Digital Signal Processing [Lab-6]

Table of Contents

- Authors: *Kshitij Srivastava(1510110200)* and *Nilambar Saha(partner)(1510110246)*
- Lab Instructor: *Dr. Ravi Kant Saini*
- Branch and Section: ECE32 Friday- 1pm to 3pm
- Email-Id: ks435@snu.edu.in

Objective:

Discrete Fourier Transform (In this lab, inputs and impulse reponse were converted to Discrete fourier transform by using the basis vectors and then output was found using Inverse DFT. Orthogonality property of the basis vectors created was also checked)

Program:

```
clc;
clear all;
close all;

% * |*Matlab Commands for finding the basis vectors for N=4*|
D=zeros(4,4);%Initializing the D matrix to zero

for i=1:4 %looping through the n in the sk[n]
    for j=1:4 %looping through the k in the sk[n]
        number=((2*pi*(i-1)*(j-1))/4);
       D(i,j)=exp( 1j*number );%Finding the number and placing in (i,j)
       %position
    end
end
```

- **Matlab Commands to see that basis vectors are orthogonal to other vectors**

```
D_har=zeros(4,4);%initializing the D_har matrix to zero

for i=1:4 %looping through the n in the sk[n]
    for j=1:4 %looping through the k in the sk[n]
        D_har(i,j)=conj(D(i,j)); %Finding the conjugate of the D Matrix
    end
```

```
end

ortho=D*transpose(D_har);%Finding the orthogonality property
ortho=ortho/4; %Dividing it by N=4 to get identity matrix
ortho=round(ortho);%rounding off the orthogonal matrix
```

- **Matlab Commands to verify-inverse of D matrix gives Hermitian of D**

```
identity=D*D_har;%Multiplying D and D Hermitian to get NI
identity=identity/4;%Dividing by N=4 to get identity matrix
round(identity);%Rounding the identity matrix
```

- **Matlab Commands for Computing the DFT and IDFT**

```
x=[1,2,3,4];%N=4 length input sequence
X=D_har*transpose(x);%Computing the DFT of the x[n]

x_found=D*X;%Computing the inverse DFT of X
x_found=x_found/4;%Dividing by N=4 to get inverse DFT
```

- **Matlab Commands for computing the y[n] from Fourier Transform**

```
x=[1,2,3,4];%N=4 length input sequence
h=[0,1,0,0];% impulse response sequence

y=conv(x,h);%Finding the convolution or output response of the system

X=D_har*transpose(x);%Computing the DFT of the x[n]
H=D_har*transpose(h);%Computing the DFT of the h[n]

Y_found=X.*H;% Element by element multiplication of X and Hs
yout=D*Y_found;%Computing the inverse DFT of Y to compute y
yout=yout/4;%Dividing it by N=4 to get y
```

- **Matlab Commands for computing the DFT from a given input data**

```
load('inputData');%Loading the input Data
load('h1.mat');%Loading the h1 impulse response
load('h2.mat');%Loading the h2 impulse response

%Finding the basis vectors for N=50
for i=1:50 %Looping through 1 to 50
    for j=1:50 %Looping through 1 to 50
        number=((2*pi*(i-1)*(j-1))/4);
        D_mat(i,j)=exp(  1j*number );%Finding the value and putting it (i,j)
       %position
    end
end

%Finding the conjugate of the basis vector for N=50
for i=1:50 %Looping through 1 to 50
    for j=1:50 %Looping through 1 to 50
        D_mat_har(i,j)=conj(D_mat(i,j)); %Finding the conjugate stored in
        %D_mat (i,j)position
    end
```

```
end

H1=D_mat_har*transpose(h1);%Finding the DFT of the h1 impulse response
H2=D_mat_har*transpose(h2);%Finding the DFT of the h2 impulse response

X_input=D_mat_har*transpose(inputData);%Finding the DFT of input x[n]

Y1=X_input.*H1;%Element by element multiplication of X and H1
Y2=X_input.*H2;%Element by element multiplication of X and H2

y1_found=D_mat*Y1;%Computing the inverse DFT of Y1 to compute y1
y1_found=y1_found/4;%Dividing it by N=50 to get y1
y1_abs=abs(y1_found);%Getting the absolute value of the y1

y2_found=D_mat*Y2;%Computing the inverse DFT of Y2 to compute y2
y2_found=y2_found/4;%Dividing it by N=50 to get y2
y2_abs=abs(y2_found);%Getting the absolute value of y2
```

