SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA

(University established under section 3of UGC Act 1956)

(Accredited with 'A' Grade by NAAC)

Enathur, Kanchipuram – 631 561

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Lecture Notes

on

DISCRETE FOURIER TRANSFORM

By

Dr.V.Jayapradha

Assistant Professor

TOPIC – DISCRETE FOURIER TRANSFORM

CONTENTS

Discrete Fourier Transform

I. Need For DFT

The DFT is one of the most powerful tools in digital signal processing which enables us to find the spectrum of a finite-duration signal.

There are many circumstances in which we need to determine the frequency content of a time-domain signal. For example, we may have to analyze the spectrum of the output of an LC oscillator to see how much noise is present in the produced sine wave. This can be achieved by the discrete Fourier transform (DFT). The DFT is usually considered as one of the two most powerful tools in digital signal processing (the other one being digital filtering), and though we arrived at this topic introducing the problem of spectrum estimation, the DFT has several other applications in DSP.

The discrete Fourier transform (DFT) is one of the most important tools in digital signal processing. This chapter discusses three common ways it is used. First, the DFT can calculate a signal's *frequency spectrum*. This is a direct examination of information encoded in the frequency, phase, and amplitude of the component sinusoids. For example, human speech and hearing use signals with this type of encoding. Second, the DFT can find a system's frequency response from the system's impulse response, and vice versa. This allows systems to be analyzed in the *frequency domain*, just as convolution allows systems to be analyzed in the *time domain*. Third, the DFT can be used as an intermediate step in more elaborate signal processing techniques. The classic example of this is *FFT convolution*, an algorithm for convolving signals that is hundreds of times faster than conventional methods.

II. Pre- Test – MCQ

1. If h(n) is the real valued impulse response sequence of an LTI system, then what is the imaginary part of Fourier transform of the impulse response?
   a) $-\sum_{k=-\infty}^{\infty} h(k) \sin \omega k$
   b) $\sum_{k=-\infty}^{\infty} h(k) \sin \omega k$
   c) $-\sum_{k=-\infty}^{\infty} h(k) \cos \omega k$
   d) $\sum_{k=-\infty}^{\infty} h(k) \cos \omega k$

2. If h(n) is the real valued impulse response sequence of an LTI system, then what is the phase of H(ω) in terms of $H_R(\omega)$ and $H_I(\omega)$?
   a) $\tan^{-1} \frac{H_R(\omega)}{H_I(\omega)}$
   b) $-\tan^{-1} \frac{H_R(\omega)}{H_I(\omega)}$
   c) $\tan^{-1} \frac{H_I(\omega)}{H_R(\omega)}$
   d) $-\tan^{-1} \frac{H_I(\omega)}{H_R(\omega)}$

3. An ideal filter should have zero gain in their stop band.
   a) True
   b) False

4. What is the period of the Fourier transform X(ω) of the signal x(n)?
   a) π
   b) 1
   c) Non-periodic
   d) 2π

5. Which of the following condition is to be satisfied for the Fourier transform of a sequence to be equal as the Z-transform of the same sequence?
   a) |z|=1
   b) |z|<1
   c) |z|>1
   d) Can never be equal

6. A signal which is a function of two or more independent variable is called

a) multi-channel Signal

b) One dimensional signal

c) Multi dimensional signal

d) two dimensional Signal

7.In fourier transform of a real signal the magnitude and phase function will be

a) Symmetric and Symmetric

b) Symmetric and Anti-Symmetric

c) Anti-Symmetric and Symmetric

d) Anti- Symmetric and Anti-Symmetric

8.A first order LTI system will behave as a

a) low pass filter

b) band pass filter

c) high pass filter

d) low pass or high pass filter

9.The fourier transform of impulse response of a system is called

a) transfer function

b) frequency response

c) forced response

d) natural response

10. The signal x(n) may be shifted in time by replacing the independent variable n by

a) n – k

b) n + k

c) 2n

d) N^2

III. Pre Requisites

- Basics of signals
- LTI- system
- Fourier Series
- Fourier Transform
- Z-Transform

IV. Concept- Theory

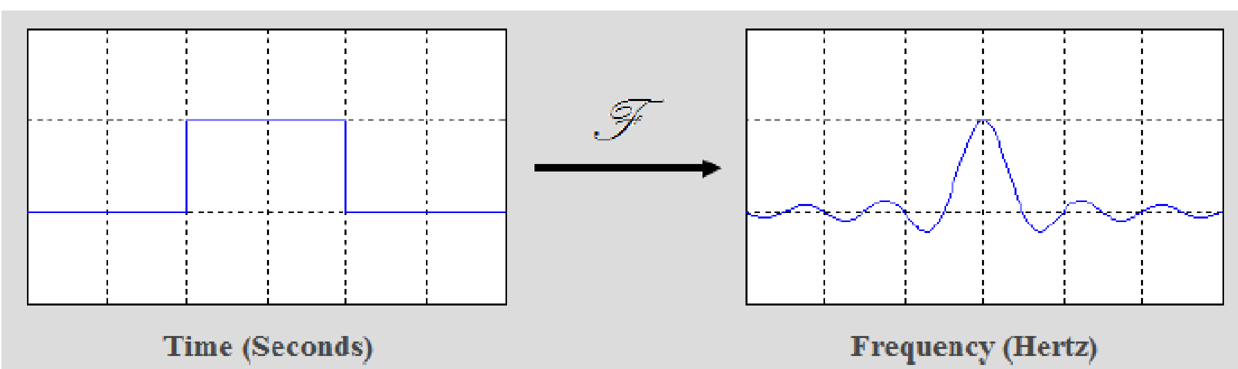
Fourier Transform

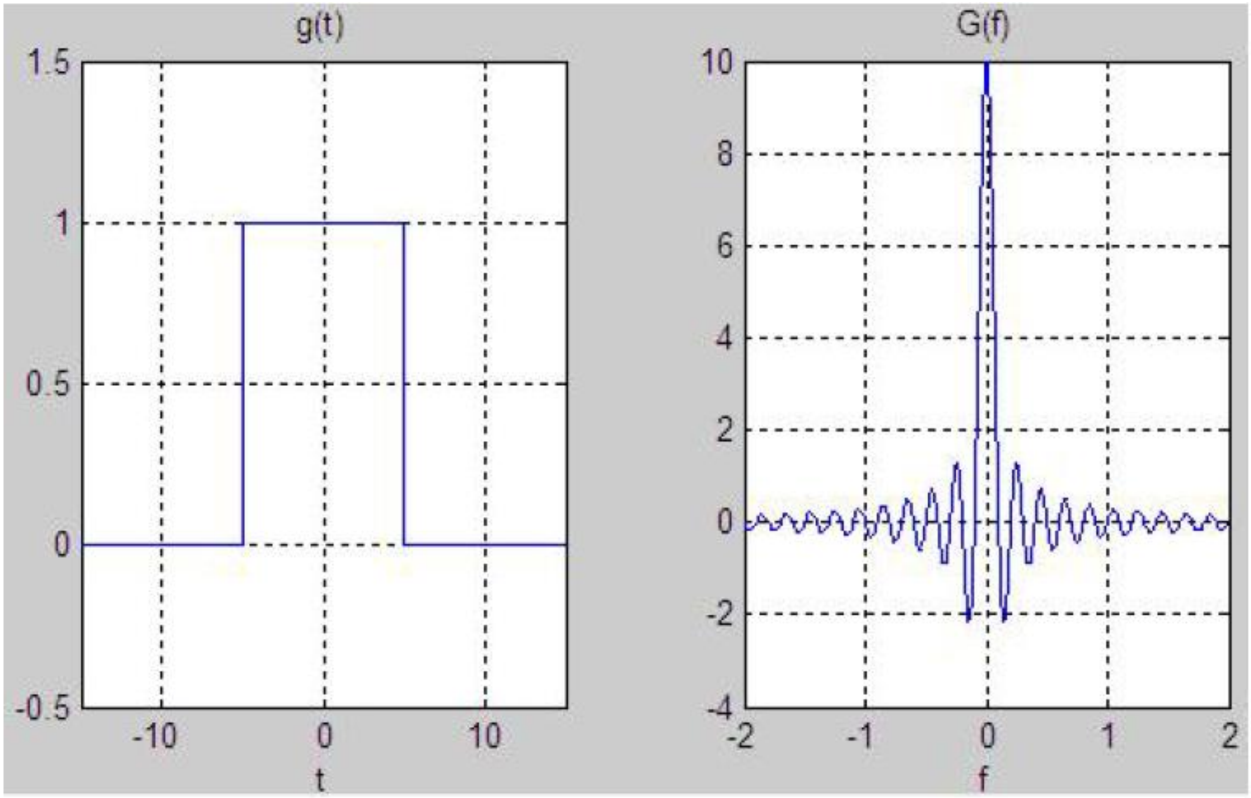
The Fourier Transform is a magical mathematical tool. The FourierTransform decomposes any function into a sum of sinusoidal basisfunctions. Each of these basis functions is a complex exponential of adifferent frequency. The Fourier Transform therefore gives us a unique wayof viewing any function - as the sum of simple sinusoids.

The Fourier Series showed us how to rewrite any periodic function into asum of sinusoids. The Fourier Transform is the extension of this idea to non-periodic functions.

$$\mathcal{F}\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft}\,dt$$

$$\mathcal{F}^{-1}\{G(f)\} = g(t) = \int^{\infty} G(f)e^{i2\pi ft}\,df$$





Conditions Transform Conditions for Existence of Fourier Transform

Any function f(t) can be represented by using Fourier transform only when the function satisfies Dirichlet's conditions. i.e.

