Amrita School of Engineering, Bengaluru
Department of Electronics and Communication Engineering
B.Tech. in Electronics and Communication Engineering

Report-2 for DSP LAB

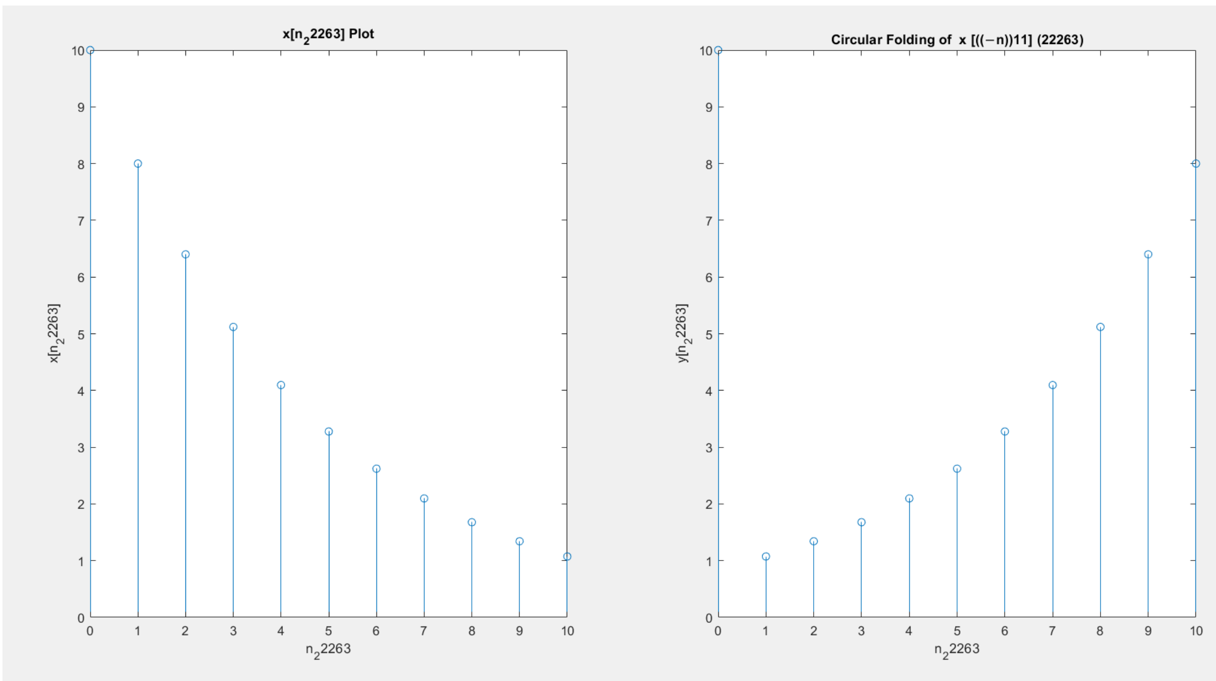
By

Abishai G

BL.EN.U4ECE22263

19ECE284/Digital Signal Processing Lab

IV Semester
Electronics and Communication Engineering


x[n_2263] Plot
x[n_2263]
n_2263
Circular Folding of x [((−n))11] (22263)
y[n_2263]
n_2263

Date:4/3/2024

Work Sheet No. 4

Title:Circular Folding, Circular Shifting, and Circular Convolution

Aim:

To realise the following operations in MATLAB

1. Circular Folding
2. Circular Shifting
3. Circular Convolution

Question:

1. Consider the signal $x[n] = 10(0.8)\ n$ defined in the interval $0 \leq n \leq 10$.

(a) Write a MATLAB function called cfold_ROLLNO to implement the circular folding operation (Hint: Take x[n], n and N as inputs).

