School of Engineering and Applied Science (SEAS)
Ahmedabad University

BTech(ICT) Digital Signal Processing (Section 1)

Laboratory Assignment-4

Enrollment No: AU1841145 **Name: Samarth Shah**

AIM : Understand different concepts of Fourier transform along with its applications.

1. Solution Problem-1

   (a) Approach: 1) In this question, sampling freqency and period were calculated based on the equation which is how in the handwritten analysis. When we consider Nyquist rate as twice of the sampling frequency then, the rate will be directly propositioned to the sampling frequency. With this increase in the number of samples, the resolution of the discrete time signal becomes higher and closer to the continuous signal.
   2) In the programming part, after calculating sampling freqency and period, along with amplitude and frequncey were taken as the input. Then the time cycle was calculated and using fft inbuilt as well as user defined function graph of DFT were plotted. In the user-defined function, inout was taken as a sinusoidal signal and then initialized with zeros. 2 for loops were considered to applying discrete summation.

   (b) Matlab Script:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab4 (Question_1) Write a program to compute the DFT of given sinusoidal wave by
     specifying amplitude, frequency, sampling frequency and number of periods to
    obtain DFT from user. Plot magnitude spectrum with horizontal axis indicating
    the analog frequency.
close all;
clc;
clear all;

fs = input('Enter fs');  % sampling frequency
Amplitude = input('Enter A:');  %amplitude
freq = input('Enter F'); %freqeuncy
NO_Period = input('Enter the no. of periods:'); %number of periods of cycles
Time = 0:(1/fs):(NO_Period - 1)/fs; %Time cycle (0:1/fs:T/fs)

sinu_signal = Amplitude*cos(2*pi*freq*Time);
% Sinusoidal Signal

%Calling fft() inbuilt function
DFT = fft(sinu_signal);

%Range
range_x_axis = [0:(NO_Period-1)];
%Plot
subplot(211);
stem(range_x_axis, DFT);
xlabel('No. of Period');
ylabel('Value of FFT');
title('DFT using inbuilt function');

%Calling user defined function
```

```
DFT = DFT_Function(sinu_signal);
%Plot
subplot(212);
stem(range_x_axis, DFT);
xlabel('No. of Period');
ylabel('Value of FFT');
title('DFT using User defined function');
```

(c) User-defined function:

```
%User defined function to calculate DFT
function output = DFT_Function(sin_cos_signal)
    %Length
    len = length(sin_cos_signal);
    %Initialize with zeros
    inti_zero = zeros(len);
    %for loop starting from 0 to (Length-1)
    for i = 0:(len-1)
        %for loop starting from 0 to (Length-1)
        for j = 0:(len-1)
            inti_zero(i+1, j+1) = exp((-1j*2*pi*i*j)/len); %Applying DTF equation
        end
    end
    output = sin_cos_signal*inti_zero;  %Multiply to get the final DFT output
end
```

(d) Hand-written Analysis:

Handwritten Analysis

Page No.
Date

⇒ $x(t) = 4\cos(100\pi t)$

from, the above equation,

the period of the signal is $T_0 = \frac{1}{50}$ s

$f_{max} = 50$ Hz

$\therefore f_{Nyq} = 2 \times f_{max} = 2 \times 50 = 100$ Hz

$\therefore$ Sampling frequency $f_s = f_{Nyq} = 100$ Hz

Number of samples within one period is:

$N_0 = T_0 f_s = \frac{1}{50} \times 100 = 5$

$\therefore$ Sampling $x(t)$ at $t = \frac{n}{N_0} T_0 = \frac{n}{250}$ — (1)

⇒ From Euler's identity $\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2}$

Suppose, $\theta = \frac{2\pi K_0 n}{N}$

$\therefore$ from the question,

$\frac{2\pi K_0 n}{N} = 100\pi t = 100\pi \times \frac{n}{250}$ (from the main eq.)

$$\boxed{\therefore \frac{K_0}{N} = \frac{1}{50}}$$

*Samarth Shah AU1841145

Page No.
Date

Basic formula of DFT is,

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

$$\therefore x(k) = \sum_{n=0}^{4} 4\left(\frac{e^{\frac{\pi N}{25}} + e^{\frac{-\pi N}{25}}}{2}\right) \times e^{-j\frac{2\pi kn}{4}}$$

After calculating above equation,

we will get,

$x(0) = 4 + 0i$
$x(1) = 8.4721 + 6.1554i$
$x(2) = -0.4721 - 1.4531i$
$x(3) = -0.4721 + 1.4531i$
$x(4) = 8.4721 - 6.1554i$

$\therefore$ Sampling frequency $= 200$ Hz
Amplitude $= 4$
Frequency $= 50$ Hz
Number of periods $= 5$

* Samarth Shah AU1841145

(e) Simulation Output:

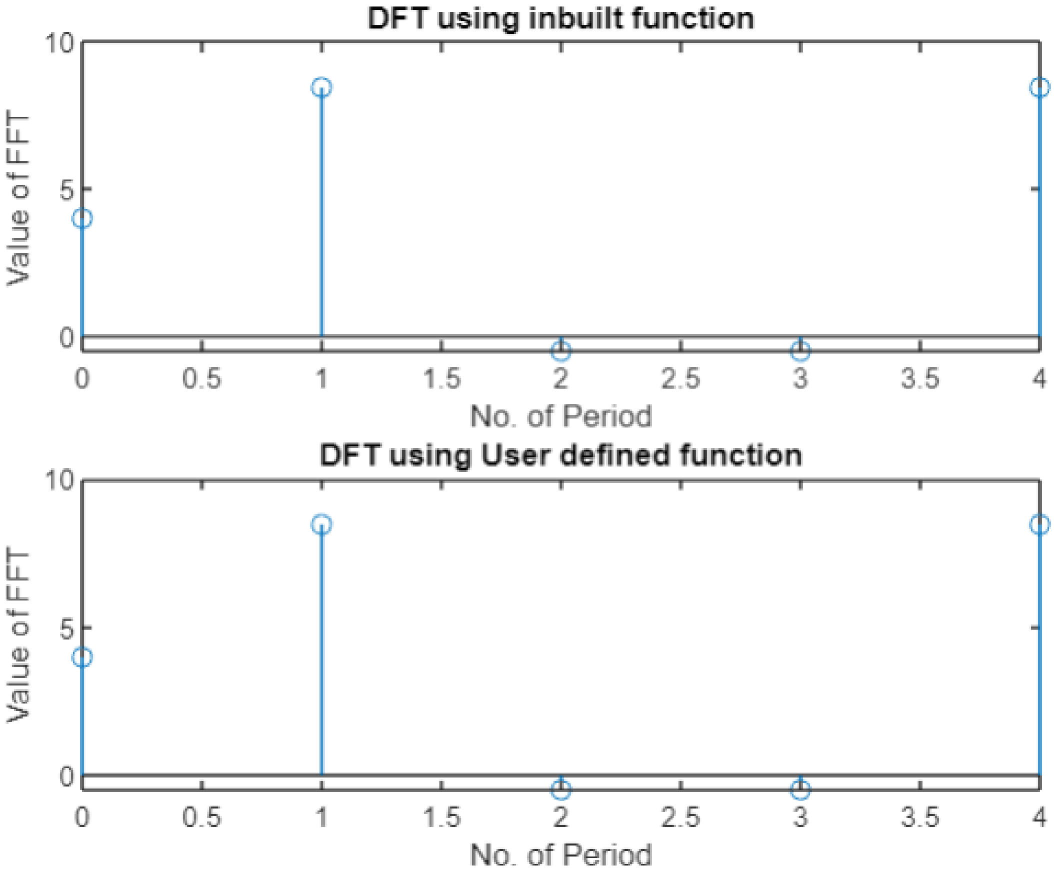

Figure: Output of DFT using in-built and User defined function

2. Solution Problem-2

   (a) Approach: In this question, DFT signal is taken same as the mentioned above. The only thing changed is that for IDFT, 2 times for loop are considered and sum it to n and k times respectively. Which will finally give the IDFT of the fourier transformed signal which would give the original signal as a result.

   (b) Matlab Script for sub-question 1:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab4 (Question_2A) MATLAB to compute DFT of following sequences and verify the
    answer by finding IFFT command.
close all;
clc;
clear all;
%Input a signal
sig = input('Enter the signal:');
len = length(sig); %Calculate length of signal
range_x_axis = 0:(len-1); %Range of the signal
%Calling user defined function
DFT = DFT_Function(sig);
%Calling ifft() inbuilt function
IDFT = ifft(DFT);
%Plot
figure;
subplot(311);
%Main signal
stem(range_x_axis, sig); %Discrete Plot
xlabel('Range'); %X-axis
ylabel('Value of Signal'); %Y-Axis
title('Input Signal');

subplot(312);
%Real part of IDFT
stem(range_x_axis, real(IDFT)); %Discrete Plot
xlabel('Range'); %X-axis
ylabel('Value of IDFT'); %Y-Axis
title('Inverse Discrete Fourier Transform');

%Plot
subplot(313);
%Only the real part of DFT
stem(range_x_axis, real(DFT)); %Discrete Plot
xlabel('Range');
ylabel('Value of DFT');
title('Discrete Fourier Transform');

%magnitude spectrum of DFT
figure;
subplot(211);
stem(range_x_axis, abs(DFT));
xlabel('Range');
ylabel('Magnitude');
title('Magnitude Spectrum of DFT');

%Phase spectrum of DFT
subplot(212);
stem(range_x_axis, angle(DFT));
xlabel('Range');
ylabel('Phase');
title('Phase Spectrum of DFT');
```