1. The function f(t) has finite number of maxima and minima.

2. There must be finite number of discontinuities in the signal f(t),in the given interval of time.

3. It must be absolutely integrable in the given interval of time i.e.

$$\int_{-\infty}^{\infty} |f(t)|\, dt < \infty|$$

Discrete Time Fourier Transforms (DTFT)

The discrete-time Fourier transform (DTFT) or the Fourier transform of a discrete–time sequence x[n] is a representation of the sequence in terms of the complex exponential sequence $e^{i\omega n}$

The by The DTFT sequence x[n] is given by

$$X(\omega) = \Sigma_{n=-\infty}^{\infty} x(n) e^{-j\omega n}$$

Inverse Transform Inverse Discrete-Time Fourier Transform

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega$$

The family of Fourier transform

A signal can be either continuous or discrete, and it can be either periodic or aperiodic. The combination of these two features generates the four categories, described below and illustrated in Figure..

Aperiodic-Continuous
This includes, for example, decaying exponentials and the Gaussian curve. These signals extend to both positive and negative infinity without repeating in a periodic pattern. The Fourier Transform for this type of signal is simply called the Fourier Transform.

Periodic-Continuous
Here the examples include: sine waves, square waves, and any waveform that repeats itself in a regular pattern from negative to positive infinity. This version of the Fourier transform is called the Fourier Series.

Aperiodic-Discrete
These signals are only defined at discrete points between positive and negative infinity, and do not repeat themselves in a periodic fashion. This type of Fourier transform is called the Discrete Time Fourier Transform.

Periodic-Discrete
These are discrete signals that repeat themselves in a periodic fashion from negative to positive infinity. This class of Fourier Transform is sometimes called the Discrete Fourier Series, but is most often called the Discrete Fourier Transform.

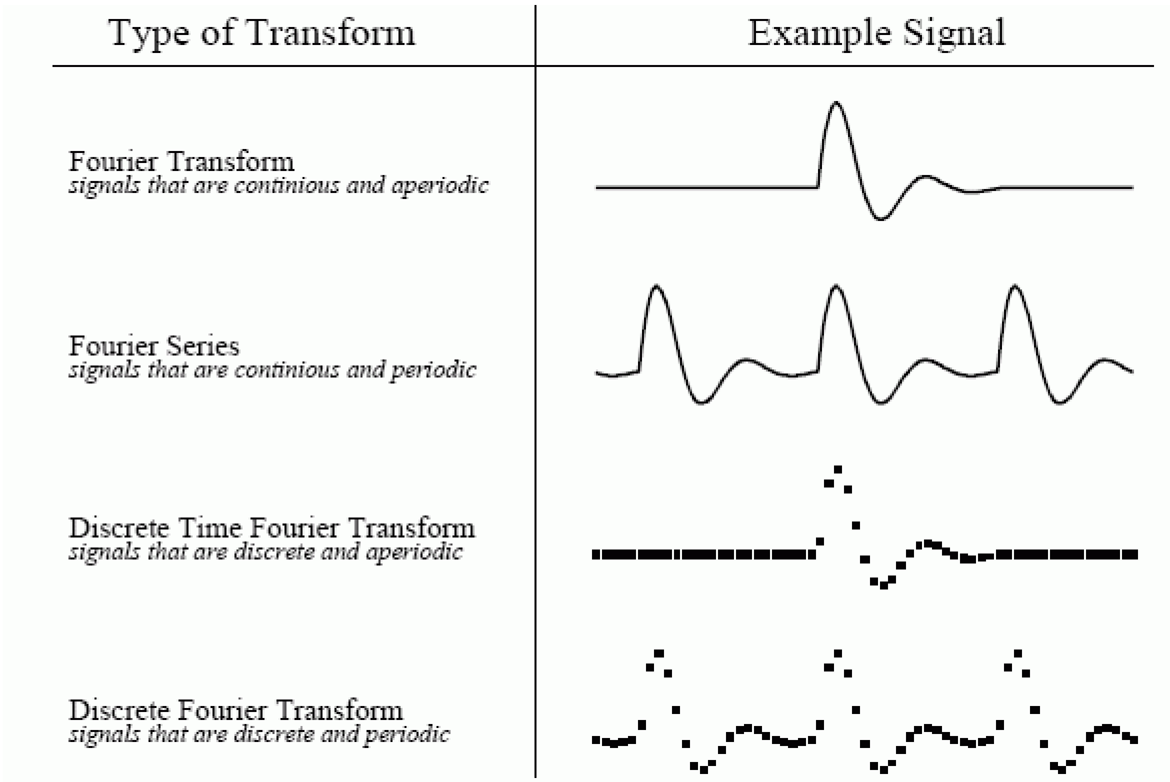
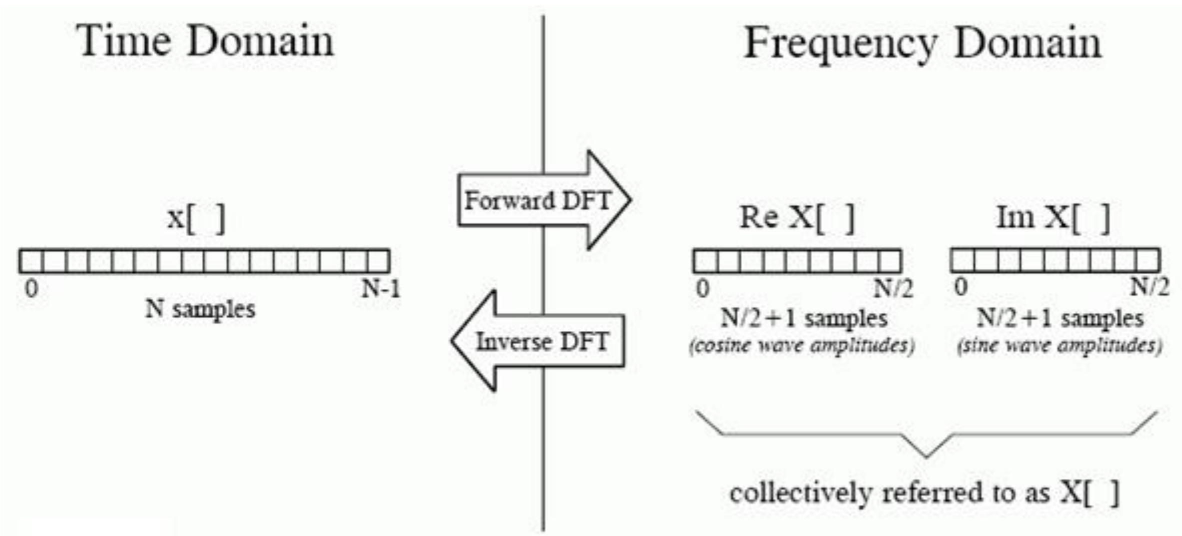


Illustration of the four Fourier transforms. A signal may be continuous or discrete, and it may be periodic or aperiodic. Together these define four possible combinations, each having its own version of the Fourier transform.

Time and Frequency Domain of DFT

The discrete Fourier transform changes an *N* point input signal into two point output signals. The input signal contains the signal being decomposed, while the two output signals contain the *amplitudes* of the component sine and cosine waves. The input signal is said to be in the **time domain**.



DFT terminology. In the time domain, $x[\ ]$ consists of N points running from 0 to $N-1$. In the frequency domain, the DFT produces two signals, the real part, written: $Re\,X[\ ]$, and the imaginary part, written: $Im\,X[\ ]$. Each of these frequency domain signals are $N/2+1$ points long, and run from 0 to $N/2$. The Forward DFT transforms from the time domain to the frequency domain, while the Inverse DFT transforms from the frequency domain to the time domain.

DFT and IDFT

DFT is used for analyzing discrete-time finite-duration signals in the frequency domain. The DFT and IDFT pair of equation is given by

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{kn}{N}} \\ x(n) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi \frac{kn}{N}} \end{aligned}$$

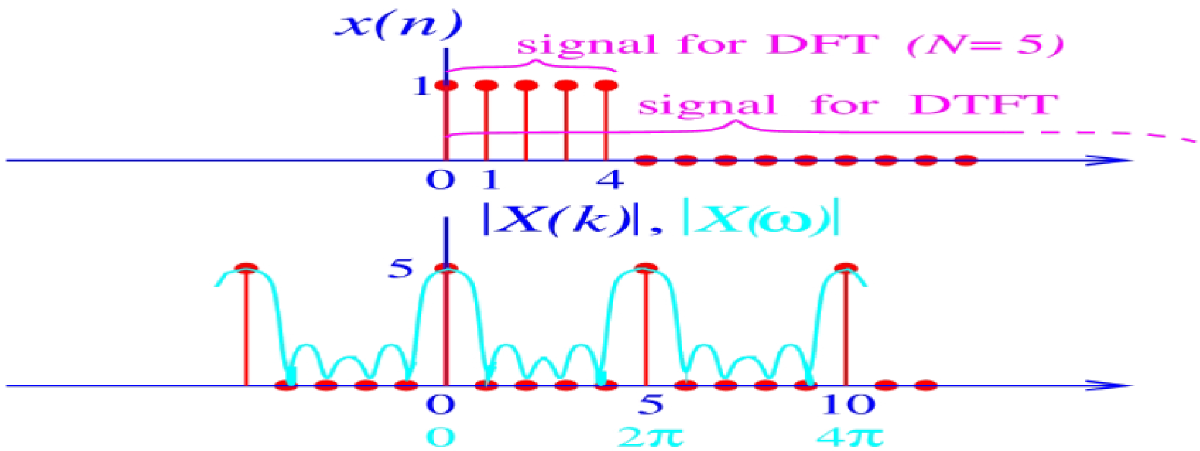
The DFT spectrum is periodic with period N (which is expected, since the DTFT spectrum is periodic as well, but with period 2π).

