Name: Do Trung Hau

Section:

Laboratory Exercise 3

DISCRETE-TIME SIGNALS: FREQUENCY-DOMAIN REPRESENTATIONS

3.1 DISCRETE-TIME FOURIER TRANSFORM

PROJECT 3.1 DTFT COMPUTATION

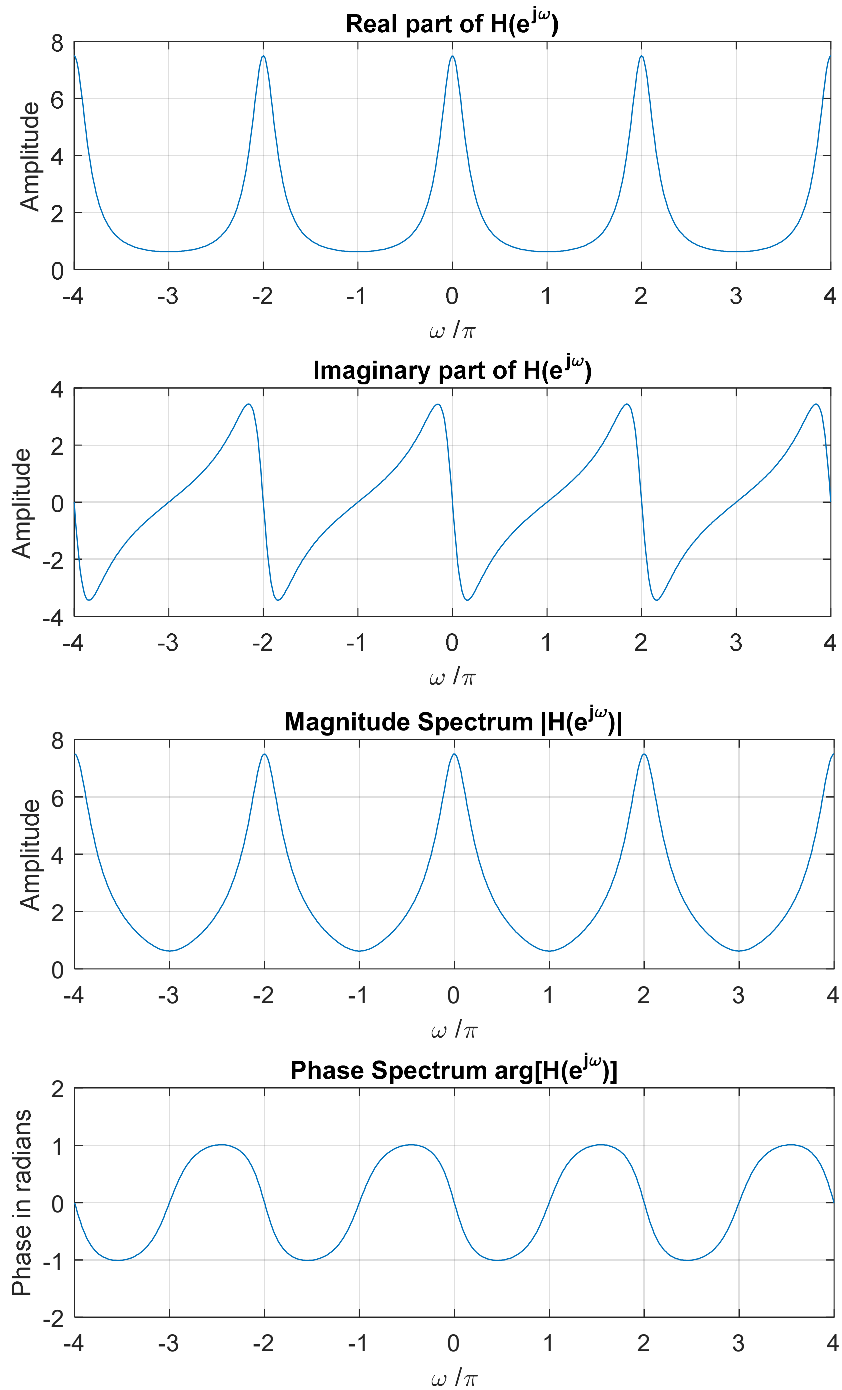
A copy of Program P3_1 is given below:

```
% Program P3_1
% Evaluation of the DTFT
clc; clear all; close all;
% Compute the frequency samples of the DTFT
w = -4*pi:8*pi/511:4*pi;
num = [2 1];den = [1 -0.6];
h = freqz(num, den, w);
% Plot the DTFT
subplot(2,1,1)
plot(w/pi,real(h));grid
title('Real part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,imag(h));grid
title('Imaginary part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
pause
subplot(2,1,1)
plot(w/pi,abs(h));grid
title('Magnitude Spectrum |H(e^{j\omega})|')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,angle(h));grid
title('Phase Spectrum arg[H(e^{j\omega})]')
xlabel('\omega /\pi');
ylabel('Phase in radians');
```

Q3.1 The expression of the DTFT being evaluated in Program P3_1 is ***a frequency response of a system***: $H\left(\omega\right)=\dfrac{2+e^{-j\omega}}{1-0.6e^{-j\omega}}$

+ The function of the `pause` command is ***used to stop MATLAB execution temporarily***

Q3.2 The plots generated by running Program P3_1 are shown below:



+ The DTFT is a ***periodic*** function of 2π.

+ Its period is 2π.

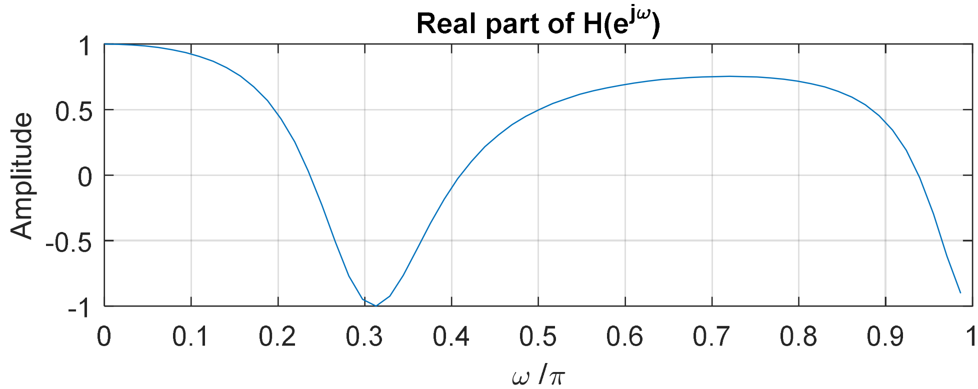
The types of symmetries exhibited by the four plots are as follows:

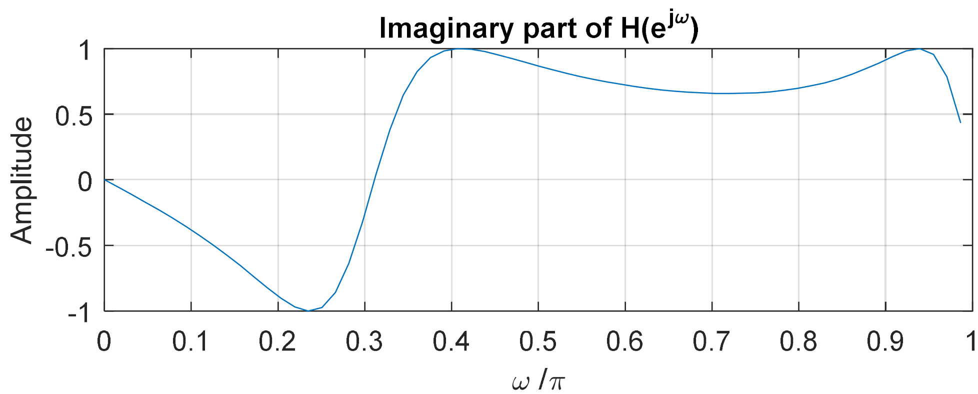
+ ***The real part and the magnitude are even symmetrics with a period of*** 2π**.**

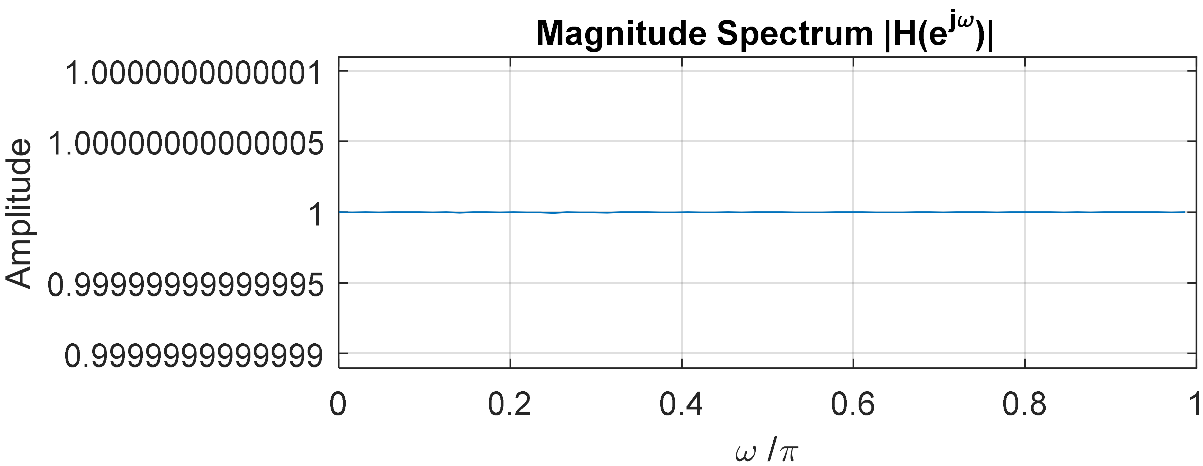
+ ***The imaginary part and the phase are odd symmetrics with a period of*** 2π**.**

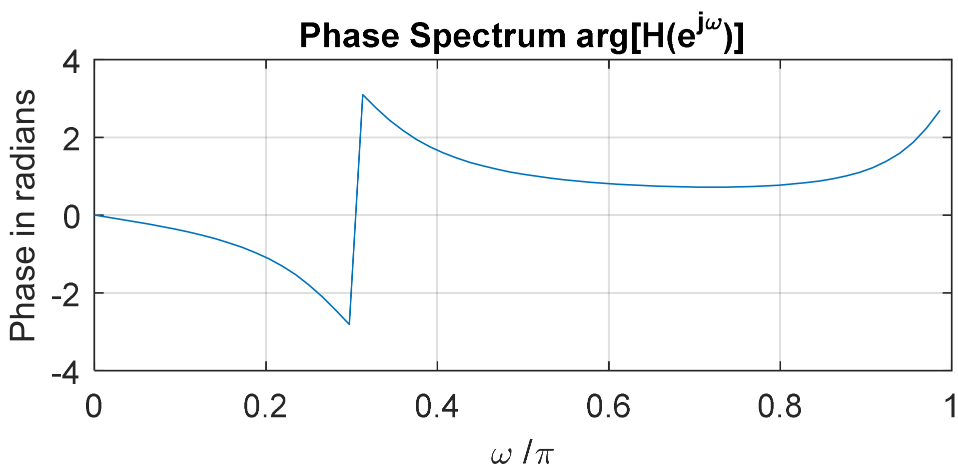
Q3.3 The required modifications to Program P3_1 to evaluate the given DTFT of Q3.3 are given below:

```
% Program P3_1
% Evaluation of the DTFT
clf; close all; clear all;
% Compute the frequency samples of the DTFT
w = 0*pi:8*pi/511:1*pi;
num = [0.7 -0.5 0.3 1];
den = [1 0.3 -0.5 0.7];
h = freqz(num, den, w);
% Plot the DTFT
subplot(2,1,1)
plot(w/pi,real(h));grid
title('Real part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,imag(h));grid
title('Imaginary part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
pause
subplot(2,1,1)
plot(w/pi,abs(h));grid
title('Magnitude Spectrum |H(e^{j\omega})|')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,angle(h));grid
title('Phase Spectrum arg[H(e^{j\omega})]')
xlabel('\omega /\pi');
ylabel('Phase in radians');
```


Real part of H(e^jω)
Amplitude
1
0.5
0
-0.5
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ω /π


Imaginary part of H(e^jω)
Amplitude
1
0.5
0
-0.5
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ω /π


Magnitude Spectrum |H(e^jω)|
Amplitude
1.0000000000001
1.00000000000005
1
0.99999999999995
0.9999999999999
0
0.2
0.4
0.6
0.8
1
ω /π


Phase Spectrum arg[H(e^jω)]
Phase in radians
4
2
0
-2
-4
0
0.2
0.4
0.6
0.8
1
ω /π

The plots generated by running the modified Program P3_1 are shown below:

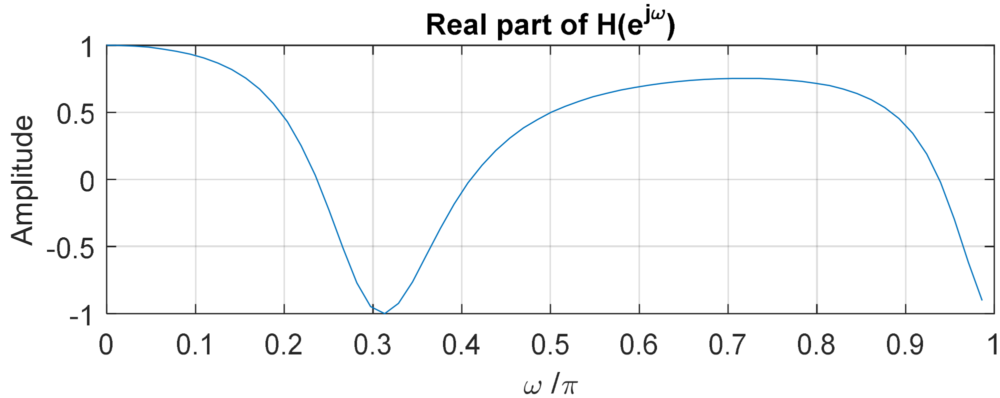
+ The DTFT is a ***periodic*** function of 2π .

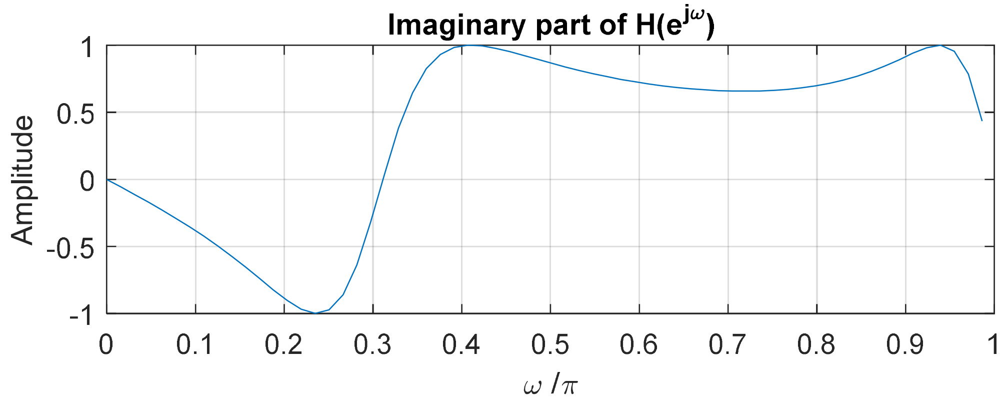
+ Its period is 2π .

