DEPARTMENT OF
ELECTRONICS & COMMUNICATION ENGINEERING
LAB MANUAL
DIGITAL SIGNAL PROCESSING LAB
IV - B. Tech. I - Semester

PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY
(Autonomous, Accredited by NBA & NAAC, an ISO 9001:2008 certified institution)
(Sponsored by Siddhartha Academy of General & Technical Education)
VIJAYAWADA – 520 007,
ANDHRA PRADESH

DSP LAB MANUAL

PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DIGITAL SIGNAL PROCESSING LAB

LIST OF EXPERIMENTS

Part – A: (Using MATLAB)

1. Linear Convolution of two sequences.
2. Circular Convolution of two sequences
3. DFT and IDFT of the given sequences.
4. Frequency response of a discrete time system described by a difference equation.
5. Fast Fourier Transform.
6. Determination of Power Density Spectrum of a given signal.
7. IIR Low Pass Digital Filter design.
8. IIR High Pass Digital Filter design.
9. FIR Low Pass Digital Filter design.
10. FIR High Pass Digital Filter design.

Part – B: (Using Code Composer Studio and TMS320C6713 DSP Starter Kit)

11. Linear Convolution.
12. Circular Convolution.
13. Generation of Sine wave & Square wave.

Part – C: Additional Programs

14. Implementation of Interpolation and Decimation.
15. Conversion of CD data to DVD data.
16. Sum of Sinusoidal signals (Gibb's Phenomenon)
17. M-Point Moving Average Filter Design

1. LINEAR CONVOLUTION OF TWO SEQUENCES

Aim: To perform linear convolution of given sequences using MATLAB.

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Step : 1 Get/Read the samples of x[n] to x.

Step : 2 Get/Read the samples of h[n] to h.

Step : 3 Get/Read the length of x to L.

Step : 4 Get/Read the length of h to M.

Step : 5 Get/Read the starting time of x[n] to N1.

Step : 6 Get/Read the starting time of h[n] to N3.

Step : 7 Determine N2= N1+L-1 and N4= N3+M-1.

Step : 8 Define time vector n as N1:N2.

Step : 9 Plot the Graph of x[n].

Step : 10 Define time vector n as N3:N4.

Step : 11 Plot the Graph of h[n].

Step : 12 Define time vector n as N1+N3:N2+N4.

Step : 13 Determine the convolution of x and h using

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

Use two for loops for the above expression

Step : 14 Plot the graph of y[n].

Procedure:

1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Input:

Enter the samples of x[n]: [1 2 3 4]

Enter the samples of h[n]: [1 2 1 2]

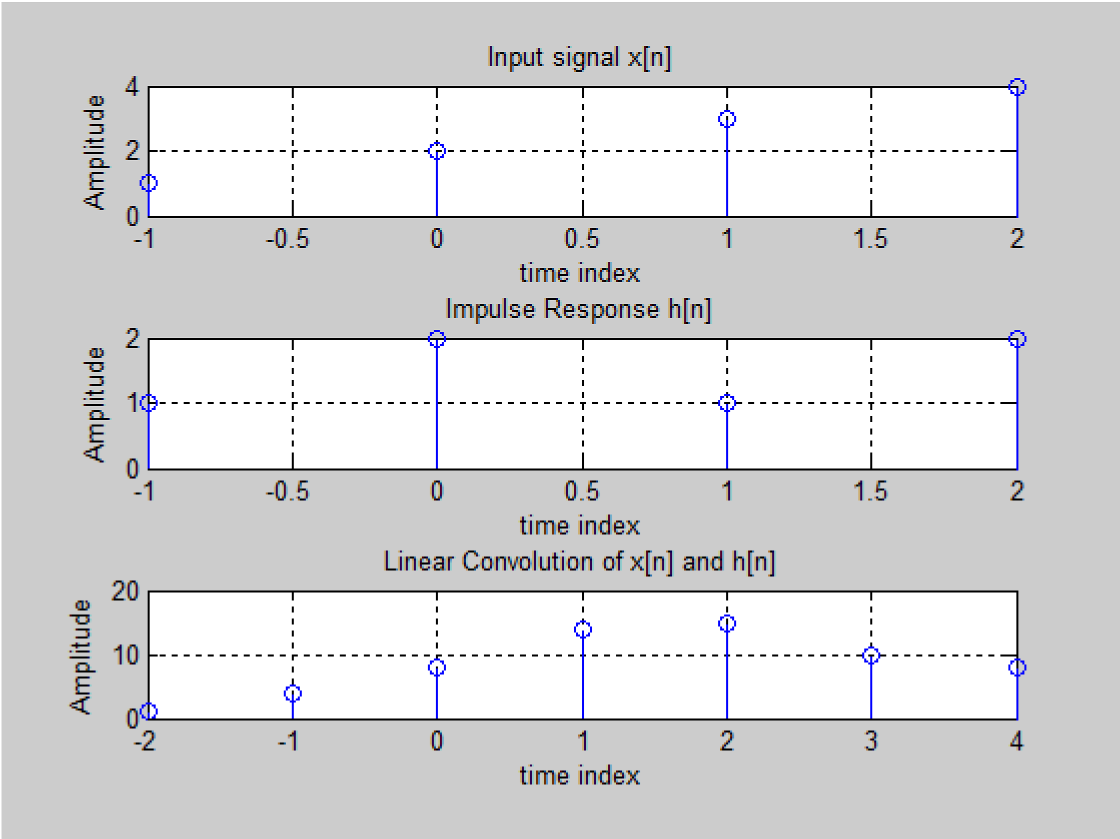
Enter starting time of x[n]: -1

Enter starting time of h[n]: -1

Output:

Linear convolution of x[n] and h[n] is

y = 1 4 8 14 15 10 8



Result: Hence linear convolution of given sequences is performed and output is observed using MATLAB.

2. CIRCULAR CONVOLUTION OF TWO SEQUENCES

Aim: To perform circular convolution of given sequences using MATLAB.

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Step : 1 Get/Read the samples of $x_1[n]$ to x1.

Step : 2 Get/Read the samples of $x_2[n]$ to x2.

Step : 3 Get/Read the length of x1 to N1.

Step : 4 Get/Read the length of x2 to N2.

Step : 5 Define time vector n as 0:N1-1.

Step : 6 Plot the Graph of x1[n].

Step : 7 Define time vector n as 0:N2-1.

Step : 8 Plot the Graph of x2[n].

Step : 9 If N1 > N2, pad N1-N2 number of zeros to x2

Else, pad N2-N1 number of zeros to x1.

Step : 10 Determine the maximum of (N1, N2) and store it N.

Step : 11 Define time vector n as 0:N-1.

