# Experiment 8 Discrete Fourier Transform, DFT

|  |
| --- |
| Objective: Computation of Fourier transform is mathematically tedious. In this laboratory exercise, MATLAB software is utilized to expedite the computation of DFT of a given signal. The discrete version of the spectrum obtained is examined. |

## Background information

A continuous-time signal is sampled to make the waveform digital so that we can easily store or process the signal in the time domain. However, the frequency spectrum of a discrete-time signal is normally not discrete (though it is periodic). In order to digitally process signals in time and frequency domains, discretization in both domains is required. In another words, the spectrum has also to be ‘sampled’ and encoded into binary code words. Discrete Fourier transform (DFT) is a mathematical tool used to obtain a discrete spectrum from a discrete-time signal.

These equations are the basic definitions for discrete Fourier transform (DFT) and the inverse discrete Fourier transform (IDFT).

Discrete Fourier transform (DFT):

 

Inverse discrete Fourier transform (IDFT):

 

where x(n) is the discrete time signal and X(k) is the discrete frequency component.

**Procedure 1: DFT of a sequence**

* 1. Given a signal *x*(*n*) = {1, 2, 0, 0, 1, 2, 0, 0}. Manually compute its 8-point DFT using the DFT formula:

 

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *k* | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| *X*(*k*) |  |  |  |  |  |  |  |  |

* 1. Enter the following codes as a MATLAB script file and save as ‘TestDFT.m’. It will compute the 8-point DFT of *x*(*n*) and measure elapsed computation time.

x=[1 2 0 0 1 2 0 0]; n=[0:7];

tic %start timer for k=0:7,

D(k+1)=sum(x.\*exp(-j\*2\*pi\*n\*k/8));

end;

toc %stop timer D

Execute the script and compare your results with those obtained in step 1-1. Are they the same? What is the elapsed time of computation?

* 1. An FFT routine is provided by MATLAB to compute the Fourier transform.

Enter the following command to understand this function.

>>help fft

* 1. Enter the following codes as a MATLAB script file and save as ‘TestFFT.m’.

x=[1 2 0 0 1 2 0 0];

tic %start timer E=fft(x);

toc %stop timer E

Compare your results with those obtained in steps 1-1 and 1-2. Are they the same? What is the elapsed computation time for FFT?

* 1. The discrete magnitude and phase spectrum of *x*(*n*) may be obtained using the following commands:

>>subplot(211),stem([0:7],abs(E),'filled');

>>ylabel('Magnitude');

>>subplot(212),stem([0:7],angle(E),'filled');

>>ylabel('Phase');

Consider another signal composed of two sine waves and sampled with a 8000-Hz sampling frequency.

>>n=0:1023;

>>x1=2\*sin(2\*pi\*500/8000\*n)+sin(2\*pi\*1000/8000\*n);

>>clf;

>>stem(x1(1:127),'filled');

>>xlabel('sample');ylabel('x1(n)');grid on;

What are the frequencies of the two sine waves?

* 1. Compute the 1024-point DFT of the signal and plot the magnitude spectrum:

>>c=fft(x1,1024);

>>stem(n,abs(c),'filled');grid on;

>>ylabel('Magnitude of x1');

Does the magnitude spectrum show two sine waves? What are the frequencies?

What is the frequency resolution of the spectrum?

What is the frequency at the sample *k* = 130 in terms of normalized (radians) and actual frequency (Hz)?

**Procedure 2: Using DFT to analyze signals**

* 1. A certain signal, sampled at 1000 Hz, has been stored as a data file, ‘sg1.mat’. Load the signal into the MATLAB workspace and perform a 256- point DFT on it:

>>load sg1;

>>y=fft(sg1,256);

>>f=[0:255]\*1000/256;

>>stem(f,abs(y));

From the frequency spectrum, can you tell what this signal, ‘sg1’, constitutes of?

* 1. Observe the signal in the time domain by:

>>plot(sg1);