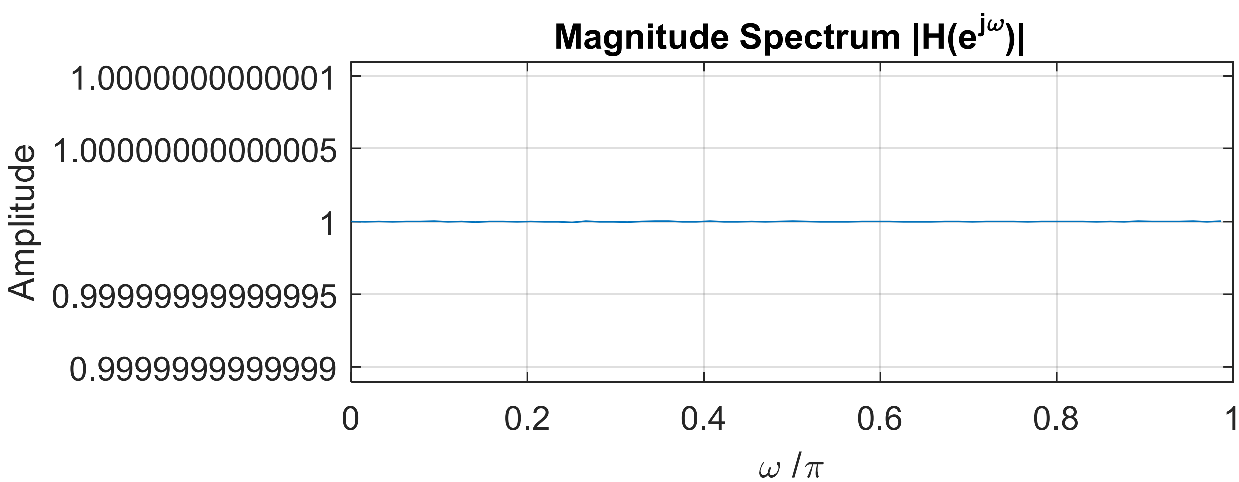
The jump in the phase spectrum is caused by ***calculating the phase response using the arctan function. The form of the phase reponse may be a rational form, then we can encounter the pole points when ploting this.***

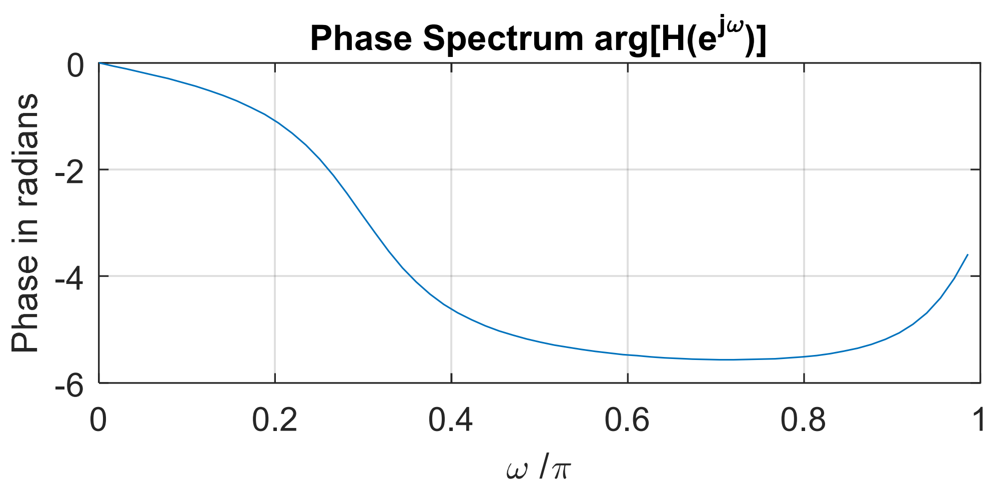
The phase spectrum evaluated with the jump removed by the command `unwrap` is as given below:

```
% Evaluation of the DTFT
clf; close all; clear all;
% Compute the frequency samples of the DTFT
w = 0*pi:8*pi/511:1*pi;
num = [0.7 -0.5 0.3 1];
den = [1 0.3 -0.5 0.7];
h = freqz(num, den, w);
% Plot the DTFT
subplot(2,1,1)
plot(w/pi,real(h));grid
title('Real part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,imag(h));grid
title('Imaginary part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
pause
subplot(2,1,1)
plot(w/pi,abs(h));grid
title('Magnitude Spectrum |H(e^{j\omega})|')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,unwrap(angle(h)));grid
title('Phase Spectrum arg[H(e^{j\omega})]')
xlabel('\omega /\pi');
ylabel('Phase in radians');
```


Real part of H(e^jω)
Amplitude
1
0.5
0
-0.5
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ω /π


Imaginary part of H(e^jω)
Amplitude
1
0.5
0
-0.5
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ω /π

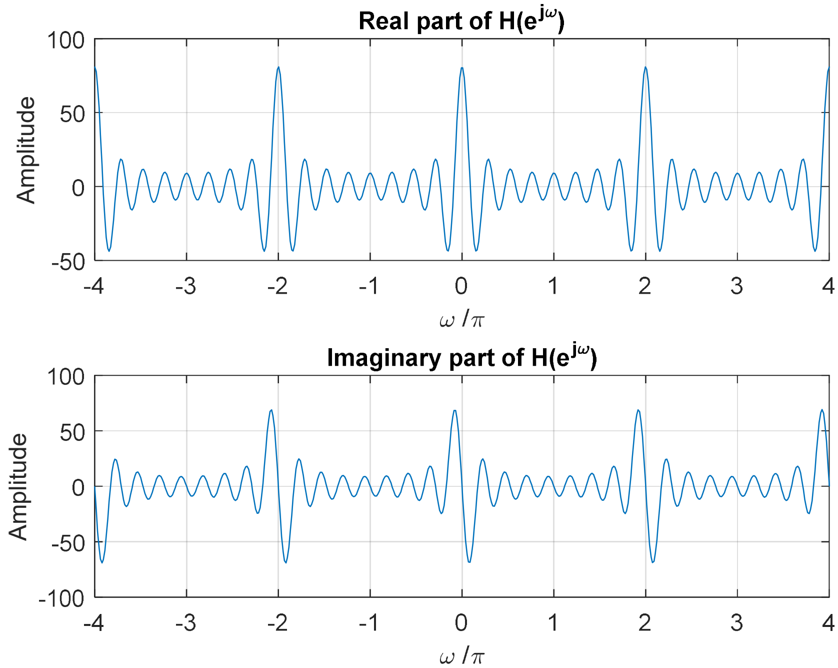

Magnitude Spectrum |H(e^jω)|
Amplitude
1.0000000000001
1.00000000000005
1
0.99999999999995
0.9999999999999
0
0.2
0.4
0.6
0.8
1
ω /π

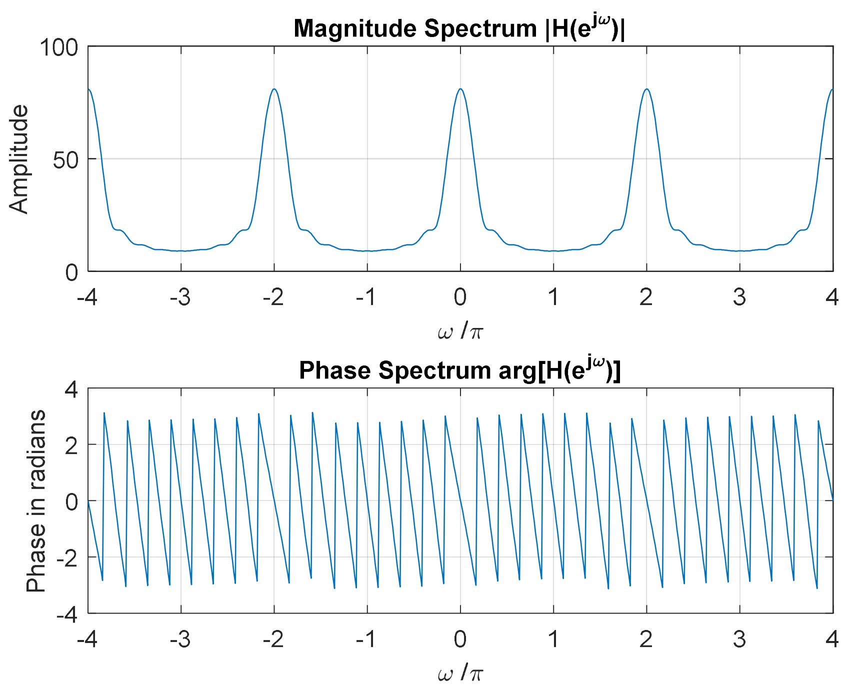

Phase Spectrum arg[H(e^jω)]
Phase in radians
0
-2
-4
-6
0
0.2
0.4
0.6
0.8
1
ω /π

Q3.4 The required modifications to Program P3_1 to evaluate the given DTFT of Q3.4 are given below:

```
% Evaluation of the DTFT
clf; close all; clear all;
% Compute the frequency samples of the DTFT
w = -4*pi:8*pi/511:4*pi;
num = [1 3 5 7 9 11 13 15 17];
den = 1;
h = freqz(num, den, w);
% Plot the DTFT
subplot(2,1,1)
plot(w/pi,real(h));grid
title('Real part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,imag(h));grid
title('Imaginary part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
pause
subplot(2,1,1)
plot(w/pi,abs(h));grid
title('Magnitude Spectrum |H(e^{j\omega})|')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,angle(h));grid
title('Phase Spectrum arg[H(e^{j\omega})]')
xlabel('\omega /\pi');
ylabel('Phase in radians');
```

The plots generated by running the modified Program P3_1 are shown below:





+ The DTFT is a ***periodic*** function of 2π .

+ Its period is 2π .

The jump in the phase spectrum ***can be explained the same as Q3.3***

Q3.5 The required modifications to Program P3_1 to plot the phase ***in degrees*** are indicated below:

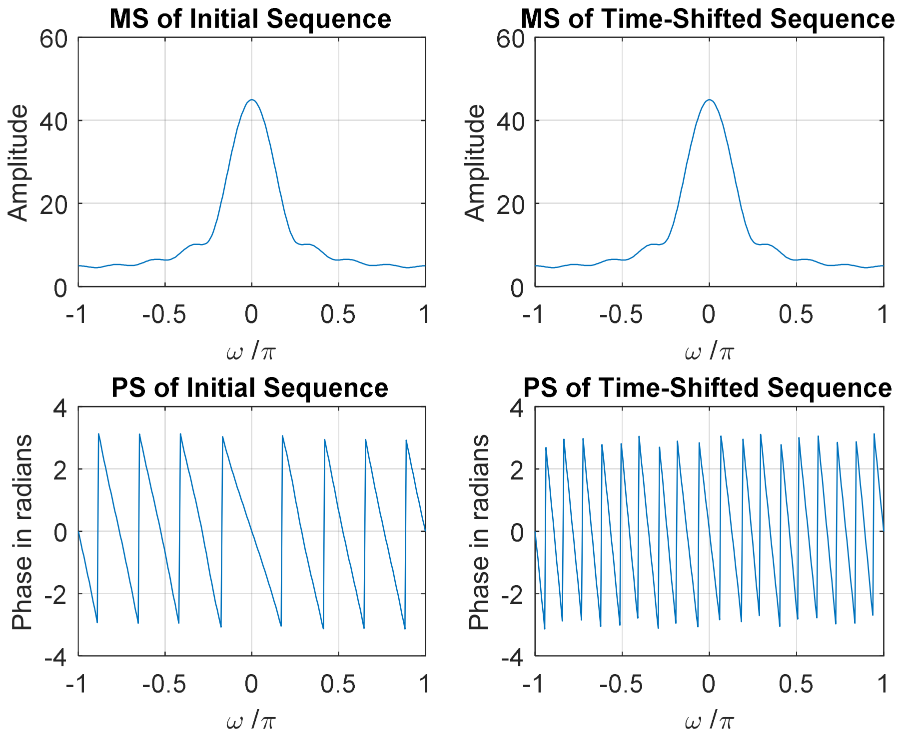
```
% Evaluation of the DTFT

clf; clear all; close all;
% Compute the frequency samples of the DTFT
w = -4*pi:8*pi/511:4*pi;
num = [2 1];den = [1 -0.6];
h = freqz(num, den, w);
% Plot the DTFT
subplot(2,1,1)
plot(w/pi,real(h));grid
title('Real part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,imag(h));grid
title('Imaginary part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
pause
subplot(2,1,1)
plot(w/pi,abs(h));grid
title('Magnitude Spectrum |H(e^{j\omega})|')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,180*angle(h)/pi);grid
title('Phase Spectrum arg[H(e^{j\omega})]')
xlabel('\omega /\pi');
ylabel('Phase in degrees');
```

PROJECT 3.2 DTFT PROPERTIES

Q3.6 The modified Program P3_2 created by adding appropriate comment statements, and adding program statements for labeling the two axes of each plot being generated by the program is given below:

```
%Time-Shifting Properties of DTFT
clf; clear all; close all;
w = -pi:2*pi/255:pi; % freqency vector for evaluating DTFT
D = 10; % Amount of time shift in samples
num = [1 2 3 4 5 6 7 8 9];
% h1 is the DTFT of original sequence
% h2 is the DTFT of the time shifted sequence
h1 = freqz(num, 1, w);
h2 = freqz([zeros(1,D) num], 1, w);
subplot(2,2,1)
% plot the DTFT magnitude of the original sequence
plot(w/pi,abs(h1));grid
title('MS of Initial Sequence')
xlabel('\omega /\pi');
ylabel('Amplitude');
% plot the DTFT magnitude of the shifted sequence
subplot(2,2,2)
plot(w/pi,abs(h2));grid
title('MS of Time-Shifted Sequence')
xlabel('\omega /\pi');
ylabel('Amplitude');
% plot the DTFT phase of the original sequence
subplot(2,2,3)
plot(w/pi,angle(h1));grid
title('PS of Initial Sequence')
xlabel('\omega /\pi');
ylabel('Phase in radians');
% plot the DTFT phase of the shifted sequence
subplot(2,2,4)
plot(w/pi,angle(h2));grid
title('PS of Time-Shifted Sequence')
xlabel('\omega /\pi');
ylabel('Phase in radians');
```

The parameter controlling the amount of time-shift is D ***parameter.***



Q3.7 The plots generated by running the modified program are given below:

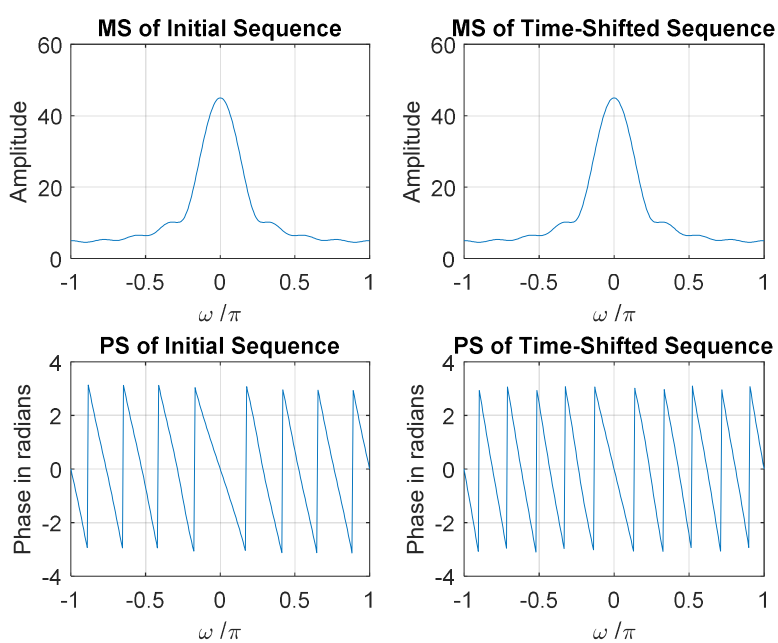
From these plots we make the following observations:

+ The magnitude spectrum remains the same before.

+ The phase spectrum has a clearly change, the slope is increased.