   (c) Matlab Script for sub-question 2 Length=4:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab4 (Question_2B) MATLAB to compute DFT of following sequences and verify the
    answer by finding IFFT command.
```

```
close all;
clc;
clear all;
Len = 4; %Length
range_x_axis = 0:(Len-1); %Range of signal

unit_ramp_1 = [(range_x_axis>=0)]; %Unit ramp signal u(n)
unit_ramp_2 = [(range_x_axis>=3)]; %Unit ramp signal u(n - 3)
final_signal = unit_ramp_1 - unit_ramp_2; %Final Signal

%Calling user defined function
DFT = DFT_Function(final_signal);
%Calling ifft() inbuilt function
IDFT = ifft(DFT);

%Plot
figure;

subplot(311);
%Final Signal
stem(range_x_axis, final_signal);%Discrete Plot
xlabel('Range');%X-axis
ylabel('Value of Signal');%Y-Axis
title('Input Signal');

subplot(312);
%Real part of IDFT
stem(range_x_axis, real(IDFT));%Discrete Plot
xlabel('Range');%X-axis
ylabel('Value of IDFT');%Y-Axis
title('Inverse Discrete Fourier Transform');

%Plot
subplot(313);
%Real part of DFT
stem(range_x_axis, real(DFT));%Discrete Plot
xlabel('Range');%X-axis
ylabel('Value of DFT');%Y-Axis
title('Discrete Fourier Transform');

%Plot Magnitude Spectrum
figure;
subplot(211);
stem(range_x_axis, abs(DFT));%Discrete Plot
xlabel('Range');%X-axis
ylabel('Magnitude');%Y-Axis
title('Magnitude Spectrum of DFT');