Example: DFT of a rectangular pulse:

$$x(n) = \begin{cases} 1, & 0 \le n \le (N-1), \\ 0, & \text{otherwise.} \end{cases}$$

the rectangular pulse is “interpreted” by the DFT as a spectral line at frequency ω= 0.

DFT and DTFT of a rectangular pulse (N=5)



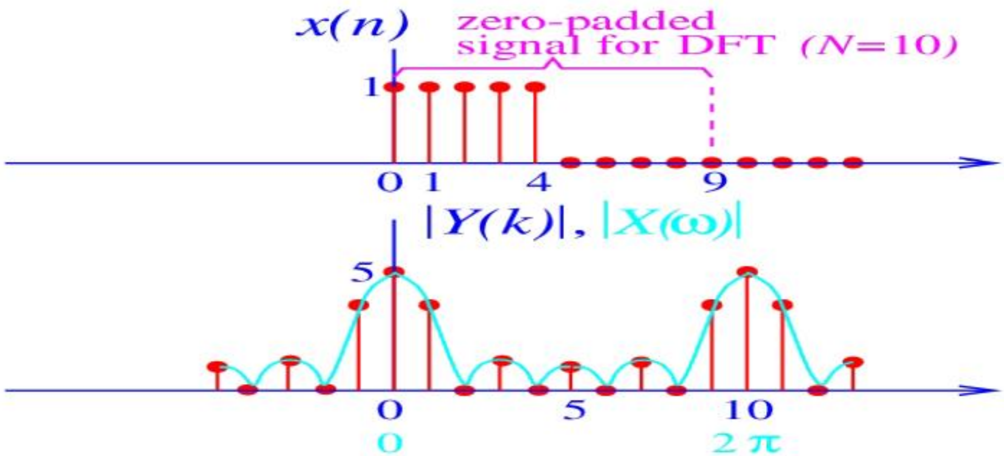
Zero Padding

What happens with the DFT of this rectangular pulse if we increase N by zero padding:

$$\{y(n)\} = \{x(0), \ldots, x(M-1), \underbrace{0, 0, \ldots, 0}_{N-M \text{ positions}}\},$$

where $x(0) = \cdots = x(M-1) = 1$.

DFT and DTFT of a Rectangular Pulse with Zero Padding ($N = 10,\ M = 5$)



Example for computing DFT.

Compute the DFT of the sequence x(n) = {1,2,3,4} sketch the magnitude and phase response.

By N-point DFT

$$X(k) = \sum_{n=0}^{N-1} X(n) e^{\frac{-2\pi kn}{N}} \quad for\ k = 0,1,2 \ldots . N-1$$

Here N=4

For K= 0, $X(0) = \sum_{n=0}^{4} X(n) e^{\frac{-2\pi(0)n}{4}}$ = x(0) + x(1) + x(2) + x(3) = 1+2+3+4 = 10

For k = 1, $X(1) = \sum_{n=0}^{4} X(n) e^{\frac{-2\pi(1)n}{4}}$

= x(0) + x(1) $e^{-j2\pi/4}$ + x(2) $e^{-j4\pi/4}$ + x(3) $e^{-j6\pi/4}$

= 1+ 2 [$\cos\frac{\pi}{2} - j\sin\frac{\pi}{2}$] +3 [$\cos\pi - j\ \sin\pi$]+ 4 [$\cos\frac{3\pi}{2} - j\ \sin\frac{3\pi}{2}$]

= 1+2(-j) + 3 (-1) +4 (j) = 1-2j-3+4j

= -2+2j

For k = 2, $X(2) = \sum_{n=0}^{4} X(n) e^{\frac{-2\pi(2)n}{4}}$

$= x(0) + x(1)\, e^{-j\pi} + x(2)\, e^{-j2\pi} + x(3)\, e^{-j3\pi}$

= 1+2(-1)+3(-1)+4(-1)=1-2+3-4=-2

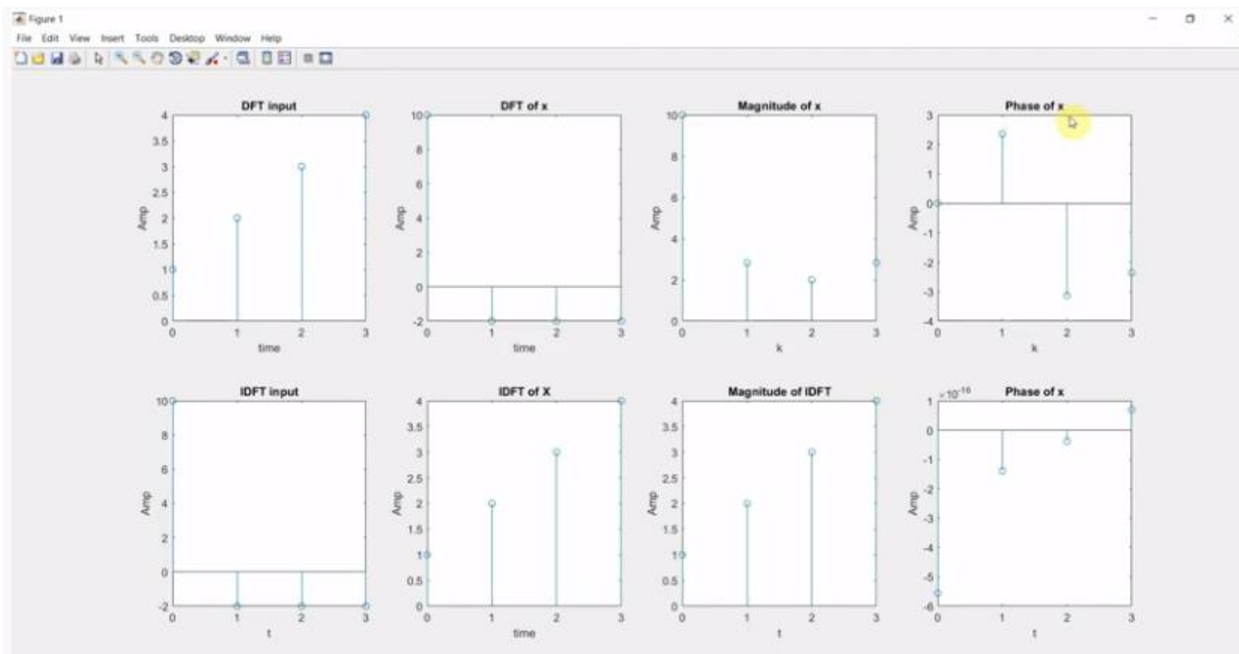
For k= 3, $X(3) = \sum_{n=0}^{4} X(n) e^{\frac{-2\pi(3)n}{4}}$

$=x(0) + x(1)\, e^{-j3\pi/2} + x(2)\, e^{-j3\pi} + x(3)\, e^{-j9\pi/2}$

= 1+2(j) +3(-1)+4(-j) = 1+2j-3-4j = -1-2j

X(K) = {10,-2+10j , -2 , -2-2j}

X(K) = $\{10, 2.28\ \angle 135\ , 2\ , 2.23\ \angle - 116.56\ \}$



Convolution Property of the Fourier Transform

The convolution of two functions in time is defined by:

$$g(t) * h(t) = \int_{-\infty}^{\infty} g(\tau) h(t-\tau) d\tau$$

The Fourier Transform of the convolution of g(t) and h(t) [with corresponding Fourier Transforms G(f) and H(f)] is given by:

$$\mathscr{F}\{g(t) * h(t)\} = G(f)H(f)$$

Theorem If a discrete-time system linear shift-invariant, T[.], has the unit sample response

T[*δ(n)*] = *h(n)* then the output *y(n)* corresponding to any input *x(n)* is given by

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) = \sum_{k=-\infty}^{\infty} x(n-k)h(k)$$

$$y(n) = x(n) * h(n) = h(n) * x\,(n)$$

The second summation is obtained by setting $m = n - k$; then for $k = -\infty$ we have $m = +\infty$, and for $k = \infty$ we have $m = -\infty$. Thus

$$\sum_{k=-\infty}^{\infty} x(k)h(n-k) = \sum_{k=-\infty}^{\infty} x(n-m)h(m) = \sum_{k=-\infty}^{\infty} x(n-k)h(k)$$

m is a dummy variable. The order of summation (forward or backward) makes no difference. hence change m to k and switch limits

[Linear Convolution]
Given the input {*x(n)*} ={1, 2, 3, 1} and the unit sample response {*h(n)*} = {4, 3, 2, 1} find the response *y(n)* = *x(n)* * *h(n)*.

Answer Since $x(k) = 0$ for $k < 0$ and $h(n-k) = 0$ for $k > n$, the convolution sum becomes

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) = \sum_{k=0}^{n} x(k)h(n-k)$$

Now *y(n)* can be evaluated for various values of *n*; for example, setting $n = 0$ gives *y(0)*.