Q3.8 Program P3_2 was run for the following value of the time-shift $D = 2$

The plots generated by running the modified program are given below:

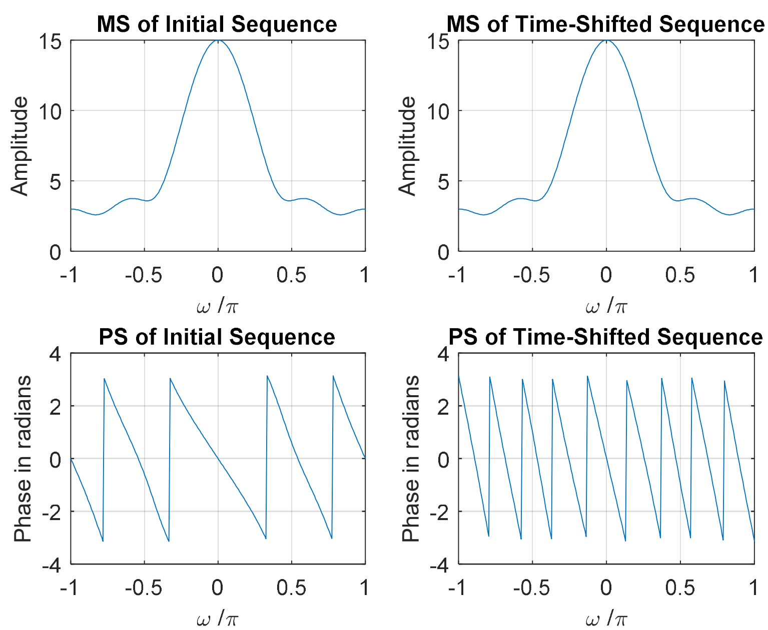


From these plots we make the following observations:

+ ***The magnitude spectrum still remains the same before.***

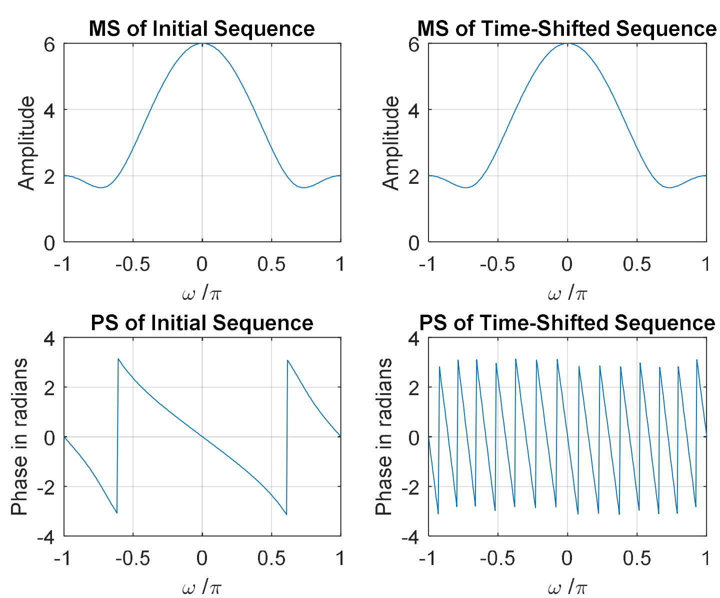
+ ***The phase spectrum has a clear change, the slope is increased but less than that of*** $D = 10$

Q3.9 Program P3_2 was run for the following values of the time-shift and for the following values of length for the sequence :



+ $L = 5\left(num = [1\ 2\ 3\ 4\ 5]\right),\ D = 5$

+ $L = 3,\ D = 12$



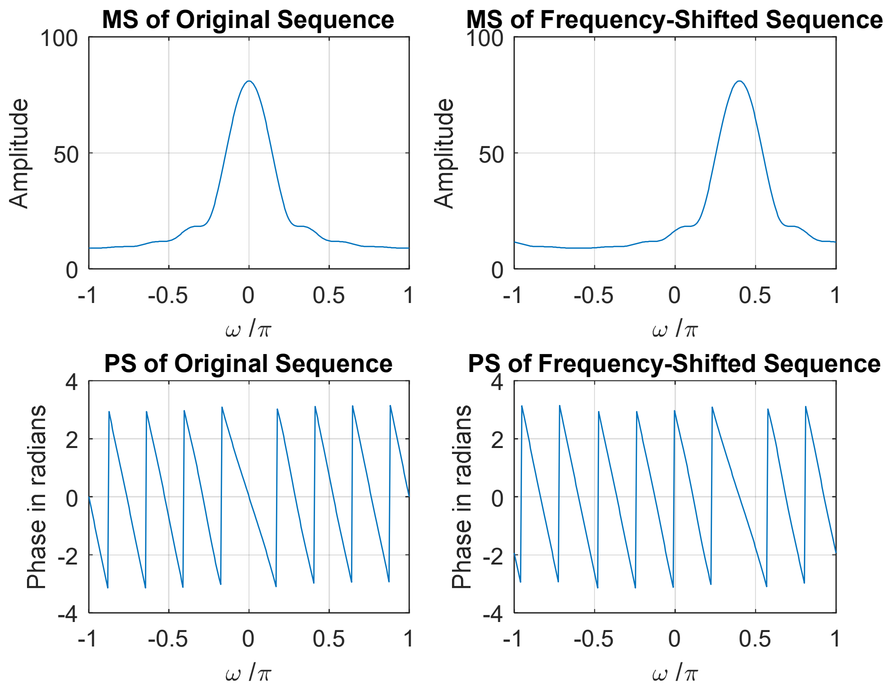
From these plots we make the following observations: ***These results are the same as previous questions.***

Q3.10 The modified Program P3_3 created by adding appropriate comment statements, and adding program statements for labeling the two axes of each plot being generated by the program is given below:

```
clf; clear all; close all;
w = -pi:2*pi/255:pi; % freqency vector for evaluating DTFT
wo = 0.4*pi; % Amount of frequency shift in radians
% h1 is the DTFT of the original sequence
% h2 is the DTFT of the frequency shifted sequence
num1 = [1 3 5 7 9 11 13 15 17];
L = length(num1);
h1 = freqz(num1, 1, w);
n = 0:L-1;
num2 = exp(wo*i*n).*num1;
h2 = freqz(num2, 1, w);
% plot the DTFT magnitude of the original sequence
subplot(2,2,1)
plot(w/pi,abs(h1));grid
title('MS of Original Sequence')
xlabel('\omega /\pi');
ylabel('Amplitude');
% plot the DTFT magnitude of the freq shifted sequence
subplot(2,2,2)
plot(w/pi,abs(h2));grid
title('MS of Frequency-Shifted Sequence')
xlabel('\omega /\pi');
ylabel('Amplitude');
% plot the DTFT phase of the original sequence
subplot(2,2,3)
plot(w/pi,angle(h1));grid
title('PS of Original Sequence')
xlabel('\omega /\pi');
ylabel('Phase in radians');
% plot the DTFT phase of the shifted sequence
subplot(2,2,4)
plot(w/pi,angle(h2));grid
title('PS of Frequency-Shifted Sequence')
xlabel('\omega /\pi');
ylabel('Phase in radians')
```

The parameter controlling the amount of frequency-shift is - w_0 ***parameter.***

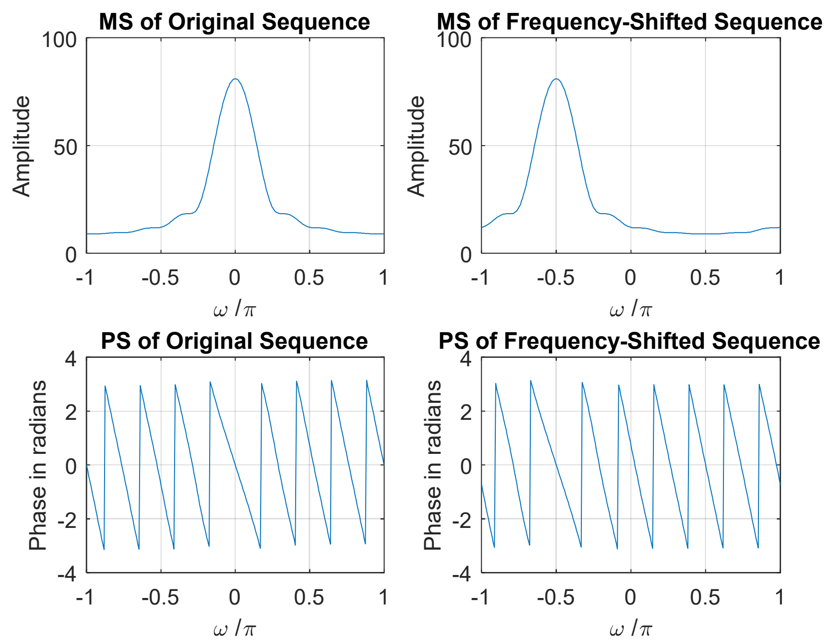
Q3.11 The plots generated by running the modified program are given below:



From these plots we make the following observations: ***Both the magnitude and phase spectra are shifted right by*** w_0**.**

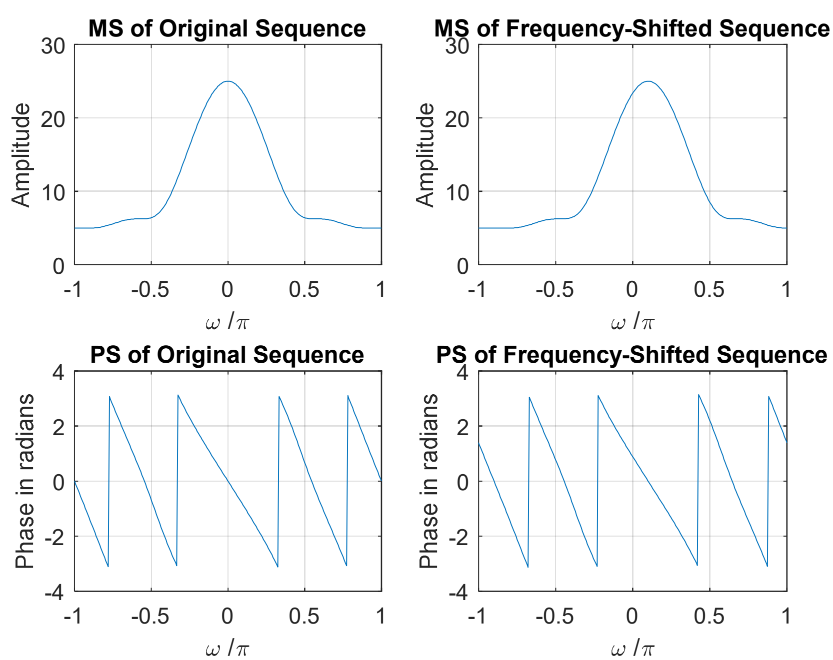
Q3.12 Program P3_3 was run for the following value of the frequency-shift - $w_0 = -0.5$

The plots generated by running the modified program are given below:



From these plots we make the following observations: ***Both the magnitude and phase spectra are shifted left by*** w_0.

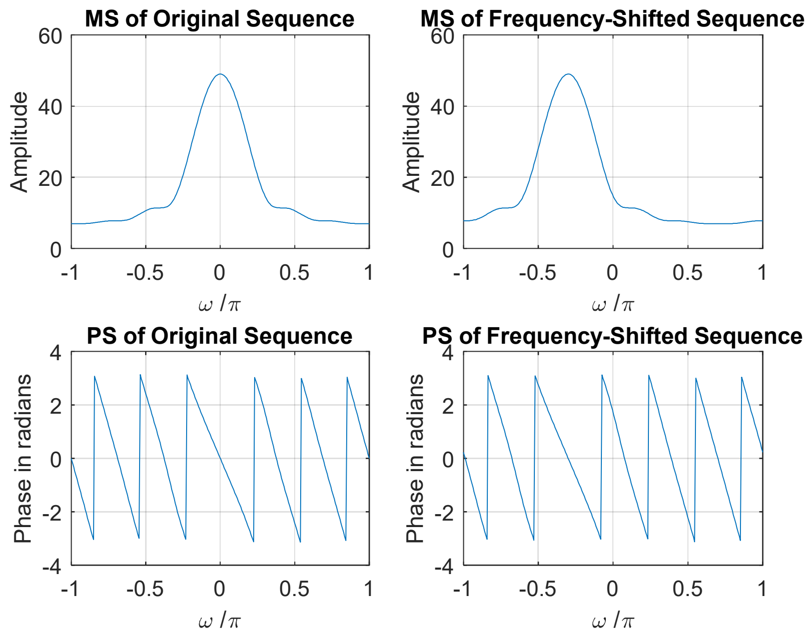
Q3.13 Program P3_3 was run for the following values of the frequency-shift and for the following values of length for the sequence :



+ $L = 5,\ w_0 = 0.1$

+ $L = 7$, $w_0 = -0.3$

The plots generated by running the modified program are given below:

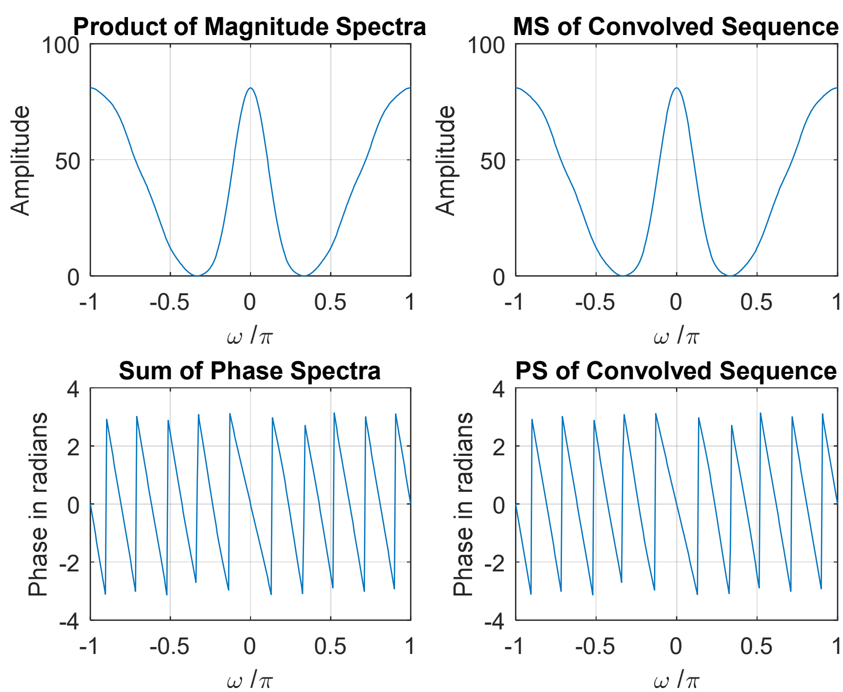


From these plots we make the following observations: ***Both the magnitude and phase spectra are shifted left/right by*** w_0.

Q3.14 The modified Program P3_4 created by adding appropriate comment statements, and adding program statements for labeling the two axes of each plot being generated by the program is given below:

```
clf; clear all; close all;
w = -pi:2*pi/255:pi; % frequency vector for evaluating DTFT
x1 = [1 3 5 7 9 11 13 15 17]; % first sequence
x2 = [1 -2 3 -2 1]; % second sequence
y = conv(x1,x2); % time domain convolution of x1 and x2
h1 = freqz(x1, 1, w); % DTFT of sequence x1
h2 = freqz(x2, 1, w); % DTFT of sequence x2
% hp is the pointwise product of the two DTFT's
hp = h1.*h2;
% h3 is the DTFT of the time domain convolution;
% it should be the same as hp
h3 = freqz(y,1,w);
% plot the magnitude of the product of the two original
spectra
subplot(2,2,1)
plot(w/pi,abs(hp));grid
title('Product of Magnitude Spectra')
xlabel('\omega /\pi');
```

```
ylabel('Amplitude');
% plot the magnitude spectrum of the time domain
convolution
subplot(2,2,2)
plot(w/pi,abs(h3));grid
title('MS of Convolved Sequence')
xlabel('\omega /\pi');
ylabel('Amplitude');
% plot the phase of the product of the two original spectra
subplot(2,2,3)
plot(w/pi,angle(hp));grid
title('Sum of Phase Spectra')
xlabel('\omega /\pi');
ylabel('Phase in radians');
% plot the phase spectrum of the time domain convolution
subplot(2,2,4)
plot(w/pi,angle(h3));grid
title('PS of Convolved Sequence')
xlabel('\omega /\pi');
ylabel('Phase in radians');
```



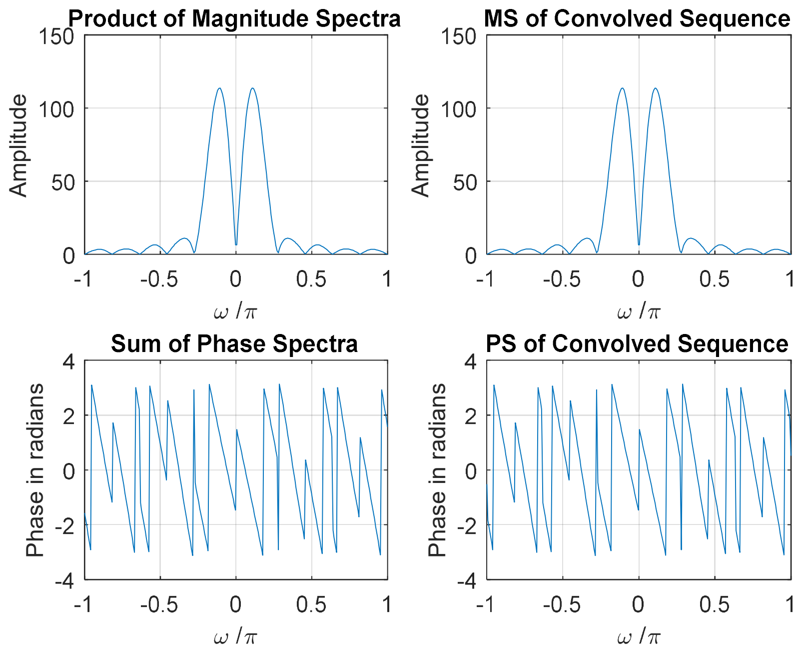
Q3.15 The plots generated by running the modified program are given below:

From these plots we make the following observations: ***Convolution Property is precisely verified.***

Q3.16 Program P3_4 was run for the following two different sets of sequences of varying lengths:

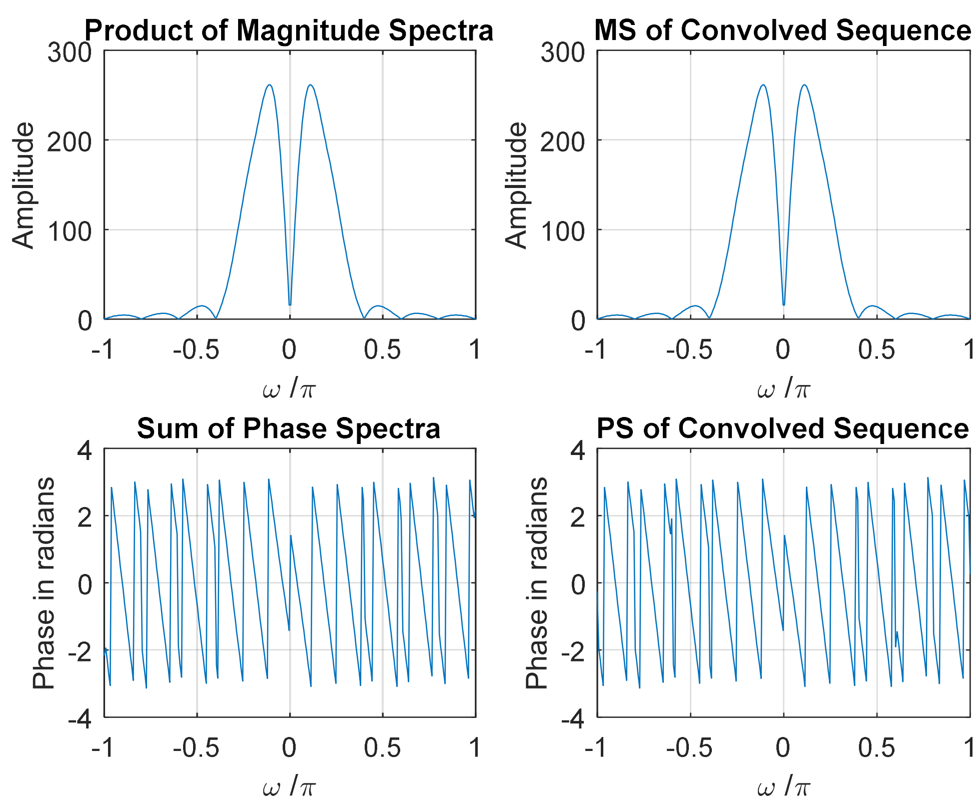
+

```
n = 0:10;
x1 = cos(0.1*pi*n);
x2 = [1 2 3 4 5 6];
```



+

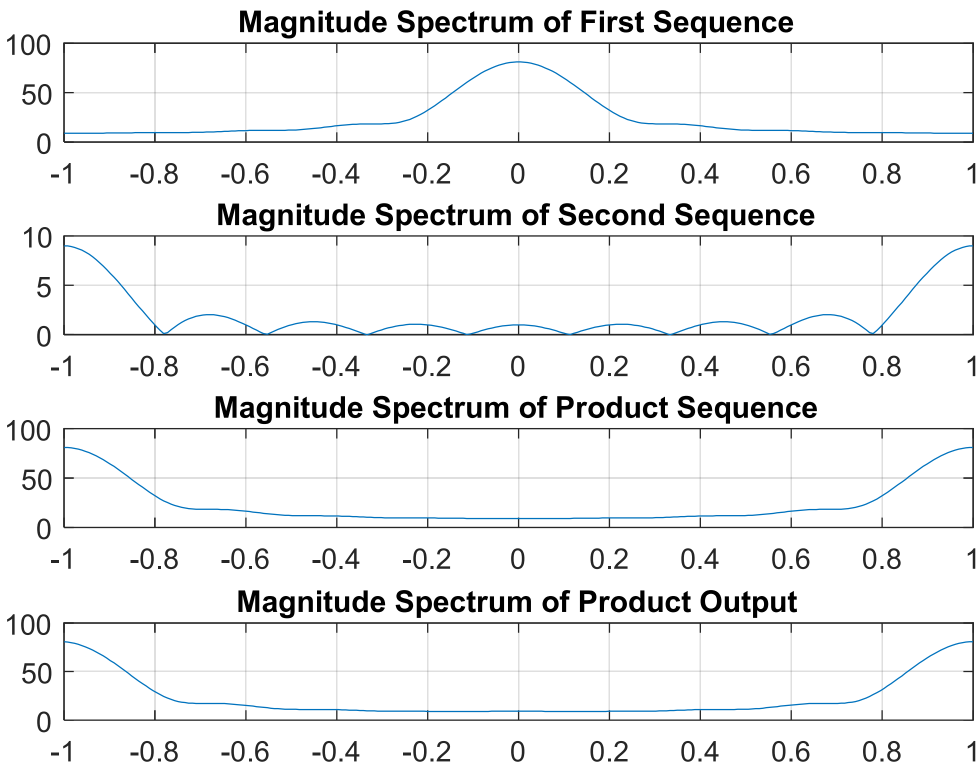
```
n = 0:10;
x1 = exp(0.1*pi*n);
x2 = sin(0.2*pi*n);
```



From these plots we make the following observations: ***Convolution Property is precisely verified.***

Q3.17 The modified Program P3_5 created by adding appropriate comment statements, and adding program statements for labeling the two axes of each plot being generated by the program is given below:

```
% Program P3_5
% Modulation Property of DTFT
clear;clc;close all;
w = -pi:2*pi/255:pi;
x1 = [1 3 5 7 9 11 13 15 17];
x2 = [1 -1 1 -1 1 -1 1 -1 1];
y = x1.*x2;
h1 = freqz(x1, 1, w);
h2 = freqz(x2, 1, w);
h3 = freqz(y,1,w);
h41 = cconv(h1,h2,256)/256;
h4 = [h41((length(h41)/2)+1:end),h41(1:length(h41)/2)];
% subplot(3,1,1)
subplot(4,1,1)
plot(w/pi,abs(h1));grid
title('Magnitude Spectrum of First Sequence')
% subplot(3,1,2)
subplot(4,1,2)
plot(w/pi,abs(h2));grid
title('Magnitude Spectrum of Second Sequence')
% subplot(3,1,3)
subplot(4,1,3)
plot(w/pi,abs(h3));grid
title('Magnitude Spectrum of Product Sequence')
subplot(4,1,4)
plot(w/pi,abs(h4));grid
title('Magnitude Spectrum of Product Output')
```



Q3.18 The plots generated by running the modified program are given below:

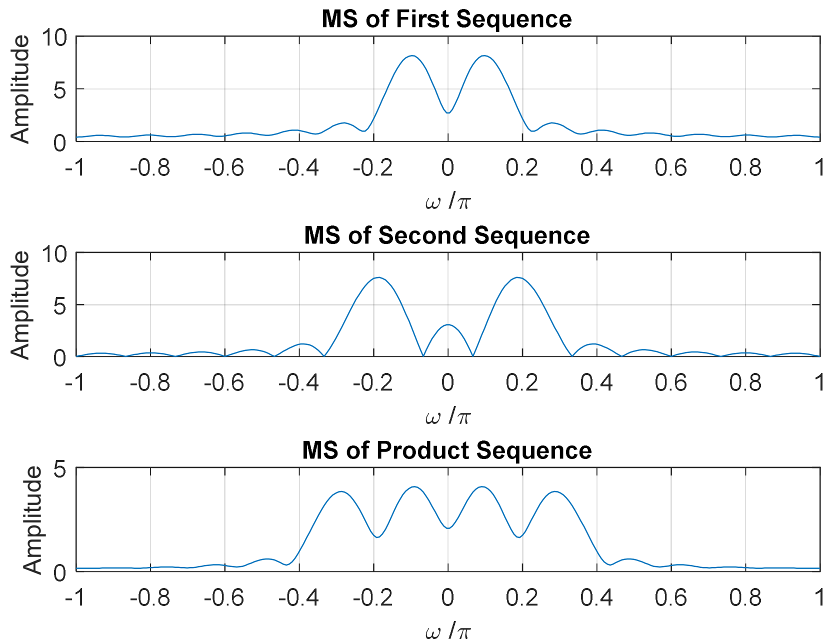
From these plots we make the following observations: ***Modulation Property is precisely verified.***

Q3.19 Program P3_5 was run for the following two different sets of sequences of varying lengths:

+

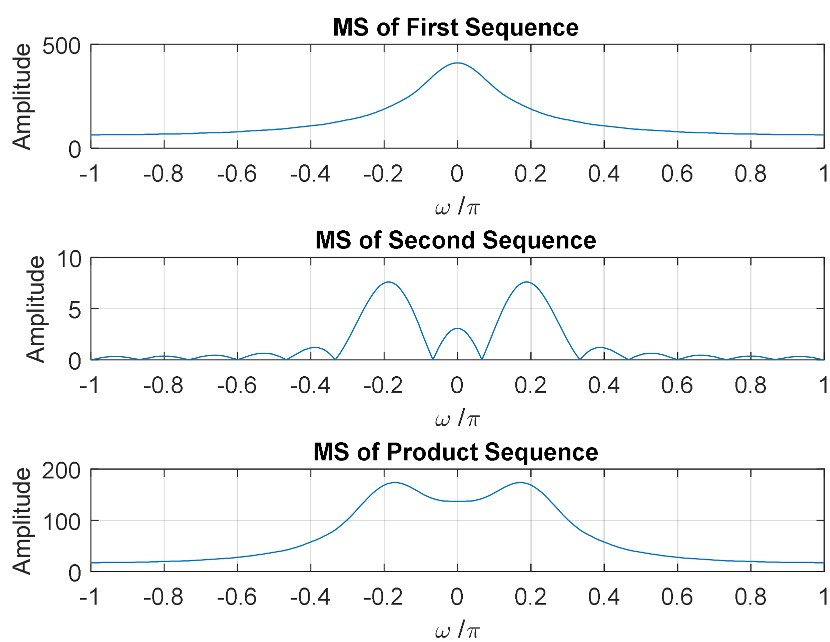
```
n=0:15;
x1 = cos(0.1*pi*n);
```

```
x2 = sin(0.2*pi*n);
```



+

```
n=0:15;
x1 = exp(0.1*pi*n);
x2 = sin(0.2*pi*n);
```



From these plots we make the following observations: ***Modulation Property is precisely verified.***

Q3.20 The modified Program P3_6 created by adding appropriate comment statements, and adding program statements for labeling the two axes of each plot being generated by the program is given below:

```
clf; clear all; close all;
w = -pi:2*pi/255:pi; % freqency vector for evaluating DTFT
num = [1 2 3 4];
L = length(num)-1;
h1 = freqz(num, 1, w); % DTFT of original ramp sequence
h2 = freqz(fliplr(num), 1, w);
h3 = exp(w*L*i).*h2;
% plot the magnitude spectrum of the original ramp sequence
subplot(2,2,1)
plot(w/pi,abs(h1));grid
title('MS of Original Sequence')
xlabel('\omega /\pi');
ylabel('Amplitude');
% plot the magnitude spectrum of the time reversed ramp
sequence
subplot(2,2,2)
plot(w/pi,abs(h3));grid
title('MS of Time-Reversed Sequence')
xlabel('\omega /\pi');
ylabel('Amplitude');
% plot the phase spectrum of the original ramp sequence
subplot(2,2,3)
plot(w/pi,angle(h1));grid
title('PS of Original Sequence')
xlabel('\omega /\pi');
ylabel('Phase in radians');
% plot the phase spectrum of the time reversed ramp
sequence
subplot(2,2,4)
plot(w/pi,angle(h3));grid
title('PS of Time-Reversed Sequence')
xlabel('\omega /\pi');
ylabel('Phase in radians');
```

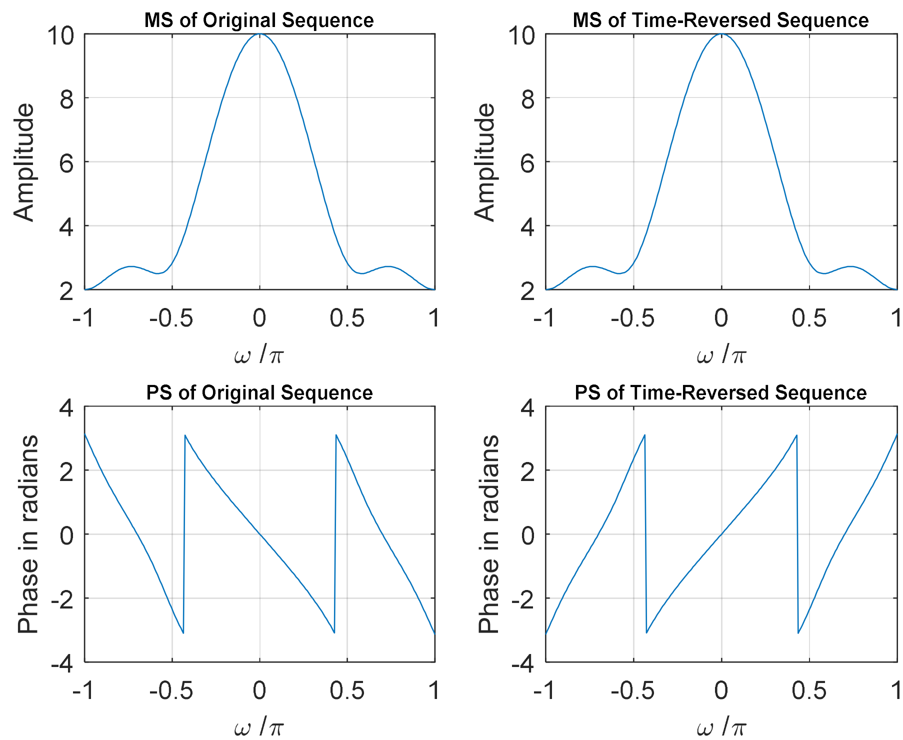
The program implements the time-reversal operation as follows

```
h2 = freqz(fliplr(num), 1, w);
h3 = exp(w*L*i).*h2;
```

Q3.21 The plots generated by running the modified program are given below:

From these plots we make the following observations: ***The Time-Reversal Property is***

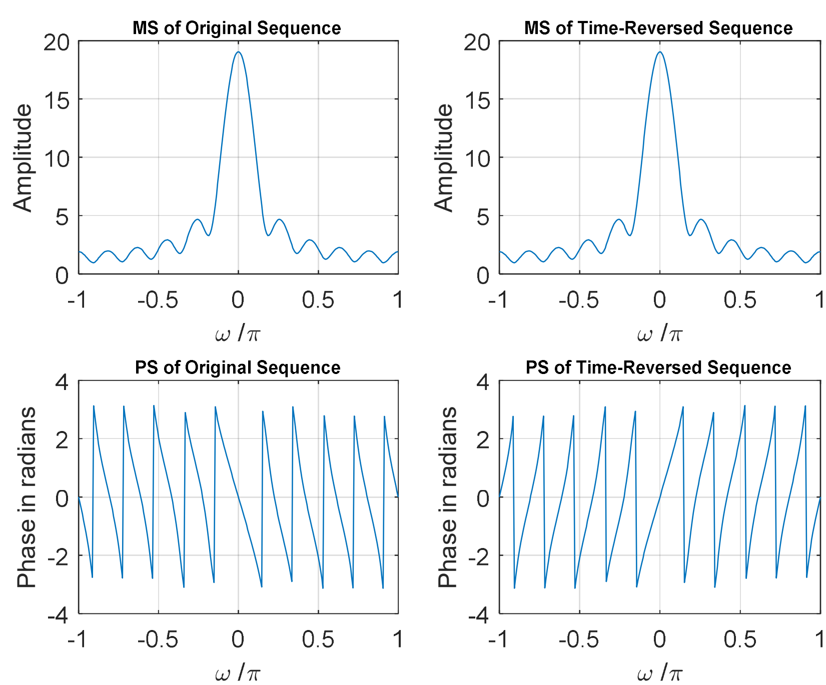
precisely verified.