(b) Using the function defined in part (a), determine and plot x [((−n))11]

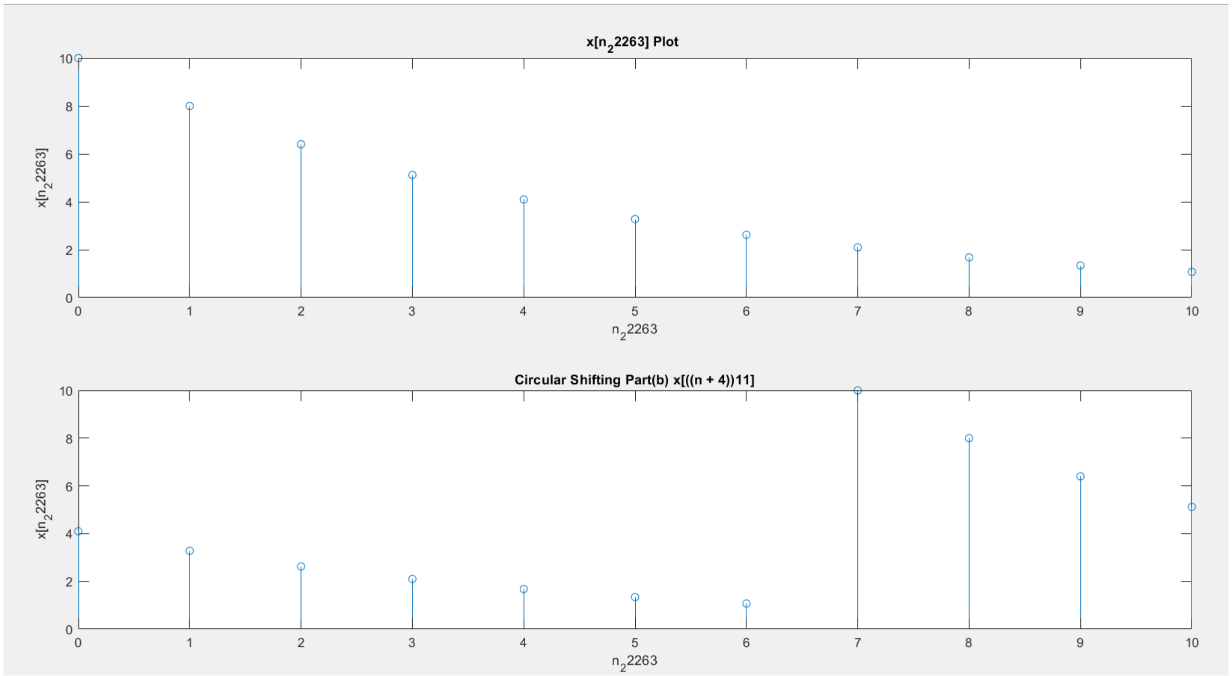
Code:

(a) **Function:**

```
function y_22263=cfold_22263(x,n,N)
y_22263=zeros(1,N);

for i_22263=1:length(x)
    y_22263(i_22263)=x(mod(-n(i_22263),N)+1);%since matrix cannot start with index
0
end
end
```

(b)

```
n_22263=0:10;
x_22263=10*(0.8).^n_22263;

y_22263=cfold_22263(x_22263,n_22263,11);

subplot(1,2,1);
stem(n_22263,x_22263);
xlabel('n_22263')
ylabel('x[n_22263]')
title('x[n_22263] Plot')

subplot(1,2,2);
stem(n_22263,y_22263);
xlabel('n_22263')
ylabel('y[n_22263]')
title('Circular Folding of  x [((-n))11] (22263)')
```


x[n_2263] Plot
x[n_2263]
n_2263
Circular Shifting Part(b) x[((n + 4))11]
x[n_2263]
n_2263

Question 2:

2. For the signal $x[n] = 10(0.8)\ n$ defined in the interval $0 \le n \le 10$,

(a) Write a MATLAB function called cshift_ROLLNO to implement circular shifting operation $x[(n - n0)N]$ (Hint: Take x[n], n, $n0$ and N as inputs)

(b) Using the function defined in part (a), Sketch x[((n + 4))11], that is, a circular shift by 4 samples toward the left.

(c) Sketch x[((n − 3))15], that is, a circular shift by 3 samples toward the right, where x[n] is assumed to be a 15-point sequence given by $x[n] = \{\ 10(0.8)\ n$ for $0 \le n \le 10$ and 0 for $11 \le n \le 14$.

(d) Sketch x[((n − 6))15], that is, a circular shift by 3 samples toward the right, where x[n] is assumed to be a 15-point sequence defined in part (c).

