Date: / /

Experiment No - 5

Aim: Implement fast fourier Transform

Theory:

The Direct fourier Transform (DTF) can be written as follows:

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

To determine the DTF of discrete signal x[n], we multiply each of its value of e raised to some function of n. We then sum the results obtained for a given n. As the name implies, the Fast fourier Transform [FFT] is an algorithm that determines DTF of an input significantly faster than computing it directly. FFT reduces number of computations needed for a problem of size N from $O(N^2)$ to $O(N \log N)$

N	1000	10^6	10^9
N^2	10^6	10^{12}	10^{18}
$N\log_2 N$	10^4	20×10^6	30×10^9

Date: / /

Suppose, we seperated the fourier Transform into even and odd indexed sub-sequence.

$n = 2r$, if even
$n = 2r+1$, if odd
where,
$r = 1, 2, \ldots, \frac{N}{2} - 1$

Later, we end up with summation of the terms. The advantage of this approach lies in the fact that the even and odd indexed sub-sequences can be completed concurrently.

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi k n}{N}}$$

$$x[k] = \sum_{r=0}^{N/2-1} x[2r] e^{-j\frac{2\pi k(2r)}{N}} + x[k] \sum_{r=0}^{N/2-1} x[2r+1] - e^{-j\frac{2\pi k(2r+1)}{N}}$$

$$x[k] = \sum_{r=0}^{N/2-1} x[2r] e^{-j\frac{2\pi k(r)}{N/2}} + x[k] \cdot e^{-\frac{2j\pi k}{N}} \sum_{r=0}^{N/2-1} x[2r+1] e^{-j\frac{2\pi k[r]}{N/2}}$$

$$x[k] = x_{even}[k] + e^{-j\frac{2\pi k}{N}} x_{odd}[k]$$