Lecture 03-8: Physical Layer IDFT and IFFT example

CS 356R

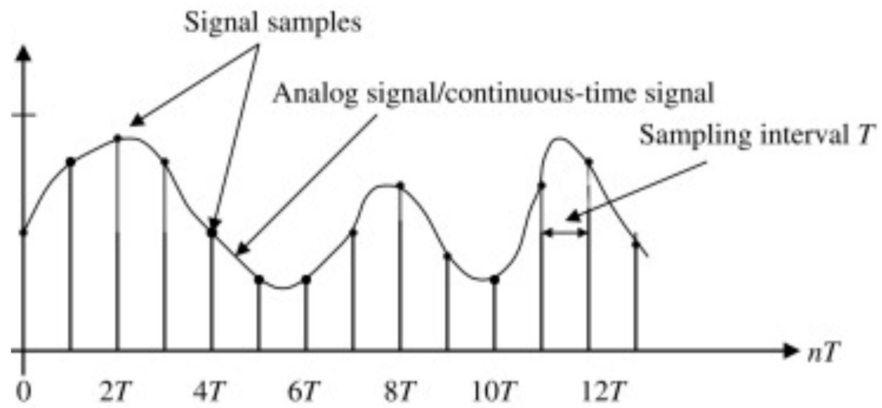
Intro to Wireless Networks

Mikyung Han

Outline

1. DFT and FFT Recap

Discrete Fourier Transform



- Let f(t) be 2L periodic continuous function in time-domain
- DFT is defined over "equally spaced" N samples of f(t)
  - Define T as the sampling interval
  - Sampling rate Fs = 1/T
  - Note NT = 2L
- Let f[0] f[1] f[2]… f[N-1] be the N samples of f(t) taken at t=0, T, 2T, 3T ...
- We can obtain the "equally spaced" Discrete-time Fourier Transform F[n]

$$F[n] = \sum_{k=0}^{N-1} f[k] e^{-j\frac{2\pi}{N}nk} \quad (n = 0 : N-1)$$

DFT Matrix

$$F[n] = \sum_{k=0}^{N-1} f[k] e^{-j\frac{2\pi}{N}nk} \quad (n = 0 : N-1)$$

- The same equation can be expressed as a matrix multiplication

freq-domain N of DTFT

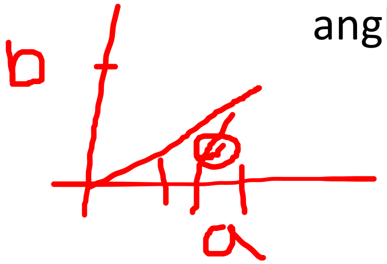
NxN DFT Matrix

time-domain
N samples of f(t)

$$\begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ \vdots \\ F[N-1] \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W & W^2 & W^3 & \dots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \dots & W^{N-2} \\ 1 & W^3 & W^6 & W^9 & \dots & W^{N-3} \\ \vdots & & & & & \\ 1 & W^{N-1} & W^{N-2} & W^{N-3} & \dots & W \end{pmatrix} \begin{pmatrix} f[0] \\ f[1] \\ f[2] \\ \vdots \\ f[N-1] \end{pmatrix}$$

where $W = \exp(-j2\pi/N)$

DFT results



angle(a+ib) = phi

- **Each F[n] values are complex value, where n=0: N-1**
  - |F[n]| represents how similar f(t) is to e^{iwt}
  - Angle formed by real(F[n]) and imag(F[n]) represents the phase offset from e^{iwt}

freq-domain N of DTFT

$$\begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ \vdots \\ F[N-1] \end{pmatrix}$$

- **n^{th} bin is mapped to actual frequency value nFs/N in Hz**
  - 0^{th} bin: 0 Hz
  - 1^{st} bin: Fs/N Hz
  - 2^{nd} bin: 2Fs/N Hz
  - …
  - …
  - $(N-1)^{th}$ bin: (N-1)Fs/N Hz

Only half of the DTFT is meaningful

- Based on Shannon Nyquist Sampling Theorem
  - To resolve all frequencies in a f(t), we must sample f(t) at $2 \times w_{max}$ where w_{max} is the highest frequency present in f(t)
  - $Fs > 2 \times w_{max}$ means

freq-domain N of DTFT

$$\begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ \vdots \\ F[N-1] \end{pmatrix}$$