Q3.22 Program P3_6 was run for the following two different sets of sequences of varying lengths:

+

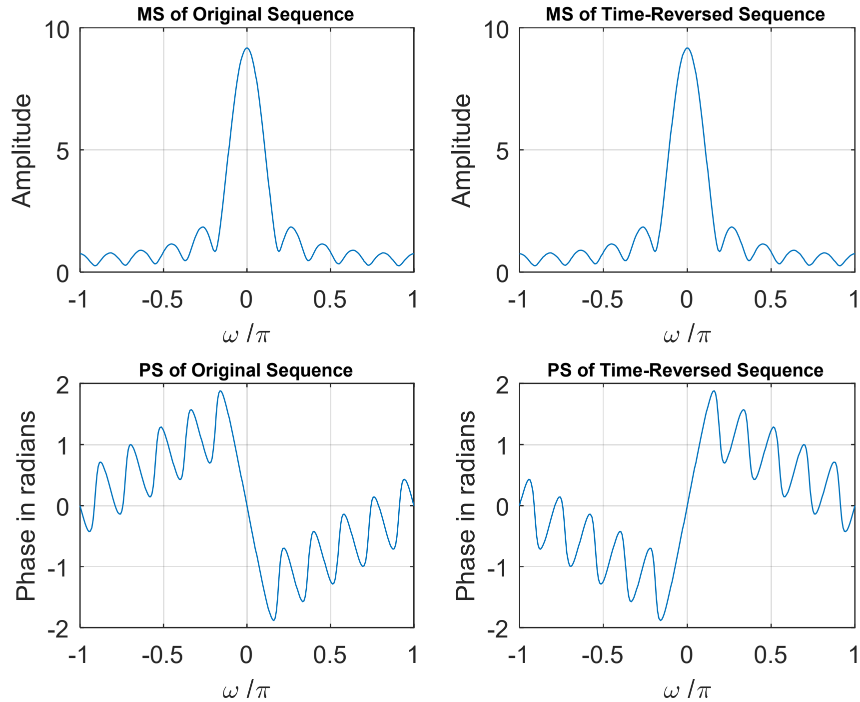
```
n = 0:10;
num = exp(0.1*n);
```



+

```
n = 0:10;
```

```
num = cos(0.1*n);
```



From these plots we make the following observations: ***The Time-Reversal Property is precisely verified.***

3.2 DISCRETE FOURIER TRANSFORM

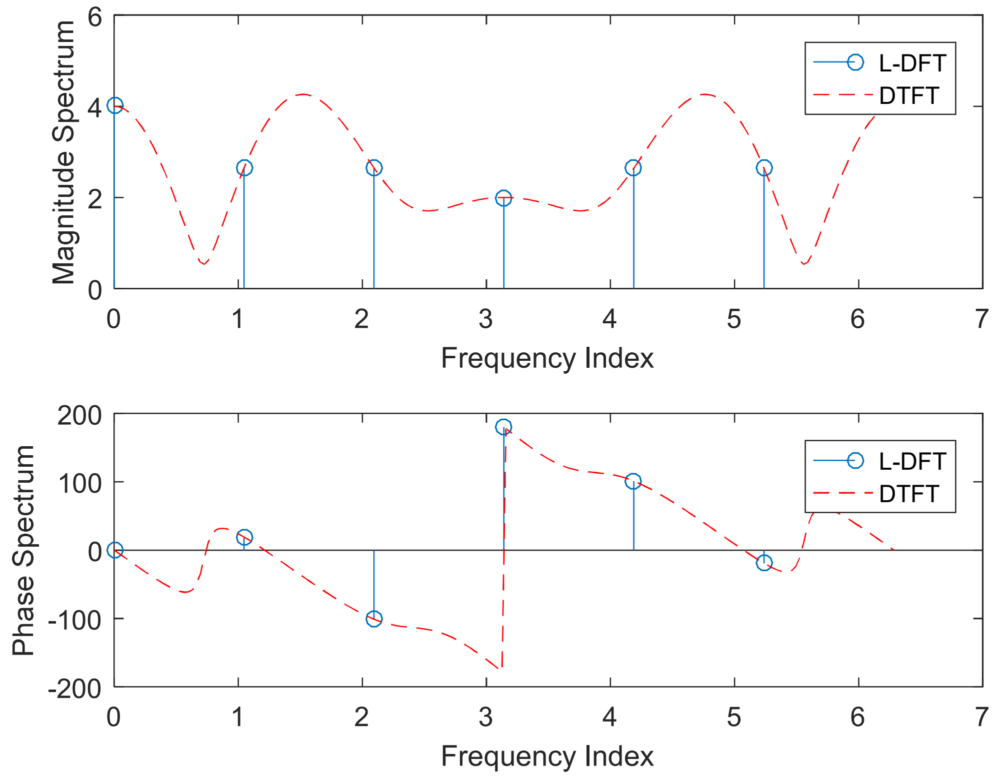
PROJECT 3.3 DFT AND IDFT COMPUTATIONS

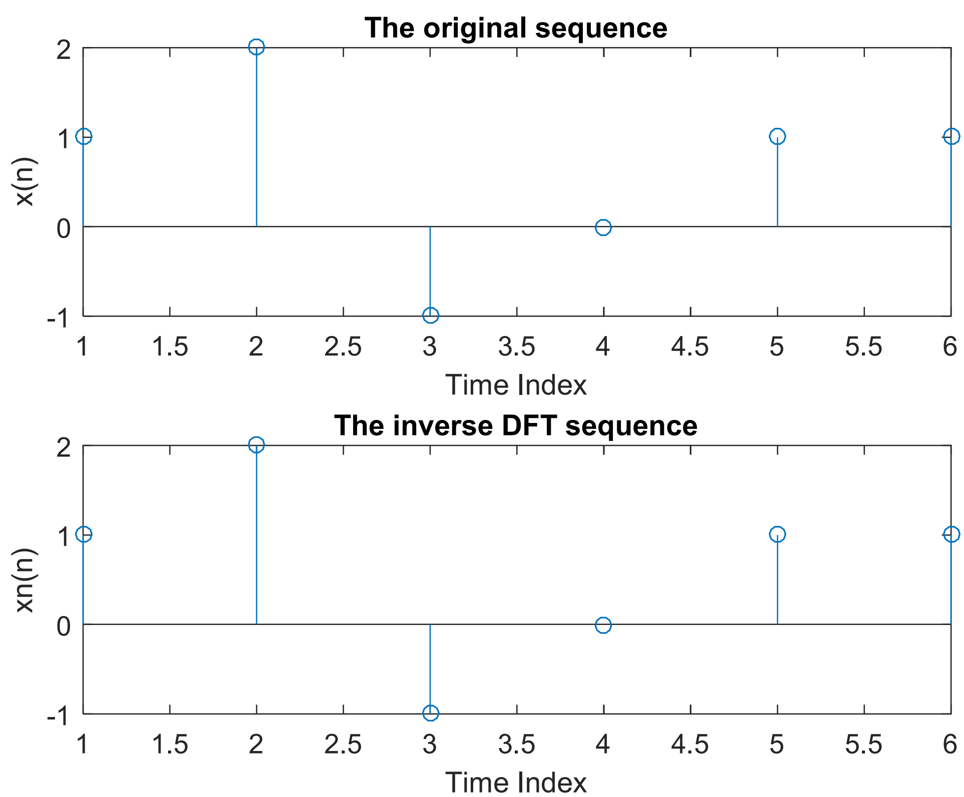
Q3.23 The MATLAB program to compute and plot the L-point DFT X[k] of a length-N sequence x[n] with L $\geq$ N and then to compute and plot the IDFT of X[k] is given below:

```
x = [1 2 -1 0 1 1];
L = 6;
clc;close all;
% Question Q3.23
% L=input('Enter the number of points L of the
DFT:');
N=length(x); % input('Enter the length N (N<=L) of
the sequence:');
if L<N
    error('L must be equal or greater than N');
    % L=input('Enter the number of points L of the
DFT:');
```

```
    % N=input('Enter the length N (N<=L) of the
sequence:');
elseif L>N
x=[x zeros(1,L-N)];% zero-padding
end
y=fft(x,L);
% DTFT
w=linspace(0,2*pi,200);
xf=freqz(x,1,w);
% DTFT
figure(1);
subplot(211);
k=(0:L-1)*2*pi/L;
stem(k,abs(y));
hold on;
plot(w,abs(xf),'r--');
legend('L-DFT','DTFT');
xlabel('Frequency Index');
ylabel('Magnitude Spectrum');
subplot(212);
stem(k,angle(y)*180/pi);
hold on;
plot(w,angle(xf)*180/pi,'r--');
legend('L-DFT','DTFT');
xlabel('Frequency Index');
ylabel('Phase Spectrum');
xn=ifft(y,L);
figure(2);
subplot(211);
stem(x);
xlabel('Time Index');
ylabel('x(n)');
title('The original sequence');
subplot(212);
stem(xn);
xlabel('Time Index');
ylabel('xn(n)');
title('The inverse DFT sequence');
```

The DFT and the IDFT pairs generated by running the program for sequences of different lengths N and for different values of the DFT length L are shown below:





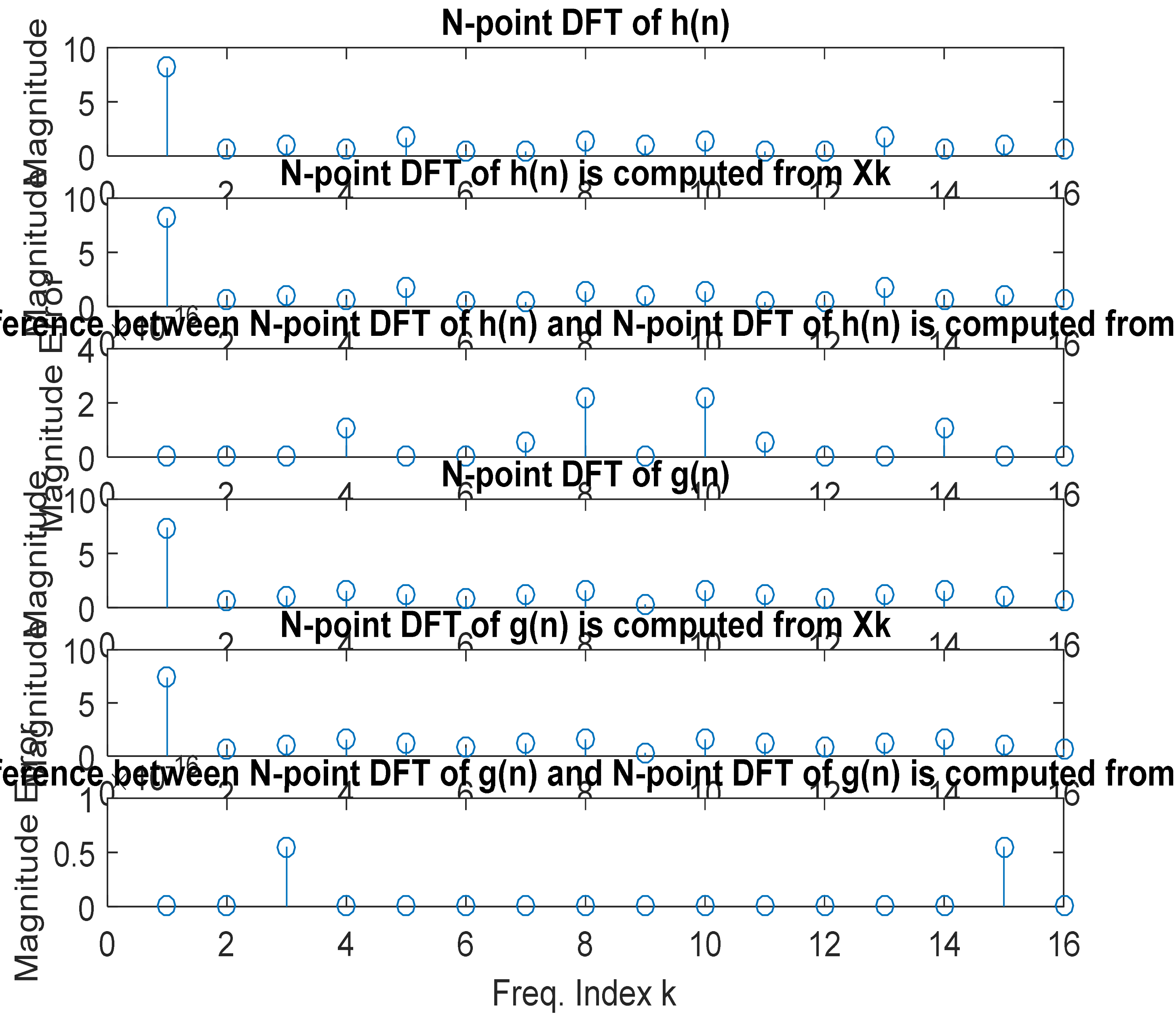
From these plots we make the following observations: ***The original sequency with the sequency implemented N – DFT and I – DFT are the same.***

Q3.24 The MATLAB program to compute the N-point DFT of two length-N real sequences using a single N-point DFT and compare the result by computing directly the two N-point DFTs is given below:

```
function Q3_24
clc;
clear all;
close all;
N=16;% N points
g=rand(1,N);
h=rand(1,N);
x=g+j*h;
Gk=fft(g,N);
Hk=fft(h,N);
Xk=fft(x,N);
Xk_cir=[Xk(1),Xk(end:-1:2)];% Circular folding of Xk
Xk_cir_conj=conj(Xk_cir);% Conjugate of Xk_cir
Gk1=(1/2)*(Xk+Xk_cir_conj);
Hk1=(-j/2)*(Xk-Xk_cir_conj);
dg=Gk-Gk1;
dh=Hk-Hk1;
subplot(611);
stem(abs(Hk));
ylabel('Magnitude');
xlabel('Freq. Index k');
title('N-point DFT of h(n)');
subplot(612);
stem(abs(Hk1));
ylabel('Magnitude');
xlabel('Freq. Index k');
title('N-point DFT of h(n) is computed from Xk');
subplot(613);
stem(abs(dh));
ylabel('Magnitude Error');
xlabel('Freq. Index k');
title('Difference between N-point DFT of h(n) and N-point DFT of
h(n) is computed from Xk');
subplot(614);
stem(abs(Gk));
ylabel('Magnitude');
xlabel('Freq. Index k');
title('N-point DFT of g(n)');
subplot(615);
stem(abs(Gk1));
ylabel('Magnitude');
xlabel('Freq. Index k');
title('N-point DFT of g(n) is computed from Xk');
subplot(616);
stem(abs(dg));
ylabel('Magnitude Error');
```

```
xlabel('Freq. Index k');
title('Difference between N-point DFT of g(n) and N-point DFT of
g(n) is computed from Xk');
end
```

The DFTs generated by running the program for sequences of different lengths N are shown below:

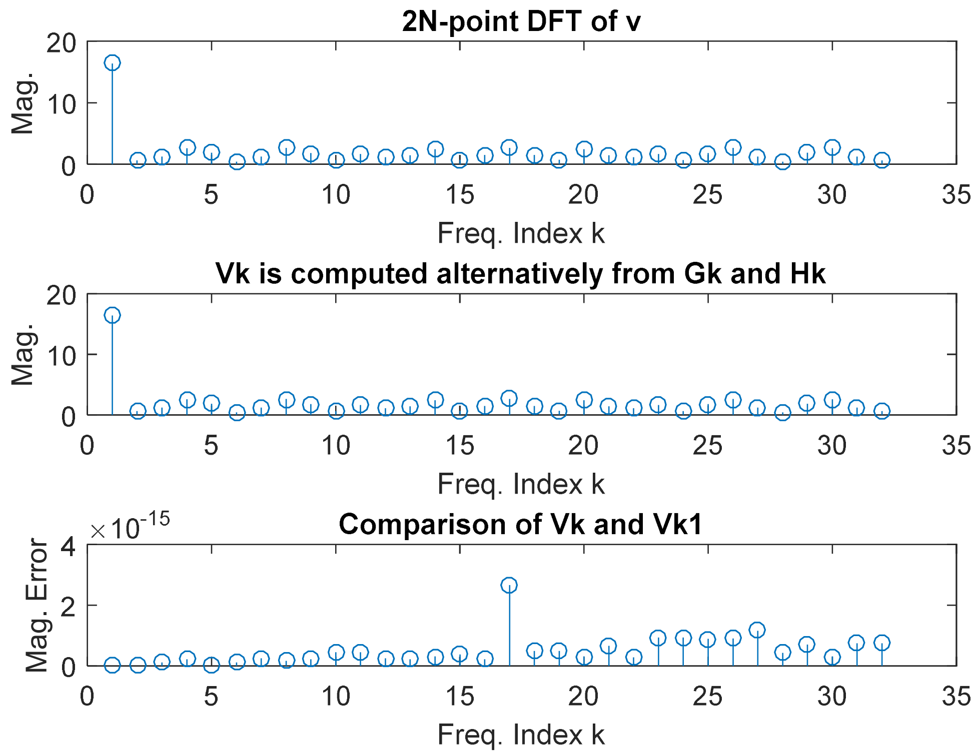


From these plots we make the following observations: ***The results obtained by computing G[k] and H[k] with a single N-point DFT or directly with two N-point DFT's are almost similar.***