Results:

- **Result for Q1 showing fourier transformation matrix**

```
D


        D =

           1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.000
           1.0000 + 0.0000i   0.0000 + 1.0000i  -1.0000 + 0.0000i  -0.0000 - 1.000
           1.0000 + 0.0000i  -1.0000 + 0.0000i   1.0000 - 0.0000i  -1.0000 + 0.000
           1.0000 + 0.0000i  -0.0000 - 1.0000i  -1.0000 + 0.0000i   0.0000 + 1.000
```

- **Result for Q2 showing the orthogonal property**

```
%Zero indicates the multiplication of basis vector with other basis vector
%One indicates the multilpication of basis vector with itself
ortho


        ortho =

             1     0     0     0
             0     1     0     0
             0     0     1     0
             0     0     0     1
```

- **Result for Q3 showing that inverse of D matrix gives hermitian of D**

```
identity
```

```
identity =

   1.0000 + 0.0000i  -0.0000 - 0.0000i   0.0000 - 0.0000i   0.0000 - 0.000
  -0.0000 + 0.0000i   1.0000 + 0.0000i  -0.0000 - 0.0000i   0.0000 - 0.000
   0.0000 + 0.0000i  -0.0000 + 0.0000i   1.0000 - 0.0000i  -0.0000 - 0.000
   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0000 + 0.0000i   1.0000 + 0.000
```

- **Result for Q4 computing the DFT of x[n]**

X

```
X =

  10.0000 + 0.0000i
  -2.0000 + 2.0000i
  -2.0000 - 0.0000i
  -2.0000 - 2.0000i
```

- **Result for Q4 computing IDFT of X**

x_found

```
x_found =

   1.0000 - 0.0000i
   2.0000 - 0.0000i
   3.0000 - 0.0000i
   4.0000 + 0.0000i
```

- **Result for Q5 computing the normal convolution**

y

```
y =

     0     1     2     3     4     0     0
```

- **Result for Q5 computing the IDFT of calculated y[n] from Y**

yout

```
yout =

   4.0000 + 0.0000i
   1.0000 - 0.0000i
   2.0000 - 0.0000i
```

```
        3.0000 + 0.0000i
```

- **Observation for Q5 about sequence obtained from normal convolution for y[n] and y[n] from DFT**