- This means frequencies Fs/2 and is meaningless!
  - 0^{th} bin: 0 Hz
  - 1^{st} bin: Fs/N Hz
  - 2^{nd} bin: 2Fs/N Hz
  - …
  - …
  - $(N-1)^{th}$ bin: (N-1)Fs/N Hz

The latter half values are the "alias" of the first half

NEGATIVE FREQUENCY ZONE
(Often it is discarded)

DFT and FFT

- FFT is a fast implementation of DFT
- Typical NxN matrix multiplication takes $O(N^2)$
- FFT's runtime is $O(N\log_2 N)$
- Basic idea
  - Re-arranging matrix as a diagnonal matrix (linear operation) and dividing into even/odd (log N each step)

$$\hat{\mathbf{f}} = \mathbf{F}_{1024}\mathbf{f} = \begin{bmatrix} \mathbf{I}_{512} & -\mathbf{D}_{512} \\ \mathbf{I}_{512} & -\mathbf{D}_{512} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{512} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{512} \end{bmatrix} \begin{bmatrix} \mathbf{f}_{\text{even}} \\ \mathbf{f}_{\text{odd}} \end{bmatrix} \qquad \mathbf{D}_{512} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \omega & 0 & \cdots & 0 \\ 0 & 0 & \omega^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \omega^{511} \end{bmatrix}$$

- This video illustrates cleverly the idea of FFT

Simply we call f_hat = fft(f), where f is 1xN vector of samples

Outline

1. Fourier Series Definition
2. Power Spectral Density

Matlab Example: A signal with phase shift

- fft_with_phase.m

Power Spectral Density

- One of the measures to analyze FFT results
- Aka Power Spectrum
- Describes how much power is present per frequency
- How to calculate
  - f_hat = fft(f)
  - PSD = f_hat. x conj(f_hat)/N
  - Let f_hat be 1x N vector where each complex value is a_k+ib_k ($k=1 \ldots N$)

Note magnitude calculates sqrt(a_k^2+bk^2)
whereas PSD calculates $\frac{1}{N}(a_k^2+b_k^2)$

Outline

1. Fourier Series Definition
2. Power Spectral Density
3. Inverse Discrete Fourier Transform

Inverse Discrete Fourier Transform (IDFT)

- The inverse of $F[n] = \sum_{k=0}^{N-1} f[k] e^{-j\frac{2\pi}{N}nk}$ is given by $f[k] = \frac{1}{N}\sum_{n=0}^{N-1} F[n] e^{+j\frac{2\pi}{N}nk}$

time-domain
N samples of f(t)

IDFT Matrix

freq-domain N of DTFT

$$
\begin{pmatrix} f[0] \\ f[1] \\ f[2] \\ \vdots \\ f[N-1] \end{pmatrix} = \frac{1}{N}\begin{bmatrix} \omega^{-0\cdot 0} & \omega^{-0\cdot 1} & \omega^{-0\cdot 2} & \cdots & \omega^{-0\cdot(N-1)} \\ \omega^{-1\cdot 0} & \omega^{-1\cdot 1} & \omega^{-1\cdot 2} & \cdots & \omega^{-1\cdot(N-1)} \\ \omega^{-2\cdot 0} & \omega^{-2\cdot 1} & \omega^{-2\cdot 2} & \cdots & \omega^{-2\cdot(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^{-(N-1)\cdot 0} & \omega^{-(N-1)\cdot 1} & \omega^{-(N-1)\cdot 2} & \cdots & \omega^{-(N-1)\cdot(N-1)} \end{bmatrix} \begin{pmatrix} F[0] \\ F[1] \\ F[2] \\ \vdots \\ F[N-1] \end{pmatrix}
$$

where $W = \exp(-j2\pi/N)$

IDFT and IFFT

- IFFT is a fast implementation of IDFT
- Almost the same operation as FFT
  - The sign is flipped on the exponent
  - Has 1/N normalizing factor
- IFFT and FFT has the same time complexity of O(N logN)

Simply we call f = ifft(f_hat) to recover the original sampled signal f

Outline

1. Fourier Series Definition
2. Power Spectral Density
3. Inverse Discrete Fourier Transform
4. Matlab Denoise Exercise using FFT and IFFT