Step : 12 Determine the circular convolution of x1 and x2 using

$$y[n] = \sum_{m=0}^{N-1} x(m) * (h(n-m))N$$

Use two for loops for the above expression

Step : 13 Plot the graph of y[n].

Procedure:

1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type 'edit' in the MATLAB prompt '>>' that appears in the Command window.
3. Write the program in the 'Edit' window and save it in 'M-file'.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Input:

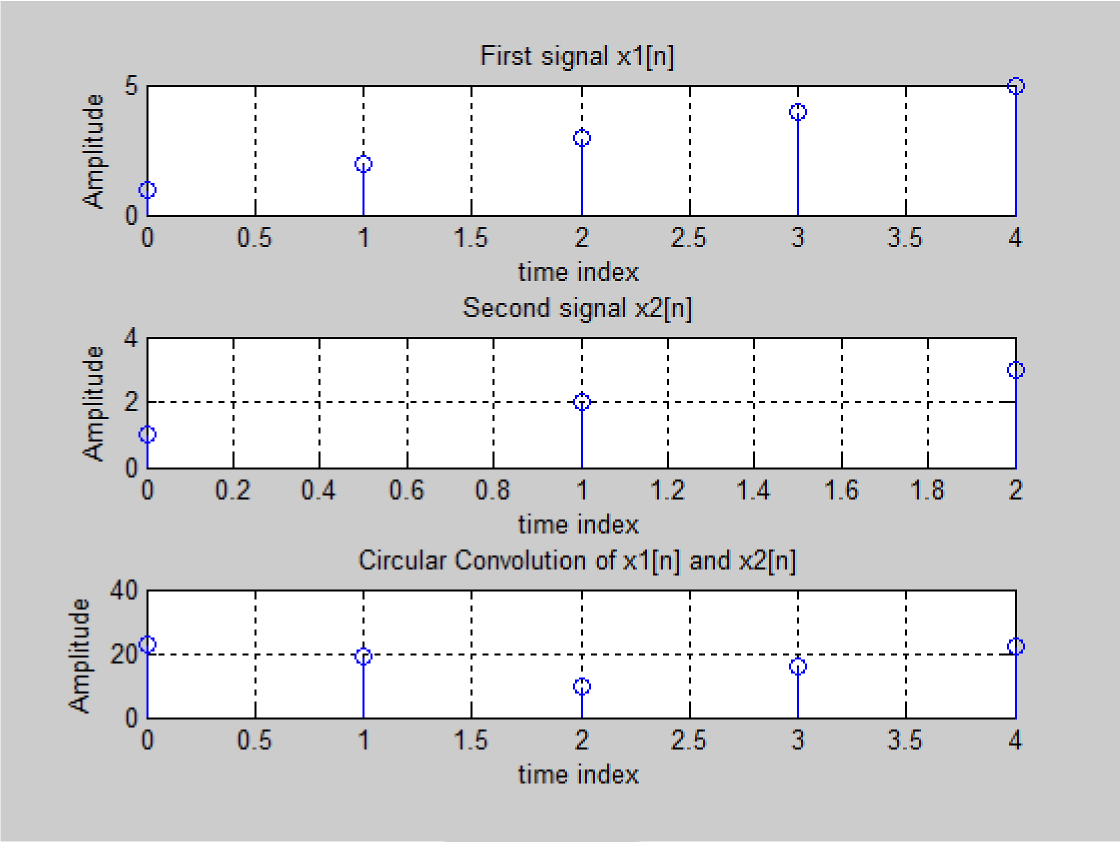
Enter the samples of x1[n]: [1 2 3 4 5]

Enter the samples of x2[n]: [1 2 3]

Output:

Circular Convolution of x1[n] and x2[n] is

y = 23 19 10 16 22



Result: Hence circular convolution of given sequences is performed and output is observed using MATLAB.

3. DFT/IDFT OF GIVEN SEQUENCE

Aim: To determine and plot the DFT/IDFT of a given sequence using MATLAB.

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Step : 1 Get/Read the samples of x[n] to x.

Step : 2 Find the length of x and store it in N.

Step : 3 Initialize the arrays xk & ixk with same size as that of x.

Step : 4 Find the DFT of the sequence x using

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

Use two for loops for the above expression

Step : 5 Find the IDFT of the sequence xk using

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

Use two for loops for the above expression

Step : 6 Plot the Graphs of x, xk and ixk.

Procedure:

1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

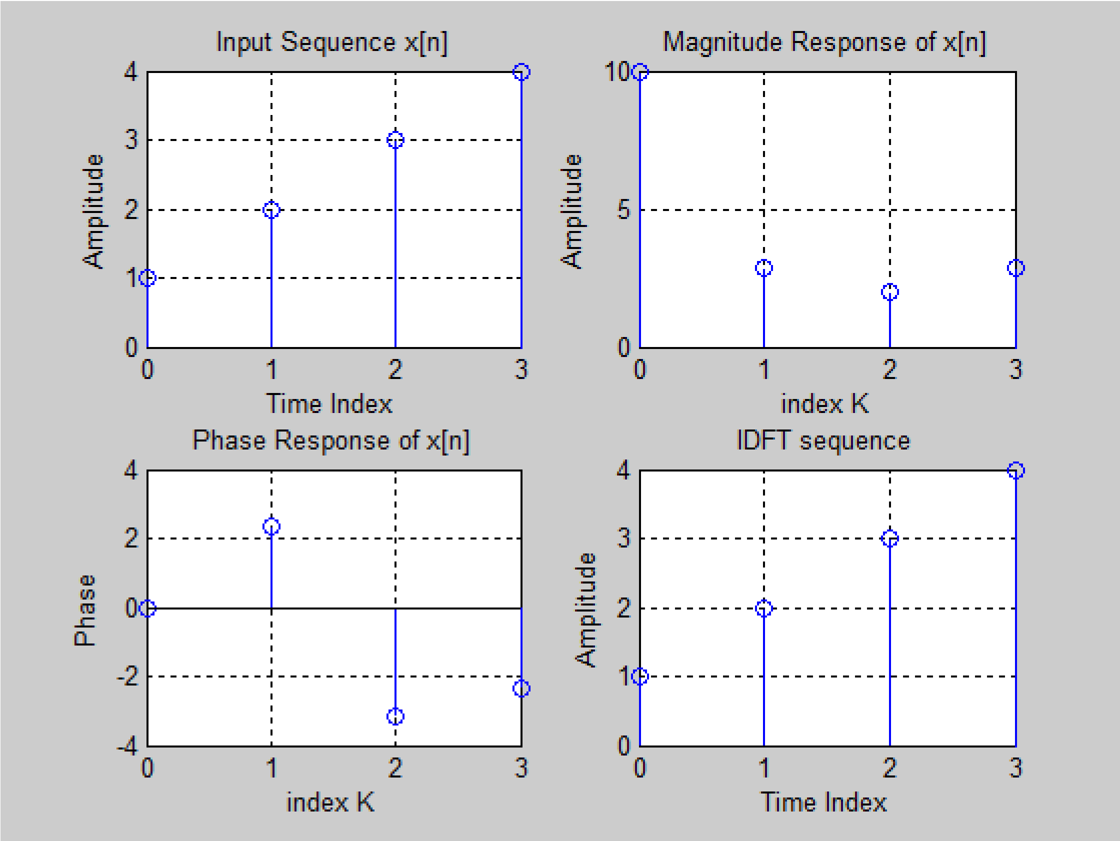
Input:

