Experiment 4

Shashwat Shah
60004220126
BE comps C22

Aim: Discrete Fourier Transform on Discrete time signals.

Theory: The discrete fourier transform is a fundamental tool in signal processing that transforms a time domain signal into its frequency domain representation. Time and frequency Domains.

In the time domain, signals are viewed as functions of time. However many practical applications require analyzing the frequency content of the signals such as communication system, audio processing and vibration analysis.

The DFT formula.

The DFT of a discrete time signal $x(n)$ is given by.

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

where

$X(k)$ is the DFT, N is the number of samples and k represent the frequency index.

Conclusion : This experiment demonstrates the utilities the DFT in understanding the spectral characte of signal including the identification of dominant frequ and periodicity. Overall, the application of the DFT in signal analysis is crucial for task such as filtering, signal reconstruction and spectral analysis in digital signal processing.

NAME: Shashwat Shah SAP ID: 60004220126
DIV/BATCH:C22
Date : 23/09/24

DIGITAL SIGNAL PROCESSING (DSP)
EXPERIMENT 04

AIM: To apply Discrete Fourier Transform on DT signal

CODE:

```
% Function to calculate DFT manually function X =

DFT_manual(x)

    N = length(x);  % Number of points
    X = zeros(1, N);  % Initialize the result array for k = 0:N-1
        for n = 0:N-1
            angle = 2 * pi * k * n / N;
            X(k+1) = X(k+1) + x(n+1) * exp(-1j * angle);
     end   end

end

N4 = 4; % N=4 N8
= 8; % N=8

% Input signal for N=4
disp('Enter 4 values for the signal (N=4):'); x4 = input('Signal:
');

if length(x4) ~= N4
    disp('Please enter exactly 4 values.'); else
    % Calculate DFT manually for N=4 dft_manual_4 =
    DFT_manual(x4); disp('Manual DFT result for
    N=4:'); disp(dft_manual_4);

    % Calculate using FFT (direct function) dft_fft_4 =
    fft(x4);
    disp('FFT result for N=4:');
    disp(dft_fft_4);

    % Compare results
    if isequal(round(dft_manual_4, 10), round(dft_fft_4, 10)) disp('The manual DFT and
        FFT results match for N=4.');
    else
        disp('The manual DFT and FFT results do not match for N=4.');
    end
end

% Input signal for N=8
disp('Enter 8 values for the signal (N=8):'); x8 = input('Signal:
');
```

```
    if length(x8) ~= N8
        disp('Please enter exactly 8 values'); else
        % Calculate DFT manually for N=8 dft_manual_8 =
        DFT_manual(x8); disp('Manual DFT result for
        N=8:'); disp(dft_manual_8);

        % Calculate using FFT (direct function) dft_fft_8 =
        fft(x8);
        disp('FFT result for N=8:');
        disp(dft_fft_8);

        % Compare results
        if isequal(round(dft_manual_8, 10), round(dft_fft_8, 10)) disp('The manual DFT and
            FFT results match for N=8.');
        else
            disp('The manual DFT and FFT results do not match for N=8.');
        end
end
```

OUTPUT:

```
Signal:
[1 2 3 4]
Manual DFT result for N=4:
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 - 0.0000i  -2.0000 - 2.0000i

FFT result for N=4:
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 + 0.0000i  -2.0000 - 2.0000i

The manual DFT and FFT results match for N=4.
Enter 8 values for the signal (N=8):
Signal:
[1 2 3 4 5 6 7 8]
Manual DFT result for N=8:
  36.0000 + 0.0000i  -4.0000 + 9.6569i  -4.0000 + 4.0000i  -4.0000 + 1.6569i  -4.0000 - 0.0000i  -4.0000 - 1.6569i  -4.0000 - 4.0000i  -4.0000 - 9.6569i

FFT result for N=8:
  36.0000 + 0.0000i  -4.0000 + 9.6569i  -4.0000 + 4.0000i  -4.0000 + 1.6569i  -4.0000 + 0.0000i  -4.0000 - 1.6569i  -4.0000 - 4.0000i  -4.0000 - 9.6569i

The manual DFT and FFT results match for N=8.
```