Code:

Part (a):

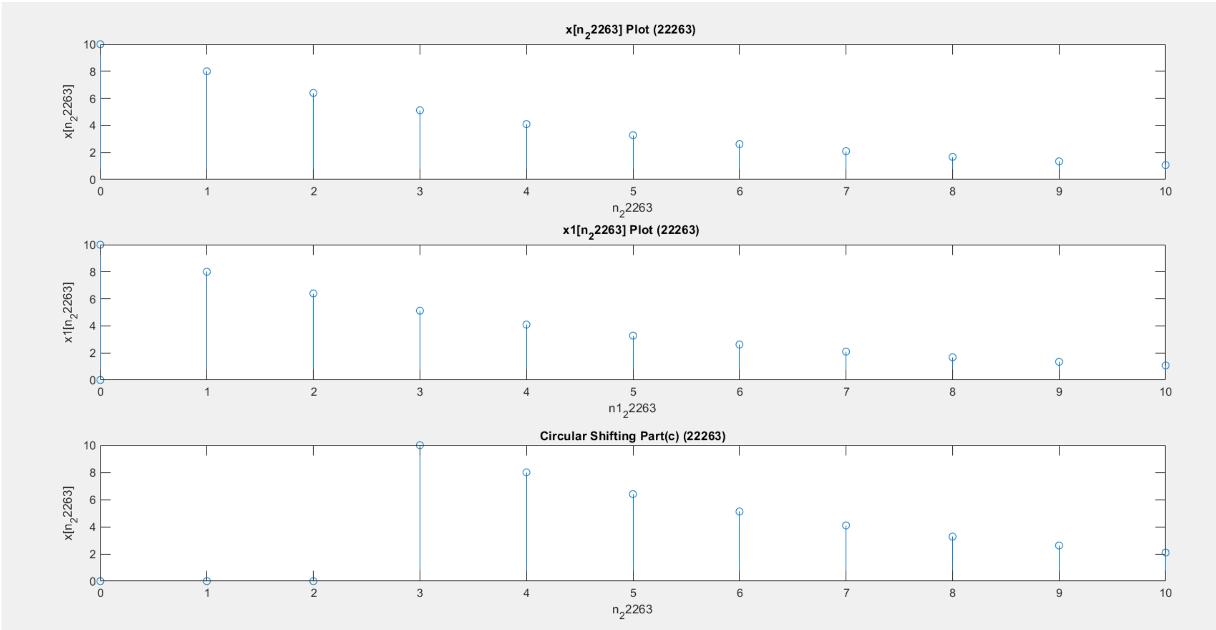
Function:

```
function y_22263=cshift_22263(x,n,N,no)
y_22263=zeros(1,N);

for i_22263=1:length(x)
    y_22263(i_22263)=x(mod(n(i_22263)-no,N)+1);
end
end
```

Part (b)

