A picture containing text, clipart

Description automatically generated

**SIGNAL PROCESSING LABORATORY RECORD (EE314)**

**Name**: Sasank Sah

**Scholar Id**: 1913142

**Sec**: B

**Semester**: 5th

**Guided By**:

Dr. TAPAN PRADHAN.

Dr. AVADH PATI.

ELECTRICAL ENGINEERING DEPARTMENT, NIT SILCHAR.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SL NO. | NAME OF EXPERIMENT | PAGE NO. | DATE OF EXPERIMENT | DATE OF SUBMISSION | REMARKS |
| 1 | Generation of Basic Signals | 3-9 | 07/09/2021 | 16/09/2021 |  |
| 2 | Verification of Sampling theorem and  Aliasing |  | 16/09/2021 | 22/09/2021 |  |
| 3 | Linear and Circular Convolution |  | 23/09/2021 | 29/09/2021 |  |
| 4 | Study of Discrete Fourier Transform (DFT) and its inverse |  | 29/09/2021 | 06/10/2021 |  |
| 5 | Computation of DFT of signals using DIT/DIF FFT Algorithm |  | 07/10/2021 | 20/10/2021 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**INDEX**

**EXP NO : 05**

**TITLE: Computation of DFT of signals using DIT/DIF FFT Algorithm**

**Date :20/10/2021**

**Objective:**

(I)To perform computation of FFT and IFFT of discrete time signa

(II)To observe the time of execution of FFT with number of points(N)

**Hardware and Software required:**

Computer

MATLAB R2021a

**Theory:**

**Fast Fourier Transform (FFT)**

Fast Fourier Transform is an efficient algorithm developed by Cooley & Tukey in 1965, used to compute the DFT with reduced computations. Due to the efficiency of FFT, it is used for spectrum analysis, convolutions, correlations, and linear filtering. FFT reduces the problem of calculating an N – point DFT to that of calculating many smaller – sized DFTs. The properties of the twiddle factor WN used in this algorithm are:

1. 𝑊𝑁 𝑘+ 𝑁 2 = 𝑒 − 𝑗2𝜋𝑘 𝑁 𝑒 −𝑗𝜋 = − 𝑊𝑁 𝑘 (Symmetry Property)

2. 𝑊𝑁 𝑘+𝑁 = 𝑒 − 𝑗2𝜋𝑘 𝑁 𝑒 −𝑗2𝜋 = 𝑊𝑁 𝑘 (periodicity property)

3. 𝑊𝑁 𝑚 = 𝑒 − 𝑗2𝜋𝑚 𝑁 = 𝑒 − 𝑗2𝜋 𝑁⁄𝑚 = 𝑊𝑁⁄𝑚

The Decimation – In – Time (DIT) and Decimation – In – Frequency (DIF) FFT algorithms use the “divide – and – conquer” approach. This is possible if the length of the sequence N is chosen as N = rm. here, r is called the radix of the FFT algorithm. The most practically implemented choice for r = 2 leads to radix – 2 FFT algorithms. So, with N = 2m, the efficient computation is achieved by breaking the N – point DFT into two 𝑁 2 - point DFTs, then breaking each 𝑁 2 - point DFT into two 𝑁 4 - point DFTs and continuing this process until 2 – point DFTs are obtained. For N=8, the Decimation – In – Time algorithm decomposition would be:

Diagram

Description automatically generated

**Decimation – In – Time (DIT) FFT algorithm**

In this algorithm, the time – domain sequence x[n] is decimated into two 𝑁 2 – point sequences, one composed of even – indexed values of x[n], and other composed of odd – indexed values of x[n].i.e.,

𝑔[𝑛] = 𝑥[2𝑛] … … …. 𝑒𝑞𝑛. 1

𝑎𝑛𝑑, ℎ[𝑛] = 𝑥[2𝑛 + 1] … … … .𝑒𝑞𝑛. 2

The N – point DFT of x[n] is given by:

A picture containing text

Description automatically generated

Text, letter

Description automatically generated

Where G(k) and H(k) are the N/2 – point DFTs of g[n] and h[n] respectively. So, G(k) and H(k) are periodic with period N/2 .i.e.,

𝐺 (𝑘 + 𝑁 2 ) = 𝐺(𝑘) … … 𝑒𝑞𝑛. 4

𝑎𝑛𝑑, 𝐻 (𝑘 + 𝑁 2 ) = 𝐻(𝑘) … … 𝑒𝑞𝑛. 5

And using the symmetry property of the twiddle factor, WN, and equations4 & 5

𝑋(𝑘 + 𝑁/2) = 𝐺(𝑘) − 𝑊𝑁 𝑘𝐻(𝑘) … … 𝑒𝑞𝑛. 6

Equations 3 and 6 result in the following butterfly diagram

A picture containing watch

Description automatically generated

**Decimation – In – Frequency (DIF)**

FFT algorithm In this algorithm, we decimate the DFT sequence X(k) into smaller and smaller subsequences (Instead of the time – domain sequence x[n]).

