Lab 6 : Efficient computation of DFT and IDFT using FFT

Aim: To obtain the efficient computation of DFT and IDFT using FFT

Software Requirement: SCI Lab

Theory:

DFT:

Discrete Fourier Transform (DFT) is used for performing frequency analysis of discrete time signals. DFT gives a discrete frequency domain representation whereas the other transforms are continuous in frequency domain. The N point DFT of discrete time signal x[n] is given by the equation

$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}}$ for k=0,1,2,…N-1

The inverse DFT allows us to recover the sequence x[n] from the frequency samples

$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{\frac{-j2\pi k}{N}}$ for n=0,1,2,…N-1

FFT:

A fast Fourier transform (FFT) is an efficient algorithm to compute the discrete Fourier transform (DFT) and its inverse. FFTs are of great importance to a wide variety of applications, from digital signal processing and solving partial differential equations to algorithms for quick multiplication of large integers. Evaluating the sums of DFT directly would take $O(N^2)$ arithmetical operations. An FFT is an algorithm to compute the same result in only O(N log N) operations. In general, such algorithms depend upon the factorization of N, but there are FFTs with O(N log N) complexity for all N, even for prime N. Since the inverse DFT is the same as the DFT, but with the opposite sign in the exponent and a 1/N factor, any FFT algorithm can easily be adapted for it as well.

Algorithm:

1) Get the input sequence
2) Find the FFT of the input sequence using SciLAB function.
3) Find the IFFT of the input sequence using SciLAB function.
4) Display the above outputs.

Program:

```
// program for calculation of FFT of a signal
clc ;
clf ;
clear all;
N = input('Enter the value of N');
x = input ('enter input sequence');
y = fft(x);
A = real(y);
B = imag(y);
mag = abs(y);
x1 = atan(imag(y),real(y));
phase = x1 *(180/ %pi ) ;
disp ('the resultant FFT sequence is ' ) ;
disp (y);
disp ('the magnitude response is ') ;
disp ( mag ) ;
disp ('the phase response is') ;
disp (phase) ;
z = ifft ( y ) ;
disp ('the resultant IFFT sequence is') ;
disp ( z );
subplot (3 ,2 ,1) ;
plot2d3 ( x );
title ( 'input sequence') ;
subplot (3 ,2 ,2) ;
plot2d3 ( A );
title ( 'FFT real sequence ' ) ;
subplot (3 ,2 ,3) ;
plot2d3 ( B );
title ('FFT imaginary sequence ' ) ;
subplot (3 ,2 ,4) ;
plot2d3 ( mag ) ;
title ( 'magnitude response ' ) ;
subplot (3 ,2 ,5) ;
plot2d3 ( phase ) ;
title ( 'phase response ') ;
subplot (3 ,2 ,6) ;
plot2d3 ( x );
```

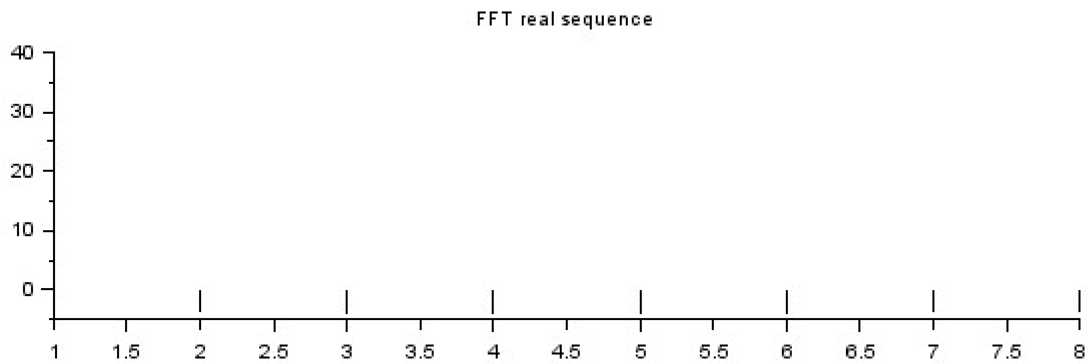
```
title ( 'IFFT sequence ' ) ;
```

Results:

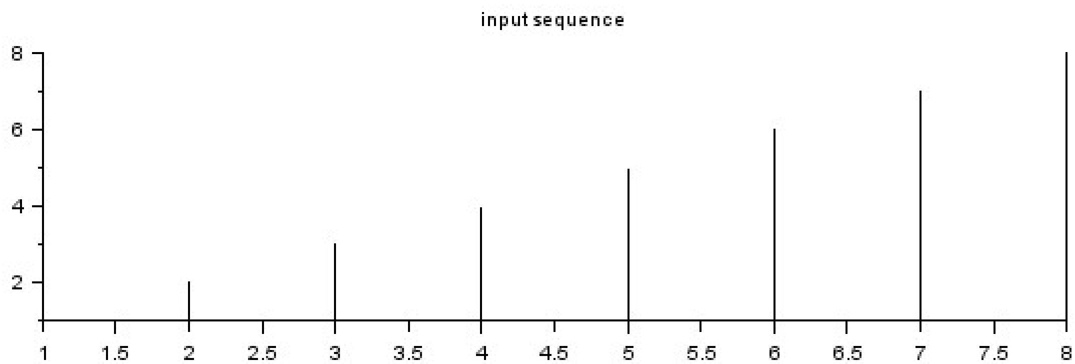
Enter the value of N 8

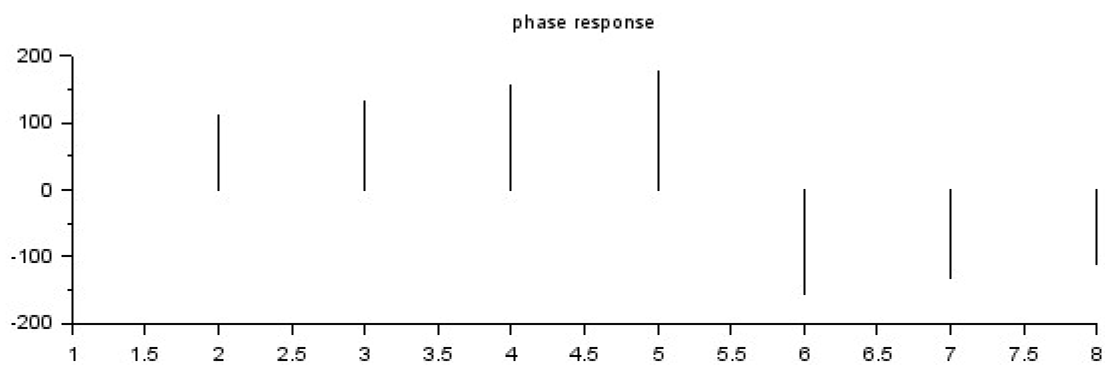
enter input sequence [1 2 3 4 5 6 7 8]

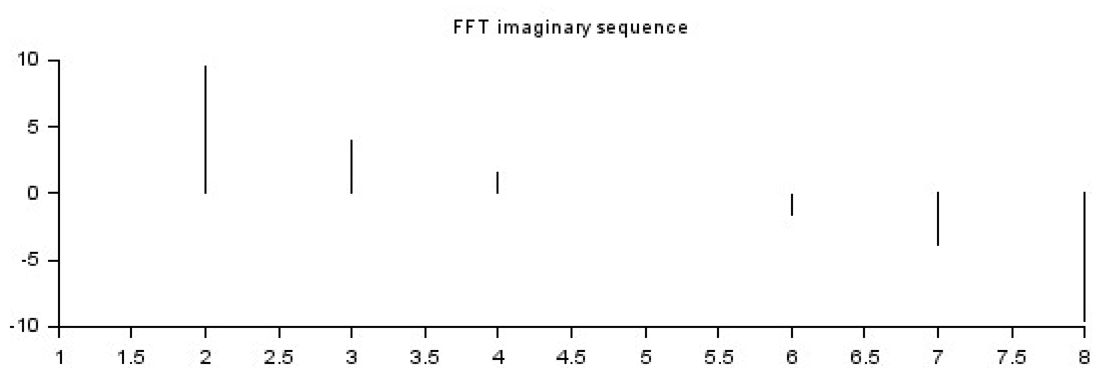
```
 "the resultant FFT sequence is "

       column 1 to 5
  36. + 0.i  -4. + 9.6568542i  -4. + 4.i  -4. + 1.6568542i  -4. + 0.i
       column 6 to 8
 -4. - 1.6568542i  -4. - 4.i  -4. - 9.6568542i

 "the magnitude response is "

       column 1 to 7
  36.   10.452504   5.6568542   4.3295688   4.   4.3295688   5.6568542
       column 8
  10.452504

 "the phase response is"

  0.   112.5   135.   157.5   180.  -157.5  -135.  -112.5

 "the resultant IFFT sequence is"

  1.   2.   3.   4.   5.   6.   7.   8.
```

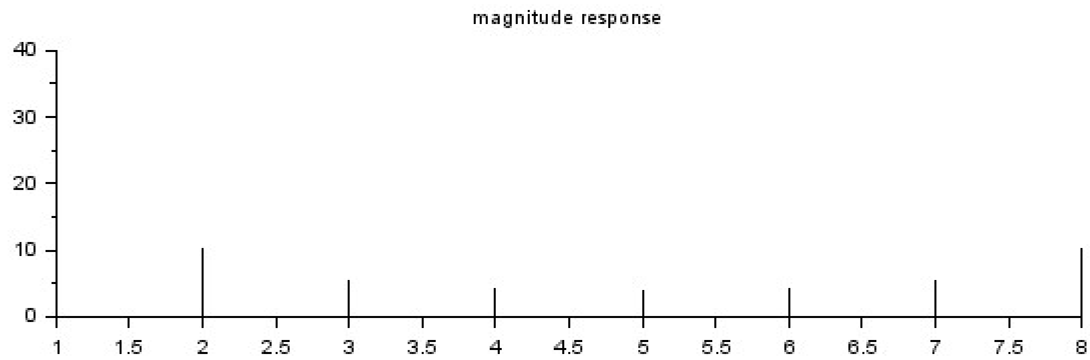

FFT real sequence
40
30
20
10
0
1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 6.5 7 7.5 8

:


input sequence
8
6
4
2
1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 6.5 7 7.5 8


phase response
200
100
0
-100
-200
1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 6.5 7 7.5 8


FFT imaginary sequence
10
5
0
-5
-10
1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 6.5 7 7.5 8



Pre-lab questions:

1. Why do we need Fourier transform in DSP?
2. What is the need of FFT ?
3. What's the difference between FFT and DFT?
4. What is "decimation-in-time" versus "decimation-in-frequency"?

Post-Lab questions:

1. Compute the 8-point FFT of the sequence x (n) = [1 2 1 2 3 4 3 4] using DIT-FFT and DIF-FFT and verify the result using Sci Code.

Result:

Lab 7 : Design of digital FIR Low Pass ,High Pass , Band Pass, band Stop filter using rectangular window

Aim: Design and implementation of FIR Filter (LP /HP /BP /BS) to meet given specifications
Using Windowing technique

a. Rectangular window

Software Requirement: SCI Lab

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, which can be in the time-domain (e.g. a matched filter) and/or the frequency domain (most common). Matched filters perform a cross-correlation between the input signal and a known pulse-shape. The FIR convolution is a cross-correlation between the input signal and a time-reversed copy of the impulse-response. Therefore, the matched-filter's impulse response is "designed" by sampling the known pulse-shape and using those samples in reverse order as the coefficients of the filter. When a particular frequency response is desired, several different design methods are common:

1. Window design method

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:

1) Get the input analog cut off frequency.
2) Get the input sampling frequency.
3) Get the filter order.
4) Find the digital cut off frequency.
5) Normalize the digital cut off frequency.
6) Using built in filter function find the time domain filter coefficients, frequency domain filter response and Frequency grid for LPF,HPF,BPF and BSF.
7) Plot the magnitude response of the filter with respect to Normalized Digital Frequency and Analog Frequency.