%Plot Phase Spectrum
subplot(212);
stem(range_x_axis, angle(DFT));%Discrete Plot
xlabel('Range');%X-axis
ylabel('Phase');%Y-Axis
title('Phase Spectrum of DFT');
```

(d) Matlab Script for sub-question 2 Length=8:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab4 (Question_2B) MATLAB to compute DFT of following sequences and verify the
   answer by finding IFFT command.

close all;
clc;
clear all;
Len = 8; %Length
range_x_axis = 0:(Len-1); %Range of signal

unit_ramp_1 = [(range_x_axis>=0)]; %Unit ramp signal u(n)
```

```
unit_ramp_2 = [(range_x_axis>=3)]; %Unit ramp signal u(n - 3)
final_signal = unit_ramp_1 - unit_ramp_2; %Final Signal

%Calling user defined function
DFT = DFT_Function(final_signal);
%Calling ifft() inbuilt function
IDFT = ifft(DFT);

%Plot
figure;

subplot(311);
%Final Signal
stem(range_x_axis, final_signal);%Discrete Plot
xlabel('Range');%X-axis
ylabel('Value of Signal');%Y-Axis
title('Input Signal');

subplot(312);
%Real part of IDFT
stem(range_x_axis, real(IDFT));%Discrete Plot
xlabel('Range');%X-axis
ylabel('Value of IDFT');%Y-Axis
title('Inverse Discrete Fourier Transform');

%Plot
subplot(313);
%Real part of DFT
stem(range_x_axis, real(DFT));%Discrete Plot
xlabel('Range');%X-axis
ylabel('Value of DFT');%Y-Axis
title('Discrete Fourier Transform');

%Plot Magnitude Spectrum
figure;
subplot(211);
stem(range_x_axis, abs(DFT));%Discrete Plot
xlabel('Range');%X-axis
ylabel('Magnitude');%Y-Axis
title('Magnitude Spectrum of DFT');

%Plot Phase Spectrum
subplot(212);
stem(range_x_axis, angle(DFT));%Discrete Plot
xlabel('Range');%X-axis
ylabel('Phase');%Y-Axis
title('Phase Spectrum of DFT');
```

(e) User-defined function:

```
%User defined function to calculate DFT
function output = DFT_Function(sin_cos_signal)
    %Length
    len = length(sin_cos_signal);
    %Initialize with zeros
    inti_zero = zeros(len);
    %for loop starting from 0 to (Length-1)
    for i = 0:(len-1)
        %for loop starting from 0 to (Length-1)
        for j = 0:(len-1)
            inti_zero(i+1, j+1) = exp((-1j*2*pi*i*j)/len); %Applying DTF equation
        end
    end
    output = sin_cos_signal*inti_zero;  %Multiply to get the final DFT output
end
```

(f) Simulation Output Sub Question-1:

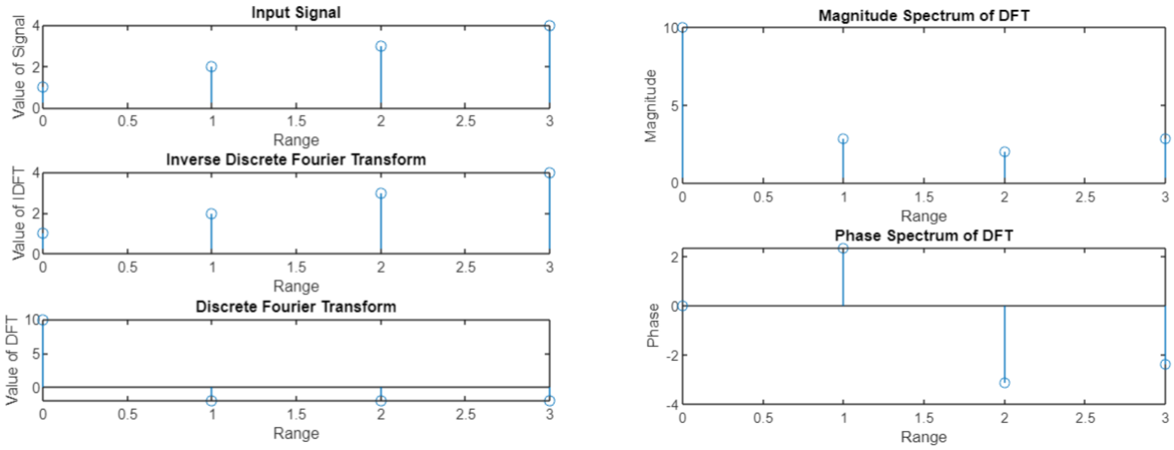

Figure 1: Sub-Question 1

(g) Simulation Output Sub-Sub Question-1:

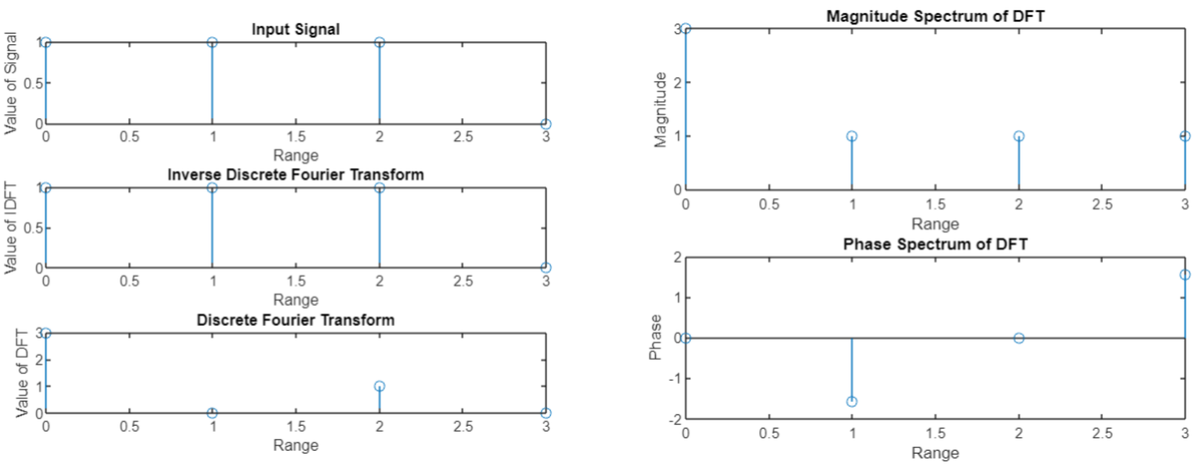

Figure 2: Sub-Sub Question-1

(h) Simulation Output Sub-Sub Question-2:

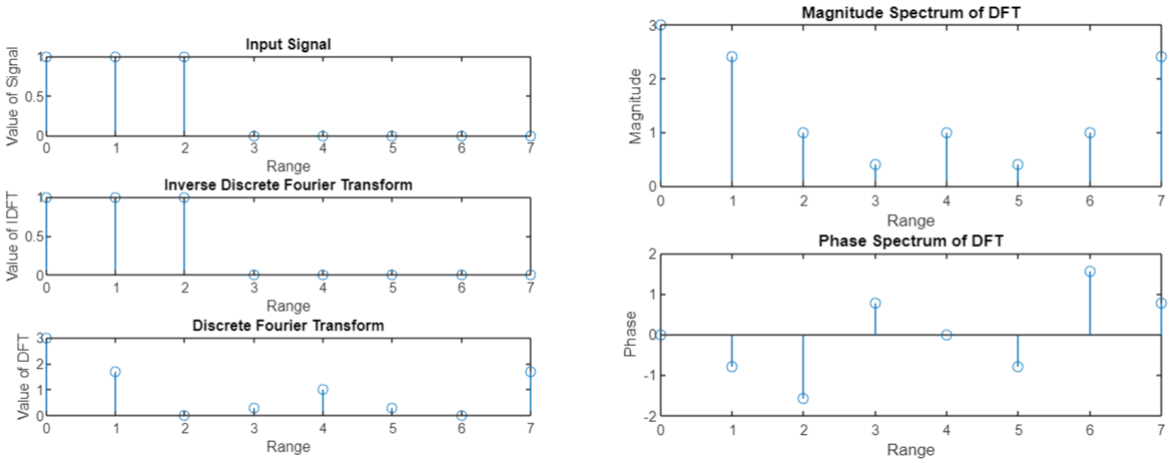

Figure 3: Sub-Sub Question-2

3. Solution Problem-3

   (a) Approach: In this question, a) Circular convolution is calculated using DFT of both signals separately. This will give us a frequesncy domain representation. The both signals were multiplied into the time domain using idft which will be the final circular convolution of the original signals. b) Linear convolution is calculating using DFT of both signal same as the stated above. Just signal lengths are to be changed as length of first seq + length of second seq -1.

   (b) Matlab Script For 1st Sub-Question:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab4 (Question_3A) Write a MATLAB program to find circular convolution of two
    sequences
close all;
clc;
clear all;

sig1 = [1,-1,-2,3,-1]; %Signal1
sig2 = [1,2,3]; %Signal2
len1=length(sig1); %Length of signal1
len2=length(sig2); %Length of signal2
Max_Len = max(len1,len2); %Maximum Length of len1 and len2

%Appending zeros
sig1 = [sig1,zeros(1, (Max_Len-len1))]; %to make both the signals of same length
sig2 = [sig2,zeros(1, (Max_Len-len2))]; %to make both the signals of same length

%Calling user defined function of DFT
DFT_signal1 = DFT_Function(sig1);
DFT_ssignal2 = DFT_Function(sig2);

%Product of values of DFT of both the signals
multipliy_signals = DFT_signal1 .* DFT_ssignal2;

%To calculate the circular convolution, IDFT of Dot Product
Circular_Convolution = IDFT_Function(multipliy_signals);
range_x_axis = 0:(Max_Len - 1);

%Calculating Circular convolution using cconv() inbuilt function
```

```
Circular_inbuilt = cconv(sig1, sig2, Max_Len);

%Plot
figure;
subplot(211);
stem(range_x_axis, Circular_Convolution);%Discrete Plot
title('Circular convolution');
xlabel('No. of periods');%X-axis
ylabel('Value');%Y-Axis

%Plot
subplot(212);
stem(range_x_axis, Circular_inbuilt);%Discrete Plot
title('Circular convolution using inbuilt function');
xlabel('No. of periods');%X-axis
ylabel('Value');%Y-Axis
```

(c) Matlab Script For 2nd Sub-Question:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab4 (Question_3B) Write a program to perform linear convolution using above DFT
    -IDFT approach based circular convolution
close all;
clc;
clear all;

sig1 = [1,-1,-2,3,-1]; %Signal1
sig2 = [1,2,3]; %Signal2
len1=length(sig1); %Length of signal1
len2=length(sig2); %Length of signal2

%Appending zeros
sig1 = [sig1,zeros(1, len2)]; %to make both the signals of same length
sig2 = [sig2,zeros(1, len1)]; %to make both the signals of same length

%Calling user defined function of DFT
DFT_signal1 = DFT_Function(sig1);
DFT_ssignal2 = DFT_Function(sig2);

%Product of values of DFT of both the signals
multipliy_signals = DFT_signal1 .* DFT_ssignal2;

%To calculate the Linear convolution, IDFT of Dot Product
Linear_Convolution = IDFT_Function(multipliy_signals);
range_x_axis = 0:(length(sig1) - 1);

%Calculating Linear convolution using cconv() inbuilt function
Linear_inbuilt = conv(sig1, sig2);

%Plot
figure;
subplot(211);
stem(range_x_axis, Linear_Convolution);%Discrete Plot
title('Linear convolution');
xlabel('No. of periods');%X-axis
ylabel('Value');%Y-Axis

%Plot
subplot(212);
stem(range_x_axis, Linear_inbuilt(1:length(sig1)));%Discrete Plot
title('Linear convolution using inbuilt function');
xlabel('No. of periods');%X-axis
ylabel('Value');%Y-Axis
```

(d) User-defined function:

```
%User defined function to calculate DFT
function output = DFT_Function(sin_cos_signal)
    %Length
```

```
    len = length(sin_cos_signal);
    %Initialize with zeros
    inti_zero = zeros(len);
    %for loop starting from 0 to (Length-1)
    for i = 0:(len-1)
        %for loop starting from 0 to (Length-1)
        for j = 0:(len-1)
            inti_zero(i+1, j+1) = exp((-1j*2*pi*i*j)/len); %Applying DTF equation
        end
    end
    output = sin_cos_signal*inti_zero;  %Multiply to get the final DFT output
end

function output = IDFT_Function(signal)

    len = length(signal);
    init_zeros = zeros(len);
    for i = 0:(len-1)
        for j = 0:(len-1)
            init_zeros(i+1, j+1) = exp((1j*2*pi*i*j)/len);
        end
    end
    output = 1/len.*(signal*init_zeros);
end
```

(e) Simulation output of Circular Convolution:

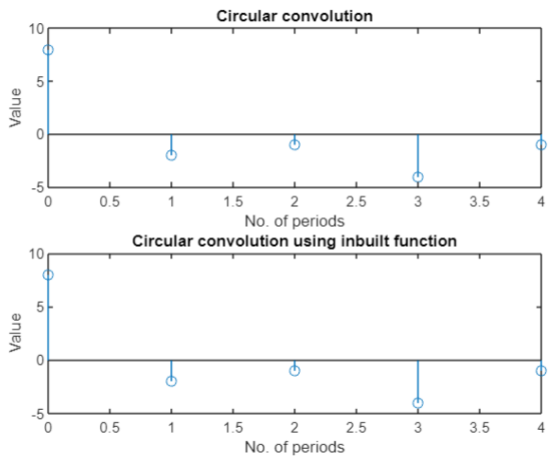

Figure:Sequences 1

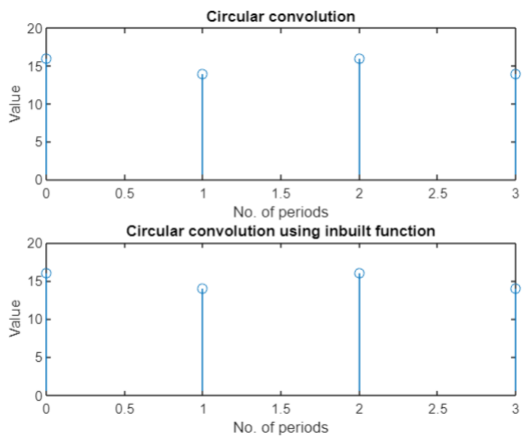

Figure:Sequences 2

(f) Simulation Output of Linear Convolution:

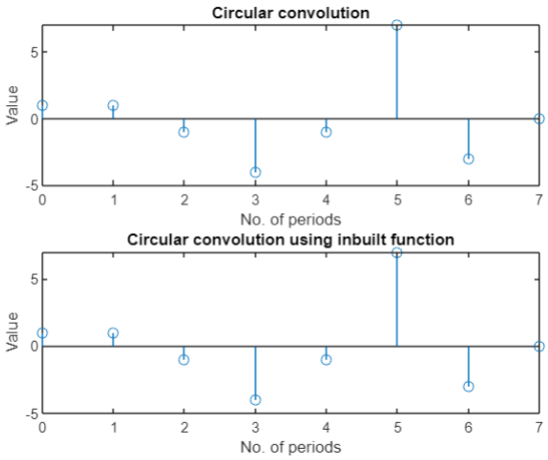

Figure:Sequences 1

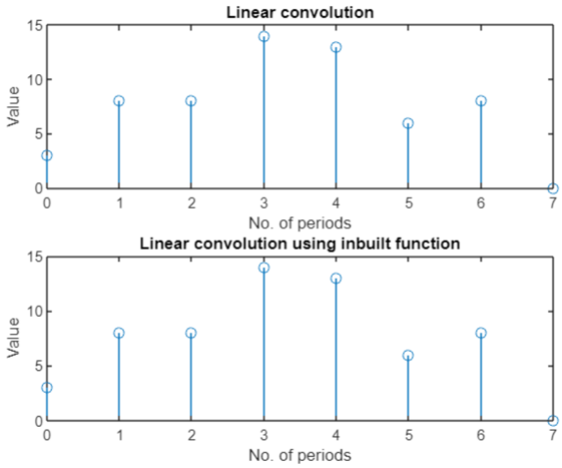

Figure:Sequences 2