Signals Systems and Communication Lab

Laboratory report submitted for the partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Electronics and Communication Engineering

by

Mohit Akhouri - 19ucc023

Course Coordinator
Dr. Navneet Upadhyay

LNMIIT
The LNM Institute of Information Technology

Department of Electronics and Communication Engineering
The LNM Institute of Information Technology, Jaipur

February 2021

Contents

Chapter *2*

Experiment - 2

2.1 Aim of the experiment

To compute and plot the Fourier spectra for the aperiodic signals

2.2 Software Used

MATLAB

2.3 Theory

2.3.1 About Aperiodic Signals :

A signal that does not repeat itself after a specific interval of time is called an aperiodic signal. By applying a limiting process, the signal can be expressed as a continuous sum (or integral) of everlasting exponentials. These signals are analysed by means of the **Fourier Transform**.

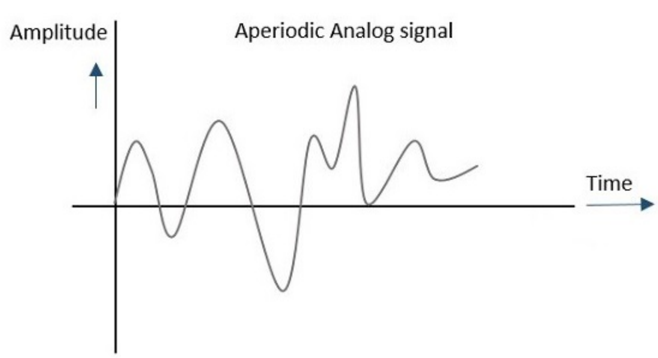


Figure 2.1 Aperiodic Analog Signal

2.3.2 About Fourier Series :

Jean B. Joseph Fourier was a French mathematician who proposed an idea that any periodic signal can be represented by addition of scaled basis signals of different frequencies(harmonics). This idea was later termed as **Fourier Series Representation**. Basically, Fourier Series is an expansion of a periodic function terms of an infinite sum of sines and cosines. It makes use of **orthagonality** relationships of sine and cosine functions. The computation and study of Fourier series is known as harmonic analysis and is extremely useful as a way to break up an arbitrary periodic function into a set of simple terms that can be plugged in, solved individually, and then recombined to obtain the solution to the original problem or an approximation to it to whatever accuracy is desired or practical. The fourier series of a periodic function f(x) of period T is :

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \frac{cos(2\pi kx)}{T} + \sum_{k=1}^{\infty} b_k \frac{sin(2\pi kx)}{T} \tag{2.1}$$

for some set of Fourier coefficients a_k and b_k defined by the integrals :

$$a_k = \frac{2}{T} \int_0^T f(x) \frac{cos(2\pi kx)}{T} dx \tag{2.2}$$

$$b_k = \frac{2}{T} \int_0^T f(x) \frac{sin(2\pi kx)}{T} dx \tag{2.3}$$

2.3.3 About Fourier Transform :

The Fourier transform expresses a function of time (a signal) as a function of frequency.The Fourier transform of a function of time itself is a complex-valued function of frequency, whose complex modulus represents the amount of that frequency present in the original function, and whose complex argument is the phase offset of the basic sinusoid in that frequency.The Fourier transform is called the frequency domain representation of the original signal.For many functions of practical interest one can define an operation that reverses this: the inverse Fourier transformation, also called Fourier synthesis, of a frequency domain representation combines the contributions of all the different frequencies to recover the original function of time. Joseph Fourier introduced the transform in his study of heat transfer, where Gaussian functions appear as solutions of the heat equation. The fourier transform of an aperiodic continuous time signal x(t) is :

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \tag{2.4}$$

Let us consider the samples of X() at regular intervals of ω_0. If X_r is the r^{th} sample the from equation 2.4 we obtain :

$$X_r = \sum_{k=0}^{N_0-1} x_k e^{-jr\Omega_0 k} \tag{2.5}$$

where $x_k = T_s x(kT_s)$,$X_r = X(r\omega_0)$ and $\Omega_0 = \omega_0 T_s$.

2.4 Code and Results

2.4.1 Exercise 1 : Creating function X = mydft(x,t_0,t_s) to generate DFT of x

```
function mydft(x,to,ts)
% This function takes three arguments :
% a sequence 'x', sampling interval 'ts' and to
% This function will calculate the discrete fourier transform of
 signal 'x'

% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

%initializing N and fs
N = to/ts;
fs = 1/ts;
%creating a empty array to store the result using zeros function
X = zeros(1,N);
%loop algorithm to calculate the DFT of signal x1(t)
for k=1:N
    sum = 0;
    for n=1:N
        sum = sum + (x(n).*(exp(-1i*2*pi*(k-1)*(n-1)/N)));
    end
    X(k) = sum;
end
f = linspace(-fs,fs,N); %defining frequency from -fs to fs
%plotting the graphs (magnitude plot and phase plot)
subplot(2,2,1);
plot(f,abs(fftshift(X)));
xlabel('Frequency (Hz) ->');
ylabel('Magnitude ->');
title('Mohit Akhouri-19ucc023','Magnitude Plot using function mydft');
grid on;
subplot(2,2,3);
plot(f,angle(X));
xlabel('Frequency (Hz) ->');
ylabel('Phase angle ->');
title('Mohit Akhouri-19ucc023','Phase Plot using function mydft');
grid on;
```

Published with MATLAB® R2020b

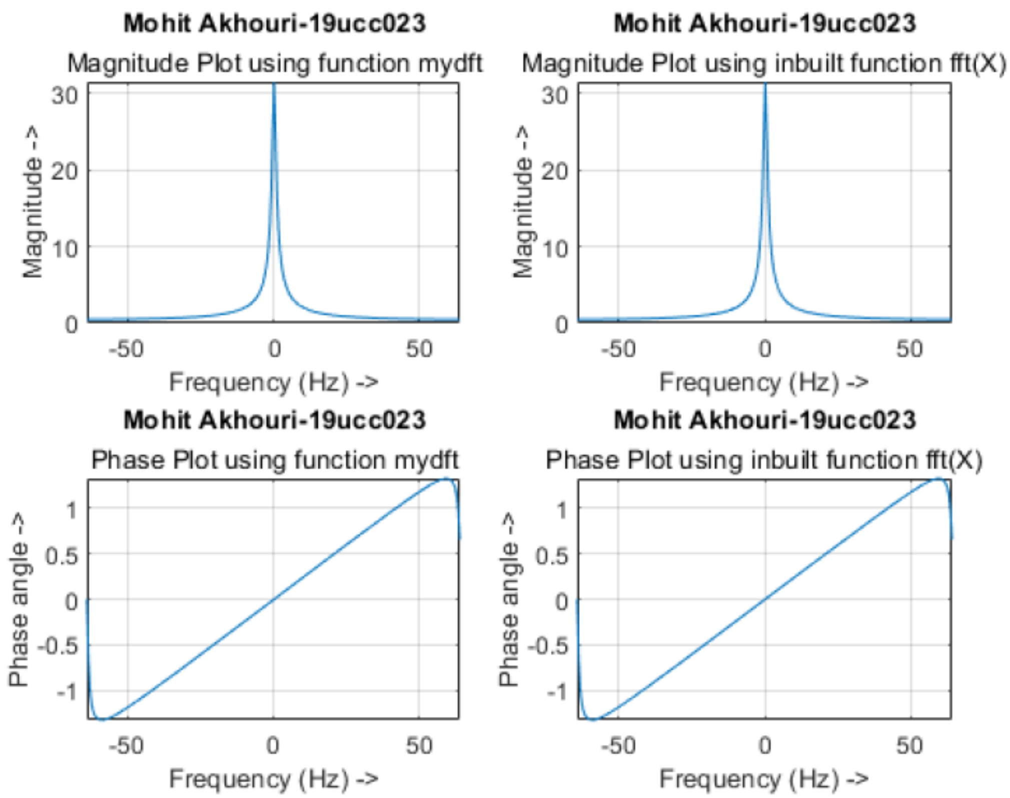
Figure 2.2 Code for the function mydft designed for finding DFT of x

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% Code for calling mydft(x,to,ts) to calculate fourier transform of x

%defining constants to,ts,fs and N
to = 4;
ts = 1/64;
fs = 1/ts;
N = to/ts;
%initializing frequency variable from -fs to fs using linspace
f = linspace(-fs,fs,N);
t = ts:ts:to;
ut = t>=0; %defining u(t) function
xt = (exp(-2*t)).*ut; %defining x1(t)
mydft(xt,to,ts);%calling function mydft to calculate dft of x1(t)
%plotting the graphs (magnitude plot and phase plot)
subplot(2,2,2);
plot(f,abs(fftshift(fft(xt))));
xlabel('Frequency (Hz) ->');
ylabel('Magnitude ->');
title('Mohit Akhouri-19ucc023','Magnitude Plot using inbuilt function
 fft(X)');
grid on;
subplot(2,2,4);
plot(f,angle(fft(xt)));
xlabel('Frequency (Hz) ->');
ylabel('Phase angle ->');
title('Mohit Akhouri-19ucc023','Phase Plot using inbuilt function
 fft(X)');
grid on;
```

Figure 2.3 Code for calculating DFT when T_0 = 4 sec and T_s = 1/64 sec



Published with MATLAB® R2020b

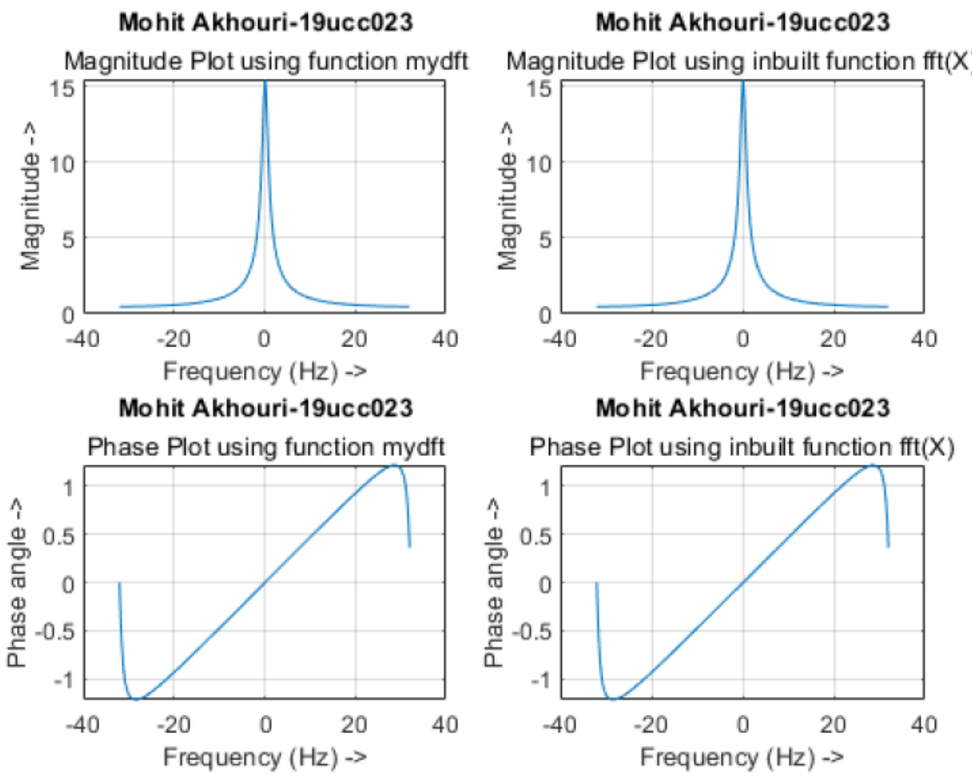
Figure 2.4 Graph of calculated DFT when T_0 = 4 sec and T_s = 1/64 sec

```
% Name = Mohit Akhouri
% Roll no = 19UCC023
% SSC LAB Batch D1 - Monday ( 2-5 pm )

% Code for calling mydft(x,to,ts) to calculate fourier transform of x

%defining constants to,ts,fs and N
to = 8;
ts = 1/32;
fs = 1/ts;
N = to/ts;
%initializing frequency variable from -fs to fs using linspace
f = linspace(-fs,fs,N);
t = ts:ts:to;
ut = t>=0; %defining u(t) function
xt = (exp(-2*t)).*ut; %defining x1(t)
mydft(xt,to,ts); %calling function mydft to calculate dft of x1(t)
%plotting the graphs (magnitude plot and phase plot)
subplot(2,2,2);
plot(f,abs(fftshift(fft(xt))));
xlabel('Frequency (Hz) ->');
ylabel('Magnitude ->');
title('Mohit Akhouri-19ucc023','Magnitude Plot using inbuilt function
 fft(X)');
grid on;
subplot(2,2,4);
plot(f,angle(fft(xt)));
xlabel('Frequency (Hz) ->');
ylabel('Phase angle ->');
title('Mohit Akhouri-19ucc023','Phase Plot using inbuilt function
 fft(X)');
grid on;
```

Figure 2.5 Code for calculating DFT when T_0 = 8 sec and T_s = 1/32 sec



Published with MATLAB® R2020b

Figure 2.6 Graph of calculated DFT when T_0 = 8 sec and T_s = 1/32 sec

2.5 Conclusion

In this experiment , we learnt the concepts of Aperiodic Signals, Fourier series representation of signals and Fourier transform concept. We tried to implement the **mydft** function to calculate the DFT of a signal ’x’. We implemented many coding concepts of MATLAB like **for loops**, **subplot and plot** and **linspace**. At last we compared the DFT calculated by our function mydft and with the inbuilt function **fft(X)** and understood the concept of Discrete fourier transform .