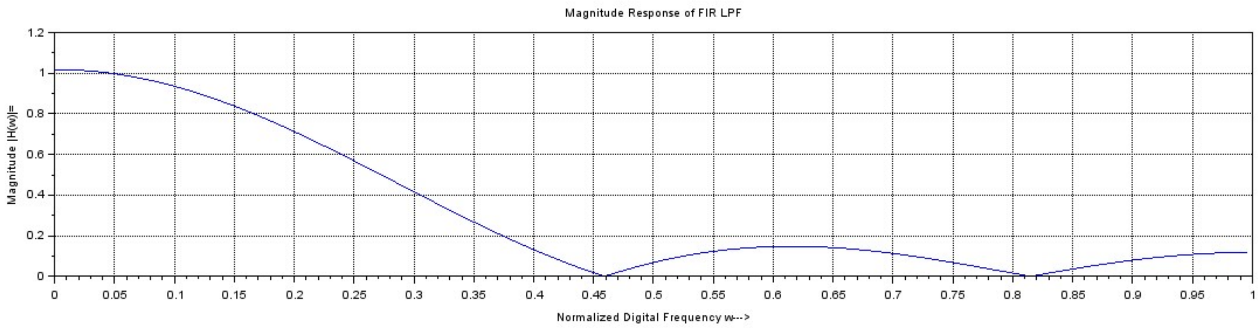
Program(a)Low pass filter

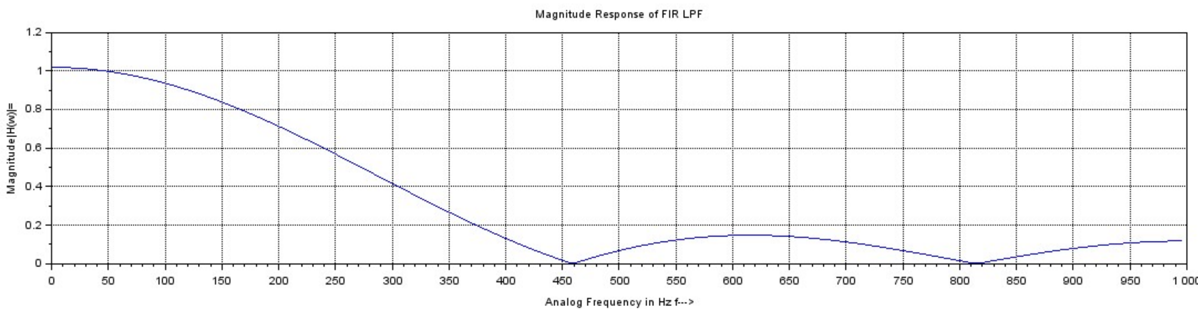
```
// To Design a Low Pass FIR Filter
// F i l t e r Length =5 , Order = 4
//Window = Rectangular Window
clc ;
clear ;
xdel(winsid());
fc=input('Enter Analog cutoff freq.in Hz=')
fs=input('Enter Analog sampling freq.in Hz=')
M=input('Enter order of filter =')
w=(2*%pi)*(fc/fs);
disp(w,'Digital cut off frequency in radians,cycles/samples');
wc=w/%pi;
```

```
disp(wc,'Normalized digital cut off frequency in cycles/samples');
[wft,wfm,fr]=wfir('lp',M+1,[wc/2,0],'re',[0,0]);
disp(wft,'Impulse Response of LPF FIR Filter:h[n]=');
// Pl o t t i n g the Magnitude Response of LPF FIR Filter
subplot(2,1,1);
plot(2*fr,wfm);
xlabel('Normalized Digital Frequency w--->')
ylabel('Magnitude  |H(w)|=')
title('Magnitude Response of FIR LPF')
xgrid(1)
subplot(2,1,2)
plot(fr*fs,wfm)
xlabel('Analog Frequency in Hz f--->')
ylabel('Magnitude|H(w)|=')
title('Magnitude Response of FIR LPF')
xgrid(1)
```

Results

```
Enter Analog cutoff freq.in Hz=250

Enter Analog sampling freq.in Hz=2000

Enter order of filter =4


   0.7853982

  "Digital cut off frequency in radians,cycles/samples"

   0.25

  "Normalized digital cut off frequency in cycles/samples"

   0.1591549   0.2250791   0.25   0.2250791   0.1591549

  "Impulse Response of LPF FIR Filter:h[n]="
```





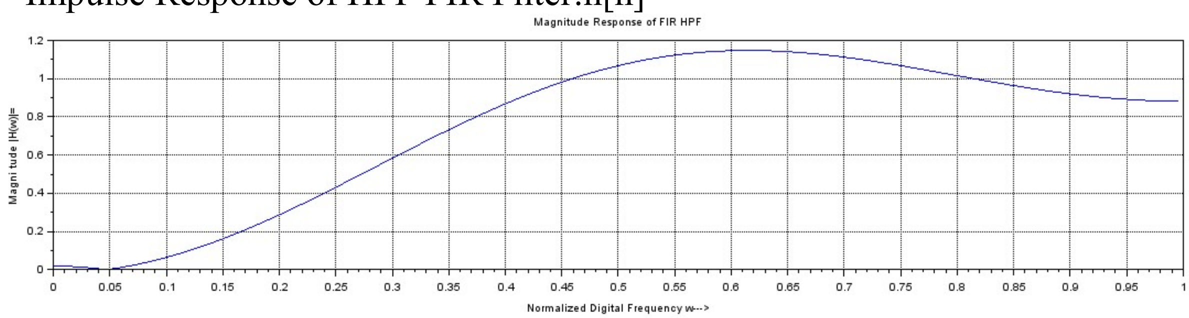
Program:(b)High pass filter

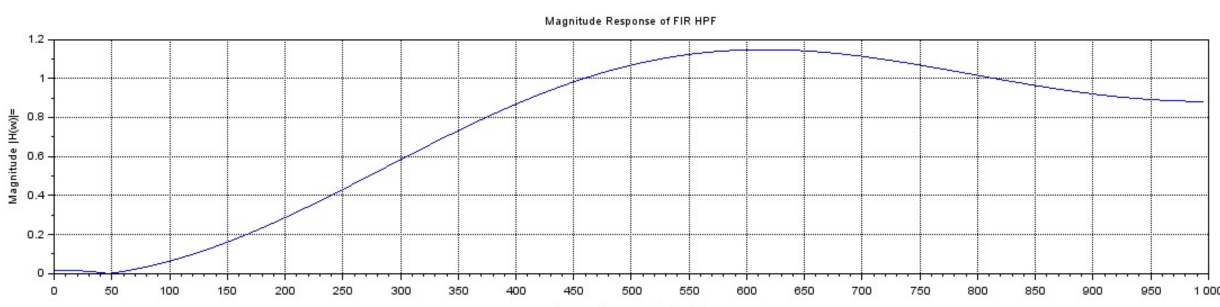
```
// To De s ign an High Pas s FIR F i l t e r
// Filter Length =5 , Order = 4
//Window = Rectangular Window
clc;
clear;
xdel(winsid());
fc=input('Enter Analog cut off freq in Hz=')// 250
fs=input('Enter Analog sampling freq in Hz=')//2000
M=input('Enter order of filter =')//4
w=(2*%pi)*(fc/fs);
disp(w,'Digital cut off frequency in radians cycles/samples');
wc=w/%pi;
disp(wc,'Normalized digital cut off frequency in cycles/samples');
[wft,wfm,fr]=wfir('hp',M+1,[wc/2,0],'re',[0,0]);
disp(wft,'Impulse Response of HPF FIR Filter:h[n]=');
// Pl o t t i n g the Magni tude Re spons e o f HPF FIR F i l t e r
subplot(2,1,1)
plot(2*fr,wfm)
```

```
xlabel('Normalized Digital Frequency w--->')
ylabel('Magni tude |H(w)|=')
title('Magnitude Response of FIR HPF')
xgrid(1)
subplot(2,1,2)
plot(fr*fs,wfm)
xlabel('Analog Frequency in Hz f--->')
ylabel ('Magnitude |H(w)|=')
title('Magnitude Response of FIR HPF')
xgrid(1)
```

