

ELC 4351: Digital Signal Processing

Liang Dong

Electrical and Computer Engineering
Baylor University

liang.dong@baylor.edu

April 6, 2017

Discrete Fourier Transform

- 1 Fourier Transform and Discrete Fourier Transform
- 2 Fast Fourier Transform Algorithms

The Fourier Transform of Discrete-Time Aperiodic Signals

Analysis Equation

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}, \quad \omega \in [-\pi, \pi) \text{ or } \omega \in [0, 2\pi)$$

Synthesis Equation

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

$X(\omega)$ is periodic with period 2π .

The Discrete Fourier Transform (DFT)

N -point DFT.

Analysis Equation

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi\frac{k}{N}n}, \quad k = 0, 1, 2, \dots, N-1$$

Synthesis Equation

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi\frac{k}{N}n}, \quad n = 0, 1, 2, \dots, N-1$$

N samples of the Fourier transform at N equally spaced frequencies.

$$\omega_k = \frac{2\pi k}{N}, \quad k = 0, 1, 2, \dots, N-1.$$

A Fast Algorithm for the DFT

Analysis Equation

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad k = 0, 1, 2, \dots, N-1$$

Synthesis Equation

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \quad n = 0, 1, 2, \dots, N-1$$

where, $W_N = e^{-j2\pi/N}$.

A Fast Algorithm for the DFT

Analysis Equation

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, \quad k = 0, 1, 2, \dots, N-1$$

To calculate one frequency sample (each k) in the analysis equation (direct Fourier transform), we need N complex multiplications and $N - 1$ complex additions.

For all N frequency samples, we need a total of N^2 complex multiplications and $N(N - 1)$ complex additions.

A Fast Algorithm for the DFT

Analysis Equation

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, \quad k = 0, 1, 2, \dots, N-1$$

For all N frequency samples, we need a total of $4N^2$ real multiplications and $N(4N - 2)$ real additions.

A Fast Algorithm for the DFT

Analysis Equation

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, \quad k = 0, 1, 2, \dots, N-1$$

The computation complexity of the DFT is proportional to N^2 .

As a comparison, the computational complexity of the FFT is proportional to $N \log N$.

Periodicity of W_N^{kn}

Use the periodicity of the sequence W_N^{kn} to reduce computation.

$$W_N^{kN} = e^{-j\frac{2\pi}{N}kN} = e^{-j2\pi k} = 1$$

$$W_N^{k(N-n)} = W_N^{-kn} = (W_N^{kn})^*$$

(complex conjugate symmetry)

$$W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$$

(periodicity)

Decimation-in-Time Fast Fourier Transform (FFT)

Considering N an integer power of 2, i.e. $N = 2^\nu$.

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k = 0, 1, \dots, N-1 \\ &= \sum_{n \text{ even}} x(n)W_N^{nk} + \sum_{n \text{ odd}} x(n)W_N^{nk} \\ &= \sum_{r=0}^{N/2-1} x(2r)W_N^{2rk} + \sum_{r=0}^{N/2-1} x(2r+1)W_N^{(2r+1)k} \\ &= \sum_{r=0}^{N/2-1} x(2r)(W_N^2)^{rk} + W_N^k \sum_{r=0}^{N/2-1} x(2r+1)(W_N^2)^{rk} \end{aligned}$$

Decimation-in-Time Fast Fourier Transform (FFT)

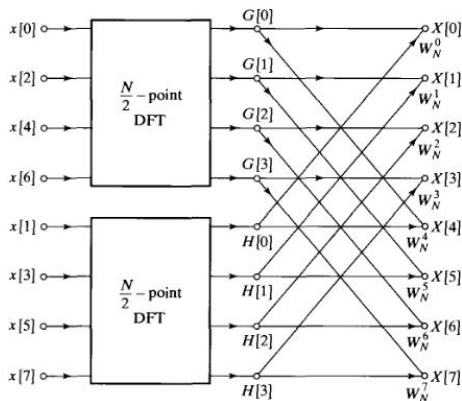
$$W_N^2 = e^{-2j\frac{2\pi}{N}} = e^{-j\frac{2\pi}{N/2}} = W_{N/2}$$

Therefore,

$$\begin{aligned} X(k) &= \sum_{r=0}^{N/2-1} x(2r)(W_N^2)^{rk} + W_N^k \sum_{r=0}^{N/2-1} x(2r+1)(W_N^2)^{rk} \\ &= \underbrace{\sum_{r=0}^{N/2-1} x(2r)W_{N/2}^{rk}}_{G(k)} + W_N^k \underbrace{\sum_{r=0}^{N/2-1} x(2r+1)W_{N/2}^{rk}}_{H(k)} \\ &= G(k) + W_N^k H(k) \end{aligned}$$

Decimation-in-Time Fast Fourier Transform (FFT)

$G(k)$ is an $(N/2)$ -point DFT of even samples $x(2r)$;
 $H(k)$ is an $(N/2)$ -point DFT of odd samples $x(2r + 1)$.