Linear Convolution of {x(n)} ={1, 2, 3, 1} and {h(n)} = {4, 3, 2, 1}

$$y(n) = \sum_{k=0}^{n} x(k)h(n-k)$$

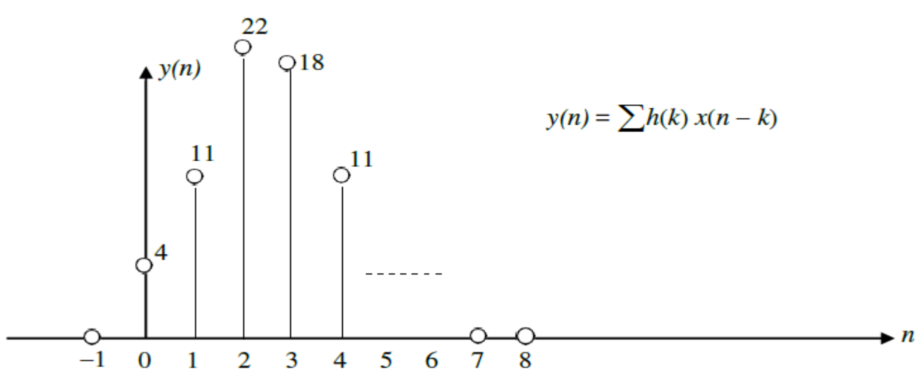
Length of x(n) is L = 4

Length of h(n) is M= 4

Length of y(n) is L= L+M-1 = 4+4-1=7

n = 0	$y(0) = \sum_{k=0}^{0} x(k)\, h(0-k)$	= x(0) h(0) = 1 . 4 = 4
n = 1	$y(1) = \sum_{k=0}^{1} x(k)\, h(1-k)$	= x(0) h(1) + x(1) h(0) = 1 . 3 + 2 . 4 = 11
n = 2	$y(2) = \sum_{k=0}^{2} x(k)\, h(2-k)$	= x(0) h(2) + x(1) h(1) + x(2) h(0) = 1 . 2 + 2 . 3 + 3 . 4 = 20
n = 3	$y(3) = \sum_{k=0}^{3} x(k)\, h(3-k)$	= x(0) h(3) + x(1) h(2) + x(2) h(1) + x(3) h(0) = 1 . 1 + 2 . 2 + 3 . 3 + 1 . 4 = 18
n = 4	$y(4) = \sum_{k=0}^{4} x(k)\, h(4-k)$	= x(0) ***h(4)*** + x(1) h(3) + x(2) h(2) + x(3) h(1) + ***x(4)*** h(0) = 1 . 0 + 2 . 1 + 3 . 2 + 1 . 3 + 0 . 4 = 11
n = 5	$y(5) = \sum_{k=0}^{5} x(k)\, h(5-k)$	= ***x(0) h(5)*** + ***x(1) h(4)*** + x(2) h(3) + x(3) h(2) + ***x(4) h(1)*** + ***x(5) h(0)*** = 3 . 1 + 1 . 2 = 5
n = 6	$y(6) = \sum_{k=0}^{6} x(k)\, h(6-k)$	= ***x(0) h(6)*** + ***x(1) h(5)*** + ***x(2) h(4)*** + x(3) h(3) + ***x(4) h(2)*** + ***x(5) h(1)*** + ***x(6) h(0)*** = 1 . 1 = 1
n = 7	$y(7) = \sum_{k=0}^{7} x(k)\, h(7-k)$	= ***x(0) h(7)*** + ***x(1) h(6)*** + ***x(2) h(5)*** + ***x(3) h(4)*** + ***x(4) h(3)*** + ***x(5) h(2)*** + ***x(6) h(1)*** + ***x(7) h(0)*** = 0
y(n) = 0 for n < 0 and n > 6		↑Product terms in ***bold italics*** are zero.

Thus y(n) ={4, 11, 20, 18, 11, 5, 1}



Tabular Method of Linear convolution

Tabular method of linear convolution

n↓	k→	–3	–2	–1	0	1	2	3	4	5	6	7	n	y(n)
	h(k) →				4	3	2	1						
	x(k) →				1	2	3	1						
0	x(0–k)	1	3	2	1								0	4
1	x(1–k)		1	3	2	1							1	11
2	x(2–k)			1	3	2	1						2	20
3	x(3–k)				1	3	2	1					3	18
4	x(4–k)					1	3	2	1				4	11
5	x(5–k)						1	3	2	1			5	5
6	x(6–k)							1	3	2	1		6	1
7	x(7–k)								1	3	2	1	7	0
.	.												.	.

Thus y(n) ={4, 11, 20, 18, 11, 5, 1}

Another method for linear convolution

Step 1 : Multiply X(n) and h(n) in the table

X	4	3	2	1
1	4	3	2	1
2	8	6	4	2
3	12	9	6	3
1	4	3	2	1

Step 2 : Add the Diagonals

Y(n) = [4, 8+3, 12+6+2 , 4+9+4+1, 3+6+2, 2+3, 1] = [4, 11, 20, 18, 11, 5, 1]

Circular convolution

The convolution property of DFT states that, the multiplication of the DFTs of two sequences is equivalent to the DFT of the circular convolution of the two sequences.

$$x_1(k)x_2(k) = DFT\ \{x_1(n) \circledast x_2(n)\}$$
$$x_3(m) = \sum_{k=0}^{n} x_1(m)x_2(m-n)$$

Methods of Circular Convolution

Generally, there are two methods, which are adopted to perform circular convolution and they are

- Concentric circle method,
- Matrix multiplication method.

Concentric Circle Method

Let x1(n)x1(n) and x2(n)x2(n) be two given sequences. The steps followed for circular convolution of x1(n)x1(n) and x2(n)x2(n) are

- Take two concentric circles. Plot N samples of x1(n)x1(n) on the circumference of the outer circle maintaining equal distance successive points in anti-clockwise direction.
- For plotting x2(n) x2(n), plot N samples of x2(n)x2(n) in clockwise direction on the inner circle, starting sample placed at the same point as 0^{th} sample of x1(n) x1(n)
- Multiply corresponding samples on the two circles and add them to get output.
- Rotate the inner circle anti-clockwise with one sample at a time.

Matrix Multiplication Method

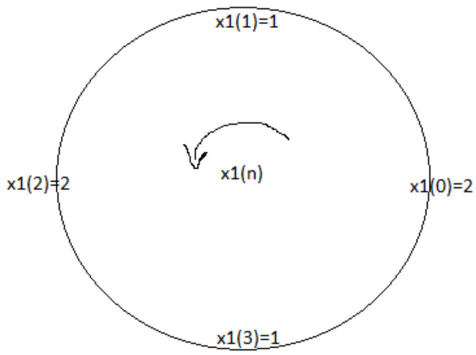
Matrix method represents the two given sequence x1(n)x1(n) and x2(n)x2(n) in matrix form.

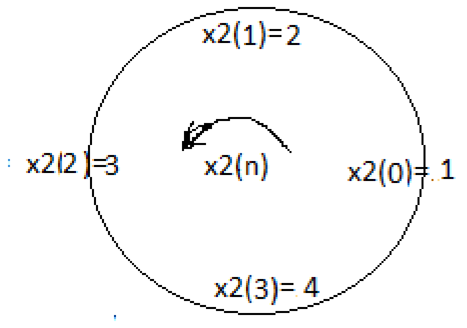
- One of the given sequences is repeated via circular shift of one sample at a time to form a N X N matrix.
- The other sequence is represented as column matrix.
- The multiplication of two matrices give the result of circular convolution.

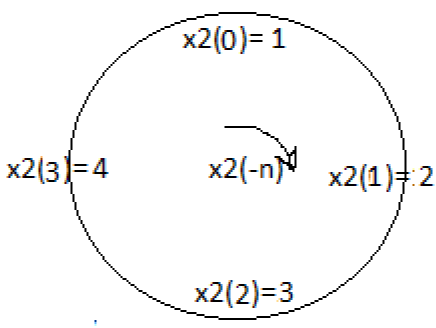
Example :

Perform Circular convolution of the two sequences $x_1(n) = \{2,1,2,1\}$ and $x_2(n) = \{1,2,3,4\}$.

Let the given sequence can be represented as points on a circle.