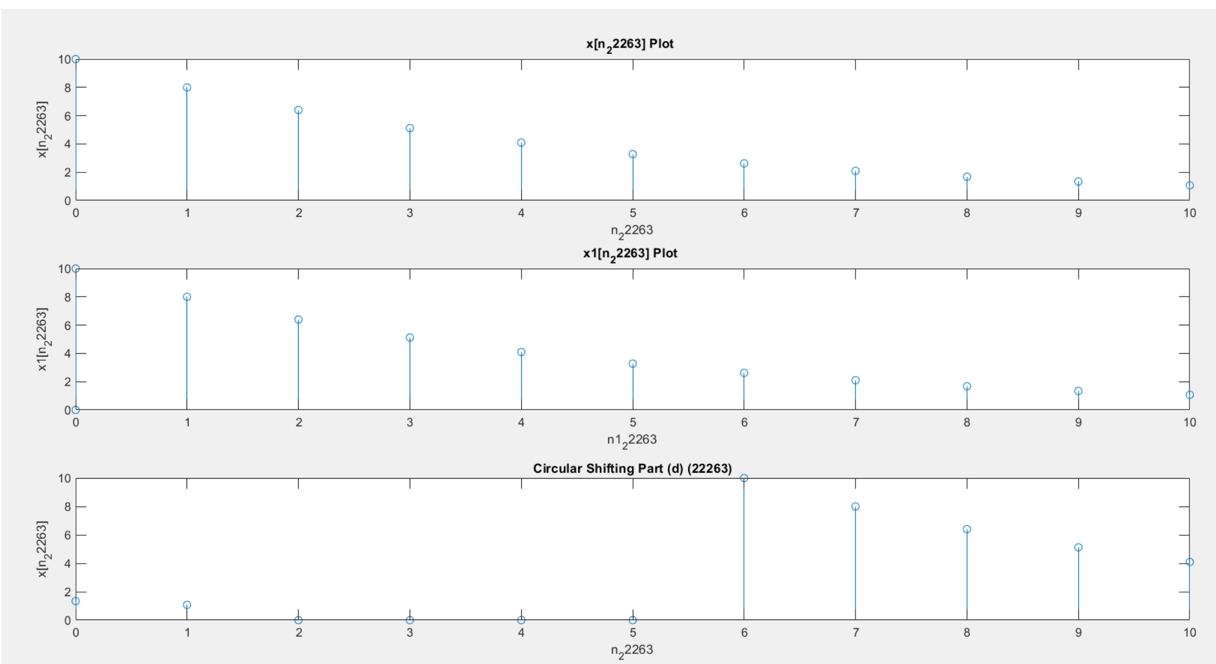
```
n_22263=0:10;
x_22263=10*(0.8).^n_22263;
no_22263=-4;
y_22263=cshift_22263(x_22263,n_22263,11,no_22263);

subplot(2,1,1);
stem(n_22263,x_22263);
xlabel('n_22263')
ylabel('x[n_22263]')
title('x[n_22263] Plot ')

subplot(2,1,2);
stem(n_22263,y_22263);
xlabel('n_22263')
ylabel('x[n_22263]')
title('Circular Shifting Part(b) x[((n + 4))11]')
```

Part(C)



x[n_2263] Plot (22263)
x[n_2263]
n_2263
x1[n_2263] Plot (22263)
x1[n_2263]
n1_2263
Circular Shifting Part(c) (22263)
x[n_2263]
n_2263

Part(d)



x[n_2263] Plot
x[n_2263]
n_2263
x1[n_2263] Plot
x1[n_2263]
n1_2263
Circular Shifting Part (d) (22263)
x[n_2263]
n_2263

Part(c)

```
n_22263=0:10;
x_22263=10*(0.8).^n_22263;
x1_22263=[x_22263,zeros(1,4)];
n1_22263=[n_22263,zeros(1,4)];
no_22263=3;
y_22263=cshift_22263(x1_22263,n1_22263,15,no_22263);

subplot(3,1,1);
stem(n_22263,x_22263);
xlabel('n_22263')
ylabel('x[n_22263]')
title('x[n_22263] Plot (22263) ')

subplot(3,1,2);
stem(n1_22263,x1_22263);
xlabel('n1_22263')
ylabel('x1[n_22263]')
title('x1[n_22263] Plot (22263) ')

subplot(3,1,3);
stem(n1_22263,y_22263);
xlabel('n_22263')
ylabel('x[n_22263]')
title('Circular Shifting Part(c) (22263)')
```

