Experiment No: 7 Date: 01/10/24

Properties of DFT

Aim:

Verify following properties of DFT using Matlab/Scilab.

1.Linearity Property

2.Parsevals Theorem

3.Convolution Property

4.Multiplication Property

Theory:

1. Linearity Property

The linearity property of the DFT states that if you have two sequences x1[n] and x2[n], and their corresponding DFTs are X1[k] and X2[k], then for any scalar a and b:

$$\mathbf{DFT\{a \cdot x1[n] + b \cdot x2[n]\} = a \cdot DFT\{x1[n]\} + b \cdot DFT\{x2[n]\}}$$

2. Parseval's Theorem

Parseval's theorem states that the total energy of a signal in the time domain is equal to the total energy in the frequency domain. For a sequence x[n] and its DFT X[k]:

$$\sum_{n=0}^{N-1} x[n]^2 = \frac{1}{N} \sum_{n=0}^{N-1} X[k]$$

3.Convolution Property

The convolution property of the DFT states that the circular convolution of two sequences in the time domain is equivalent to the element-wise multiplication of their DFTs in the frequency domain:

$$\mathbf{DFT\{x1[n] \circledast x2[n]\} = DFT\{x1[n]\} \cdot DFT\{x2[n]\}}$$

4. Multiplication Property

The multiplication property of DFT states that pointwise multiplication in the time domain corresponds to circular convolution in the frequency domain:

$$\mathbf{DFT\{x1[n] \cdot x2[n]\} = \frac{1}{N} DFT\{x1[n]\} \circledast DFT\{x2[n]\}}$$

Program:

1. Linearity Property

```
clc;
clear all; close
all;
x=input("enter first sequence");
h=input("enter sequence sequence:");
lx=length(x); lh=length(h); if lx>lh
h=[h zeros(1,lx-lh)] else     x=[x
zeros(1,lh-lx)] end
a=input("enter value of 'a':");
b=input("enter value of 'b':");
lhs=fft((a.*x)+(b.*h));
rhs=a.*fft(x)+b.*fft(h); disp('LHS');
disp(lhs); disp('RHS'); disp(rhs); if
lhs==rhs     disp('Linearity property
verified'); else
    disp('Linearity property not verified'); end
```

2. Parseval's Theorem `clc; clear all; close all; x=input("enter first`

```
sequence:");       h=input("enter
second               sequence:");
N=max(length(x),length(h)); xn=[x
zeros(1,N-length(x))];      hn=[h
zeros(1,N-length(h))];
lhs=sum(xn.*conj(hn));
rhs=sum(fft(xn).*conj(fft(hn)))/
N;     disp('LHS');     disp(lhs);
disp('RHS');     disp(rhs);     if
lhs==rhs
    disp("Parseval's Theorem verified"); else
    disp("Parseval's Theorem not verified"); end
```

3.Convolution Property

```
clc;  clear  all;
close   all;   x=input("enter   first
sequence");  h=input("enter  sequence
sequence:");        N=max(length(x),
length(h));      xn=[x      zeros(N-
length(x))];      hn=[h      zeros(N-
length(h))];
Xn=fft(xn); Hn=fft(hn);
lhs=cconv(xn,hn,N);
rhs=ifft(Xn.*Hn);
disp('LHS');
disp(lhs);
disp('RHS');
disp(rhs); if lhs==rhs
disp('Circular
Convolution verified')
else
```

```
disp('Circular
Convolution not
verified'); end
```

4. Multiplication Property

```
clc; clear all; close all; x=input("enter
first sequence"); h=input("enter sequence
sequence:"); N=max(length(x), length(h));
xn=[x zeros(N-length(x))]; hn=[h zeros(N-
length(h))]; lhs=fft(xn.*hn); Xn=fft(xn);
Hn=fft(hn); rhs=(cconv(Xn,Hn,N))/N;
disp('LHS'); disp(lhs); disp('RHS');
disp(rhs); if lhs==rhs
disp('Multiplication property verified');
else
    disp('Multiplication property not verified'); end
```

Result:

Performed and verified the following properties of DFT:

1.Linearity Property

2.Parsevals Theorem

3.Convolution Property

4.Multiplication Property

Observation:

1. Linearity Property enter first

sequence[1 2 3 4] enter sequence

sequence:[1 1 1 1] x =

1 2 3 4 enter

value of 'a':2 enter

value of 'b':3

LHS

32.0000 + 0.0000i -4.0000 + 4.0000i -4.0000 + 0.0000i -4.0000 - 4.0000i

RHS

32.0000 + 0.0000i -4.0000 + 4.0000i -4.0000 + 0.0000i -4.0000 - 4.0000i

Linearity property verified

2. Parseval's Theorem enter

first sequence:[1 2 3 4] enter

second sequence:[1 1 1 1]

LHS

10

RHS

10

Parseval's Theorem verified

3.Convolution Property enter

first sequence[1 2 3 4] enter

sequence sequence:[1 1 1 1]

LHS

10 10 10 10

RHS

10 10 10 10

Circular Convolution verified

4.Multiplication Property enter

first sequence[1 2 3 4] enter

sequence sequence:[1 1 1 1]

LHS

Columns 1 through 3

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

Column 4

-2.0000 - 2.0000i

RHS

Columns 1 through 3

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

Column 4

-2.0000 - 2.0000i

Multiplication property verified

Experiment No: 8 Date: 08/10/24 **OVERLAP ADD AND OVERLAP SAVE METHOD**

Aim:

Implement overlap add and overlap save method using Matlab/Scilab.

Theory:

Both the Overlap-Save and Overlap-Add methods are techniques used to compute the convolution of long signals using the Fast Fourier Transform (FFT). The direct convolution of two signals, especially when they are long, can be computationally expensive. These methods allow us to break the signals into smaller blocks and use the FFT to perform the convolution more efficiently.