Q3.25 The MATLAB program to compute the 2N-point DFT of a length-2N real sequence using a single N-point DFT and compare the result by computing directly the 2N-point DFT is shown below:

```
function Q3_25

clc;
clear all;
close all;
N=16; % N-point DFT
v=rand(1,2*N);
g=v(1:2:length(v)); % g is selected from the even positions of v
h=v(2:2:length(v)); % h is selected from the odd positions of v
Gk=fft(g,N);
Hk=fft(h,N);
Vk=fft(v,2*N);
W=exp(-j*2*pi/(2*N)); % W2N
k=0:2*N-1;
Vk1=[Gk,Gk]+(W.^k).*[Hk Hk];
dv=Vk-Vk1;
subplot(311);
stem(abs(Vk));
ylabel('Mag.');
xlabel('Freq. Index k');
title('2N-point DFT of v');
subplot(312);
stem(abs(Vk1));
ylabel('Mag.');
xlabel('Freq. Index k');
title('Vk is computed alternatively from Gk and Hk');
subplot(313);
stem(abs(dv));
ylabel('Mag. Error');
xlabel('Freq. Index k');
title('Comparison of Vk and Vk1');
end
```

The DFTs generated by running the program for sequences of different lengths 2N are shown below:



From these plots we make the following observations: ***Two methods result in the same destination.***

PROJECT 3.4 DFT PROPERTIES

Q3.26 The purpose of the command `rem` in the function `circshift` is ***used to remainder after division.***

Q3.27 The function `circshift` operates as follows:

Q3.28 The purpose of the operator ~= in the function ***circonv*** is ***used to consider the binary relationship NOT EQUAL operator. It returns 1 if A and B are not equal and vice versal.***

Q3.29 The function *`circonv`* operates as follows:

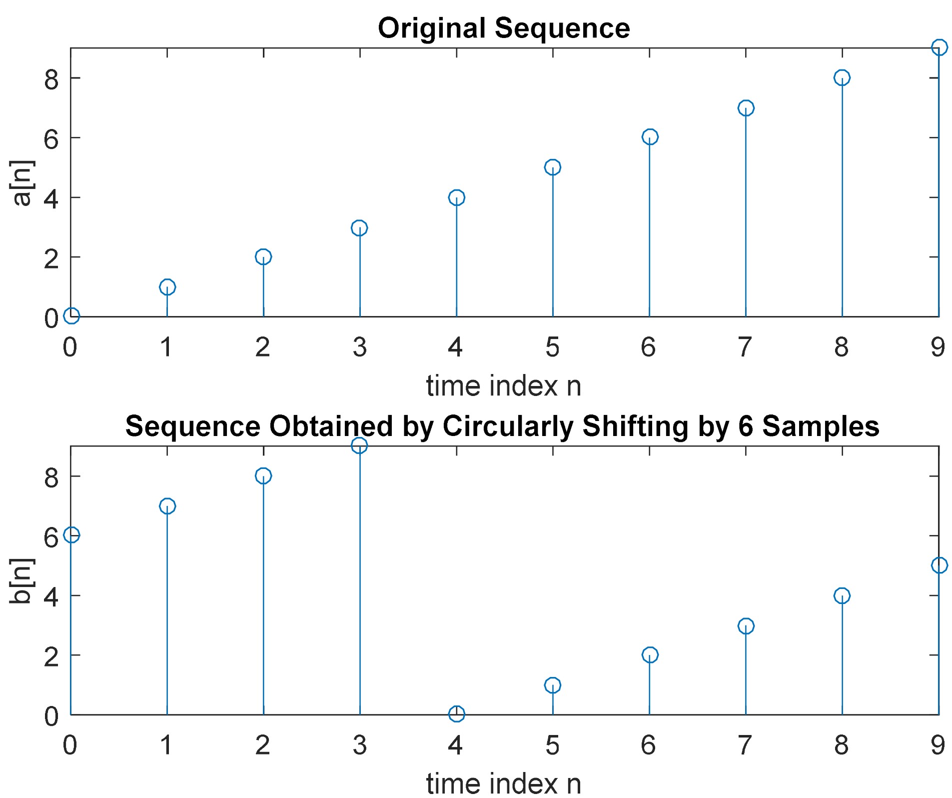
Q3.30 The modified Program P3_7 created by adding appropriate comment statements, and adding program statements for labeling each plot being generated by the program is given below:

```
% Illustration of Circular Shift of a Sequence

clf; clear all; close all;
% initialize shift amount M
M = 6;
% initialize sequence a to be shifted
a = [0 1 2 3 4 5 6 7 8 9];
b = circshift(a,M); % perform the circular shift
```

```
L = length(a)-1;
% plot the original sequence a and the circularly shifted
sequence b
n = 0:L;
subplot(2,1,1);
stem(n,a);axis([0,L,min(a),max(a)]);
title('Original Sequence');
xlabel('time index n');
ylabel('a[n]');
subplot(2,1,2);
stem(n,b);axis([0,L,min(a),max(a)]);
title(['Sequence Obtained by Circularly Shifting by
',num2str(M),' Samples']);
xlabel('time index n');
ylabel('b[n]');
```

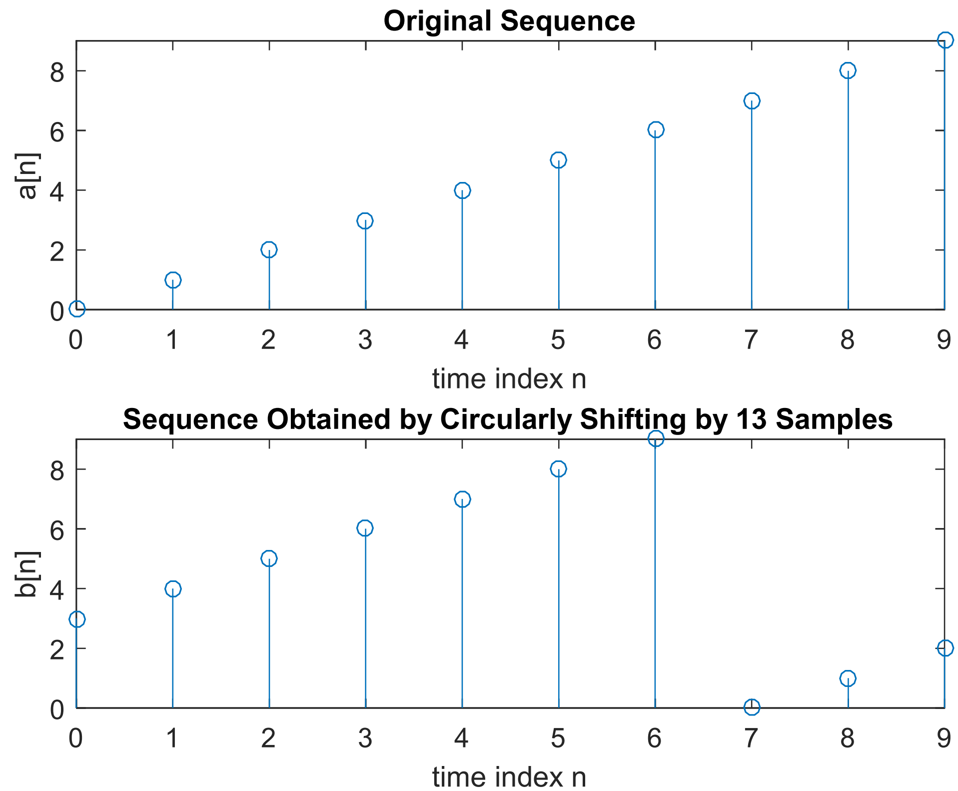
The parameter determining the amount of time-shifting is - M

If the amount of time-shift is greater than the sequence length then - ***If M > Length(a) is equivalent to shifting left by M mod Length(a), because after moving one circle, it still returns to the old position.***



Q3.31 The plots generated by running the modified program are given below:

+ Replace M from 6 to 13, we get this plot:



From these plots we make the following observations: ***We have*** $13 = 10 + 3$***, with 10 is a length of a sequency, so after moving 10 samples, we return the old position and then we shift the sequency left by 3 samples.***

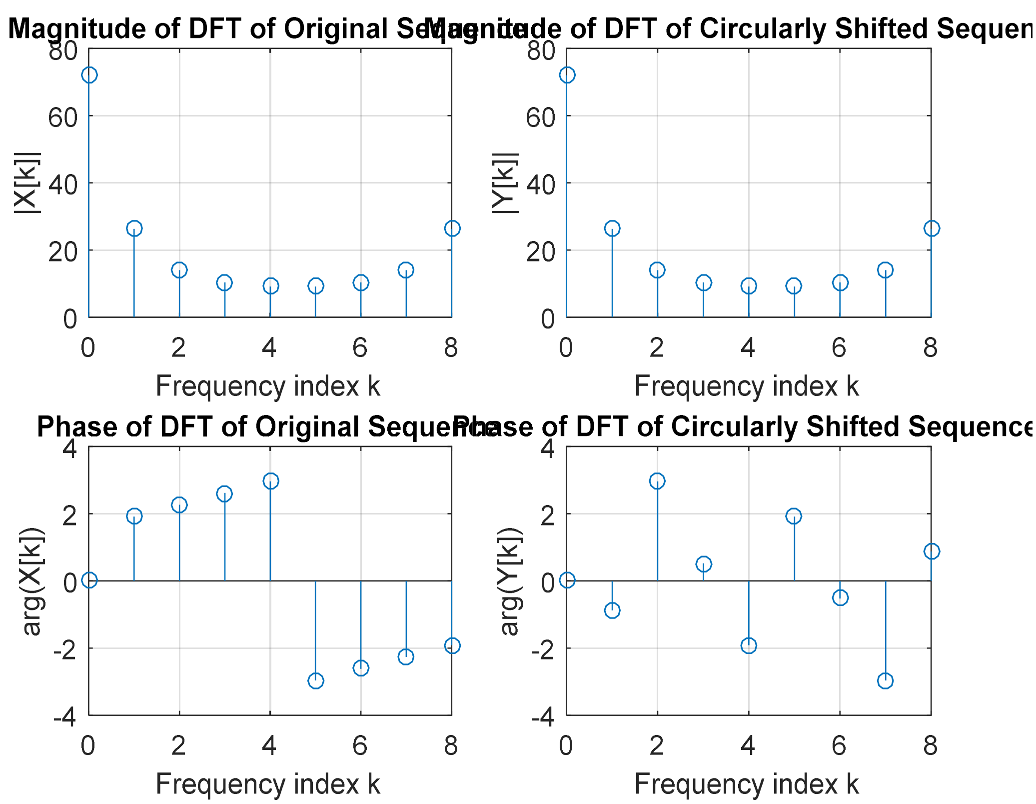
Q3.32 The modified Program P3_8 created by adding appropriate comment statements, and adding program statements for labeling each plot being generated by the program is given below:

```
% Circular Time-Shifting Property of DFT

clf; clear all; close all;
x = [0 2 4 6 8 10 12 14 16]; % original sequence x
N = length(x)-1; n = 0:N; % time index vector
% set y equal to the circular shift left of x
y = circshift(x,5);
XF = fft(x); % DFT of x
YF = fft(y); % DFT of y
subplot(2,2,1);
% plot the spectral magnitudes of the original and shifted
sequences
stem(n,abs(XF));grid;
title('Magnitude of DFT of Original Sequence');
xlabel('Frequency index k');
```

```
ylabel('|X[k]|');
subplot(2,2,2);
stem(n,abs(YF));grid;
title('Magnitude of DFT of Circularly Shifted Sequence');
xlabel('Frequency index k');
ylabel('|Y[k]|');
% plot the spectral phases of the original and shifted
sequences
subplot(2,2,3);
stem(n,angle(XF));grid;
title('Phase of DFT of Original Sequence');
xlabel('Frequency index k');
ylabel('arg(X[k])');
subplot(2,2,4);
stem(n,angle(YF));grid;
title('Phase of DFT of Circularly Shifted Sequence');
xlabel('Frequency index k');
ylabel('arg(Y[k])');
```

The amount of time-shift is 5.

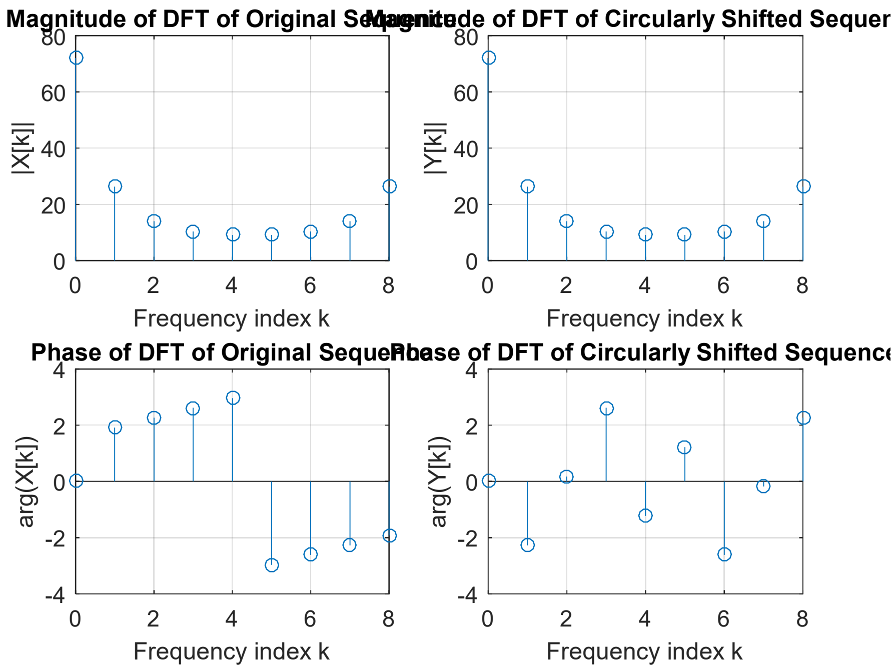
Q3.33 The plots generated by running the modified program are given below:



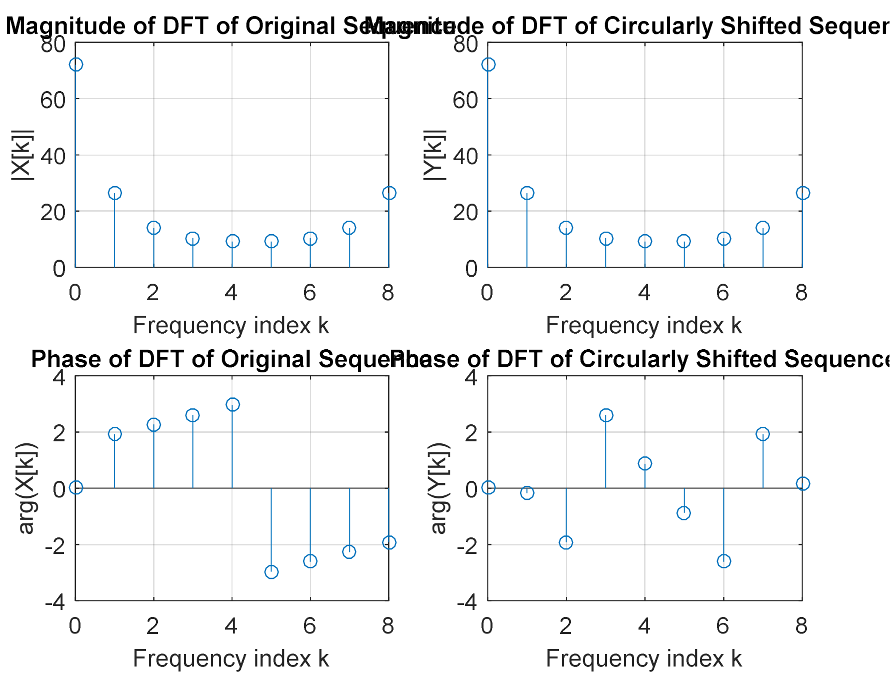
From these plots we make the following observations: ***The Circular Time-Shifting Property is verified exactly.***