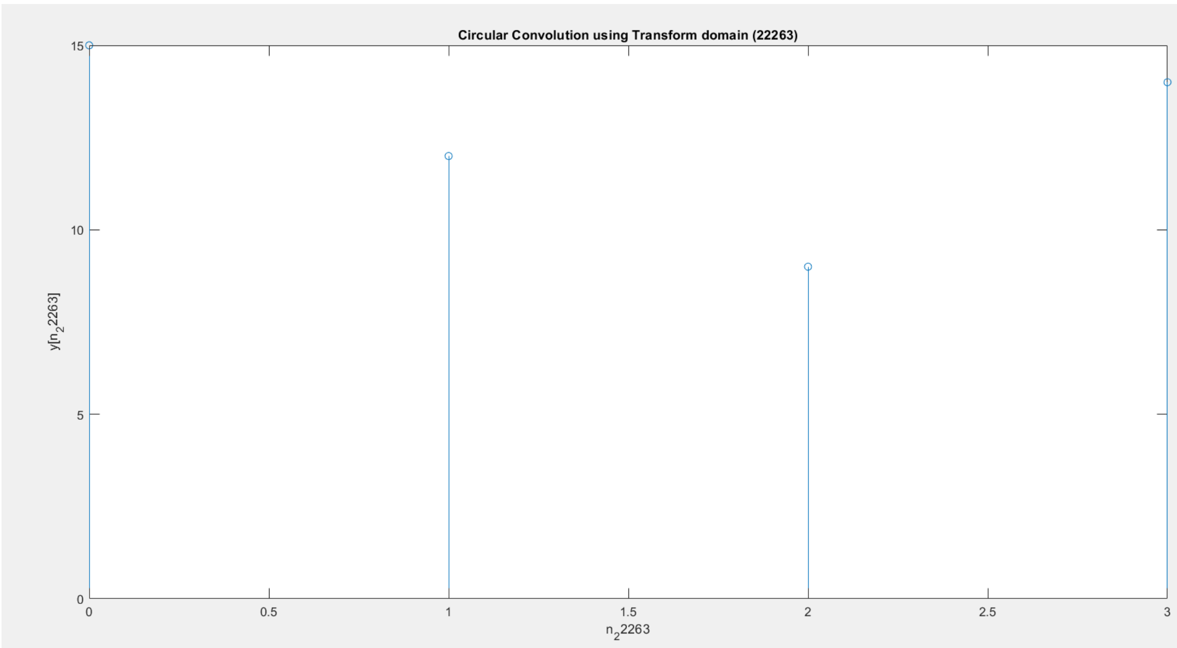
Part(d)

```
n_22263=0:10;
x_22263=10*(0.8).^n_22263;
x1_22263=[x_22263,zeros(1,4)];
n1_22263=[n_22263,zeros(1,4)];
no_22263=6;
y_22263=cshift_22263(x1_22263,n1_22263,15,no_22263);

subplot(3,1,1);
stem(n_22263,x_22263);
xlabel('n_22263')
ylabel('x[n_22263]')
title('x[n_22263] Plot ')

subplot(3,1,2);
stem(n1_22263,x1_22263);
xlabel('n1_22263')
ylabel('x1[n_22263]')
title('x1[n_22263] Plot ')

subplot(3,1,3);
stem(n1_22263,y_22263);
xlabel('n_22263')
ylabel('x[n_22263]')
title('Circular Shifting Part (d) (22263)')
```

Choice entering:

```
>> Q3
Enter your choice = 0
>>
```

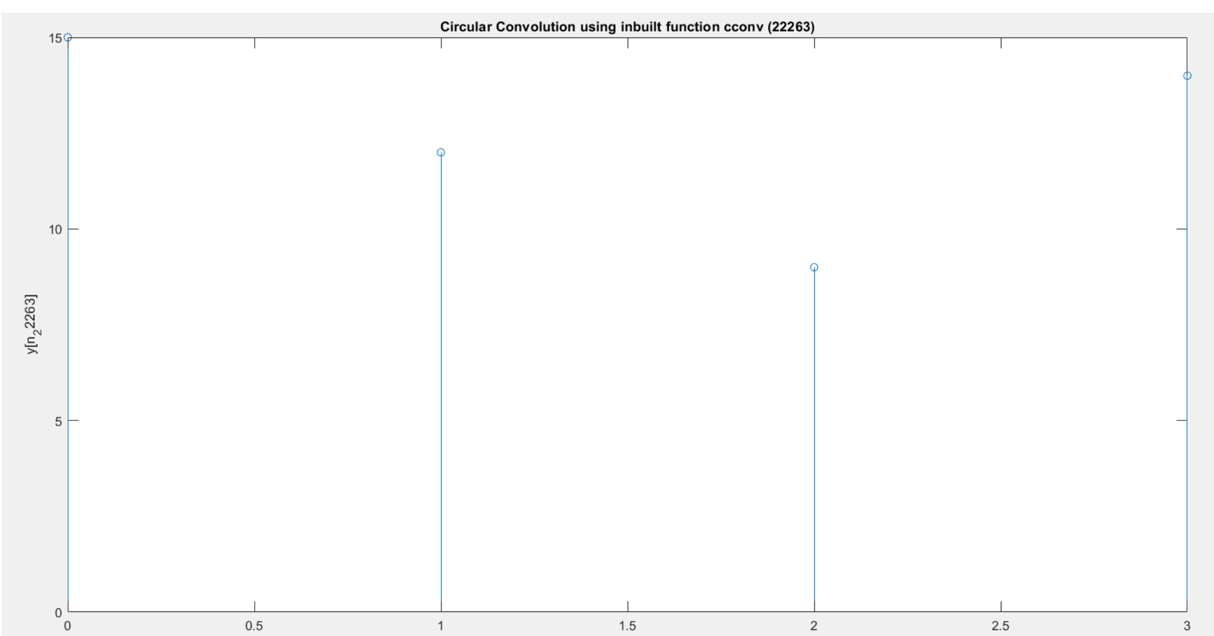
Using Transform Domain:



Choice Entering:

```
>> Q3
Enter your choice = 1
>>
```

Using inbuilt cconv function:



Question 3

There are two ways to write a function for circular convolution:

(a). In the transform domain (The frequency-domain implementation uses the fact that the circular convolution of x[n] and h[n] is equivalent to the multiplication of their DFTs, i.e., Y [k] = X[k].H[k] and take the inverse DFT of Y [k]).

(b). In the time domain (Using the inbuilt function cconv in MATLAB).

Develop a MATLAB function, called cconv_time_freq_ROLLNO that implements both methods. The MATLAB function will require four inputs: the signal vectors x[n] and h[n], the length of the circular convolution N, and a variable ind that indicates which method to use. The function returns a single output y[n]. Consider x[n] = {1, 2, 2,0} and h[n] = {1, 2, 3, 4}. Using the function cconv_time_freq_ROLLNO compute the circular convolution of x[n] and h[n] by both approaches and verify your results

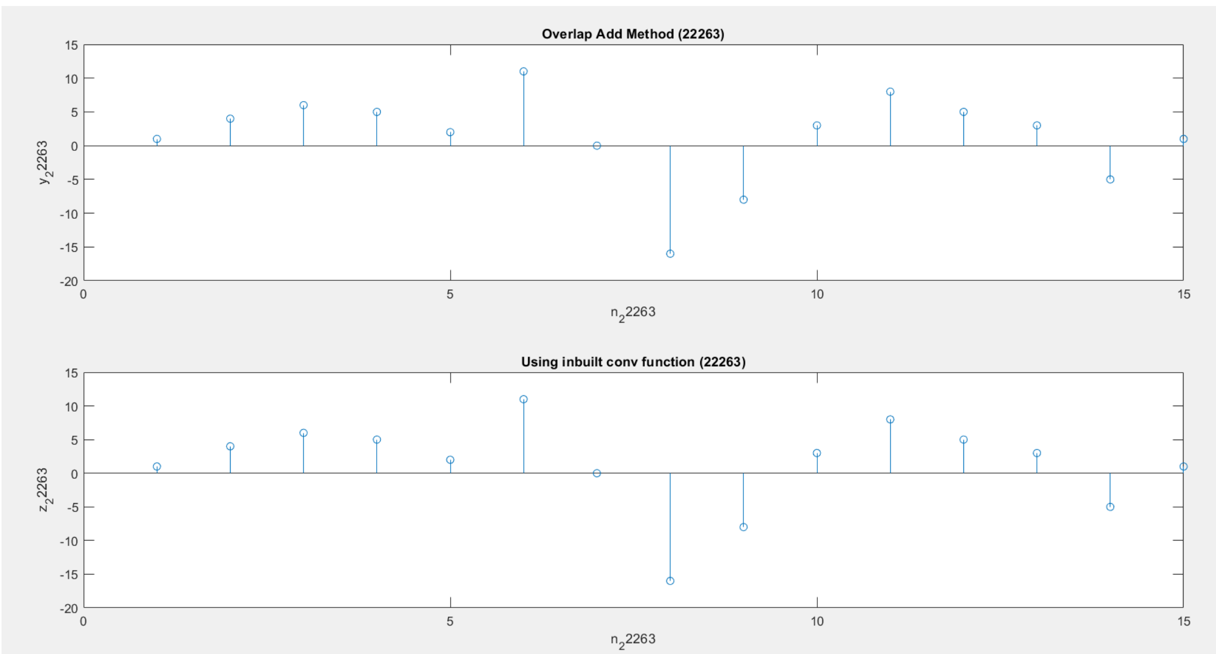
Code:

Function:

```
function y_22263=cconv_time_freq_22263(x_22263,h_22263,N_22263,ind_22263)
n_22263=0:N_22263-1;
if ind_22263==0
    X_22263=fft(x_22263,N_22263);
    H_22263=fft(h_22263,N_22263);

    Y_22263=X_22263.*H_22263;
    y_22263=ifft(Y_22263,N_22263);
    stem(n_22263,y_22263);
    xlabel('n_22263')
    ylabel('y[n_22263]')
    title('Circular Convolution using Transform domain (22263)')
elseif ind_22263==1
    y_22263=cconv(x_22263,h_22263,N_22263);
    stem(n_22263,y_22263);
    xlabel('n_22263')
    ylabel('y[n_22263]')
    title('Circular Convolution using inbuilt function cconv (22263)')

end

end
```

Main:

```
x_22263=[1,2,2,0];
h_22263=[1,2,3,4];

N_22263=4;

ind_22263=input('Enter your choice');

y_22263=cconv_time_freq_22263(x_22263,h_22263,N_22263,ind_22263);
```



Overlap Add Method (22263)
y22263
n22263
Using inbuilt conv function (22263)
z22263
n22263

Date:15/3/2024

Work Sheet No. 5

Title: Linear Filtering using Overlap add and Overlap save methods

Aim:

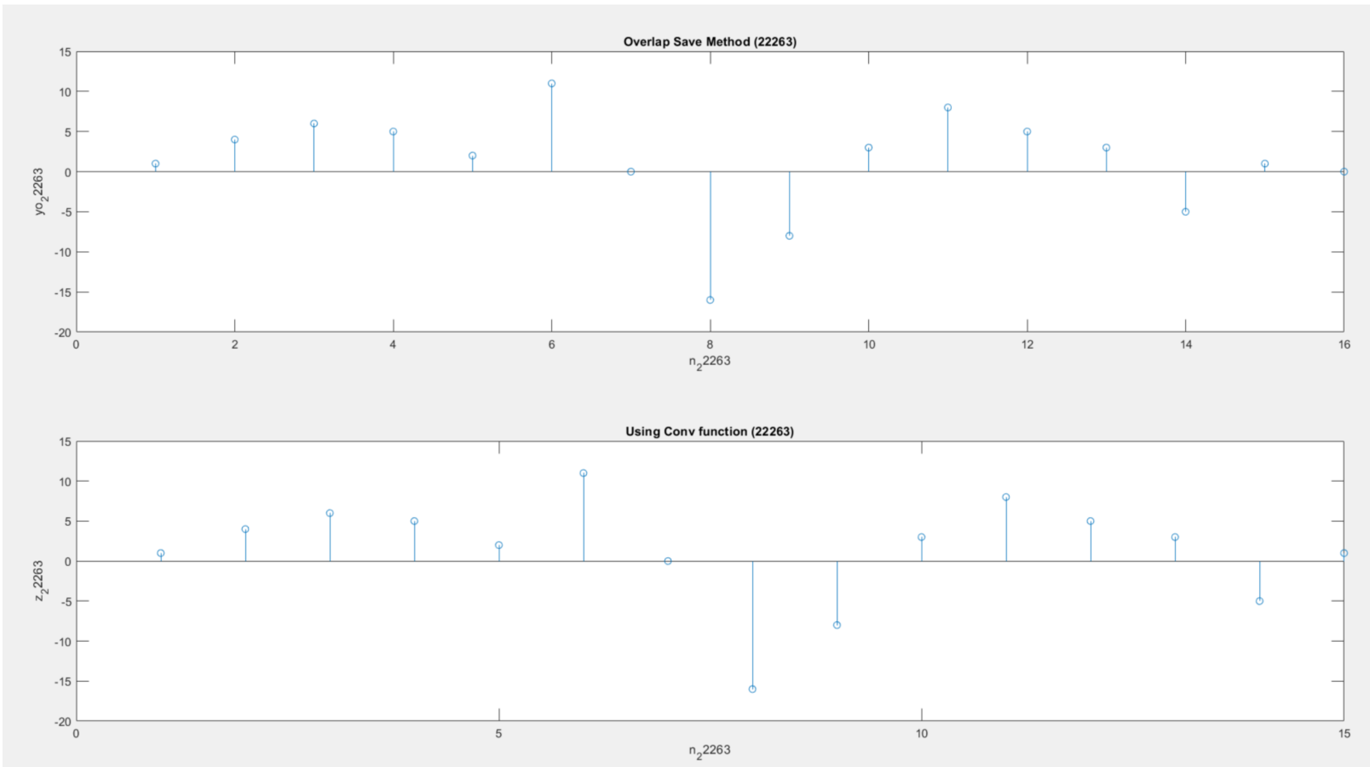
To perform block convolution on the given sequence by using overlap add/ save method in MATLAB