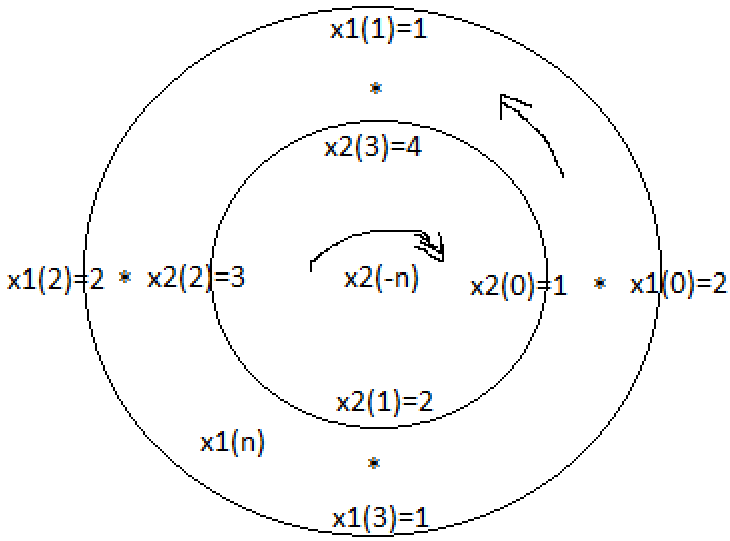


The graphical method of computing $x_3(n)$ as follows

$$x_3(m) = \sum_{k=0}^{n} x_1(m)x_2(m-n)$$

When m=0,

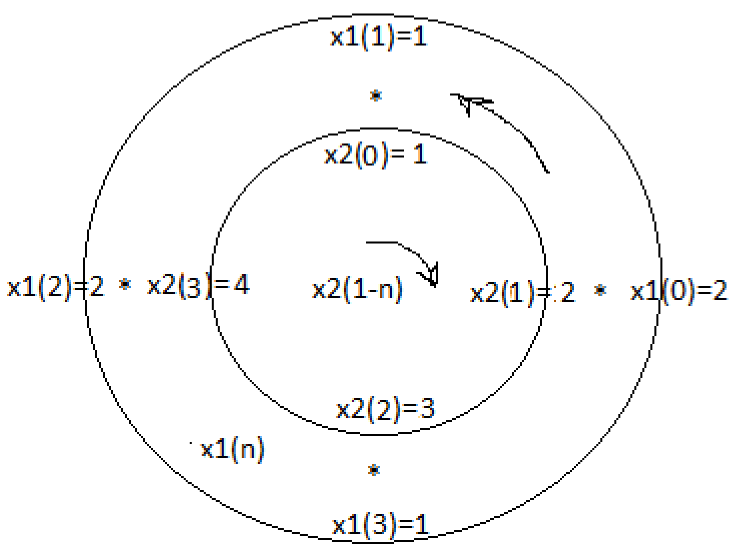
$$x_3(m) = \sum_{k=0}^{n} x_1(m)x_2(0-n)$$



X_3 (0)= 2*1 + 1*4 + 2*3 + 1*2 = 2+4+6+2 = 14

When m=1,

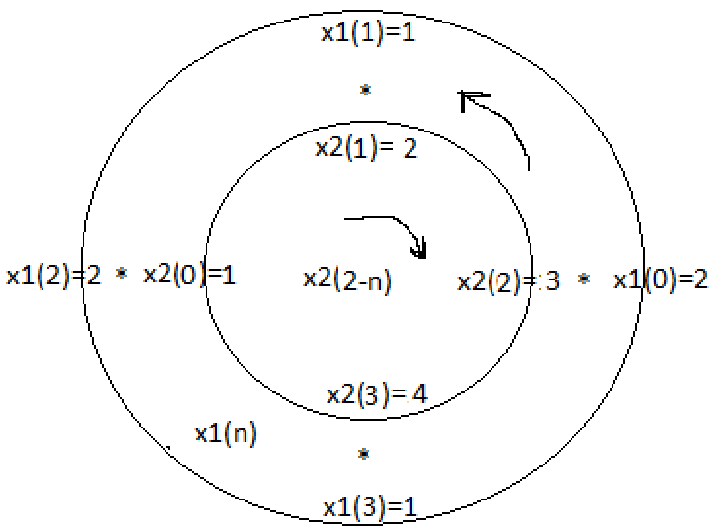
$$x_3(m) = \sum_{k=0}^{n} x_1(m)x_2(1-n)$$



X_3 (1)= 1*1 + 2*4 + 1*3 + 2*2 = 1+8+3+4 = 16

When m=2,

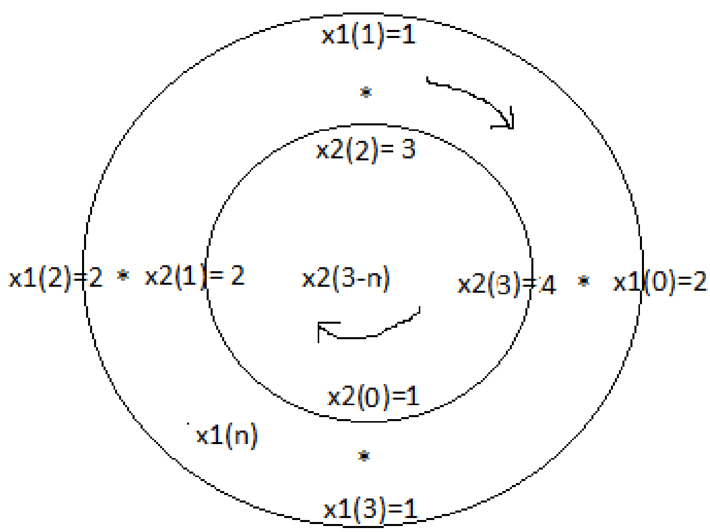
$$x_3(m) = \sum_{k=0}^{n} x_1(m)x_2(2-n)$$



X_3 (2) = 1*2 + 4*1 + 3*2 + 2*1 = 2+4+6+2 = 14

When m=3,

$$x_3(m) = \sum_{k=0}^{n} x_1(m)x_2(3-n)$$



X_3 (3) = 1*1 + 2*4 + 1*3 + 2*2 = 1+8+3+4 = 16

X_3 (n) = {14,16,14,16}

Circular convolution by Matrix method.

The sequence $x_1(n)$ can be arranged as a column vector of order N x 1 and using the samples of $x_2(n)$ the N xN matrix is formed.

$$\begin{bmatrix} x_2(0) & x_2(3) & x_2(2) & x_2(1) \\ x_2(1) & x_2(0) & x_2(3) & x_2(2) \\ x_2(2) & x_2(1) & x_2(0) & x_2(3) \\ x_2(3) & x_2(2) & x_2(1) & x_2(0) \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix} = \begin{bmatrix} x_3(0) \\ x_3(1) \\ x_3(2) \\ x_3(3) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 14 \\ 16 \\ 14 \\ 16 \end{bmatrix}$$

X_3 (n) = {14,16,14,16}

Comparison of Linear and Circular convolution

Comparison points	Linear Convolution	Circular Convolution
Shifting	Linear shifting	Circular shifting
Samples in the convolution result	N1+N2−1	Max(N1,N2)
Finding response of a filter	Possible	Possible with zero padding
Periodicity	Aperiodic	Periodic

Fast Fourier Transform Algorithms

The time taken to evaluate a DFT on a digital computer depends principally on the number of multiplications involved, since these are the slowest operations. With the DFT, this number is directly related to N^2 (matrix multiplication of a vector), where N is the length of the transform. For most problems, N is chosen to be at least 256 in order to get a reasonable approximation for the spectrum of the sequence under consideration – hence computational speed becomes a major consideration.
Highly efficient computer algorithms for estimating Discrete Fourier Transforms have been developed since the mid-60's. These are known as Fast Fourier Transform (FFT) algorithms and they rely on the fact that the standard DFT involves a lot of redundant calculations:

- Basic idea is to split the sum into 2 subsequences of length N/2 and continue all the way down until you have N/2 subsequences of length 2

  Thus rewrite the DFT equation as,

$$X(k) = \sum_{n=0}^{N-1} X(n) e^{\frac{-2\pi kn}{N}} \quad for\ k = 0,1,2 \ldots .N-1$$

$$X(k) = \sum_{n=0}^{N-1} x(n) W^{kn} \quad \longleftarrow W = e^{-j\frac{2\pi}{N}}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W^{-kn}.$$

Thus $W_N^{kn} = e^{-j2\pi/N}$ called the Twiddle Factor W_N^k

General form for 4- bit FFT

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{matrix} & n=0 & n=1 & n=2 & n=3 \\ k=0 \\ k=1 \\ k=2 \\ k=3 \end{matrix} \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

Steps for Radix-2 DIT and DIF algorithm

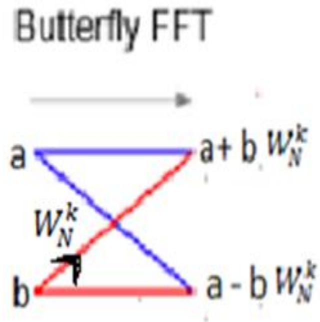
DIT-FFT

- The number of input samples $N=2^M$,M is an integer.
- The input sequence is shuffled through bit reversal.
- number of stages is given by M= $\log_2$ N.
- each stage consists of N/2 butterflies.
- I/O for each butterflies are separated by 2^{m-1} samples. (m is stage index).
- The number of complex addition is N $\log_2$ N .
- the number of complex multiplication is N/2 $\log_2$ N.
- Twiddle factor exponential are the funtion of stage index m and is given by $k = \frac{Nt}{2^m} \quad t = 0,1,2 \ \ldots. \ 2^{m-1} - 1$.
- The number of set or section in stage is 2^{M-m} .
- Exponent repeat factor ERF is 2^{M-m}.
- The output sequence is natural order.

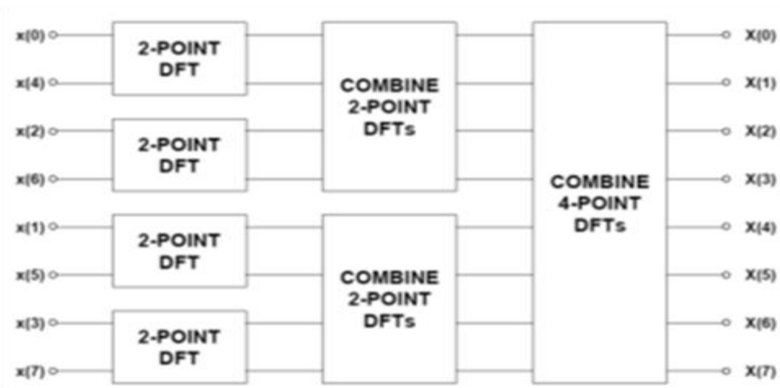
DIF-FFT

- The number of input samples $N=2^M$,M is an integer.
- The input sequence is natural order.
- number of stages is given by M= $\log_2$ N.
- each stage consists of N/2 butterflies.
- I/O for each butterflies are separated by 2^{M-m} samples. (m is stage index).
- The number of complex addition is N $\log_2$ N .
- the number of complex multiplication is N/2 $\log_2$ N.
- Twiddle factor exponential are the funtion of stage index m and is given by $k = \frac{Nt}{2^{M-m+1}} \quad t = 0,1,2 \ \ldots. \ 2^{M-m} - 1$.
- The number of set or section in stage is 2^{M-1} .
- Exponent repeat factor ERF is 2^{M-1} .
- The output sequence is shuffled through bit reversal.

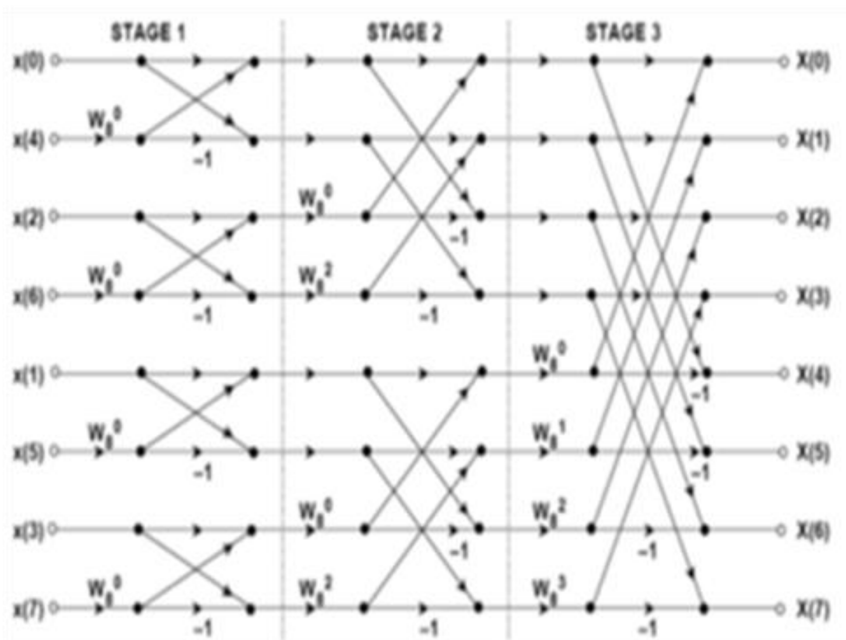
FFT Algorithms - Eight point DIT-FFT



Butterfly Function

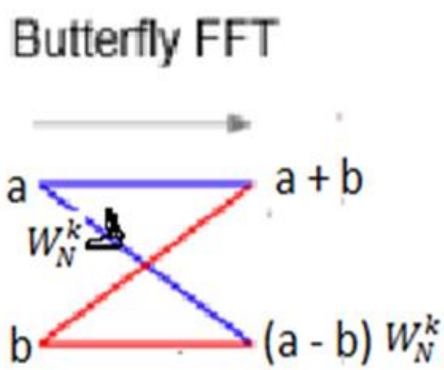


8- point

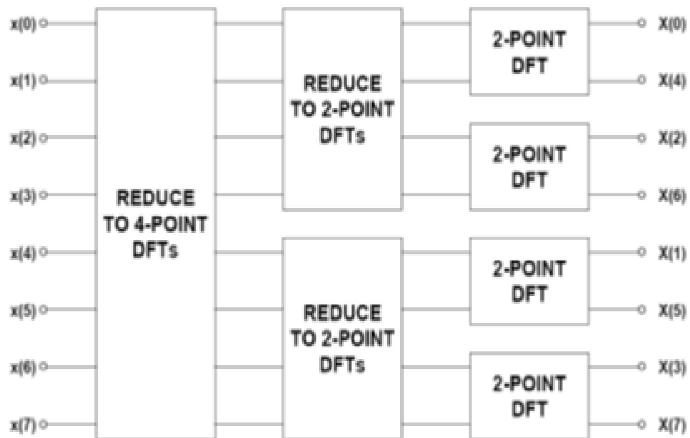


DIT-FFT

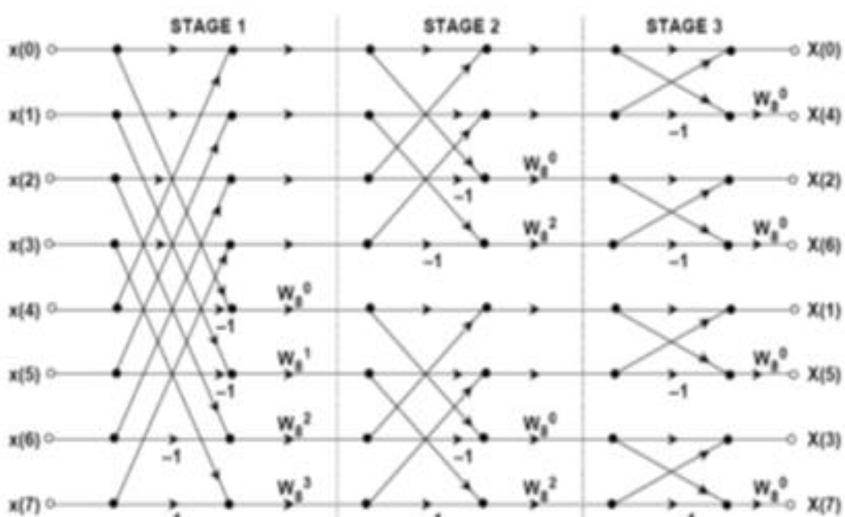
FFT Algorithms - Eight point DIF-FFT



Butterfly Function



8- point

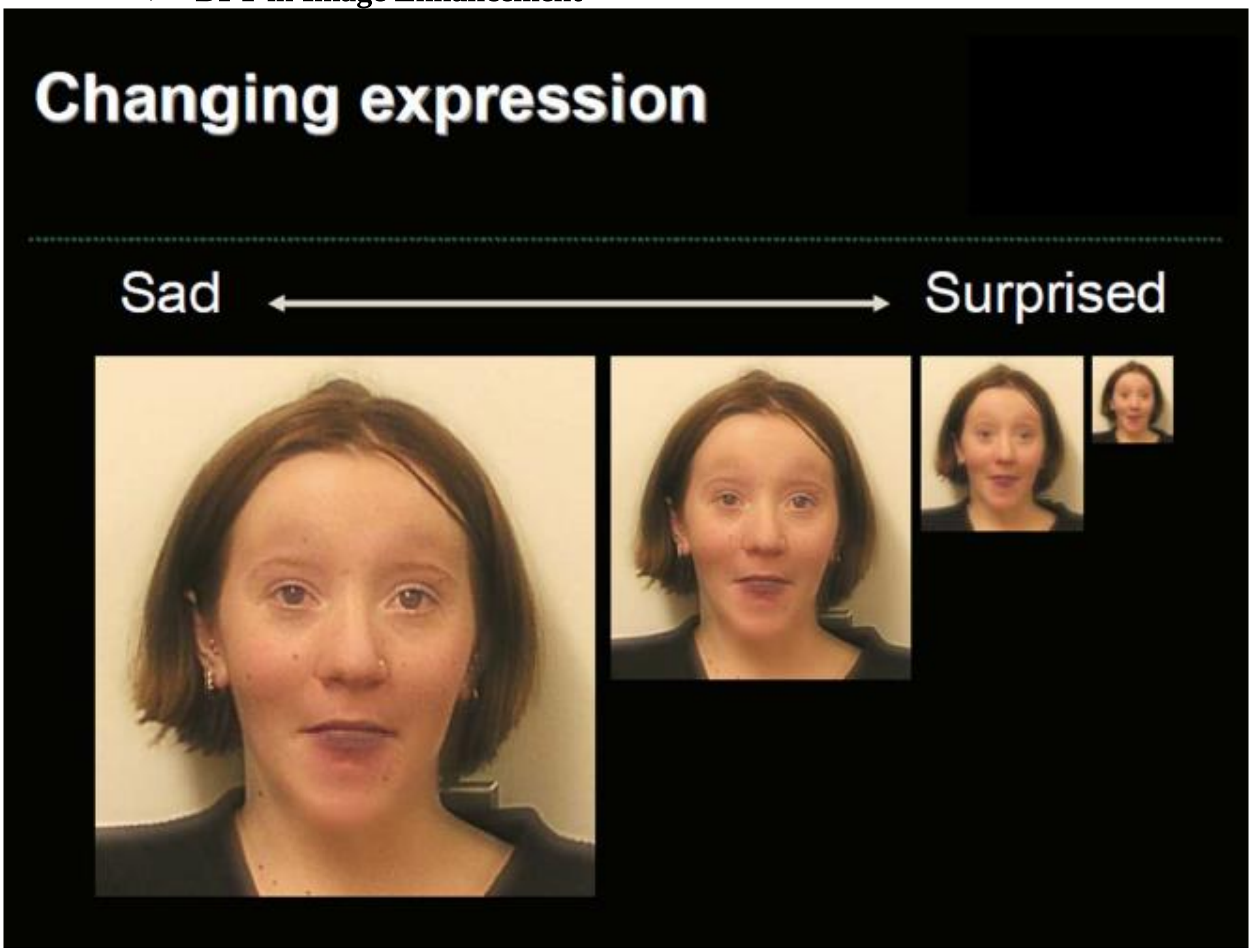


DIF-FFT

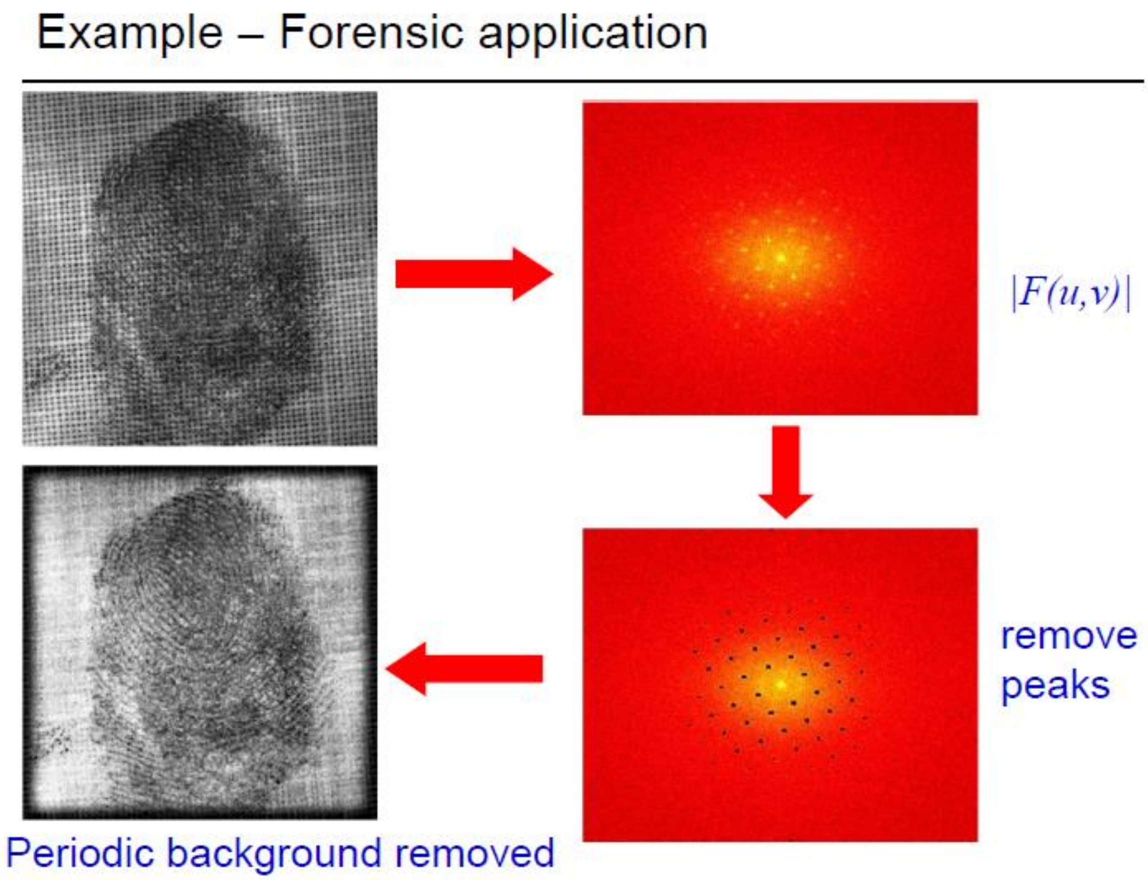
V. Applications

1. **Biomedical** – Spectral analysis of EEG and ECG Signals
2. **Image Processing** – two dimensional filtering on images for image enhancement, finger print matching, etc.
3. **Seismology** - Spectrum Analysis of seismic signals , can be used to predict the earth quake.
4. **Consumer electronics** – Music system, Karoake system etc.

- **DFT in Image Enhancement**



- **Forensic Application**



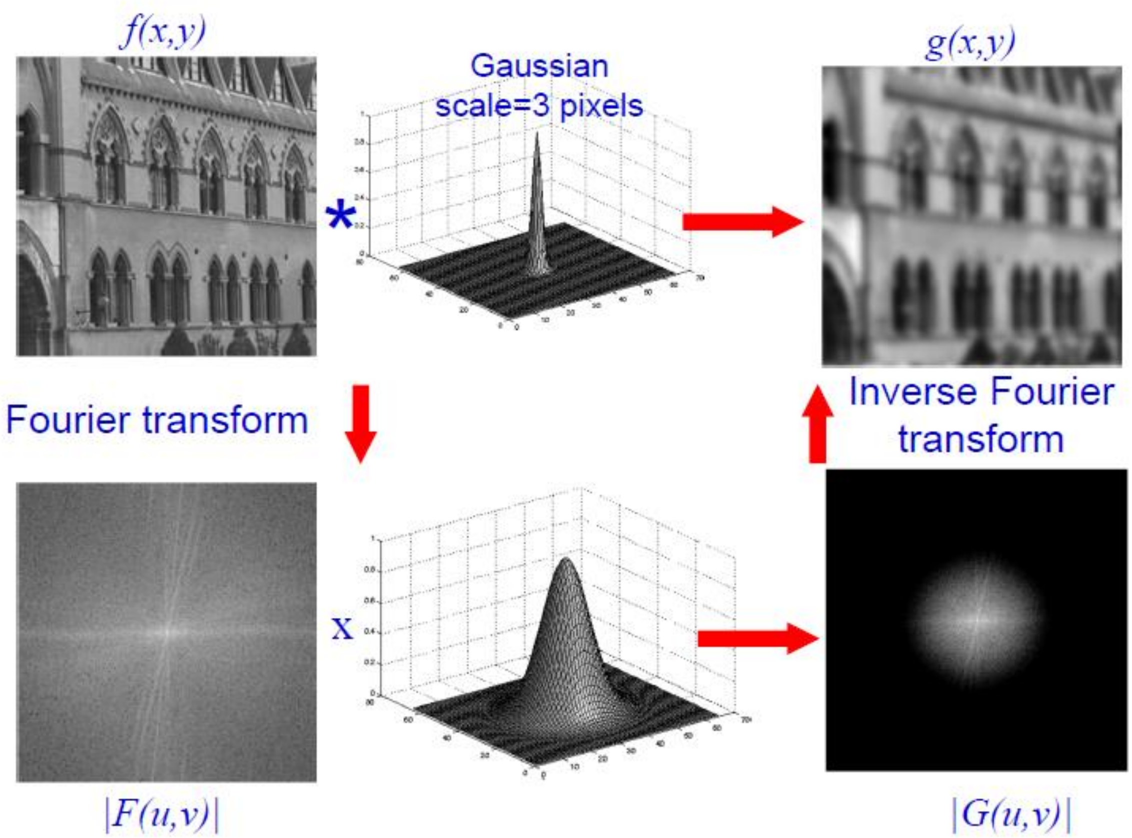
- **Convolution Theorem**

Convolution theorem

$$f(x,y) * h(x,y) \quad \Leftrightarrow \quad F(u,v)H(u,v)$$

Space convolution = frequency multiplication

In words: the Fourier transform of the convolution of two functions is the product of their individual Fourier transforms



VI. Laboratory Examples

SPECTRUM ANALYSIS USING DFT

Aim:

To write MATLAB program for spectrum analyzing signal using DFT.

PROCEDURE:

1. Start the MATLAB program.
2. Open new M-file
3. Type the program
4. Save in current directory
5. Compile and Run the program
6. If any error occurs in the program correct the error and run it again
7. For the output see command window\ Figure window
8. Stop the program.

PROGRAM: (Spectrum Analysis Using DFT)

```
N=input('type length of DFT= ');
T=input('type sampling period= ');
freq=input('type the sinusoidal freq= ');
k=0:N-1;
f=sin(2*pi*freq*1/T*k);
F=fft(f);
stem(k,abs(F));
grid on;
xlabel('k');
ylabel('X(k)');
```

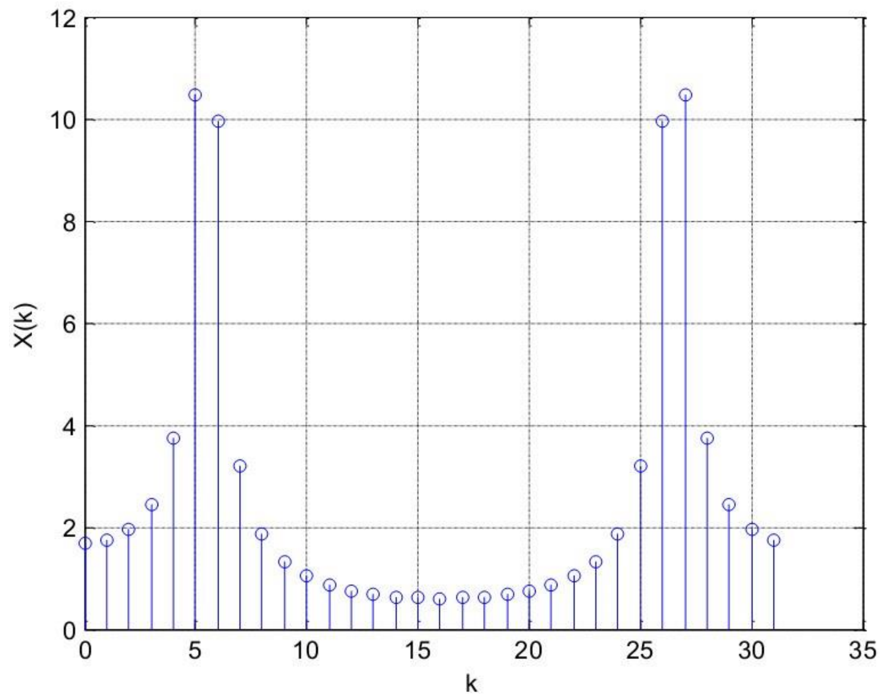
INPUT:

type length of DFT=32

type sampling period=64

type the sinusoidal freq=11

OUTPUT: (Spectrum Analysis Using DFT)



RESULT:

Thus the Spectrum Analysis of the signal using DFT is obtained using MATLAB.

Fourier analysis of 2D Image

Aim:

To write MATLAB program for Fourier analysis of 2D image.

Program:

```
% fourier analysis of 2D signals (Image in gray scale)
Clear all
I = imread('img1.png');
I = rgb2gray(I); % create a gray scale image
[r,  c] = size (I) ;
for i = 1: r
X(i , :) = fft (I (i,:) );  % calculate fourier analysis of each row
End
for j = 1: c
Y (:, j) = fft (X (:,j) );  % calculate fourier analysis of each column
End
Figure (1)
Imshow (I)
Figure (2)
M=Y;
M= fftshift (M); % shift low frequencies to the centre
Ab= abs (M);  % magnitude
Ab = (Ab – min (min (Ab))) ./ (max(max(Ab))) . *255;
% normalized to display the magnitude
Imshow (Ab) % Display
```

Result:

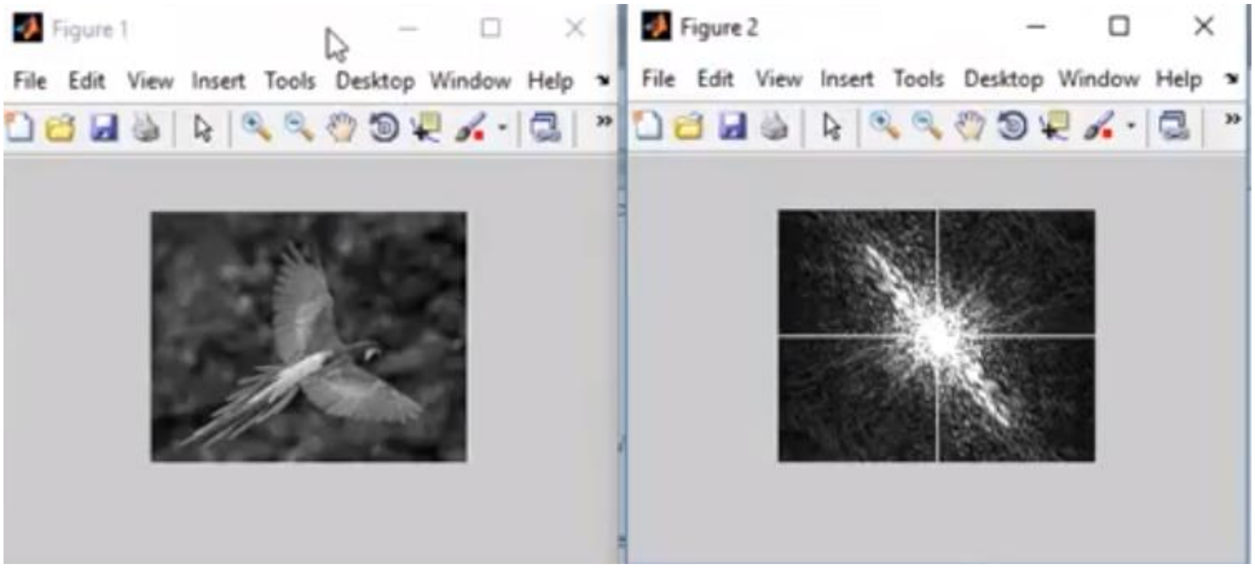


Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 2
File Edit View Insert Tools Desktop Window Help

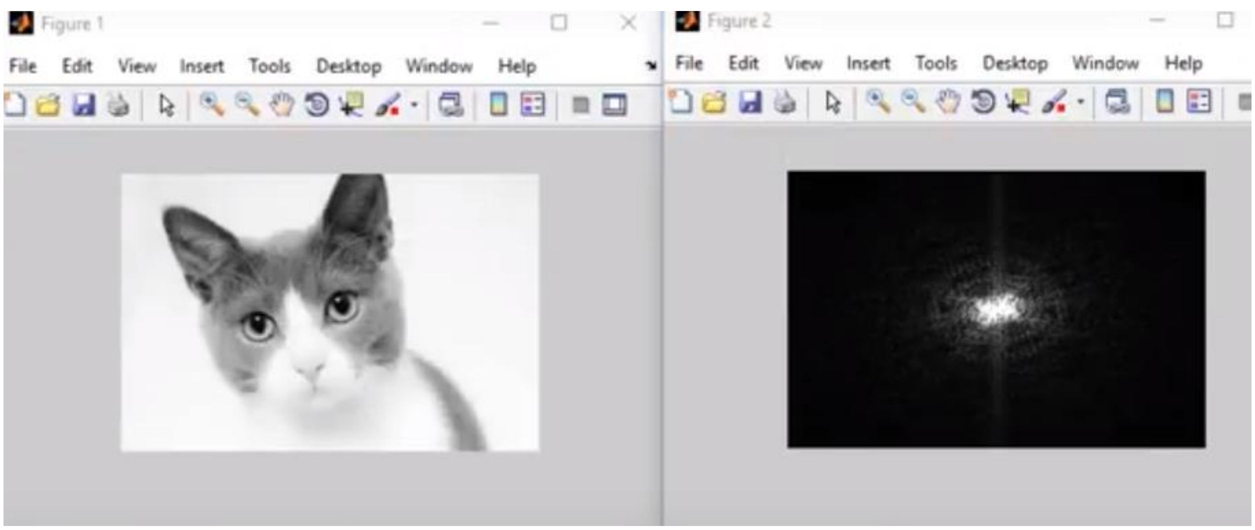


Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 2
File Edit View Insert Tools Desktop Window Help

VII. Post Test

1. The 4-point discrete Fourier Transform (DFT) of a discrete time sequence {1, 0, 2, 3} is

   a) [0,-2+2j,2,-2-2j]

   b) [2,2+2j,6,2-2j]

   c) [6,1-3j,2,1+3j]

   d) [6,-1+3j,0,-1-3j]

2. For an N-point FFT algorithm with $N=2m$, which one of the following statements is TRUE?

   (a) It is not possible to construct is signal flow graph with both input and output in normal order

   (b) The number of butterflies in the mth state is N/m

   (c) In-place computation requires storage of only 2N node data

   (d) Computation of a butterfly requires only one complex multiplication

3. Which of the following is true regarding the number of computations requires to compute an N-point DFT?

   a) N complex multiplications and N(N-1) complex additions

   b) N complex additions and N(N-1) complex multiplications

   c) N complex multiplications and N(N+1) complex additions

   d) N complex additions and N(N+1) complex multiplications

4. Divide-and-conquer approach is based on the decomposition of an N-point DFT into successivelysmaller DFTs. This basic approach leads to FFT algorithms.

   a) True

   b) False

5. Which is called a periodic convolution

   a) Circular convolution

   b) Linear convolution

   c) Both linear and circular convolution

   d) None of the above

6. In the following statements which is incorrect regarding FFT
   a) In FFT algorithm the N point DFT is decomposed into successively smaller DFTs.
   b) In N point DFT using Radix-2 FFT , the decimation is performed m times, where $m=\log_2 N$.
   c) Both DIT and DIF algorithms involve same number of computations.
   d) Bit reversing is required for only DIF algorithm.

7. The natural signals like speech signals, EEG,ECG,EMG, etc cannot be expressed by mathematical equations.
   a) True
   b) False

8. The convolution of FFT is called
   a) Fast convolution
   b) Slow convolution
   c) Linear convolution
   d) Circular convolution

9. In DFT computation using radix-2 FFT, the value of N should be
   a) $N=m^2$
   b) $N= 2^m$
   c) $N=2m$
   d) $N=m/2$

10. The number of complex addition in radix-2 FFT is given by
   a) $(N/2)\log_2 N$
   b) $N \log_2 N$
   c) $N(N-1)$
   d) N^2

VIII. Conclusion- Outcome

- To evaluate system response of a system using Z-transform, convolution methods, frequency transformation technique, DFT, DIF-FFT or DIT-FFT algorithm;
- Thorough understanding of frequency domain analysis of discrete time signals.
- To apply DFT for the analysis of digital signals & systems
- To apply DFT for the analysis of digital images.

IX. Video Link

1. **DFT and IDFT**
   https://www.youtube.com/watch?v=lovfWfCHMLc

2. **Impact of Phase in FFT**
   https://www.youtube.com/watch?v=G9ZCWSyLvYY

X. Assignment

Study the action of filters on a real image using FFT.