Results:

Enter Analog cut off freq in Hz=250
Enter Analog sampling freq in Hz=2000
Enter order of filter =4
 0.7853982
 "Digital cut off frequency in radians cycles/samples"
 0.25
 "Normalized digital cut off frequency in cycles/samples"
 -0.1591549 -0.2250791 0.75 -0.2250791 -0.1591549
 "Impulse Response of HPF FIR Filter:h[n]="





Program:(c)Band pass filter

```
// To Design an Band Pass FIR Filter
// F i l t e r Length =5 , Order = 4
//Window = Rectangular Window
clc ;
clear;
xdel( winsid());
fc1=input('Enter Analog lower cut off freq.in Hz=') // 250
fc2=input('Enter Analog higher cut off freq. in Hz=') // 600
fs=input('Enter Analog sampling freq.in Hz=')//2000
M=input('Enter order of filter =')//4
w1=(2*%pi)*(fc1/fs);
w2=(2*%pi)*(fc2/fs);
disp(w1,'Digital lower cut off frequency in radians cycles/samples');
disp(w2,'Digital higher cut off frequency in radians cycles/samples');
wc1=w1/%pi;
wc2= w2/%pi;
disp(wc1,'Normalized digital lower cut off frequency in cycles/ samples');
disp(wc2,'Normalized digital higher cut off frequency in cycles / samples');
[wft,wfm,fr]=wfir('bp' ,M+1,[wc1/2,wc2/2],'re',[0,0]);
disp(wft,'Impulse Response of BPF FIR Filter : h[ n]= ');
// Pl o t t i n g the Magni tude Re sponse of HPF FIR Filter
subplot(2,1,1)
plot(2*fr,wfm)
xlabel('Normalized Digital Frequency w--->' )
ylabel('Magnitude|H(w)|=')
title('Magnitude Response of FIR BPF')
xgrid(1)
subplot(2,1,2)
plot(fr*fs,wfm)
xlabel('Analog Frequency in Hz f --->' )
ylabel('Magnitude  |H(w)|= ')
title ('Magnitude Response of FIR BPF')
xgrid(1)
```

Results:
Enter Analog lower cut off freq.in Hz=250

Enter Analog higher cut off freq. in Hz=600

Enter Analog sampling freq.in Hz=2000

Enter order of filter =4

0.7853982

"Digital lower cut off frequency in radians cycles/samples"

1.8849556

"Digital higher cut off frequency in radians cycles/samples"

0.25

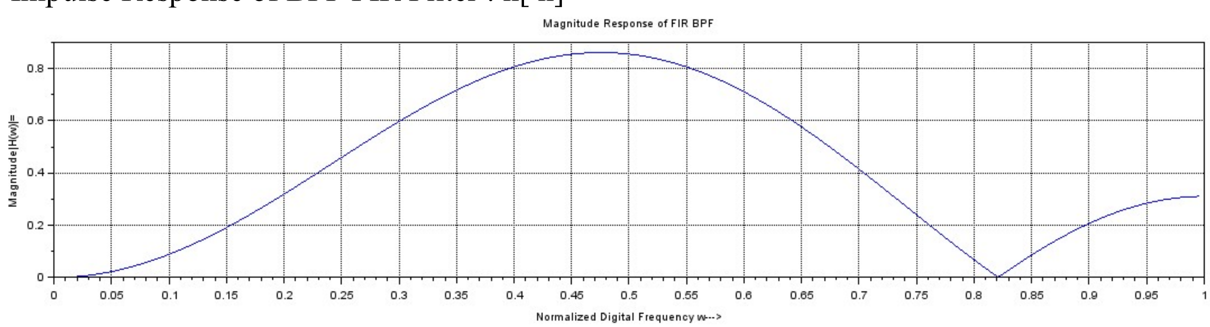
"Normalized digital lower cut off frequency in cycles/ samples"

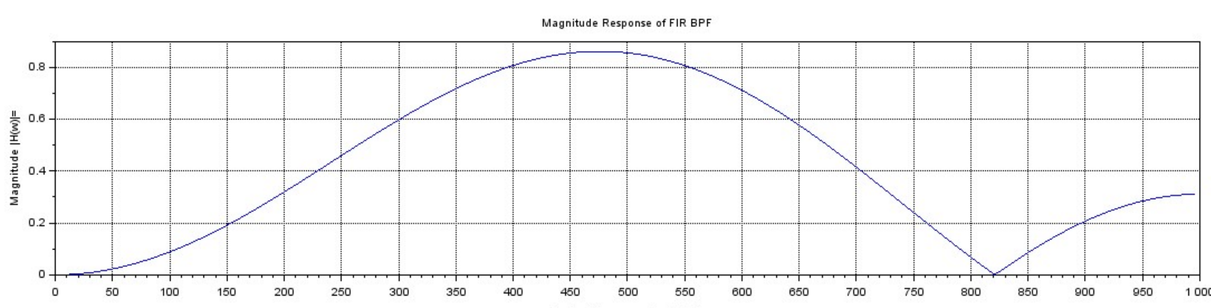
0.6

"Normalized digital higher cut off frequency in cycles / samples"

-0.2527039   0.0776516   0.35   0.0776516 -0.2527039

"Impulse Response of BPF FIR Filter : h[n]= "





Program:(d)Band stop filter

```
// Filter Length =5 , Order = 4
//Window = Rectangular Window
clc ;
clear ;
xdel ( winsid ());
fc1 = input ('Enter Analog lower cut off freq in Hz=') // 250
fc2 = input ('Enter Analog higher cut off freq in Hz=') // 600
fs = input ('Enter Analog sampling freq in Hz=') //2000
M = input ('Enter order of filter =') // 4
w1 = (2* %pi )*( fc1/fs);
w2 = (2* %pi )*( fc2/fs);
disp (w1,'Digital lower cut off frequency in radians cycles/samples' );
disp(w2 , 'Digital higher cut off frequency in radians.cycles/samples ' );
wc1 = w1/%pi;
wc2 = w2/%pi;
disp (wc1 , 'Normalized digital lower cut off frequency in cycles / samples
' );
disp (wc2 , 'Normalized digital higher cut off frequency in cycles / samples
' );
[wft,wfm,fr]= wfir('sb',M+1,[wc1/2,wc2/2],'re',[0,0]);
disp (wft , 'Impulse Response of BSF FIR Filter : h [ n]= ' );
// Pl o t t i n g the Magni tude Re spons e o f HPF FIR F i l t e r
subplot (2 ,1 ,1)
plot (2*fr , wfm )
 xlabel ( ' Normalized Digital Frequency w--->' )
ylabel ( 'Magnitude |H(w)|= ' )
title ( 'Magnitude Response of FIR BSF ' )
xgrid (1)
subplot (2 ,1 ,2)
plot (fr*fs , wfm )
xlabel ( ' Analog Frequency in Hz f --->')
ylabel ('Magnitude |H(w) |= ')
title ('Magnitude Response of FIR BSF')
xgrid (1)
```

Results:

Enter Analog lower cut off freq in Hz=250

Enter Analog higher cut off freq in Hz=600

Enter Analog sampling freq in Hz=2000

Enter order of filter =4

0.7853982

"Digital lower cut off frequency in radians cycles/samples"

1.8849556

"Digital higher cut off frequency in radians.cycles/samples "

0.25

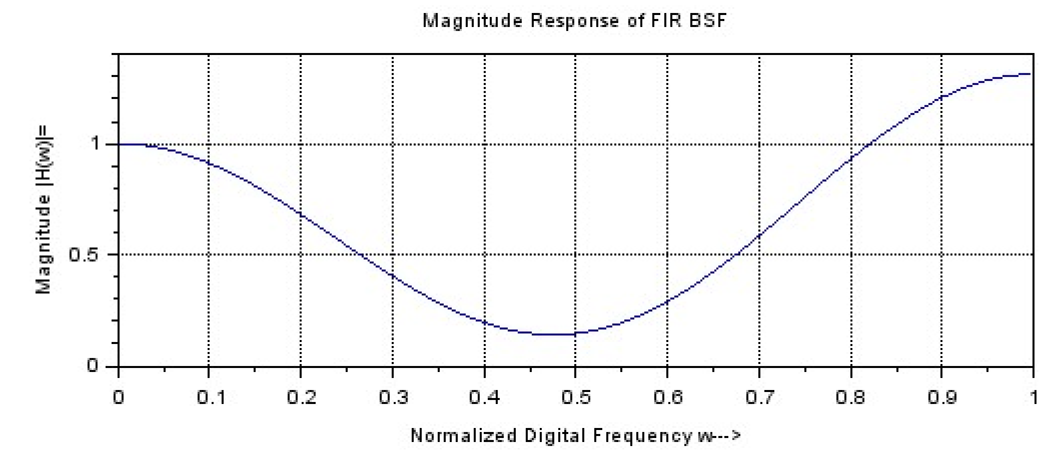
"Normalized digital lower cut off frequency in cycles / samples "

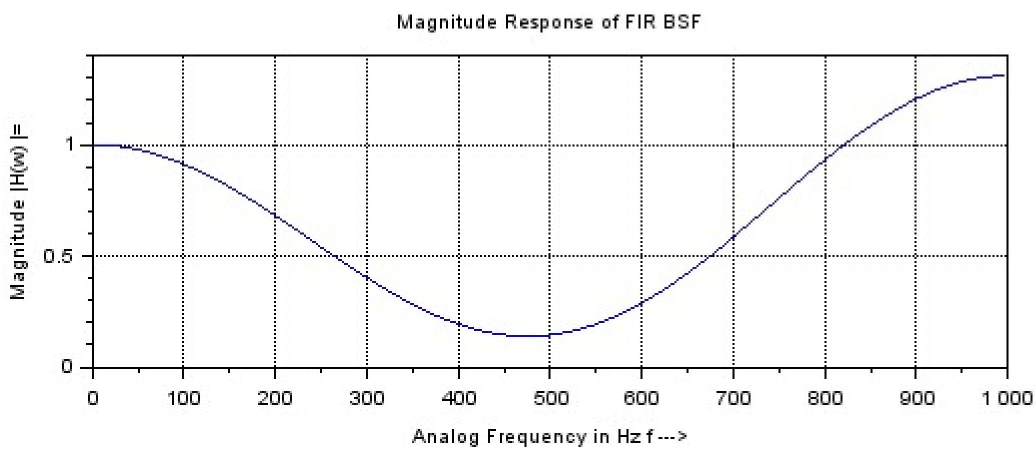
0.6

"Normalized digital higher cut off frequency in cycles / samples "

0.2527039 -0.0776516   0.65 -0.0776516   0.2527039

"Impulse Response of BSF FIR Filter : h [n]= "


Magnitude Response of FIR BSF
Magnitude |H(w)|=
Normalized Digital Frequency w--->
0
0.5
1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


Magnitude Response of FIR BSF
Magnitude |H(w) |=
Analog Frequency in Hz f --->
0
0.5
1
100
200
300
400
500
600
700
800
900
1 000