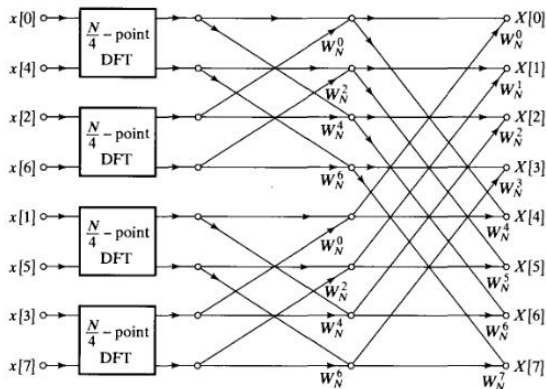


Decimation-in-Time Fast Fourier Transform (FFT)

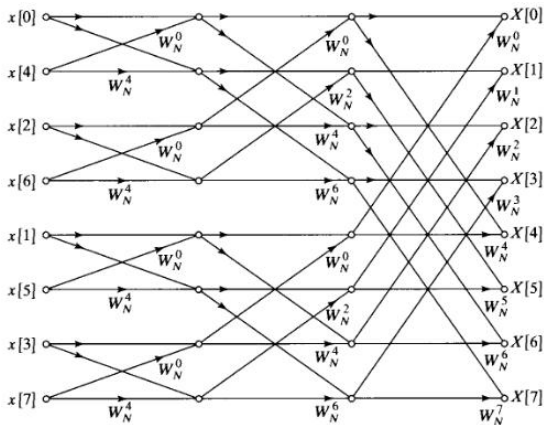
Continue...

$$G(k) = \sum_{l=0}^{N/4-1} g(2l)W_{N/4}^{lk} + W_{N/2}^k \sum_{l=0}^{N/4-1} g(2l+1)W_{N/4}^{lk}$$
$$H(k) = \underbrace{\sum_{l=0}^{N/4-1} h(2l)W_{N/4}^{lk}}_{(N/4)\text{-point DFT}} + W_{N/2}^k \underbrace{\sum_{l=0}^{N/4-1} h(2l+1)W_{N/4}^{lk}}_{(N/4)\text{-point DFT}}$$

Decimation-in-Time Fast Fourier Transform (FFT)



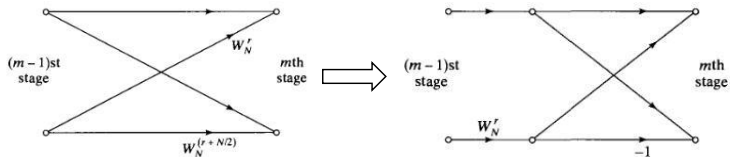
Decimation-in-Time Fast Fourier Transform (FFT)



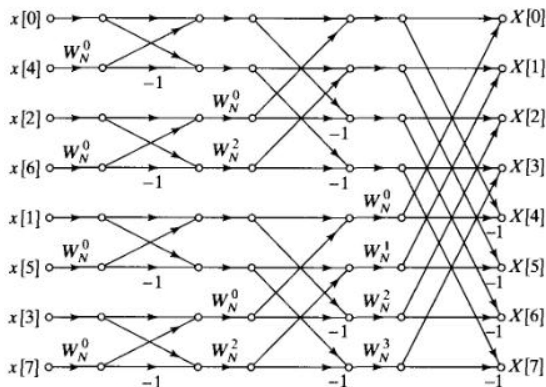
Symmetry of W_N

$$W_N^{N/2} = e^{-j\frac{2\pi}{N}\frac{N}{2}} = e^{-j\pi} = -1$$

$$W_N^{r+N/2} = W_N^{N/2} W_N^r = -W_N^r$$



Decimation-in-Time Fast Fourier Transform (FFT)



Bit-reverse Reordering

$x(n)$'s index n	binary
0	000
4	100
2	010
6	110
1	001
5	101
3	011
7	111

Decimation-in-Frequency Fast Fourier Transform (FFT)