Enter the sequence x(n): [1 2 3 4]

Output:

DFT of the given sequence x[n] is

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



Result: Hence DFT/IDFT of given sequence is determined and outputs are observed using MATLAB.

4. FREQUENCY RESPONSE OF A DISCRETE TIME SYSTEM

Aim: To plot & observe the frequency response of first order and second order discrete-time LTI systems described by the difference equations

a) y(n) - 0.5 y(n-1) = x(n)
b) y(n) + 0.8 y(n-1) + 0.125 y(n-2) = x(n) + 2x(n-1)

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Step : 1 Get/Read the coefficients of x(n) to a.

Step : 2 Get/Read the coefficients of y(n) to b.

Step : 3 Define the frequency range vector w.

Step : 4 Find the frequency response of the filter by using the coefficients b and a.

Step : 5 Calculate the magnitude of the frequency response

Step : 6 Plot the magnitude response of the filter

Step : 7 Calculate the phase response of the filter

Step : 8 Plot the phase response of the filter

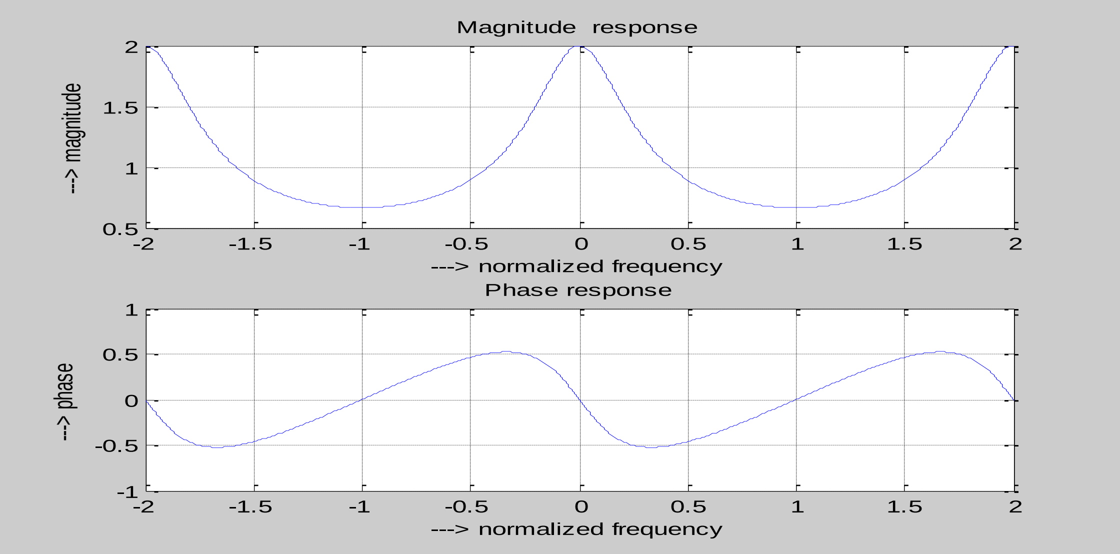
Procedure:

1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Input 1:

enter the coefficients of x(n): [1]
Enter the coefficients of y(n): [1, -0.5]

Output 1:

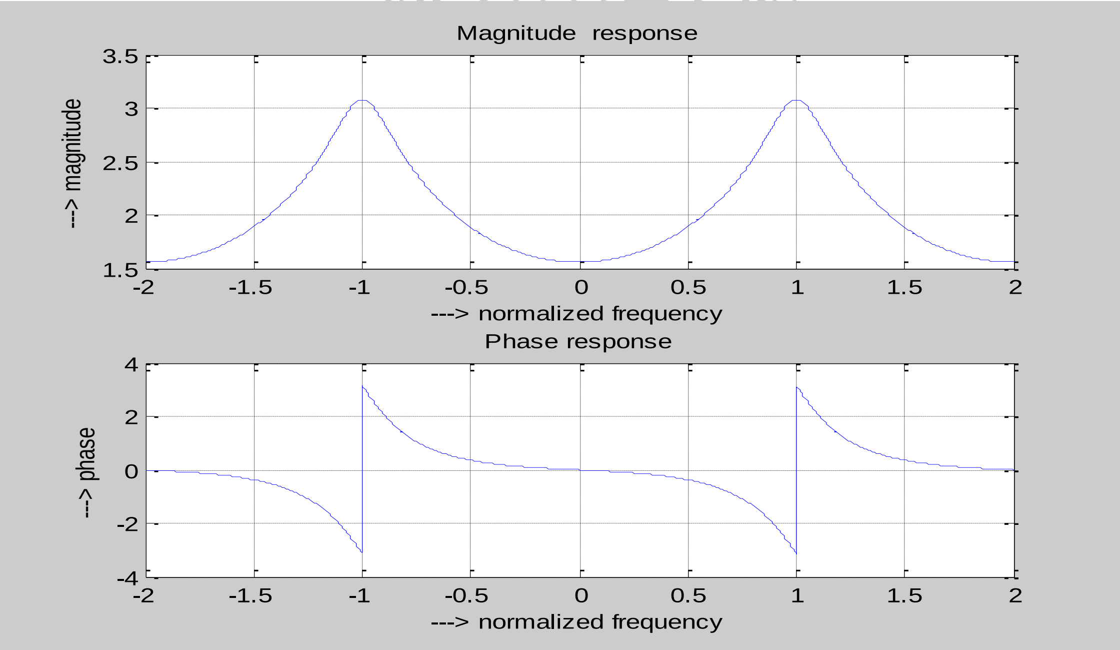


Input 2:

Enter the coefficients of x(n): [1 2]

Enter the coefficients of y(n): [1, 0.8, 0.125]

Output 2:



Result: Hence the frequency response of a first order and second order discrete-time LTI systems are observed and plotted using MATLAB.

5. FAST FOURIER TRANSFORM

Aim: To compute the Fast Fourier transform of a given signal using MATLAB.

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Step : 1 Get/Read the length of sequence x[n] to N.

Step : 2 Get/Read the samples of sequence x[n] to x.

Step : 3 Find the length of x and store it in N1.

Step : 4 If N1< N then pad N-N1 number of zeros to x.

Step : 5 Initialize the time vector n from 0:N-1.

Step : 6 Find the FFT of the given sequence and store it in y.

Step : 7 Determine the magnitude of y and store it in ‘my’.

Step : 8 Determine the phase of y and store it ‘py’.

Step : 9 Plot the Graphs of x, my and py.

Procedure:

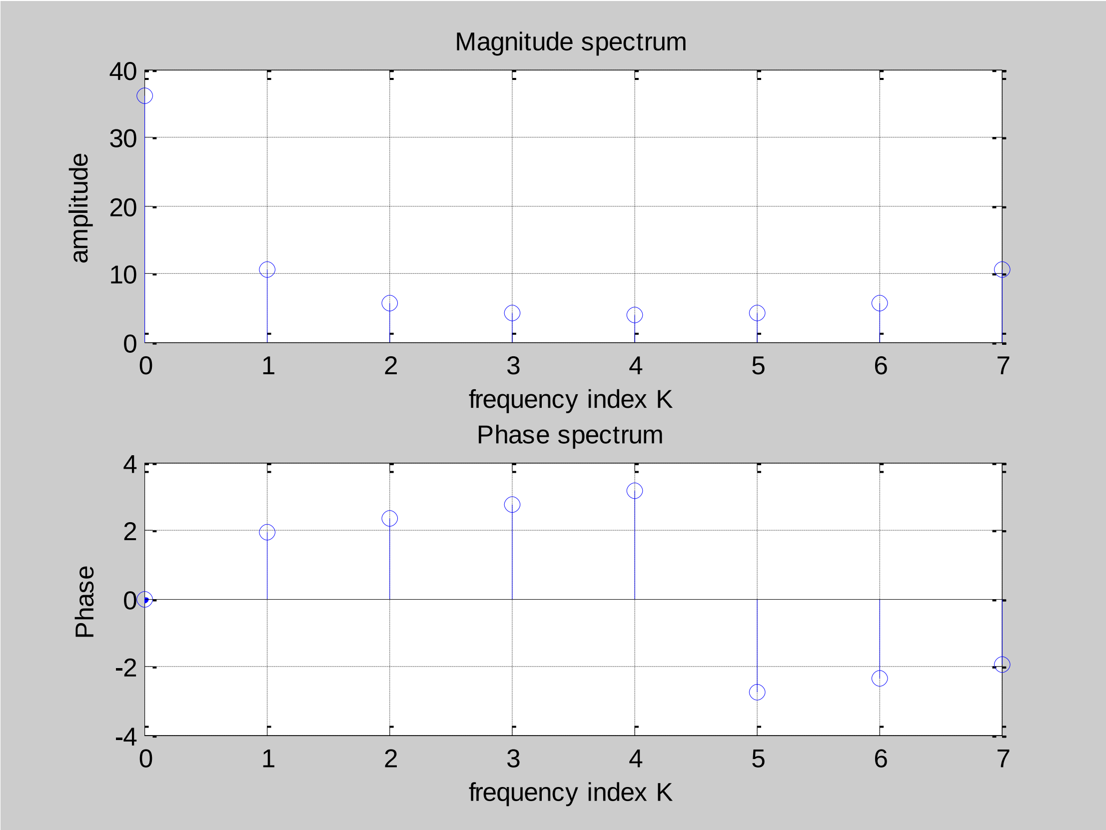
1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Input:

enter the length of sequence: 8
enter the samples of sequence: [1 2 3 4 5 6 7 8]

Output:

FFT of the given sequence x[n] is
Columns 1 through 4
36.0000 + 0.0000i -4.0000 + 9.6569i -4.0000 + 4.0000i -4.0000 + 1.6569i
Columns 5 through 8
-4.0000 + 0.0000i -4.0000 - 1.6569i -4.0000 - 4.0000i -4.0000 - 9.6569i



Result: Hence Fast Fourier transform of a given signal is computed and its spectrum is plotted using MATLAB

6. POWER DENSITY SPECTRUM OF GIVEN SIGNAL

Aim: To determine the power spectral density of a given input signal using MATLAB.

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Step : 1 Get/Read the number of samples of x[n] to N.

Step : 2 Get/Read the frequency of sinusoidal signal to 'f'.

Step : 3 Define a time vector n from 0:N-1.

Step : 4 Determine x using $x[n] = \sin(2\pi fn/N)$.

Step : 5 Determine autocorrelation of x and store it in 'y'.

Step : 6 Find the FFT of 'y' and store it in 'sx'.

Step : 7 Determine the magnitude of 'sx' and store it 'msx'.

Step : 8 Define the frequency vector f from 0:N-1.

Step : 9 Plot the Graphs of 'x' and 'msx'.

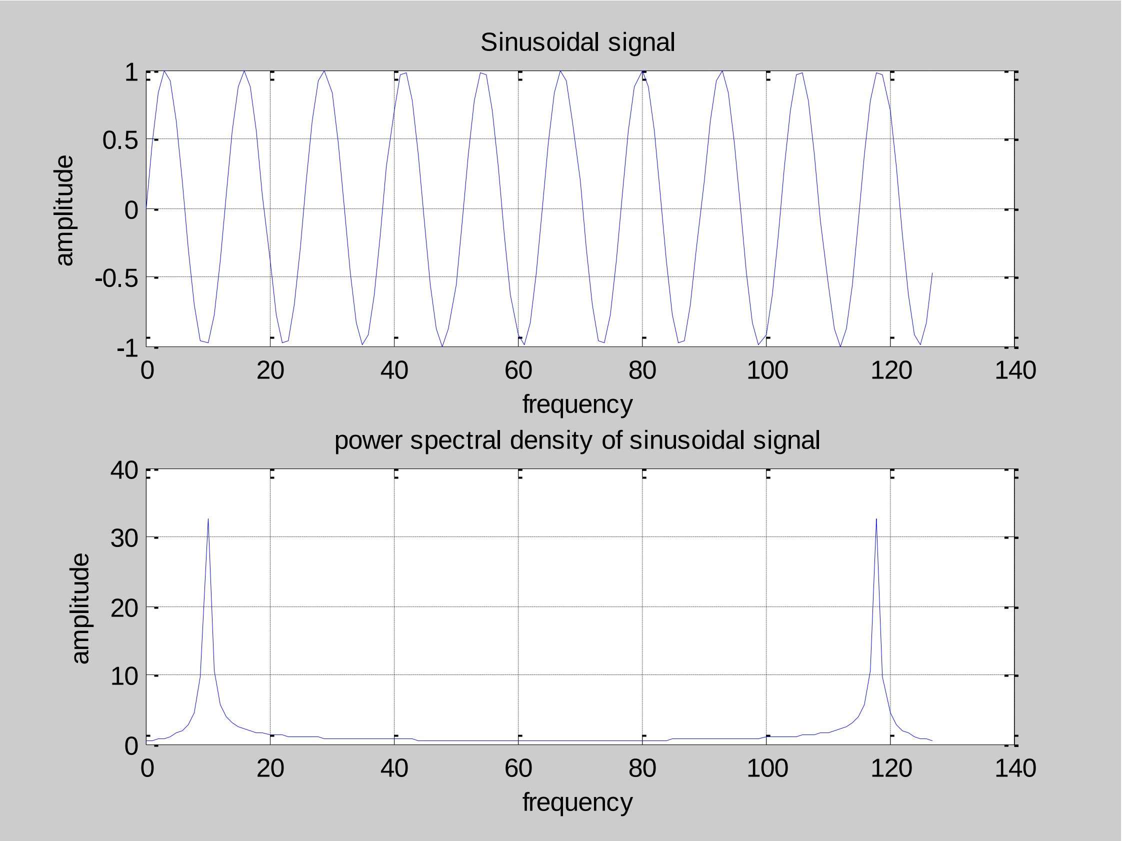
Procedure:

1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type 'edit' in the MATLAB prompt '>>' that appears in the Command window.
3. Write the program in the 'Edit' window and save it in 'M-file'.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Input:

enter the length of sequence: 128
enter the frequency of sinusoidal: 10

Output:



Result: Hence the power spectral density of a sinusoidal signal is determined and its spectrum is observed using MATLAB.

7. DESIGN OF IIR LOW PASS DIGITAL FILTER

Aim: To design and plot the frequency response of IIR low pass digital filter using Butterworth & Chebyshev approximations.

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Step : 1 Get/Read the pass band frequency of LPF to ‘fp’.

Step : 2 Get/Read the stop band frequency of LPF to ‘fs’.

Step : 3 Get/Read the pass band ripple of LPF to ‘rp’.

Step : 4 Get/Read the stop band ripple of LPF to ‘rs’.

Step : 5 Get/Read the sampling frequency to ‘f’.

Step : 6 Normalize the pass band and stop band frequencies using

wp = 2*fp/f; ws = 2*fs/f;

Step : 7 Define the frequency vector ‘w’ from 0:π.

Step : 8 Determine the order ‘n’ and cut off frequency ‘wn’ of the filter using Butterworth approximation.

Step : 9 Determine the coefficients of digital filter [b, a] using the order ‘n’ and cutoff frequency ‘wn’.

Step : 10 Determine the frequency response of low pass filter ‘H’ using the coefficients [b, a].

Step : 11 Determine the magnitude of H in dB and store it in ‘mag’.

Step : 12 Determine the phase of H and store it in ‘phase’.

Step : 13 Plot the Graphs of ‘mag’ and ‘phase’.

Step : 14 Repeat the steps 8 to 13 for Chebyshev type I & type II approximations.

Procedure:

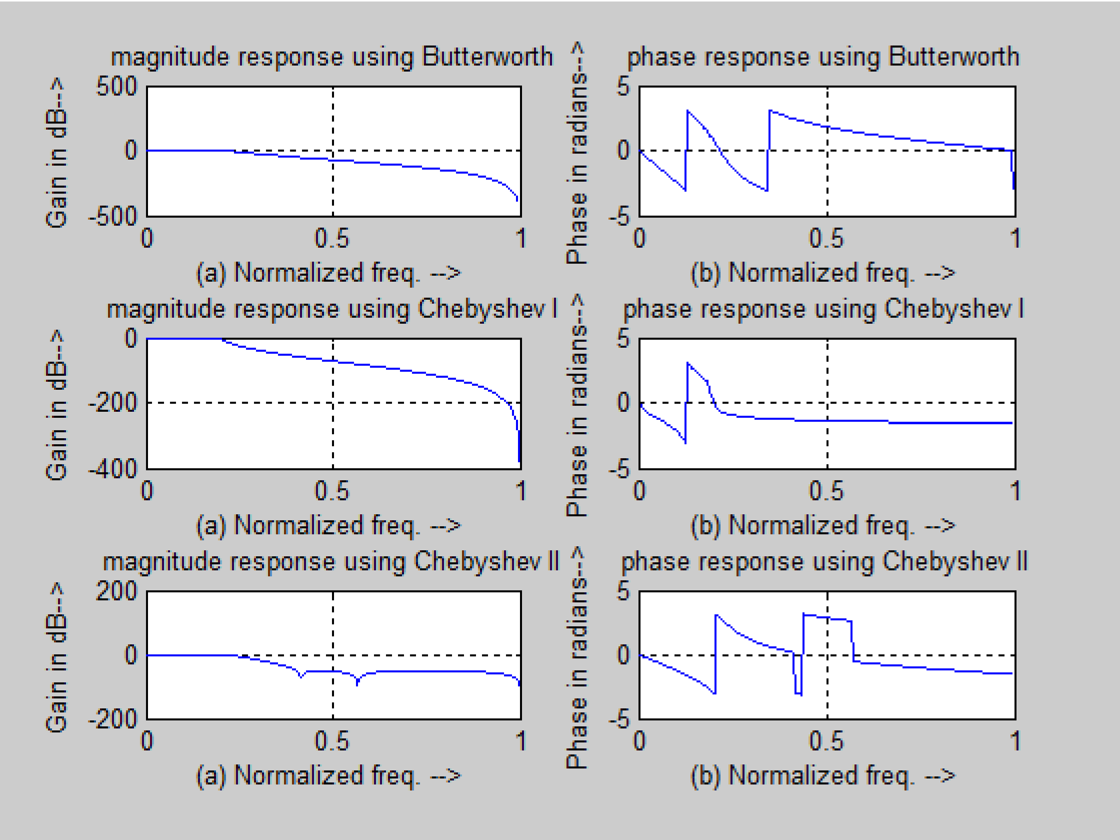
1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the input in the command window.

6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Input:

enter the passband ripple: 3
enter the stopband ripple: 50
enter the passband frequency: 1000
enter the stopband frequency: 2000
enter the sampling frequency: 10000

Output:



Result: Hence an IIR low pass digital filter is designed and its frequency response is observed using MATLAB.

8. DESIGN OF IIR HIGH PASS DIGITAL FILTER

Aim: To design and plot the frequency response of IIR high pass digital filter using Butterworth & Chebyshev approximations.

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Step : 1 Get/Read the pass band frequency of HPF to ‘fp’.

Step : 2 Get/Read the stop band frequency of HPF to ‘fs’.

Step : 3 Get/Read the pass band ripple of HPF to ‘rp’.

Step : 4 Get/Read the stop band ripple of HPF to ‘rs’.

Step : 5 Get/Read the sampling frequency to ‘f’.

Step : 6 Normalize the pass band and stop band frequencies using

wp = 2*fp/f; ws = 2*fs/f;

Step : 7 Define the frequency vector ‘w’ from 0:π.

Step : 8 Determine the order ‘n’ and cut off frequency ‘wn’ of the filter using Butterworth approximation.

Step : 9 Determine the coefficients of digital filter [b, a] using order ‘n’ and cutoff frequency ‘wn’.

Step : 10 Determine the frequency response of the high pass filter ‘H’ using the coefficients [b, a].

Step : 11 Determine the magnitude of H in dB and store it in ‘mag’.

Step : 12 Determine the phase of H and store it in ‘phase’.

Step : 13 Plot the Graphs of ‘mag’ and ‘phase’.

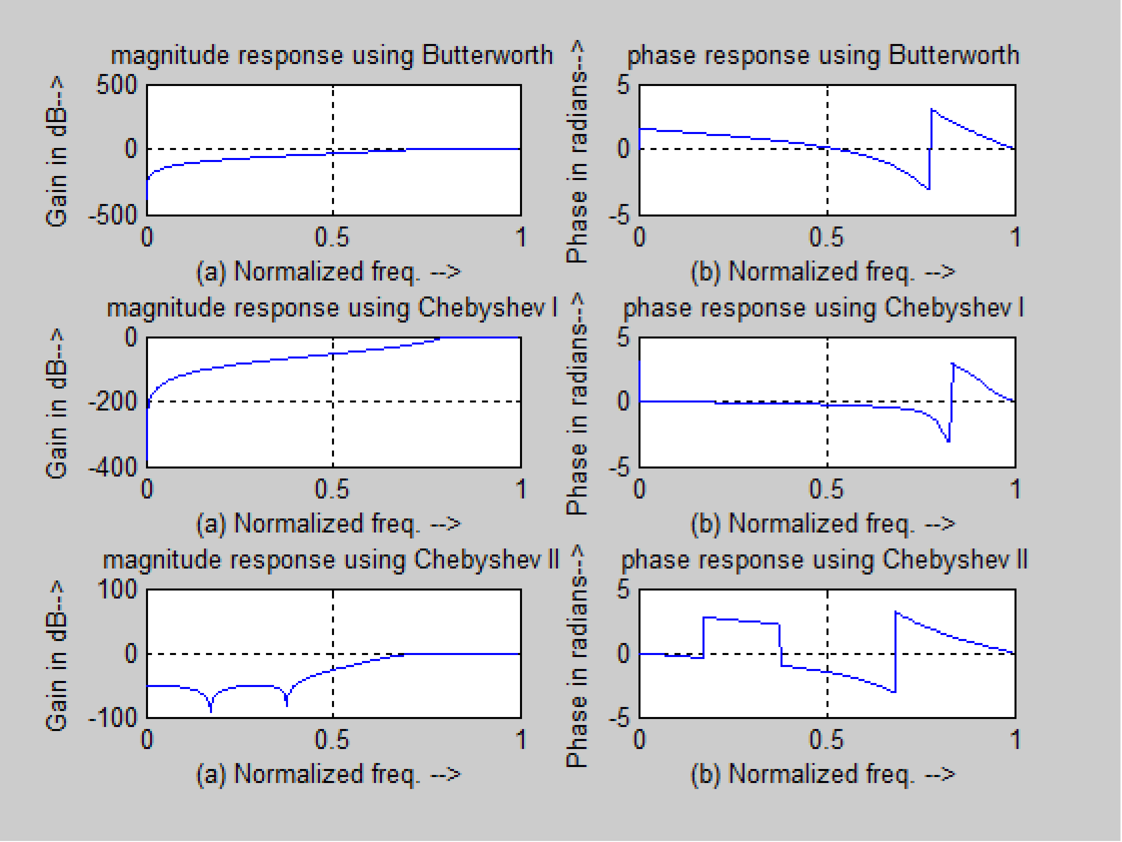
Step : 14 Repeat the steps 8 to 13 for Chebyshev type I & type II approximations.

Procedure:

1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Input:
enter the passband ripple: 2
enter the stopband ripple: 50
enter the passband frequency: 2000
enter the stopband frequency: 1000
enter the sampling frequency: 5000

Output:



Result: Hence an IIR high pass digital filter is designed and its frequency response is observed using MATLAB.

9. DESIGN OF FIR LOW PASS DIGITAL FILTER

Aim: To design and plot the frequency response of FIR low pass digital filter using windowing technique.

Equipment Required:
PC loaded with MATLAB software

Algorithm:

Step : 1 Get/Read the pass band frequency of LPF to ‘fp’.

Step : 2 Get/Read the stop band frequency of LPF to ‘fs’.

Step : 3 Get/Read the pass band ripple of LPF to ‘rp’.

Step : 4 Get/Read the stop band ripple of LPF to ‘rs’.

Step : 5 Get/Read the sampling frequency to ‘f’.

Step : 6 Normalize the pass band and stop band frequencies using

```
wp = 2*fp/f;  ws = 2*fs/f;
```

Step : 7 Define the frequency vector ‘w’ from 0:π.

Step : 8 Determine the order ‘n’ of the filter using

```
num = -20*log10(sqrt(rp*rs))-13;  dem = 14.6*(fs-fp)/f
n = ceil(num/dem)
```

Step : 9 Make sure that the order of the filter is always odd.

```
if(rem(n,2)~=0)
        n1=n;  n=n-1;
else
        n1=n+1;
end
```

Step : 10 Determine the coefficients ‘b’ of digital LPF using Rectangular window and fir1 functions.

Step : 11 Determine the frequency response of low pass filter ‘H’ using the coefficients ‘b’.

Step : 12 Determine the magnitude of H in dB and store it in ‘mag’.

Step : 13 Determine the phase of H and store it in ‘phase’.

Step : 14 Plot the Graphs of ‘mag’ and ‘phase’.

Step : 15 Repeat the steps 10 to 14 for triangular, Hanning, Hamming, Blackman and Kaiser window functions.

