

DSP Lab - Assignment 4

Name - Subroto Banerjee

Roll - ECE/22160 [1014]

Sub - DSP Lab

QUESTIONS:

1. Generate two discrete-time sequences $x_1[n]$ and $x_2[n]$ by taking sample inputs from the user. Compute the Circular Convolution of $x_1[n]$ and $x_2[n]$ using the Circulant Matrix method and store the result in $y_1[n]$. Plot $y_1[n]$.
2. For sequences $x_1[n]$ and $x_2[n]$ above, compute the Circular Convolution $y_2[n]$ by multiplying their respective DFTs and taking the IDFT (you may use the matrix method to perform DFT/IDFT operations for example). Verify whether $y_2[n] = y_1[n]$.

Answers:

1) Code:

```
clc; clear; close all;
x1 = input('Enter the first sequence x1[n] (e.g., [1 2 3]): ');
x2 = input('Enter the second sequence x2[n] (e.g., [4 5 6]): ');

N = max(length(x1), length(x2));
x1 = [x1(:); zeros(N - length(x1), 1)];
x2 = [x2(:); zeros(N - length(x2), 1)];
disp('x1[n] after zero padding:');
disp(x1. ');
disp('x2[n] after zero padding:');
disp(x2. ');

circ_matrix = zeros(N, N);
for i = 1:N
    circ_matrix(:, i) = circshift(x1, i - 1);
end

y1 = circ_matrix * x2;
disp('y1[n] Using Circulant Matrix:');
disp(y1. ');

figure;
stem(0:N-1, x1, 'filled');
xlabel('n'); ylabel('x_1[n]');
title('Input x_1 Sequence');
```

```

grid on;
figure;
stem(0:N-1, x2, 'filled');
xlabel('n'); ylabel('x_2[n]');
title('Input x_2 Sequence');
grid on;

figure;
stem(0:N-1, y1, 'filled');
xlabel('n'); ylabel('y_1[n]');
title('Circular Convolution via Circulant Matrix');
grid on;

```

Output Plots:

- Command Window:

```

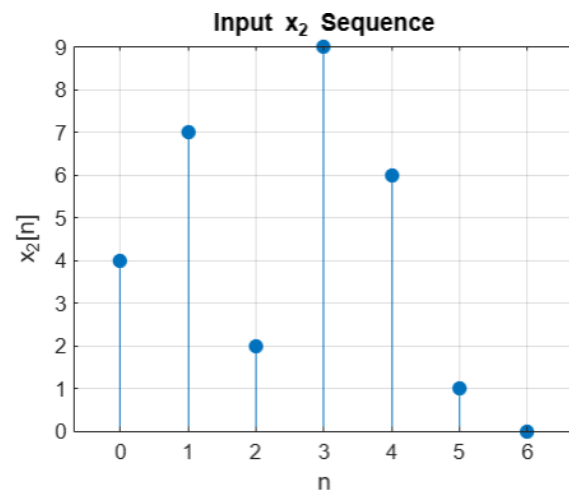
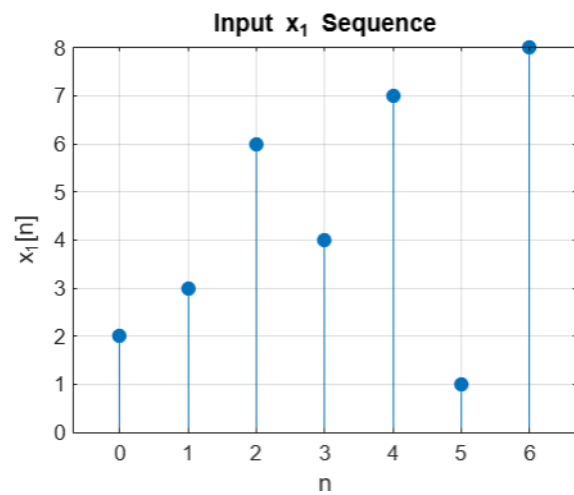
Command Window
Enter the first sequence x1[n] as a vector (e.g., [1 2 3]):
[2 3 6 4 7 1 8]
Enter the second sequence x2[n] as a vector (e.g., [4 5 6]):
[4 7 2 9 6 1]
x1[n] after zero padding:
    2    3    6    4    7    1    8

x2[n] after zero padding:
    4    7    2    9    6    1    0

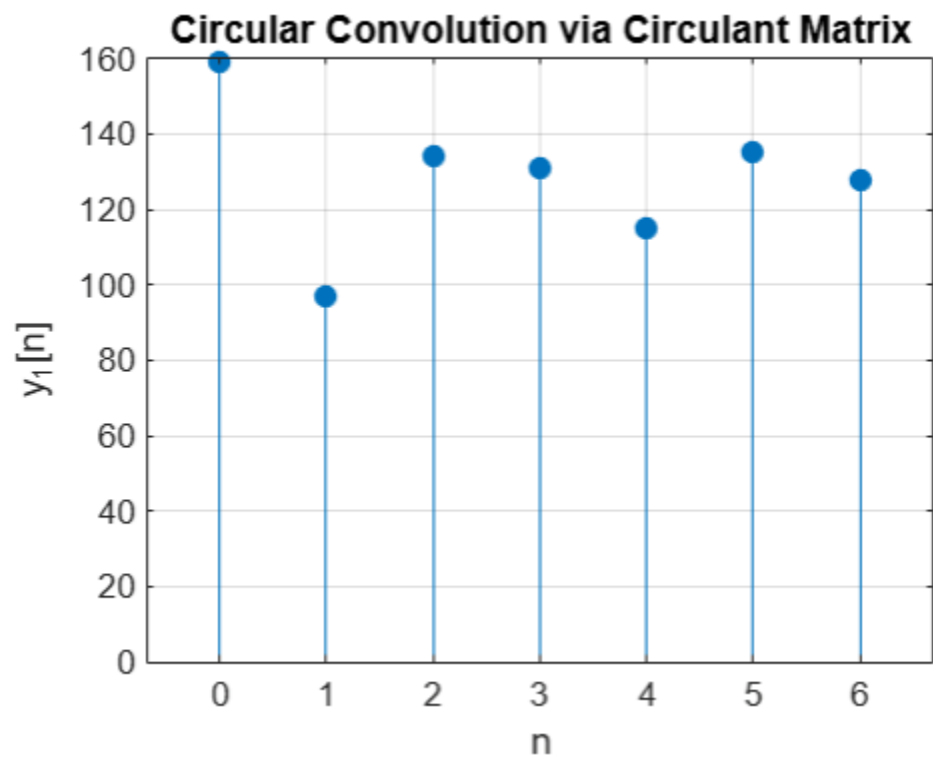
y1[n] Using Circulant Matrix:
   159    97   134   131   115   135   128

```

- x1 and x2 input plot:



- **Output Plot:**



2) Code:

```
clc; clear; close all;
x1 = input('Enter the first sequence x1[n] (e.g., [1 2 3]): ');
x2 = input('Enter the second sequence x2[n] (e.g., [4 5 6]): ');

N = max(length(x1), length(x2));
x1 = [x1(:); zeros(N - length(x1), 1)];
x2 = [x2(:); zeros(N - length(x2), 1)];
disp('x1[n] after zero padding:');
disp(x1. ');
disp('x2[n] after zero padding:');
disp(x2. ');

circ_matrix = zeros(N, N);
for i = 1:N
    circ_matrix(:, i) = circshift(x1, i - 1);
end
y1 = circ_matrix * x2;

W = exp(-1j * 2 * pi / N);
DFT_matrix = zeros(N, N);
for k = 0:N-1
    for n = 0:N-1
        DFT_matrix(k+1, n+1) = W^(k * n);
    end
end

X1 = DFT_matrix * x1;
X2 = DFT_matrix * x2;
Y_freq = X1 .* X2;
```

```

IDFT_matrix = conj(DFT_matrix) / N;

y2 = IDFT_matrix * Y_freq;
disp('y1[n] from circulant matrix method:');
disp(y1.');
disp('y2[n] from DFT-IDFT method:');
disp(real(y2.'));

if max(abs(y1 - y2)) < 1e-10
    disp('y1[n] and y2[n] are equal (within numerical tolerance).');
else
    disp('y1[n] and y2[n] are NOT equal.');
```

```

end

figure;
stem(0:N-1, y1, 'filled');
xlabel('n'); ylabel('y_1[n]');
title('Circular Convolution via Circulant Matrix');
grid on;

figure;
stem(0:N-1, real(y2), 'filled');
xlabel('n'); ylabel('y_2[n]');
title('Convolution via DFT/IDFT');
grid on;
```

Output Plots:

- **Command Window:**

```

Command Window
Enter the first sequence x1[n] as a vector (e.g., [1 2 3]):
[2 3 6 4 7 1 8]
Enter the second sequence x2[n] as a vector (e.g., [4 5 6]):
[4 7 2 9 6 1]
x1[n] after zero padding:
    2    3    6    4    7    1    8

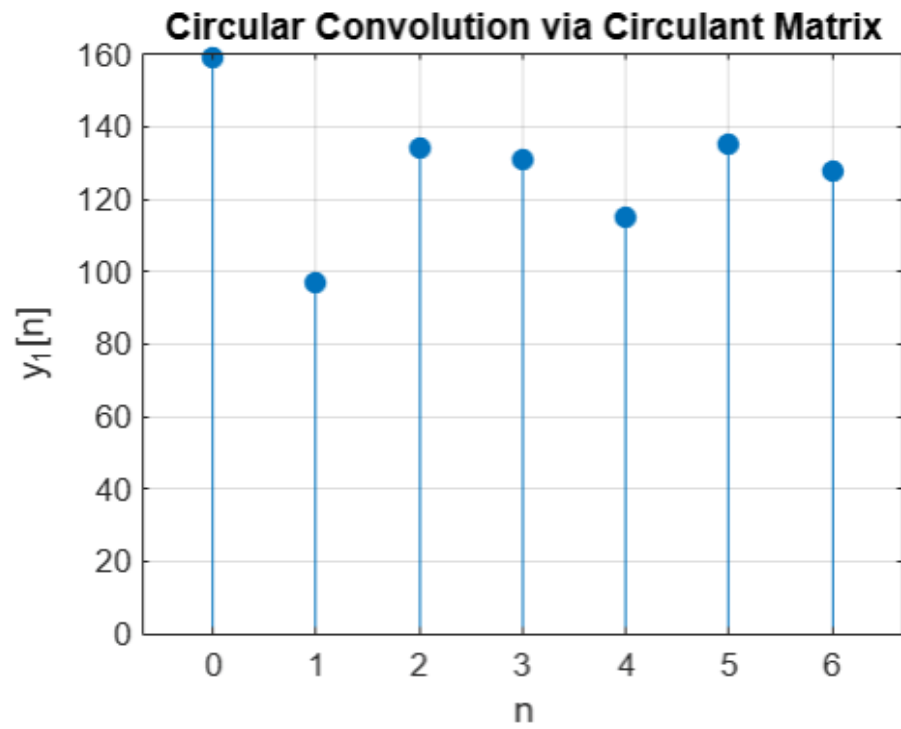
x2[n] after zero padding:
    4    7    2    9    6    1    0

y1[n] from circulant matrix method:
   159    97   134   131   115   135   128

y2[n] from DFT-IDFT method:
  159.0000  97.0000  134.0000  131.0000  115.0000  135.0000  128.0000

y1[n] and y2[n] are equal (within numerical tolerance).
```

- $y_1[n]$ - via Circulant Matrix:



- $y_2[n]$ - via DFT/IDFT:

