EXP.NO: 1

TO FIND DFT / IDFT OF GIVEN DT SIGNAL

AIM: To find Discrete Fourier Transform and Inverse Discrete Fourier Transform of given digital signal.

SOFTWARE: MATLAB /PYTHON/SCILAB

1.For only first exp give description of the software used

2.Write about libraries, functions and commands used

Basic equation to find the DFT of a sequence is given below.

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}$$

$$where \quad W_N^{nk} = e^{-j\frac{2\pi nk}{N}} \quad \text{[TWIDDLE FACTOR]}$$

Basic equation to find the IDFT of a sequence is given below. (Write it)

ALGORITHM:

Step I: Get the input sequence.

Step II: Find the DFT of the input sequence using direct equation of DFT.

Step III: Find the IDFT using the direct equation.

Step IV: Plot DFT and IDFT of the given sequence using command stem.

Step V: Display the above outputs.

EXP.NO: 2

TO FIND CIRCULAR CONVOLUTION USING DFT/IDFT METHOD

AIM: To find the circular convolution using DFT & IDFT method.

SOFTWARE: MATLAB /PYTHON/SCILAB

Write about libraries, functions and commands used

THEORY:

Convolution is a mathematical operation used to express the relation between input and output of an LTI system.

It relates input, output and impulse response of an LTI system as y(n)=x(n)∗h(n) Where

y (n) = output of LTI x (n) = input of LTI h (n) = impulse response of LTI

Discrete Convolution y(n)=x(n)∗h(n)

Discrete Convolution

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

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

Methods for circular convolution

i) Concentric circle method

ii) DFT/IDFT method

iii)Matrix multiplication method (explain using one example)

ALGORITHM:

Step I: Give input sequence x1[n].

Step II: Give impulse response sequence x2(n)

Step III: Find the convolution y[n] using DFT /IDFT method.

a. First find dft of x1n and x2n using twiddle factor matrix for DFT.
b. Let X1K is dft of x1n and X2K is dft of x2n.
c. Then YK=X1K*X2K
d. Then find IDFT of YK using Twiddle factor matrix for IDFT.
e. This will be yn i..e circularly convolved sequence yn

Step IV: Plot x1[n], x2[n], y[n].

EXP.NO: 3

To study Fast Fourier Transform Algorithms.

AIM: a) To find the DFT using DIT FFT algorithm of any 8-point sequence

b) To find IDFT using DIF FFT algorithm for the result obtained in part a)

SOFTWARE: MATLAB /PYTHON/SCILAB

Write about libraries, functions and commands used in this code

THEORY:

DFT of a sequence

(Write the conventional equation of DFT)

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$$

Where

N = Length of sequence.

k = Frequency Coefficient.

x(n) = Samples in time domain.

FFT: -Fast Fourier transform.

There are two methods.

1. Decimation in time (DIT) FFT.
2. Decimation in Frequency (DIF) FFT.

Why we need FFT?

=>to convert signal into individual spectral components and thereby provides frequency information about the signal.

The no of multiplications in DFT = N*N

The no of Additions in DFT = N (N-1).

For FFT. The no of multiplication = N/2 log ...N.

The no of additions = N log ...N.

ALGORITHM:

Step I: Give input sequence x[n] for N=8.

Step II: Find the FFT implementing signal flow graph equation for DIT FFT algorithm.

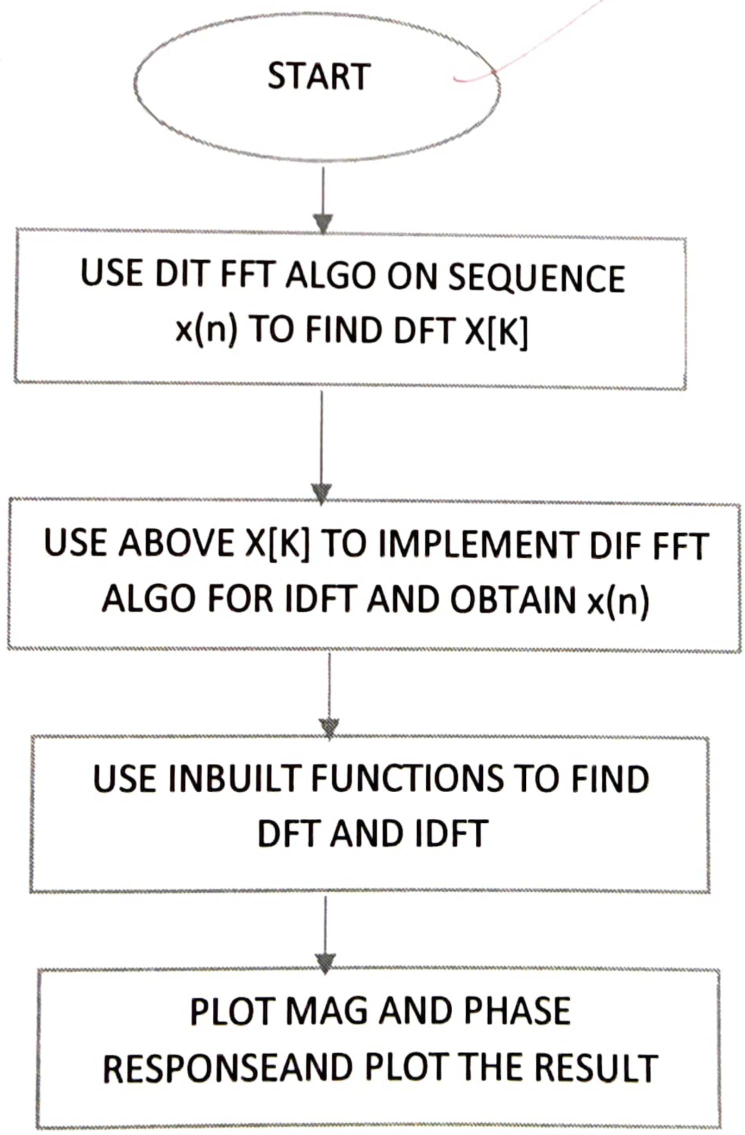
Step III: Find the IDFT implementing signal flow graph equation for DIF FFT algorithm.

Step IV: Also, find FFT and IFFT using MATLAB commands fft and ifft.

Step V: Plot magnitude and phase response.

Step V: Display and write the analysis of results.

FLOW CHART:



EXP.NO: 4

TO FIND FREQUENCY RESPONSE OF A GIVEN SYSTEM GIVEN IN (DIFFERENTIAL EQUATION FORM)

AIM: To find frequency response of a given system in differential equation form.

SOFTWARE: MATLAB /PYTHON/SCILAB

Write about libraries, functions and commands used in this code

THEORY:

Systems respond differently to inputs of different frequencies. Some systems may amplify components of certain frequencies, and attenuate components of other frequencies. The way that the system output is related to the system input for different frequencies is called the *frequency response* of the system.

Since the frequency response is a complex function, we can convert it to polar notation in the complex plane. This will give us a magnitude and an angle. We call the angle the *phase*.

Amplitude Response:

For each frequency, the magnitude represents the system's tendency to amplify or attenuate the input signal.

$$A(\omega) = |H(j\omega)|$$

Phase Response:

The phase represents the system's tendency to modify the phase of the input sinusoids.

$$\phi(\omega) = \angle H(j\omega).$$

The phase response, or its derivative the group delay, tells us how the system delays the input signal as a function of frequency.

ALGORITHM:

Step I: Give numerator coefficients of the given transfer function or difference equation.

Step II: Give denominator coefficients of the given transfer function or difference equation

Step III: Pass these coefficients to MATLAB command freqz to find frequency response.

Step IV: Find magnitude and phase response using MATLAB commands abs and angle.

Step V: Plot magnitude and phase response.

EXP.NO: 5

IMPLEMENTATION OF LP FIR FILTER

AIM: To study and determine the coefficient of FIR filter using window technique.

SOFTWARE: MATLAB /PYTHON/SCILAB

Write about libraries, functions and commands used in this code

THEORY:

FIR filters are digital filters with finite impulse response. They are also known as non-recursive digital filters as they do not have the feedback.

An FIR filter has two important advantages over an IIR design:

- Firstly, there is no feedback loop in the structure of an FIR filter. Due to not having a feedback loop, an FIR filter is inherently stable. Meanwhile, for an IIR filter, we need to check the stability.

- Secondly, an FIR filter can provide a linear-phase response. As a matter of fact, a linear-phase response is the main advantage of an FIR filter over an IIR design otherwise, for the same filtering specifications; an IIR filter will lead to a lower order.

FIR FILTER DESIGN

An FIR filter is designed by finding the coefficients and filter order that meet certain specifications, when a particular frequency response is desired, several different design methods are common:

1. Window design method
2. Frequency Sampling method
3. Weighted least squares design

WINDOW DESIGN METHOD

In the window design method, one first designs an ideal IIR filter and then truncates the infinite impulse response by multiplying it with a finite length window function. The result is a finite impulse response filter whose frequency response is modified from that of the IIR filter.

ALGORITHM:

Step I: Give input specification find the cutoff frequency

Step II: Determine the length N if not given.

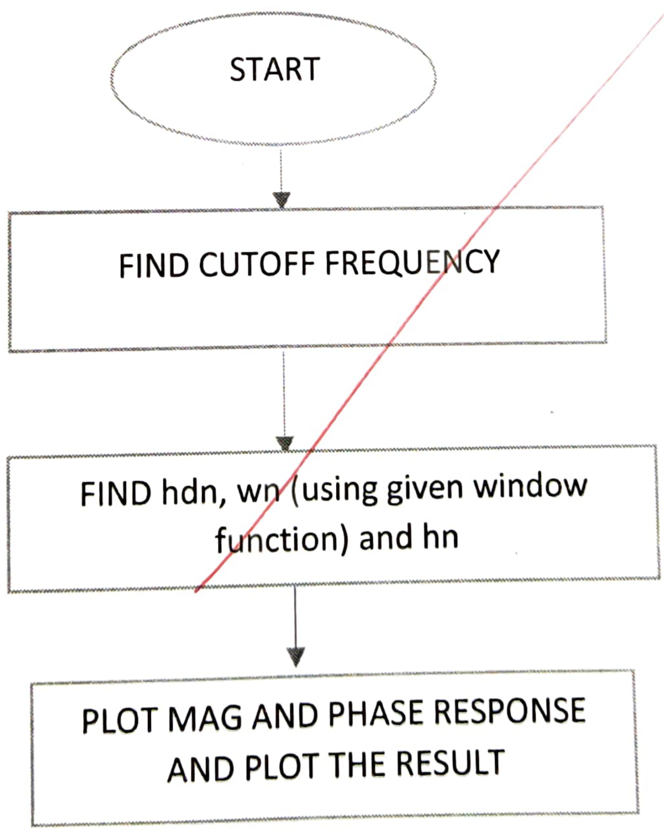
Step III: find hdn

Step IV: find hn by using hdn and wn.

Step V: Plot magnitude and phase response.

Step V: Display and write the analysis of results.

FLOW CHART:



EXP.NO: 6

IMPLEMENTATION OF LP IIR FILTER

Aim: To Implement LP IIR Filter For A Given Transfer Function.

Software: MATLAB /Python/SCILAB

Write about Libraries, Functions and Commands Used In This Code

Theory:

IIR Filters Are Digital Filters With Infinite Impulse Response. Unlike Fir Filters, They Have The Feedback (A Recursive Part Of A Filter) And Are Known As Recursive Digital Filters Therefore.

For This Reason, IIR Filters Have Much Better Frequency Response Than Fir Filters Of The Same Order. Unlike Fir Filters, Their Phase Characteristic Is Not Linear Which Can Cause A Problem To The Systems Which Need Phase Linearity. For This Reason, It Is Not Preferable To Use IIR Filters In Digital Signal Processing When The Phase Is Of The Essence. Otherwise, When The Linear Phase Characteristic Is Not Important, The Use Of IIR Filters Is An Excellent Solution.

There Is One Problem Known As A Potential Instability That Is Typical Of IIR Filters Only. Fir Filters Do Not Have Such A Problem As They Do Not Have The Feedback. For This Reason, It Is Always Necessary To Check After The Design Process Whether The Resulting IIR Filter Is Stable Or Not.

IIR Filter Design

For The Given Specifications To Design A Digital IIR Filter, First We Need To Design Analog Filter (Butterworth Or Chebyshev). The Resultant Analog Filter Is Transformed To Digital Filter By Using Either "Bilinear Transformation Or impulse Invariant Transformation".

Algorithm:

Step I : Enter The Pass Band Ripple (Rp) And Stop Band Ripple (Rs).

Step II : Enter The Pass Band Frequency (Wp) And Stop Band Frequency (Ws).

Step III : Get The Sampling Frequency (Fs).

Step IV: Calculate Normalized Pass Band Frequency, and Normalized Stop Band Frequency W1 and W2 Respectively. W1 = 2 * WP /Fs W2 = 2 * Ws /Fs Step

V : Make Use Of The Following Function To Calculate Order Of Filter Butterworth Filter Order [N,Wn]=Buttord(W1,W2,Rp,Rs) Chebyshev Filter Order [N,Wn]=Cheb1ord(W1,W2,Rp,Rs)

Step Vi : Design An Nth Order Digital Low Pass Butterworth Or Chebyshev Filter Using The Following Statements. Butterworth Filter [B, A]=Butter (N, Wn) Chebyshev Filter [B,A]=Cheby1 (N, 0.5, Wn)

Step Vii : Find The Digital Frequency Response Of The Filter By Using 'Freqz( )' Function

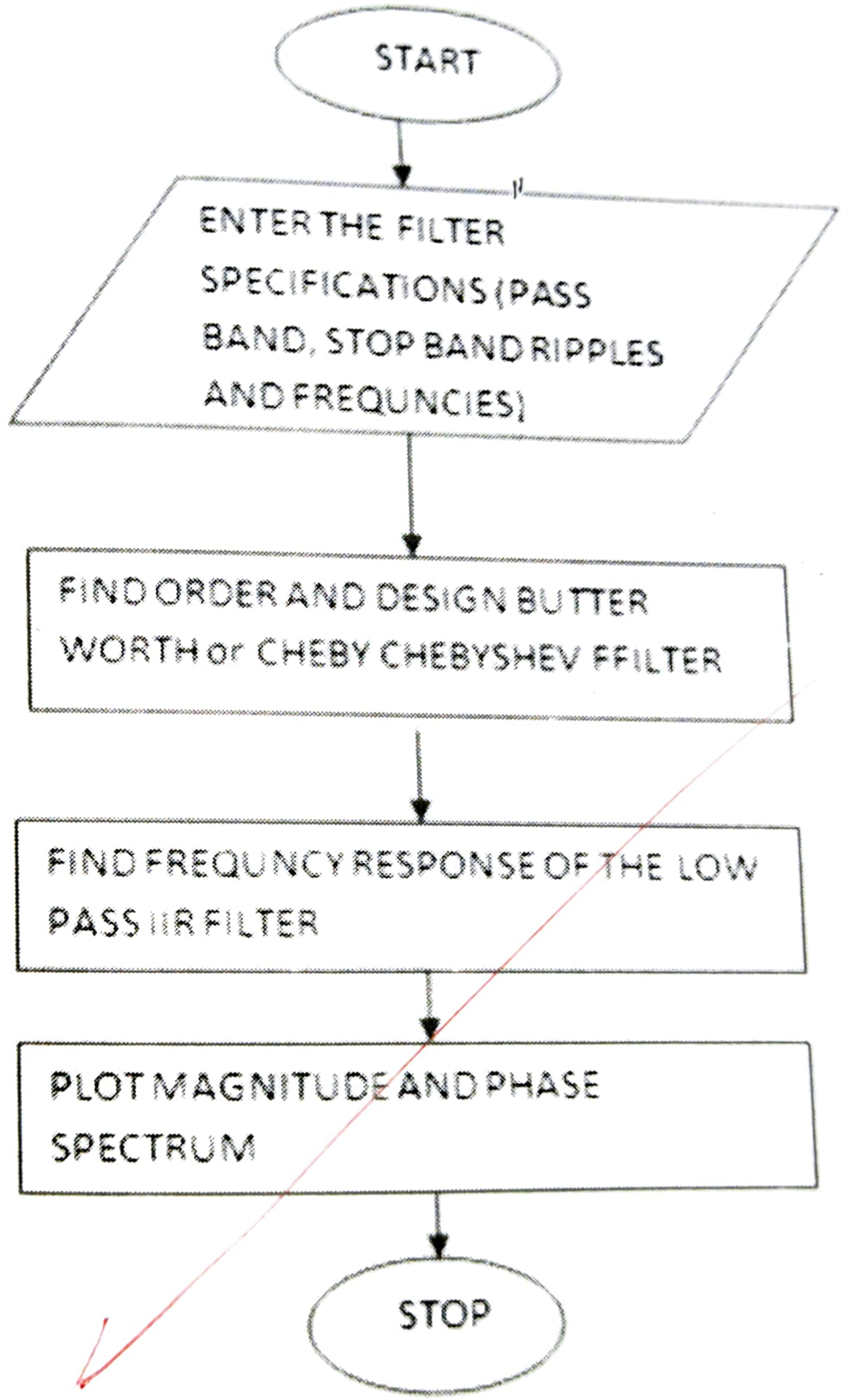
Step Viii : Calculate The Magnitude Of The Frequency Response In Decibels (Db) Mag=20*Log10 (Abs (H))

Step Ix : Plot The Magnitude Response [Magnitude In Db Vs Normalized Frequency]

Step X : Calculate The Phase Response Using Angle (H)

Step Xi : Plot The Phase Response [Phase In Radians Vs Normalized Frequency (Hz)].

Flow Chart:



EXP.NO: 7

UNDERSTANDING HOW TO READ A SOUND FILE AND ANALYSE THE EFFECT OF NOISE ON IT

AIM: To read and plot different sound signals and study the effect of addition of noise in test signal

SOFTWARE: MATLAB /PYTHON/SCILAB

Write about libraries, functions and commands used in this code

ALGORITHM:

Step I: Read a sound file

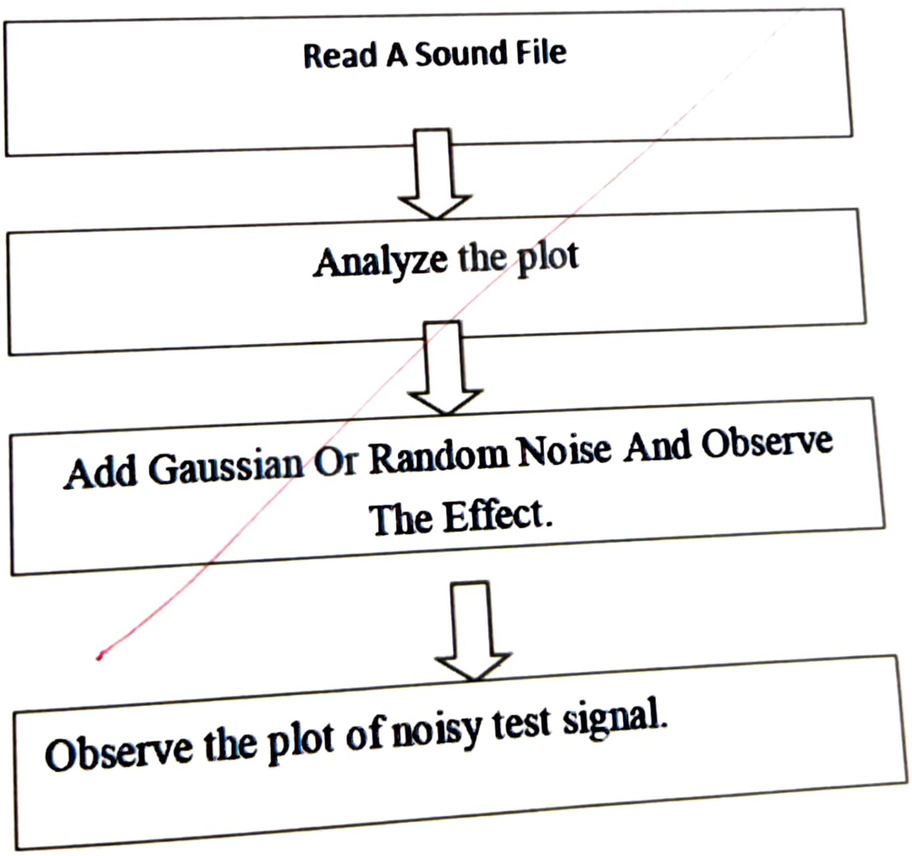
Step II: Analyze the plot

Step III: Add Gaussian or random noise

Step IV: Observe the effect.

Step V: Observe the plot of noisy test signal.

FLOW CHART:



EXP.NO: 8

IMPLEMENTATION OF DECIMATION PROCESS

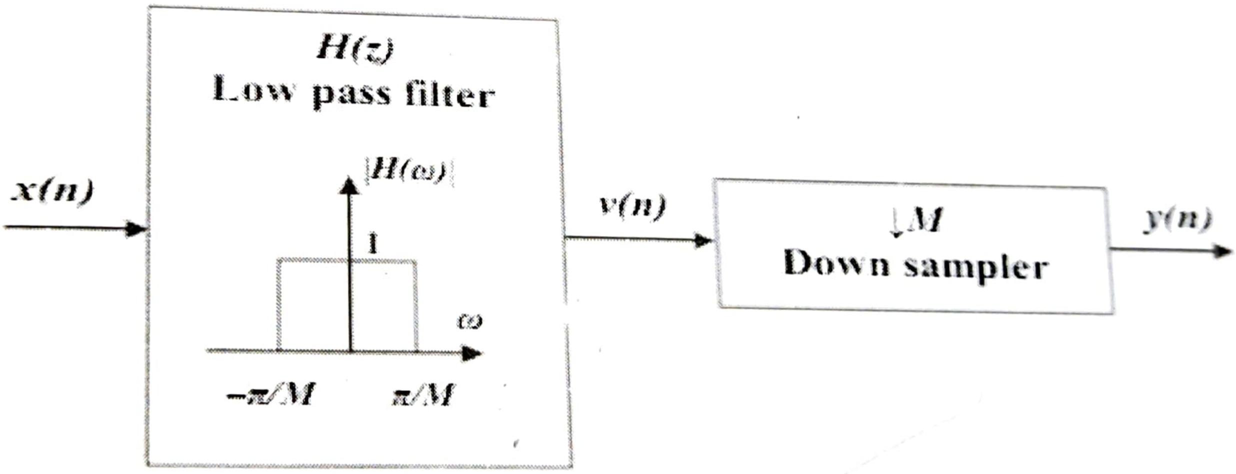
AIM: Program to verify the decimation of given sequence.

SOFTWARE: MATLAB /PYTHON/SCILAB

Write about libraries, functions and commands used in this code

THEORY:

"Decimation" is the process of reducing the sampling rate. "Down sampling" is a more specific term which refers to just the process of throwing away samples, without the lowpass filtering operation. The most immediate reason to decimate is simply to reduce the sampling rate at the output of one system so a system operating at a lower sampling rate can input the signal. But a much more common motivation for decimation is to reduce the cost of processing: the calculation and/or memory required to implement a DSP system generally is proportional to the sampling rate, so the use of a lower sampling rate usually results in a cheaper implementation.



ALGORITHM:

Step I : Define down sampling factor and input frequencies f1 and f2

Step II : Represent input sequence with frequencies f1 and f2

Step III : Perform the decimation on the input signal using the Matlab command decimate.

Step IV : Plot the input and output sequence

FLOW CHART: