

**Independent University Bangladesh**

Department of Electrical and Electronics Engineering

**Lab Report** **08**

Name: Injamamul Haque Sourov

Id: 1820170

Course code: EEE 321L

Couse name: Digital Signal Processing Lab

Lab no: 07

Lab title: Study on DTFT, Circular Folding and Circular Convolution

Date: 6/1/2021

1. Discrete Fourier Transform function definition (dft.m):

% function to perform discrete Fourier transform

function [Xk] = dft(xn, N)

n = 0:N-1;

k = 0:N-1;

WN = exp(-j\*2\*pi/N);

nk = n'\*k;

WNnk = WN.^nk;

Xk = xn\*WNnk;

end

1. DTFT and 4 point DFT of x(n) = [1 1 1 1]

Code:

% signal definition

n1 = 0;

n2 = 3;

n = n1:n2;

x = [1 1 1 1];

% parameters for DTFT

M = 200; % number of frequency points (0,π)

k = 0:M;

w = (4\*pi/M)\*k;

% parameters for 4 point DFT

N = 4;

m = 0:N-1;

xn = [1 1 1 1];

% compute DTFT

X = x\*exp(-j\*n'\*w);

magX = abs(X);

phX = angle(X);

% compute DFT

Xk = dft(xn, N);

magXk = abs(Xk);

phaXk = angle(Xk)\*180/pi;

% Plotting

figure(1); % for DTFT

subplot(2,1,1);

plot(w/pi,magX)

title('Magnitude of X(w)', 'fontsize', 15);

xlabel('w', 'fontsize', 15);

ylabel('X(w)', 'fontsize', 15);

grid;

subplot(2,1,2);

plot(w/pi,PhaX);

title('Phase of X(w)', 'fontsize', 15);

xlabel('w', 'fontsize', 15);

grid;

figure(2); % for DFT

subplot(2,1,1);

stem(m,magXk)

title('Magnitude of X(k)', 'fontsize', 15);

xlabel('k', 'fontsize', 15);

ylabel('X(k)', 'fontsize', 15);

grid;

subplot(2,1,2);

stem(m,phaXk)

title('Angle of X(k)', 'fontsize', 15);

xlabel('k', 'fontsize', 15);

grid;

Outputs:





1. Function definition of circular folding (circfold.m):

% function to compute N point circular folding

function x2 = circfold(x1, N)

n = 0:N-1;

x2 = x1(mod(-n, N)+1);

end

1. Circular folding of x(n) = {1,3,5,7,9,-7,-5,-3,-1}

Code:

% circular folding

x1 = [1 3 5 7 9 -7 -5 -3 -1];

N = length(x1);

x2 = circfold(x1, N);

% ploting

subplot(2,1,1);

stem(0:N-1, x1)

title('x\_1(n)', 'fontsize', 15);

xlabel('n', 'fontsize', 15);

ylabel('x(n)', 'fontsize', 15);

grid;

subplot(2,1,2);

stem(0:N-1, x2)

title('x\_2(n)', 'fontsize', 15);

xlabel('n', 'fontsize', 15);

ylabel('x((-n))\_N', 'fontsize', 15);

grid;

Outputs:



1. 11 point circular folding of x(n) = 10(0.9) n , 0 <= n <= 10

Code:

n = 0:10;

x = 10\*(0.9).^n;

% circular folding

y = circfold(x, 11);

% DFT

Y = dft(y,11);

% Plotting

subplot(2,1,1);

stem(n, real(Y))

title('Real{DFT[(x((-n))\_1\_1)]}', 'fontsize', 15);

xlabel('k', 'fontsize', 15);

grid;

subplot(2,1,2);

stem(n, imag(Y))

title('Imaginary{DFT[(x((-n))\_1\_1)]}', 'fontsize', 15);

xlabel('k', 'fontsize', 15);

grid;

Outputs:



1. Assignment: Circular convolution
   1. Function definition:

% function to compute N-point circular convolution of two signals

function [] = circ\_conv( x1,x2,N )

% padd signals with 0 to make N points each

x1pad = [x1 zeros(1,N-length(x1))];

x2pad = [x2 zeros(1,N-length(x2))];

% conv in t domain = multiplication in f domain

cconv = ifft(fft(x1pad).\*fft(x2pad));

y = cconv;

% plot

stem(y)

title('Circular Convolution')

xlabel('n')

grid on

end

Call and output:

% arbitrary signals

x1 = [1 2 3 4];

x2 = [4 3 2 1];

% 5 point circular conv

circ\_conv(x1,x2,5)