Text

Description automatically generated

Text, letter

Description automatically generated

**MATLAB PROGRAM:**

**1.Computation of FFT of discrete time signals**

clc;

clear all;

close all;

x=input('Enter the seq');

n=input('Enter the length of the sequence');

X=fft(x,n);

stem(X);

xlabel('real');

ylabel('img');

title('FAST FOURIER TRANSFORM');

display(X);

**OUTPUT:**

**X =**

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

**Chart

Description automatically generated**

**2. Computation of IFFT (or IDFT) of signals**

clc;

close all;

clear all;

X=input('Enter the seq');

n=input('Enter the length of the sequence');

x=ifft(X,n);

stem(x);

xlabel('real');

ylabel('img');

title('INVERSE FAST FOURIER TRANSFORM');

display(x);

**OUTPUT:**

**x =**

**2.5000 + 0.0000i -0.5000 - 0.5000i -0.5000 + 0.0000i -0.5000 + 0.500**

**Chart, histogram

Description automatically generated**

**3.Plotting Execution time of FFT of random discrete time signals**

clear all;

clc;

close all;

Nmax = 3000;

fft\_time=zeros(1,Nmax);

for n=1:1:Nmax

x=rand(1,n);

t=clock;

fft(x);

fft\_time(n)=etime(clock,t);

end

n=[1:1:Nmax];

bar(n,fft\_time)

xlabel('N');

ylabel('Time in Sec');

title('FFT execution times')

OUTPUT:

Chart, histogram

Description automatically generated

**ASSIGNMENTS :**

1. **Write a program to compute a four point DFT using DIT FFT algorithm.**

**CODE:**

clc;

close all;

clear all;

x=input('enter sequence');

N=length (x);

a = zeros (1, N) ;

z = zeros (1, N) ;

y=[1:N] ;

p=bitrevorder(y) ;

t=expm (-i\*2\*pi/N);

for i=1:N

b=p(i);

a(b)=x(i);

end

for m = 1:log2(N)

k = 0 : (2^ (m-1))-1;

j =1 ;

i=1;

for q =1:N / 2

z(i)=a(i)+a(i+2^(m-1))\*t^k(j);

z(i+2^(m-1))=a(i)-a(i+2^(m-1))\*t^k(j);

j=j+1;

if (mod(i, 2^ (m-1))==0)

i=i+2^(m-1)+1;

j=1;

else i=i+1;

end

end

a=z;

end

k = 0:1:N - 1 ;

subplot (2,1,1);

stem (k, abs (a));

xlabel('k');

ylabel('a');

title('magnitude response')

subplot (2,1,2);

stem(k, angle(a));

title('phase response');

**OUTPUT:**

**Chart

Description automatically generated**

**2. What do you understand by “in place” computation? How is it useful to make the FFT algorithm efficient?**

* This efficient use of memory is important for designing fast hardware to calculate the FFT. The term in-place computation is used to describe this memory usage.

## Decimation in Time Sequence

In this structure, we represent all the points in binary format i.e. in 0 and 1. Then, we reverse those structures. The sequence we get after that is known as bit reversal sequence. This is also known as decimation in time sequence.

Table

Description automatically generated

3. Compare an N-point DFT and N-point FFT in terms of their complexities.Compare the number of complex/ real multiplication, complex/ real addition and comment on the comparative data.

* The FFT is an efficient implementation of DFT and is used in digital image processing. FFT is applied to convert an image from the spatial domain to the frequency domain. Applying filters to images in frequency domain is computationally faster than to do the same in the spatial domain

For the computation of N Fourier coefficients, the number of complex multiplications and additions required is proportional to N2. The computational complexity in the implementation of DFT can be reduced from N2 to Nlog2N by a decomposition procedure.

For example, when N = 512, the direct DFT computational complexity proportional to N 2 = 262144, whereas the FFT computational complexity is proportional to Nlog2N = 2048. This means FFT is 32 times faster than DFT. [262144/2048 = 32].

**4. What is the difference and relation between DFT and FFT?**

* Difference and Relation between DFT and FFT

|  |  |
| --- | --- |
| **DFT** | **FFT** |
| DFT stands for discrete Fourier transform. | FFT stands for fast Fourier transform |
| DFT is a discrete version of Fourier transform. | FFT is a much efficient and fast version of Fourier transform |
| DFT is useful in spectrum estimation, convolution, etc. | FFT is useful in sound engineering, seismology, etc. |
| DFT establishes a relationship between the time domain and the frequency domain representation. | FFT is an implementation of DFT |
| DFT is a mathematical algorithm which transforms time-domain signals to frequency domain components | FFT algorithm consists of several computation techniques including DFT. |

**Conclusion:**

This experiment helped me to understand the theoretical concepts of Fast Fourier Transform (FFT) and how efficiently we can calculate FFT using the DIT/DFT algorithm by visualizing them by plotting the signals using MATLAB and having better grasp on the concepts on how these concepts are used in practical situations. In the last assignment, I understood more precisely about DFT and FFT as we compared them both about the differences and relations between them helped to understand the concepts.