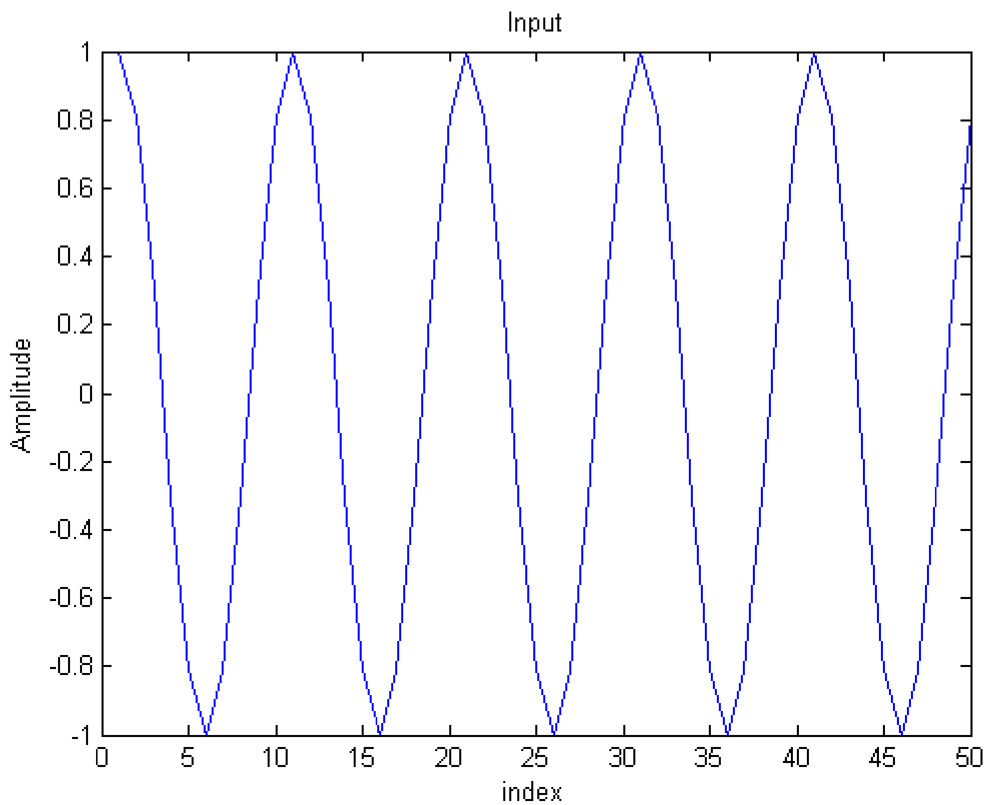
```
%The y[n] got from the normal convolution has support of 7 while the y[n]
%got from DFT has the support of 4. Also the output got from the DFT is
%circular shifted version of output.
```

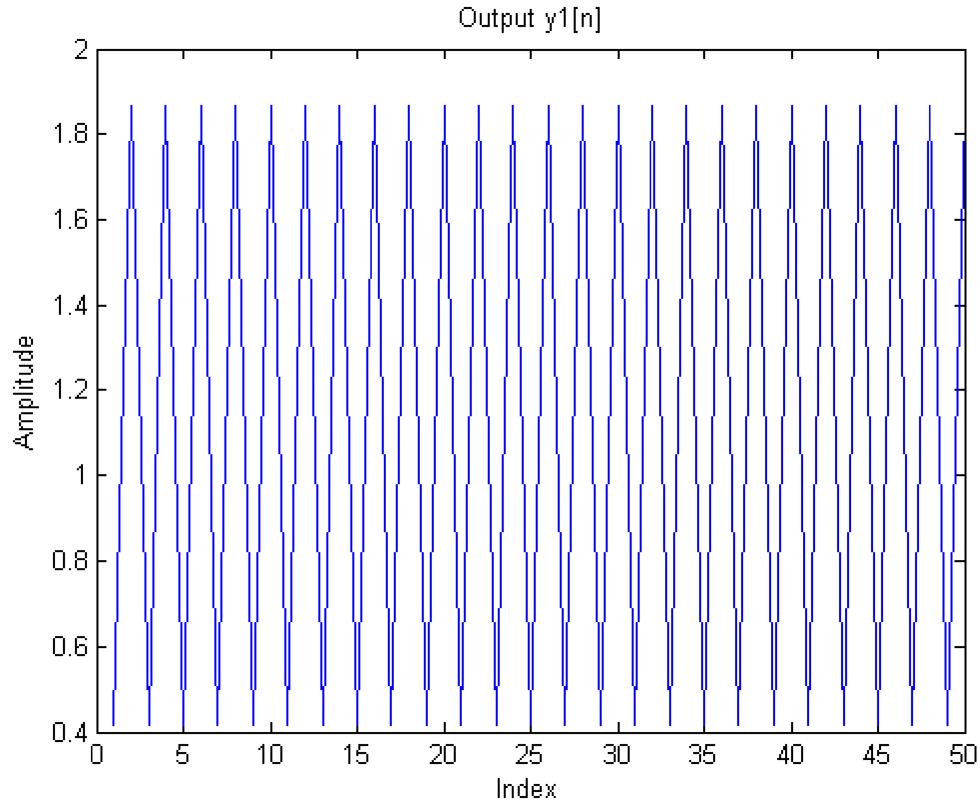
- **Plot for Q6 showing the input x[n]**

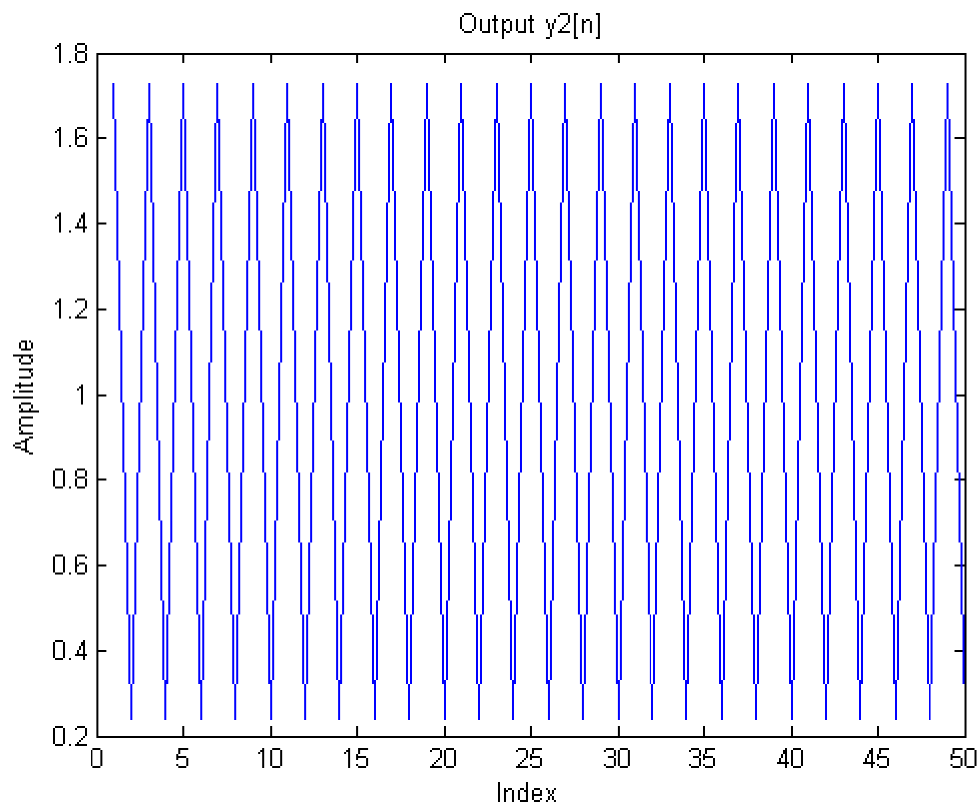
```
figure;plot(inputData);
title('Input');xlabel('index');ylabel('Amplitude');

% * |*Plot for Q6 showing the output's absolute value y1[n]*|
figure;plot(y1_abs);
title('Output y1[n]');xlabel('Index');ylabel('Amplitude');

% * |*Plot for Q6 showing the output's absolute value y2[n]*|
figure;plot(y2_abs);
title('Output y2[n]');xlabel('Index');ylabel('Amplitude');
```




Output y1[n]
Amplitude
Index
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0
5
10
15
20
25
30
35
40
45
50


Output y2[n]
Amplitude
Index
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
5
10
15
20
25
30
35
40
45
50

- **Observation for Q6 about signal obtained y1[n] and y2[n]**

```
%The impulse response h1 and h2 increase the frequency of input x[n]. So
%the number of oscillation has increased very much while the there has also
%been a change in the amplitude of the signals
```

Published with MATLAB® R2014a