Applied Signal Processing
Lecture 5

Tomas McKelvey

Department of Electrical Engineering
Chalmers University of Technology

Agenda

- Fast Fourier Transform
- Using FFT for filtering streaming signals
  - Overlap and Add method
- Frequency Analysis
- Window effects

The complexity of DFT

The DFT is defined as

$$\text{DFT:} \quad X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}}$$

where the integer index k ranges from 0 to $N-1$.

From a computationally complexity point of view, the multiplications dominate the cost

A brute force calculation of the DFT yields N^2 multiplications

Fast Fourier Transform (FFT)

FFT is a computational procedure to calculate the DFT

DFT of a length N sequence $\{x(n)\}_{n=0}^{N-1}$ is defined as

$$\text{DFT:} \quad X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} = \sum_{n=0}^{N-1} x(n) W_N^{kn}$$

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

- index k corresponds to a frequency $\omega_k = \frac{2\pi k}{N\Delta t} = \frac{k}{N}\omega_s$
- $X(k)$ is $N-$periodic, i.e. $X(k) = X(k+N)$.

We note that $W_N^{2kn} = W_{N/2}^{kn}$

Assume $N = 2^p$

Split the DFT sum in even and odd time indices

$$
\begin{aligned}
X(k) &= \sum_{n=0}^{N/2-1} x(2n)\underbrace{W_N^{k2n}}_{W_{N/2}^{kn}} + \sum_{n=0}^{N/2-1} x(2n+1)\underbrace{W_N^{k(2n+1)}}_{W_N^k W_{N/2}^{kn}} \\
&= \underbrace{\sum_{n=0}^{N/2-1} x(2n)W_{N/2}^{kn}}_{X_e(k)} + W_N^k \underbrace{\sum_{n=0}^{N/2-1} x(2n+1)W_{N/2}^{kn}}_{X_o(k)} \\
&= X_e(k) + W_N^k X_o(k)
\end{aligned} \tag{1}
$$

We note that $X_e(k)$ and $X_o(k)$ are DFT of length $N/2$.

$\Rightarrow X_e(k+N/2) = X_e(k)$ and $X_o(k+N/2) = X_o(k)$

FFT final stage

Given $X_e(k)$ and $X_o(k)$ are already calculated for $k = 0, 1, \ldots, N/2 - 1$ we get the final result by

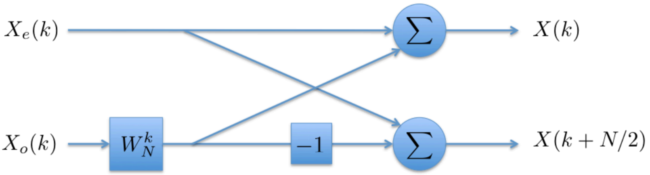
$$X(k) = X_e(k) + W_N^k X_o(k), \quad k = 0, 1, \ldots N/2 - 1$$

for indices $k' = N/2, \ldots, N - 1$ we have (since X_e and X_o are periodic)

$$\begin{aligned} X(k + N/2) &= X_e(k) + W_N^{k+N/2} X_o(k) \\ &= X_e(k) - W_N^k X_o(k), \quad k = 0, \ldots, N/2 - 1 \end{aligned}$$

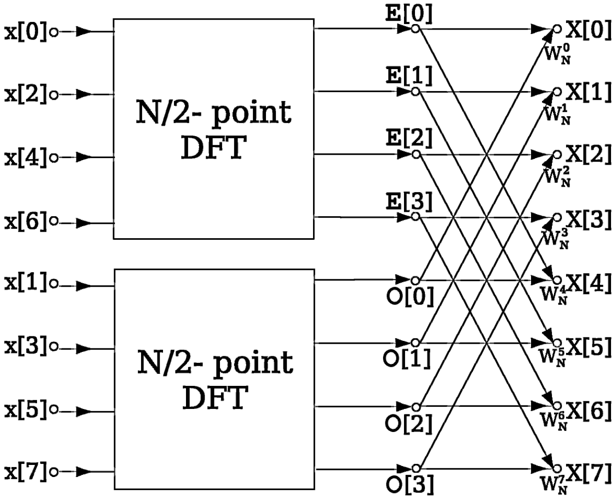
In total we need to perform $N/2$ complex multiplications for this final stage.

FFT Butterfly



$$X(k) = X_e(k) + W_N^k X_o(k)$$
$$X(k+N/2) = X_e(k) - W_N^k X_o(k), \quad k = 0, \ldots, N/2-1$$

W_N^k is known as the "twiddle factor".


x[0]
x[2]
x[4]
x[6]
N/2- point DFT
x[1]
x[3]
x[5]
x[7]
N/2- point DFT
E[0]
E[1]
E[2]
E[3]
O[0]
O[1]
O[2]
O[3]
X[0]
X[1]
X[2]
X[3]
X[4]
X[5]
X[6]
X[7]
W_N^0
W_N^1
W_N^2
W_N^3
W_N^4
W_N^5
W_N^6
W_N^7

FFT as a recursive tree

In the next stage we perform 4 DFT, each of length $N/4$. And to assemble the results we need $2 * N/4 = N/2$ multiplications.

$N/4$ to calculate X_e and $N/4$ to calculate X_o.

In the stage that follows we perform 8 DFT, each of length $N/8$. And to assemble the results we need $4 * N/8 = N/2$ multiplications.

In the final stage we have $N/2$ DFTs of length 2 which have the trivial result (for each of the N/2 DFTs)

$$X(0) = x(0) + x(1)$$
$$X(1) = x(0) - x(1)$$

And to assemble the results again we need $N/2$ multiplications.

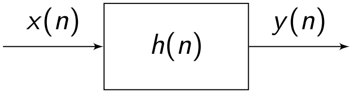
FFT summary

We have $N = 2^p$ or $p = \log_2 N$.

We have $p - 1$ stages in which each requires $N/2$ complex multiplications

In total we need $N/2(p-1) = N/2(\log_2 N - 1) \sim N \log_2 N$ multiplications.

Data len. (N)	DFT compl. (N^2)	FFT compl. ($N/2 \log_2 N$)
8	64	12
64	4096	192
128	16384	448
1024	1048576	5120



$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

Basic idea:

- Divide the input into consecutive blocks of some finite size N
- Use the *finite signals* strategy discussed in last lecture on each consecutive block
- Assemble the resulting output signals in the correct way

Overlap and add method:

- Each block will generate $N + M - 1$ long output
- The “tail” of size $M - 1$ need to be added to the beginning of the output of the next block.