

Solution Using Eq. (4.1) $X(e^{j\omega}) = \sum_{i=1}^{n} \delta(n)e^{-j\omega n}$ $X(e^{j\omega}) = \sum_{n=-\infty}^{-1} \delta(n)e^{-j\omega n} + \sum_{n=0}^{0} \delta(n)e^{-j\omega n} + \sum_{n=1}^{\infty} \delta(n)e^{-j\omega n}$ $\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & otherwise \end{cases}$ • since • Hence, $X(e^{j\omega}) = \sum_{n=1}^{\infty} \delta(n)e^{-j\omega n} = e^{-j\omega * 0} = 1$ n=0

Problem statement 2: Find the Fourier transform of an LTI system whose input x(n) is a unit step u(n)

Solution Using Eq. (4.1) $U(e^{j\omega}) = \sum_{\substack{n=-\infty \\ 0}}^{\infty} u(n)e^{-j\omega n}$ $U(e^{j\omega}) = \sum_{\substack{n=-\infty \\ n=-\infty}}^{-1} u(n)e^{-j\omega n} + \sum_{\substack{n=0 \\ n=-\infty}}^{\infty} u(n)e^{-j\omega n}$

• since

$$u(n) = \begin{cases} 1, & n \ge 0\\ 0, & otherwise \end{cases}$$