Questions:

1. Compute the block convolution using overlap add method for the input sequence x[n] = [1 2 -1 2 3 -2 -3 -1 1 1 2 -1] and the impulse response h[n] = [1 2 3 -1]. Assume input signal block size to be 4. Compare the obtained result with that obtained from the inbuilt function conv.

Code:

```
x_22263 = [1,2,-1,2,3,-2,-3,-1,1,1,2,-1];
h_22263 = [1,2,3,-1];
L_22263 = 4;
M_22263 = length(h_22263);
B_22263 = round(length(x_22263)/L_22263);
h1_22263 = [h_22263, zeros(1, L_22263-1)];
xi_22263 = zeros(B_22263, L_22263+M_22263-1);
yi_22263 = zeros(B_22263,L_22263+M_22263-1);
for i=1:B_22263
xi_22263(i,:) = [x_22263((i-1)*L_22263+1:i*L_22263), zeros(1,M_22263-1)];
yi_22263(i,:) = cconv(xi_22263(i,:), h1_22263,L_22263+M_22263-1);
end
y_22263 = [];
for i=1:B_22263
if(i==1)
y_22263 = [y_22263,yi_22263(i,1:L_22263), yi_22263(i,L_22263+1:L_22263+M_22263-
1)+yi_22263(i+1,1:M_22263-1)];
elseif(i==B_22263)
y_22263 = [y_22263,yi_22263(i,M_22263:L_22263+M_22263-1)];
else
y_22263 = [y_22263,yi_22263(i,M_22263:L_22263),
yi_22263(i,L_22263+1:L_22263+M_22263-1)+yi_22263(i+1,1:M_22263-1)];
end
end
subplot(2,1,1)
stem(y_22263)
xlabel('n_22263')
ylabel('y_22263')
title('Overlap Add Method (22263)')

z_22263=conv(x_22263,h_22263);

subplot(2,1,2)
stem(z_22263)
xlabel('n_22263')
ylabel('z_22263')
title('Using inbuilt conv function (22263)')
```



Overlap Save Method (22263)
yo_2 2263
n_2 2263
Using Conv function (22263)
z_2 2263
n_2 2263

Question 2.

Perform block convolution for the sequences given in Qn.1 using overlap save method with same input signal block size and verify the result

Code:

```
x_22263 = [1,2,-1,2,3,-2,-3,-1,1,1,2,-1];
h_22263 = [1,2,3,-1];
L_22263 = 4;
M_22263 = length(h_22263);
B_22263 = round(length(x_22263)/L_22263)+1;
h1_22263 = [h_22263, zeros(1,L_22263-1)];
x1_22263 = zeros(B_22263,L_22263+M_22263-1);
y1_22263 = zeros(B_22263, L_22263+M_22263-1);
for i=1:B_22263
if(i==1)
x1_22263(i,:) = [zeros(1, M_22263-1), x_22263(1:L_22263)];
elseif(i==B_22263)
x1_22263(i,:) = [x1_22263(i-1,L_22263+1:L_22263+M_22263-1), zeros(1,L_22263)];
else
x1_22263(i,:) = [x1_22263(i-1,L_22263+1:L_22263+M_22263-1),x_22263((i-
1)*L_22263+1:i*L_22263)];
end
y1_22263(i,:) = cconv(x1_22263(i,:), h1_22263, L_22263+M_22263-1);
end
yo_22263 = [];
for i=1:B_22263
yo_22263 = [yo_22263,y1_22263(i,M_22263:L_22263+M_22263-1)];
end

subplot(2,1,1)
stem(yo_22263);
xlabel('n_22263')
ylabel('yo_22263')
title('Overlap Save Method (22263)')

z_22263=conv(x_22263,h_22263);

subplot(2,1,2)
stem(z_22263);
xlabel('n_22263')
ylabel('z_22263')
title('Using Conv function (22263)')
```