Q3.34 The plots generated by running the modified program for the following two different amounts of time-shifts, with the amount of shift indicated, are shown below:

+ $D = 3$



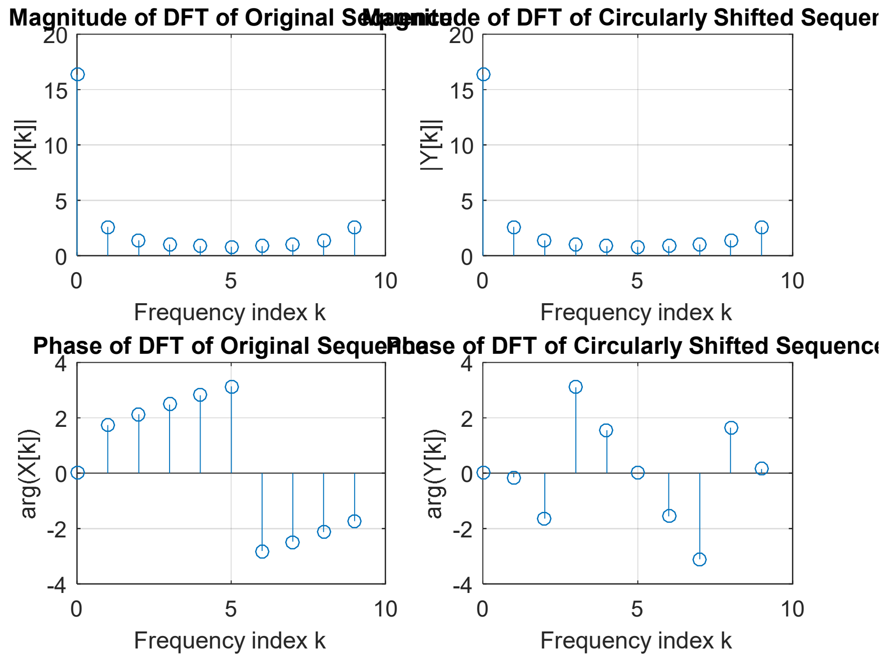
+ $D = -3$



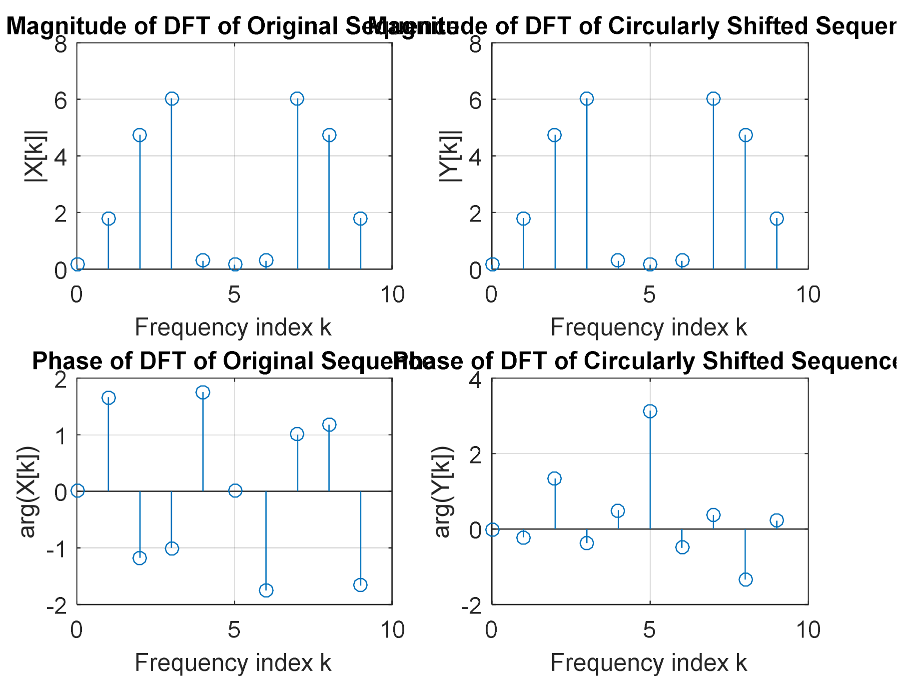
From these plots we make the following observations: ***The Circular Time-Shifting Property is verified exactly.***

Q3.35 The plots generated by running the modified program for the following two different sequences of different lengths, with the lengths indicated, are shown below:

+
```
n= 0:9;
x = exp(0.1*n);
```



+
```
    n= 0:9;
    x = cos(n)+sin(2*n);
```



From these plots we make the following observations: ***The Circular Time-Shifting Property is verified exactly.***

Q3.36 A copy of Program P3_9 is given below along with the plots generated by running this program:

```
% Circular Convolution Property of DFT
g1 = [1 2 3 4 5 6]; g2 = [1 -2 3 3 -2 1];
ycir = Circonv(g1,g2); %chu C in moi chiu nha ☹(((
disp('Result of circular convolution = ');disp(ycir)
G1 = fft(g1); G2 = fft(g2);
yc = real(ifft(G1.*G2));
disp('Result of IDFT of the DFT products = ');disp(yc)
```

```
Result of circular convolution =

    12    28    14     0    16    14
Result of IDFT of the DFT products =
    12    28    14     0    16    14
```

From these plots we make the following observations: ***The Circular Convolution Property is verified exactly.***

Q3.37 Program P3_9 was run again for the following two different sets of equal-length sequences:

```
g1 = [1 -1 1 -1 1 -1];
g2 = [-1 1 -1 1 -1 1];
```

The plots generated are shown below:

```
Result of circular convolution =

    -6     6    -6     6    -6     6
Result of IDFT of the DFT products =
    -6     6    -6     6    -6     6
```

From these plots we make the following observations: ***The Circular Convolution Property is verified exactly.***

Q3.38 A copy of Program P3_10 is given below along with the plots generated by running this program:

```
% Program P3_10

% Linear Convolution via Circular Convolution
g1 = [1 2 3 4 5];g2 = [2 2 0 1 1];
g1e = [g1 zeros(1,length(g2)-1)];
g2e = [g2 zeros(1,length(g1)-1)];
```

```
ylin = Circonv(g1e,g2e);
disp('Linear convolution via circular convolution = 
');disp(ylin);
y = conv(g1, g2);
disp('Direct linear convolution = ');disp(y)
```

```
Linear convolution via circular convolution =

     2     6    10    15    21    15     7     9     5
Direct linear convolution =
     2     6    10    15    21    15     7     9     5
```

From these plots we make the following observations: ***Two methods make the same results.***

Q3.39 Program P3_10 was run again for the following two different sets of sequences of unequal lengths:

```
g1 = [1 -1 2 -2 3 -3];
g2 = [-3 3 -2 2 -1 1 5 5];
```

The plots generated are shown below:

```
Linear convolution via circular convolution =

 -3 6 -11 16 -22 28 -17 16 -6 6 2 0 -15
Direct linear convolution =
 -3 6 -11 16 -22 28 -17 16 -6 6 2 0 -15
```

From these plots we make the following observations: ***Two methods make the same results***

Q3.40 The MATLAB program to develop the linear convolution of two sequences via the DFT of each is given below:

```
% Linear Convolution via Circular Convolution

g1 = [1 2 3 4 5];
g2 = [2 2 0 1 1];
g1e = [g1 zeros(1,length(g2)-1)];
g2e = [g2 zeros(1,length(g1)-1)];
G1EF = fft(g1e);
G2EF = fft(g2e);
ylin = real(ifft(G1EF.*G2EF));
disp('Linear convolution by DFT = ');disp(ylin);
```

The plots generated by running this program for the sequences of Q3.38 are shown below:

```
  Linear convolution by DFT =

  Columns 1 through 7

    2.0000    6.0000   10.0000   15.0000   21.0000
15.0000    7.0000

  Columns 8 through 9

    9.0000    5.0000
```

From these plots we make the following observations: ***We can find the linear convolution of two sequences via the DFT.***

The plots generated by running this program for the sequences of Q3.39 are shown below:

```
Linear convolution via circular convolution =

 -3 6 -11 16 -22 28 -17 16 -6 6 2 0 -15
Direct linear convolution =
 -3 6 -11 16 -22 28 -17 16 -6 6 2 0 -15
```

From these plots we make the following observations: ***These methods make the same final results.***

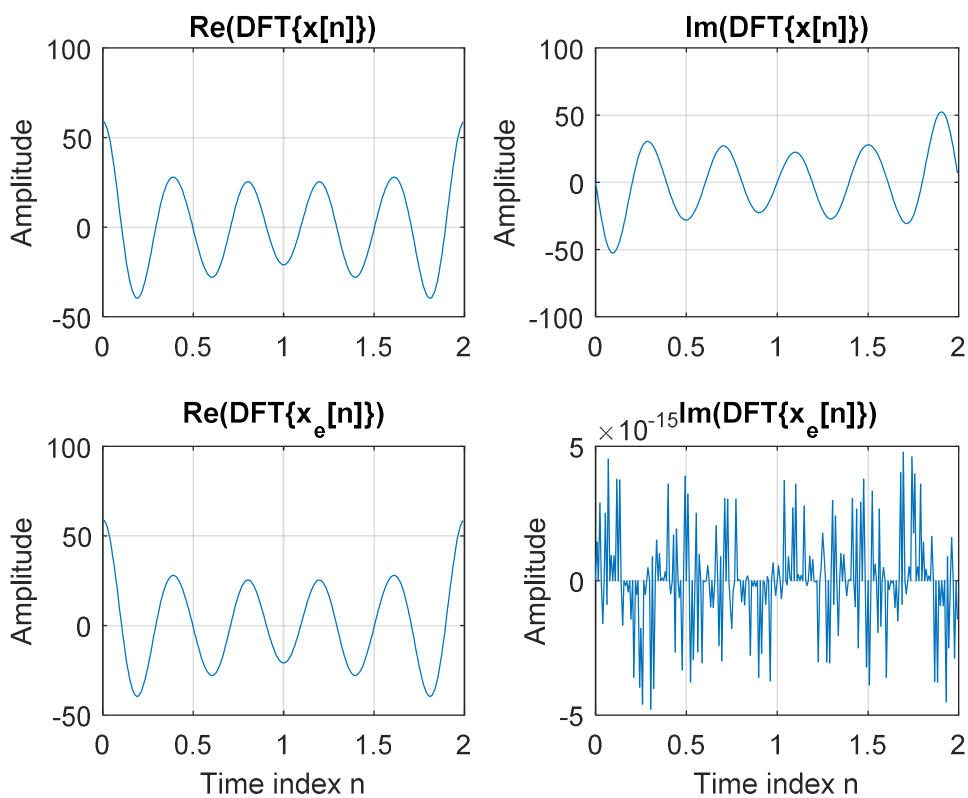
Q3.41 A copy of Program P3_11 is given below:

```
% Relations between the DFTs of the Periodic Even

% and Odd Parts of a Real Sequence
x = [1 2 4 2 6 32 6 4 2 zeros(1,247)];
x1 = [x(1) x(256:-1:2)];
xe = 0.5 *(x + x1);
XF = fft(x);
XEF = fft(xe);
clf;
k = 0:255;
subplot(2,2,1);
plot(k/128,real(XF)); grid;
ylabel('Amplitude');
title('Re(DFT\{x[n]\})');
subplot(2,2,2);
plot(k/128,imag(XF)); grid;
ylabel('Amplitude');
title('Im(DFT\{x[n]\})');
subplot(2,2,3);
```

```
plot(k/128,real(XEF)); grid;
xlabel('Time index n');ylabel('Amplitude');
title('Re(DFT\{x_{e}[n]\})');
subplot(2,2,4);
plot(k/128,imag(XEF)); grid;
xlabel('Time index n');ylabel('Amplitude');
title('Im(DFT\{x_{e}[n]\})');
```

The relation between the sequence x1[n] and x[n] is – ***x1[n] is a circular reversal version of x[n].***

Q3.42 The plots generated by running Program P3_11 are given below:

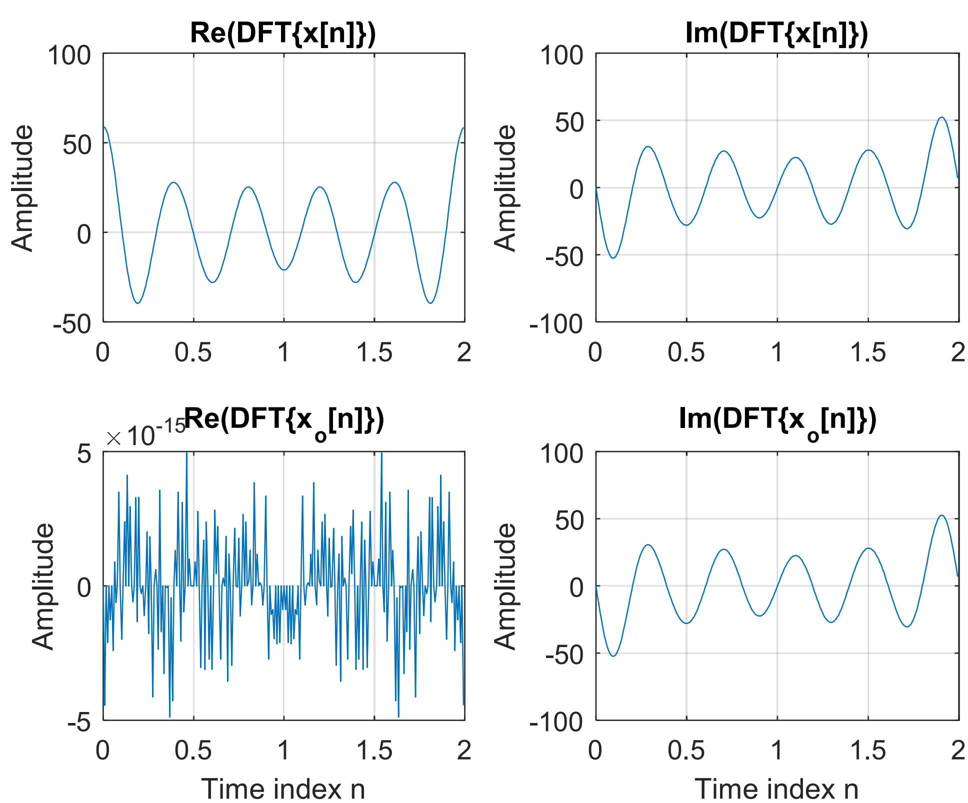


The imaginary part of XEF is approximately equal to zero. This result can be explained as follows:

Q3.43 The required modifications to Program P3_11 to verify the relation between the DFT of the periodic odd part and the imaginary part of XEF are given below along with the plots generated by running this program:

```
% Relations between the DFTs of the Periodic Even

% and Odd Parts of a Real Sequence
x = [1 2 4 2 6 32 6 4 2 zeros(1,247)];
x1 = [x(1) x(256:-1:2)];
```

```
xo = 0.5 *(x - x1);
XF = fft(x);
XOF = fft(xo);
clf;
k = 0:255;
subplot(2,2,1);
plot(k/128,real(XF)); grid;
ylabel('Amplitude');
title('Re(DFT\{x[n]\})');
subplot(2,2,2);
plot(k/128,imag(XF)); grid;
ylabel('Amplitude');
title('Im(DFT\{x[n]\})');
subplot(2,2,3);
plot(k/128,real(XOF)); grid;
xlabel('Time index n');ylabel('Amplitude');
title('Re(DFT\{x_{o}[n]\})');
subplot(2,2,4);
plot(k/128,imag(XOF)); grid;
xlabel('Time index n');ylabel('Amplitude');
title('Im(DFT\{x_{o}[n]\})');
```



From these plots we make the following observations: ***The properties shown in R3.17 was verified exactly.***

Q3.44 A copy of Program P3_12 is given below:

```
% Program P3_12

% Parseval's Relation
x = [(1:128) (128:-1:1)];
XF = fft(x);
a = sum(x.*x)
b = round(sum(abs(XF).^2)/256)
```

The values for a and b we get by running this program are:

```
 a =

       1414528
b =
       1414528
```

Q3.45 The required modifications to Program P3_11 are given below:

```
% Parseval's Relation

x = [(1:128) (128:-1:1)];
XF = fft(x);
a = sum(x.*x)
b = round(sum(XF.*conj(XF))/256)
```

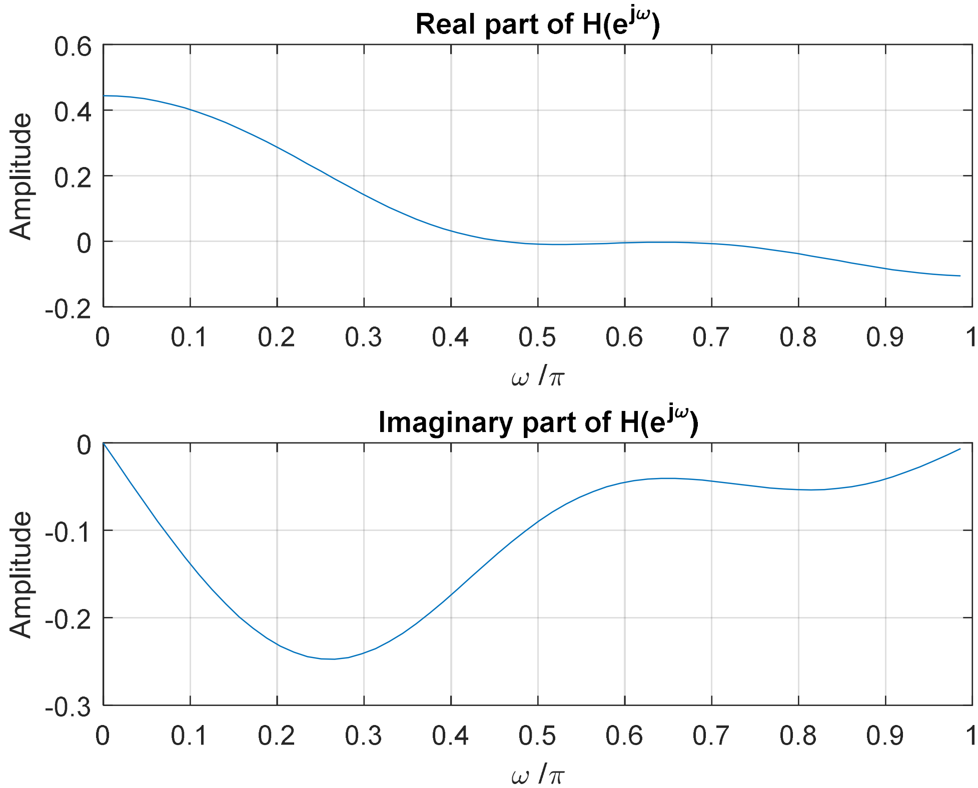
```
a =

       1414528
b =
       1414528
```

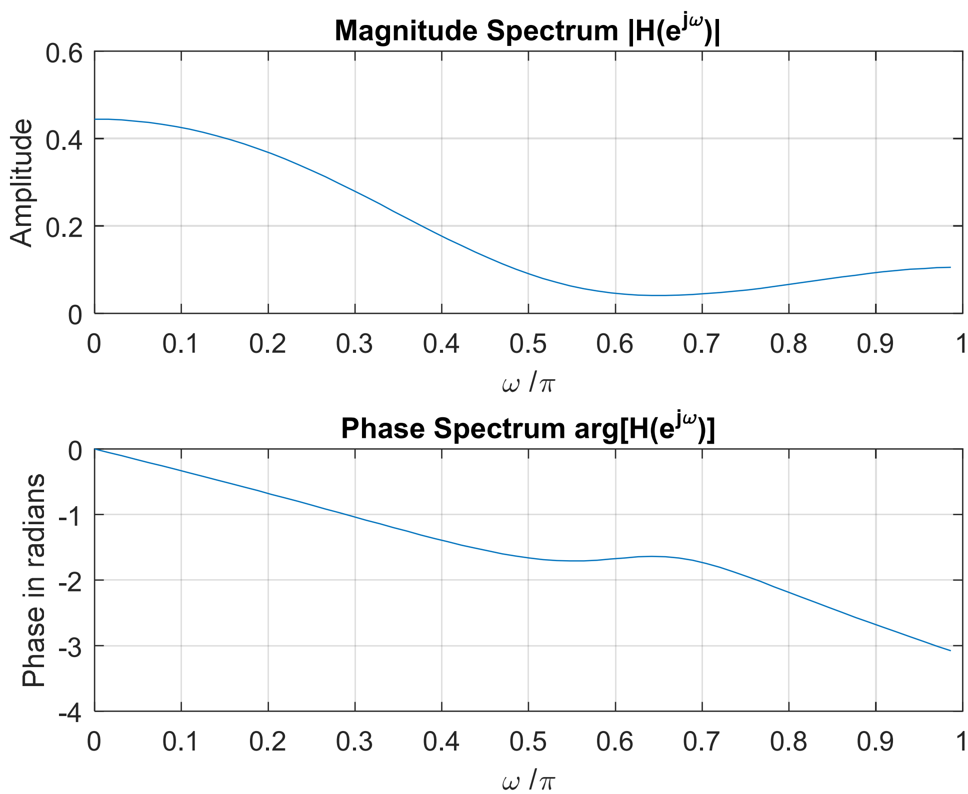
3.3 Z-TRANSFORM

PROJECT 3.5 ANALYSIS OF Z-TRANSFORMS

Q3.46 The frequency response of the z-transform obtained using Program P3_1 is plotted below:

```
clf; clear all; close all;

% Compute the frequency samples of the DTFT
w = 0:8*pi/511:pi;
num = [2 5 9 5 3];
den = [5 45 2 1 1];
h = freqz(num, den, w);
% Plot the DTFT
subplot(2,1,1)
plot(w/pi,real(h));grid
title('Real part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,imag(h));grid
title('Imaginary part of H(e^{j\omega})')
xlabel('\omega /\pi');
ylabel('Amplitude');
pause
subplot(2,1,1)
plot(w/pi,abs(h));grid
title('Magnitude Spectrum |H(e^{j\omega})|')
xlabel('\omega /\pi');
ylabel('Amplitude');
subplot(2,1,2)
plot(w/pi,angle(h));grid
title('Phase Spectrum arg[H(e^{j\omega})]')
xlabel('\omega /\pi');
ylabel('Phase in radians');
```


Real part of H(e^jω)
Amplitude
ω /π
Imaginary part of H(e^jω)
Amplitude
ω /π


Magnitude Spectrum |H(e^jω)|
Amplitude
ω /π
Phase Spectrum arg[H(e^jω)]
Phase in radians
ω /π

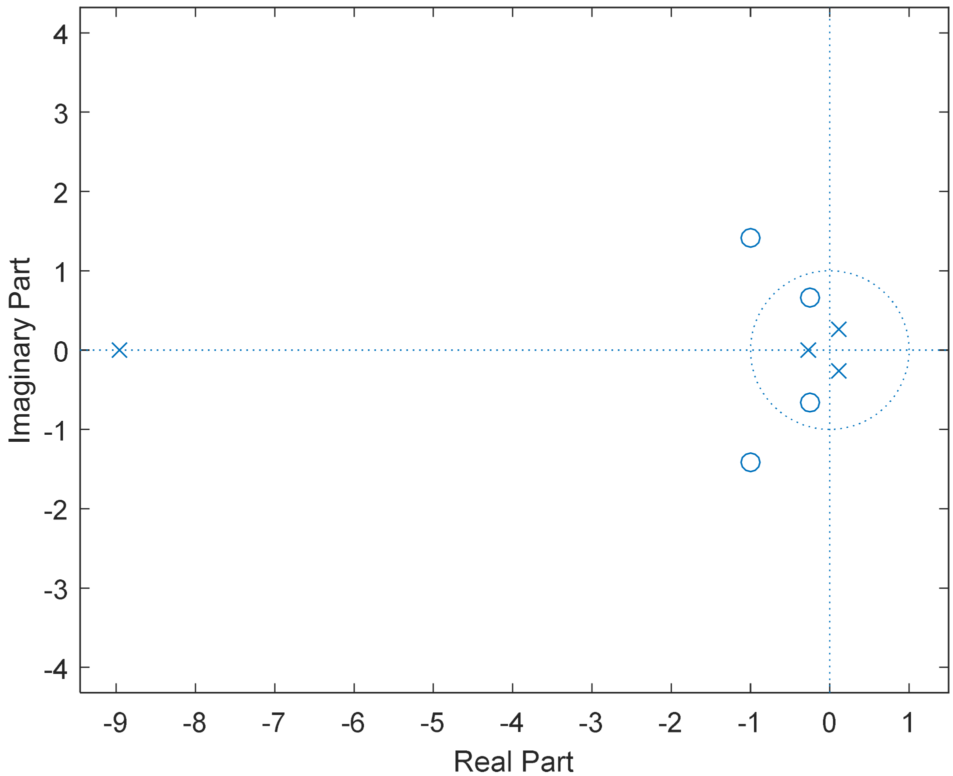
Q3.47 The MATLAB program to compute and display the poles and zeros, to compute and display the factored form, and to generate the pole-zero plot of a rational z-transform is given below:

```
function Q3_47
%clc;
%clear all;
close all;
% num. and den. coefficients of Tranfer Function of DF sys.
num=[2 5 9 5 3];
den=[5 45 2 1 1];
% poles and zeros of sys.
[z,p,k]=tf2zp(num,den);
disp('Zeros are:');
disp(num2str(z));
disp('Poles are:');
disp(num2str(p));
% factorized form
disp('TF is factorized as a sum of the following partial
fractions:')
for i=1:length(p)
    disp([['(',num2str(z(i)),')'],'/',['(','1-
',['(',num2str(p(i)),')'],'z^-1',')']]);
end
% pole-zero plot
figure(1);
zplane(num,den);
% phi=linspace(0,2*pi,200);
% figure(2);
% z=zeros(length(p),200);
% for k=1:length(p)
%     z(k,:)=abs(p(k))*exp(j*phi);
%     polarplot(phi,z(k,:));
%     hold on;
% end
end
```

Using this program we obtain the following results on the z-transform G(z) of Q3.46:

```
Zeros are:
   -1+1.4142i
   -1-1.4142i
-0.25+0.66144i
-0.25-0.66144i
Poles are:
 -8.9576+0i
-0.27177+0i
 0.11466+0.2627i
 0.11466-0.2627i
TF is factorized as a sum of the following partial fractions:
(-1+1.4142i)/(1-(-8.9576)z^-1)
(-1-1.4142i)/(1-(-0.27177)z^-1)
```

```
(-0.25+0.66144i)/(1-(0.11466+0.2627i)z^-1)
(-0.25-0.66144i)/(1-(0.11466-0.2627i)z^-1)
```



Q3.48 From the pole-zero plot generated in Question Q3.47, the number of regions of convergence (ROC) of G(z) are - ***4.***

All possible ROCs of this z-transform are sketched below:

-8.9576 + 0.0000i → $\left|z_{p_0}\right| = 8.9576$

-0.2718 + 0.0000i → $\left|z_{p_1}\right| = 0.2718$

0.1147 + 0.2627i → $\left|z_{p_2}\right| = \sqrt{0.1147^2 + 0.2627^2} = 0.2866$

0.1147 - 0.2627i → $\left|z_{p_3}\right| = \sqrt{0.1147^2 + 0.2627^2} = 0.2866$

+ $\left|z\right| < 0.2718$

+ $0.2718 < \left|z\right| < 0.2866$

+ $0.2866 < \left|z\right| < 8.9576$

+ $\left|z\right| > 8.9576$

From the pole-zero plot it can be seen that the DTFT - ***Whether DTFT exists or not is based not only on the poles but also on the region of convergence.***

Q3.49 The MATLAB program to compute and display the rational z-transform from its zeros, poles and gain constant is given below:

```
clf; clear all; close all;
% initialize
z = [0.3 2.5 -0.2+i*0.4 -0.2-i*0.4]';
p = [0.5 -0.75 0.6+i*0.7 0.6-i*0.7]';
k = 3.9;
[num den] = zp2tf(z,p,k)
```

The rational form of a z-transform with the given poles, zeros, and gain is found to be:

```
num =
    3.9000    -9.3600    -0.6630    -1.0140     0.5850
den =
    1.0000    -0.9500     0.1750     0.6625    -0.3187
```

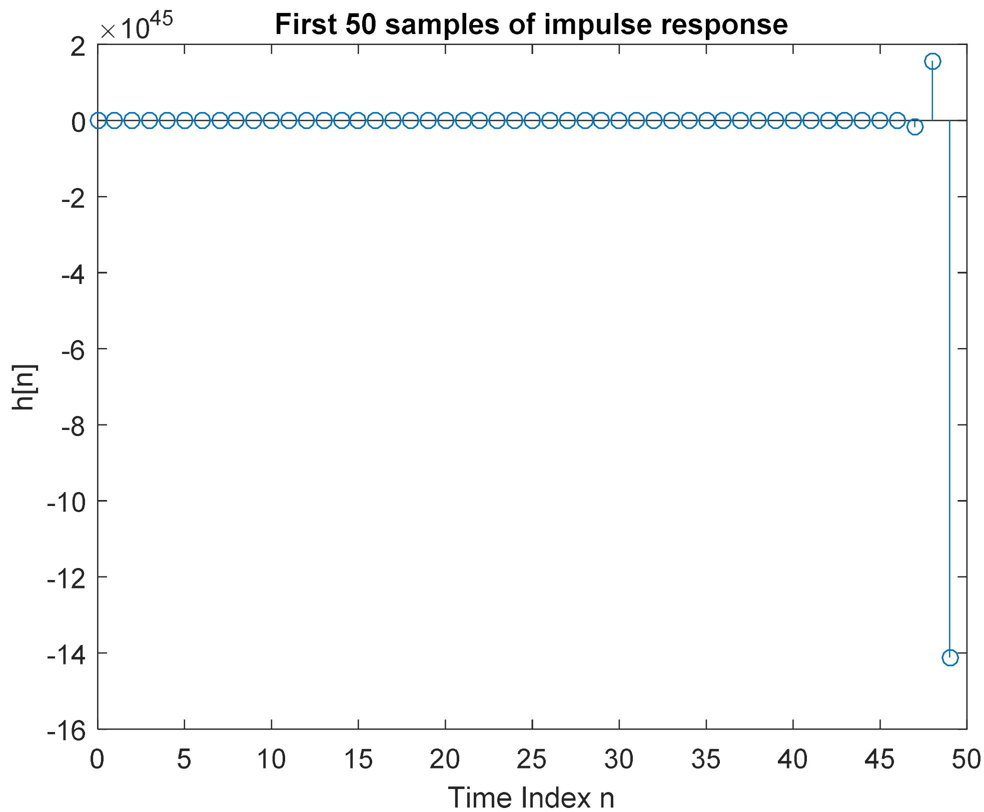
$$H(z) = \frac{3.9 - 9.36z^{-1} - 0.663z^{-2} - 1.014z^{-3} + 0.585z^{-4}}{1 - 0.95z^{-1} + 0.175z^{-2} + 0.6625z^{-3} - 0.3187z^{-4}}$$

PROJECT 3.6 INVERSE Z-TRANSFORM

Q3.50 The MATLAB program to compute the first L samples of the inverse of a rational z-transform is given below:

```
clf; clear all; close all;
% initialize
num = [2 5 9 5 3];
den = [5 45 2 1 1];
% Query user for parameter L
L = input('L = ? ');
% find impulse response
[g t] = impz(num,den,L);
%plot the impulse response
stem(t,g);
title(['First ',num2str(L),' samples of impulse
response']);
xlabel('Time Index n');
ylabel('h[n]')
```

The plot of the first 50 samples of the inverse of G(z) of Q3.46 obtained using this program is sketched



below:

Q3.51 The MATLAB program to determine the partial-fraction expansion of a rational z-transform is given below:

```
clf; clear all; close all;
% initialize
num = [2 5 9 5 3];
den = [5 45 2 1 1];
% partial fraction expansion
[r p k] = residuez(num,den)
```

The partial-fraction expansion of G(z) of Q3.46 obtained using this program is shown below:

```
r =
   0.3109 + 0.0000i
  -1.0254 - 0.3547i
  -1.0254 + 0.3547i
  -0.8601 + 0.0000i
p =
  -8.9576 + 0.0000i
   0.1147 + 0.2627i
   0.1147 - 0.2627i
  -0.2718 + 0.0000i
```

```
k =
    3.0000
```

$$H\left(z\right)=3+\frac{0.3109}{1+8.9576z^{-1}}+\frac{-1.0254-0.3547i}{1-\left(0.1147+0.2627i\right)z^{-1}}+\frac{-1.0254+0.3547i}{1-\left(0.1147-0.2627i\right)z^{-1}}+\frac{-0.8601}{1+0.2718z^{-1}}$$

From the above partial-fraction expansion we arrive at the inverse z-transform g[n] as shown below:

Date: 1/11/2023 **Signature: Do Trung Hau**