Procedure:

1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Input:

enter passband ripple: 0.02

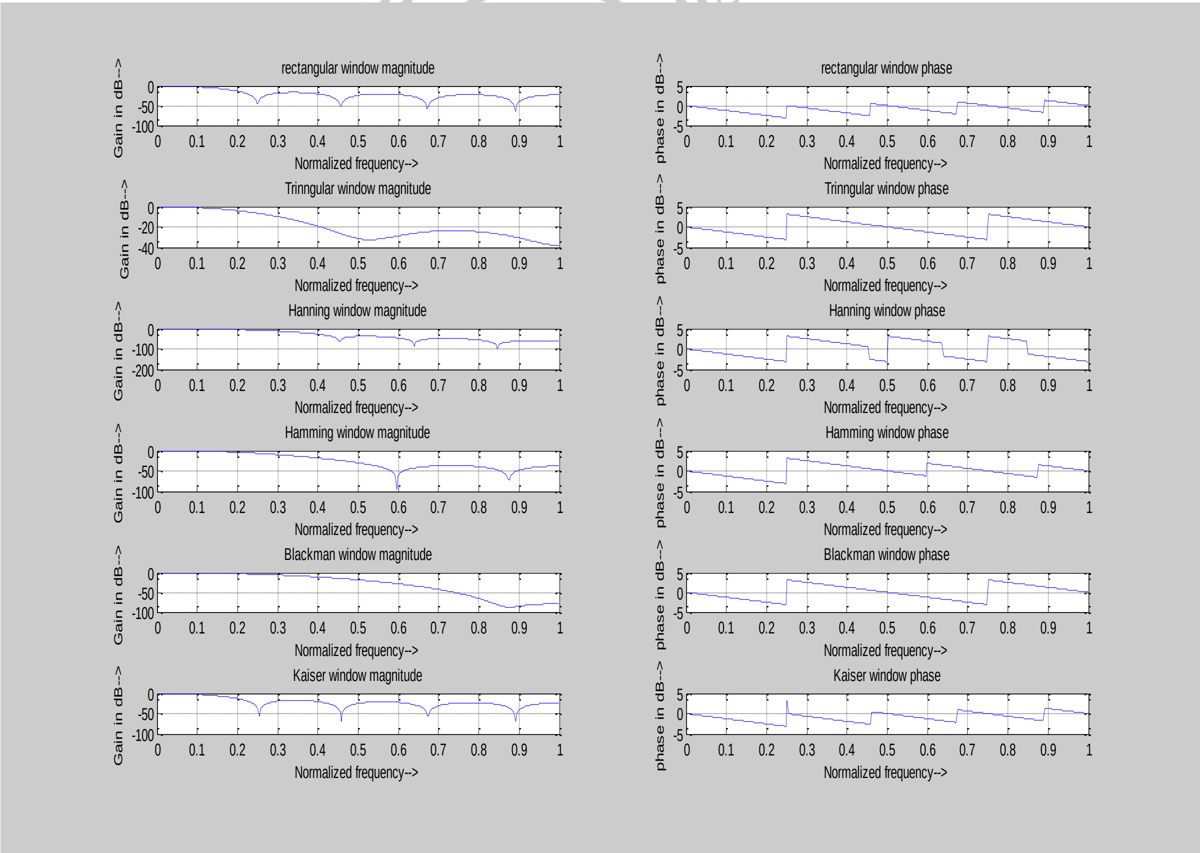
enter the stopband ripple: 0.01

enter passband frequency: 500

enter stopband frequency: 2000

enter sampling frequency: 8000

Output:



Result: Hence an FIR low pass digital filter is designed and its response is observed using MATLAB.

10. DESIGN OF FIR HIGH PASS DIGITAL FILTER

Aim: To design and plot the frequency response of FIR high pass filter using windowing technique.

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Step : 1 Get/Read the pass band frequency of HPF to ‘fp’.

Step : 2 Get/Read the stop band frequency of HPF to ‘fs’.

Step : 3 Get/Read the pass band ripple of HPF to ‘rp’.

Step : 4 Get/Read the stop band ripple of HPF to ‘rs’.

Step : 5 Get/Read the sampling frequency to ‘f’.

Step : 6 Normalize the pass band and stop band frequencies using

wp = 2*fp/f; ws = 2*fs/f;

Step : 7 Define the frequency vector w from 0:π.

Step : 8 Determine the order ‘n’ of the filter using

num=-20*log10(sqrt(rp*rs))-13; dem=14.6*(fp-fs)/f
n=ceil(num/dem)

Step : 9 Make sure that the order of the filter is always odd.

Step : 10 Determine the coefficients ‘b’ of digital HPF using Rectangular window and fir1 functions.

Step : 11 Determine the frequency response of high pass filter ‘H’ using the coefficients ‘b’.

Step : 12 Determine the magnitude of H in dB and store it in ‘mag’.

Step : 13 Determine the phase of H and store it in ‘phase’.

Step : 14 Plot the Graphs of ‘mag’ and ‘phase’.

Step : 15 Repeat the steps 10 to 14 for triangular, Hanning, Hamming, Blackman and Kaiser Window functions.

Procedure:

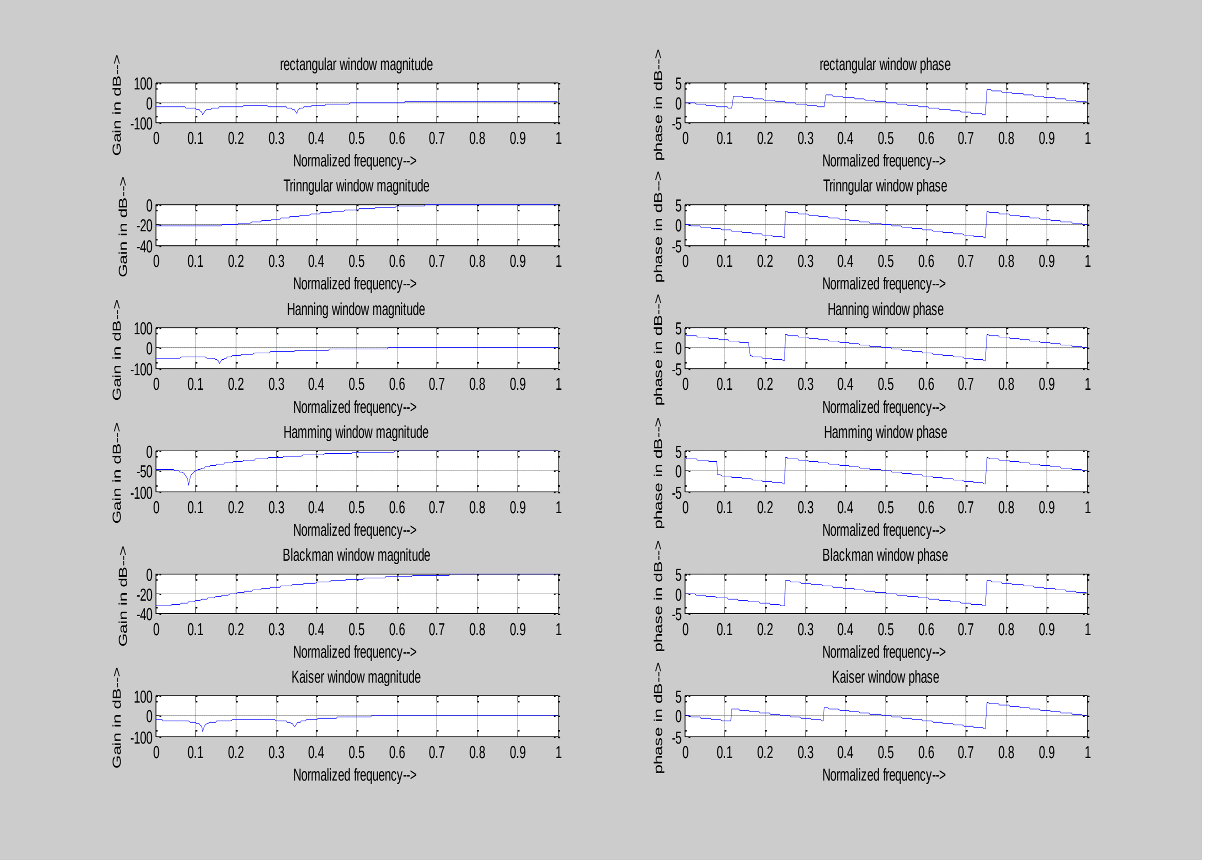
1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the input in the command window.

6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Input:

enter passband ripple: 0.02
enter the stopband ripple: 0.01
enter passband frequency: 5000
enter stopband frequency: 1000
enter sampling frequency: 20000

Output:



Result: Hence an FIR high pass digital filter is designed and its Frequency Response is observed using MATLAB.

11. LINEAR CONVOLUTION

Aim: To perform linear convolution of given sequences using TMS320C6713 DSP Processor.

Equipment Required:

PC loaded with Code Composer Studio software, TMS320C6713 DSP Starter kit

Algorithm:

Step : 1 Get/Read the length of input sequence to N1.

Step : 2 Get/Read the length of impulse response to N2.

Step : 3 Get/Read Input Signal Samples to x[n].

Step : 4 Get/Read Impulse Response samples to h[n]

Step : 5 Calculate the length of output sequence N using

N=N1+N2-1

Step : 6 Perform Linear convolution using the formula

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

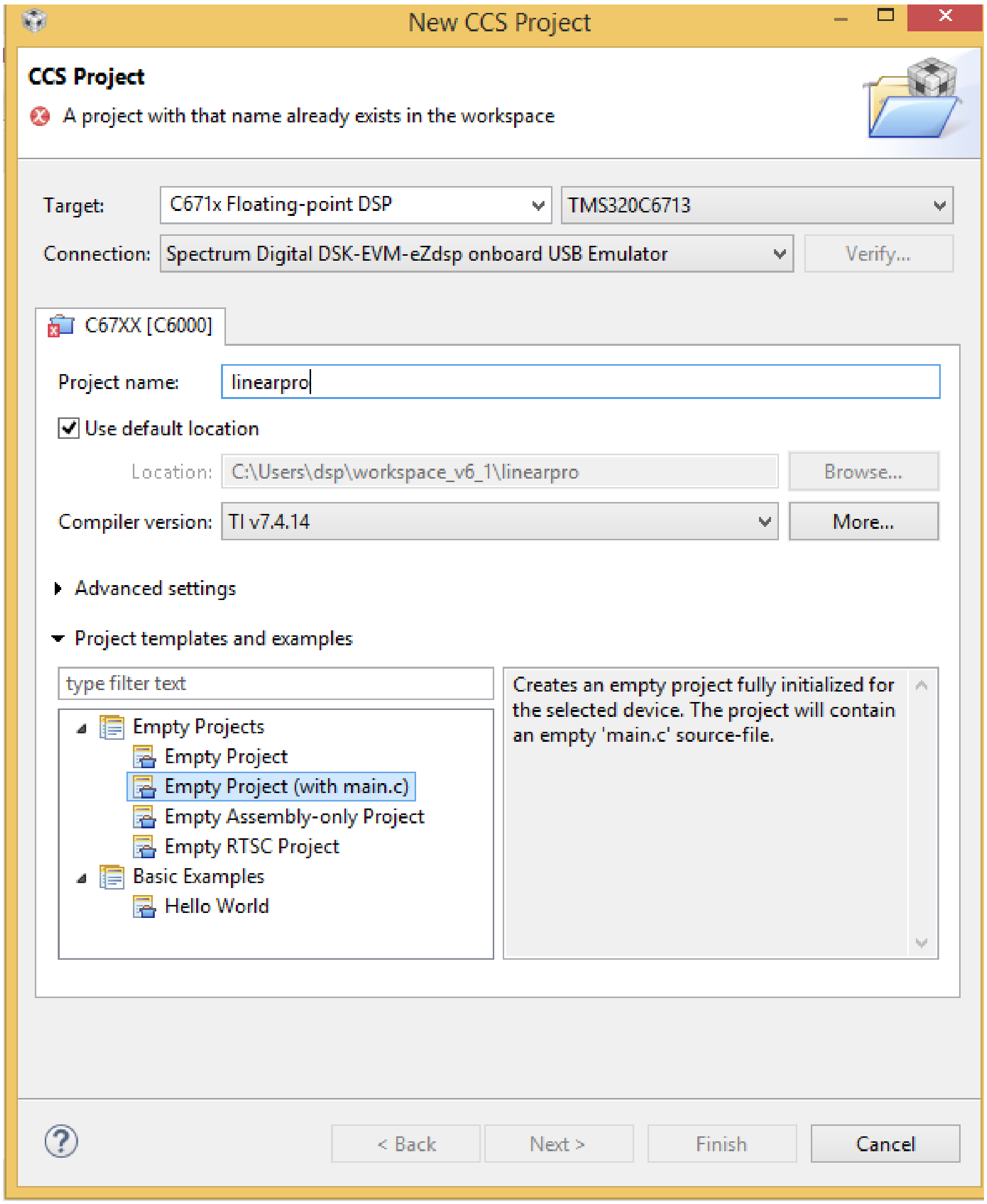
Step : 7 Display the values of y[n].

Step : 8 Plot the graphs for x[n], h[n] and y[n].

Procedure:

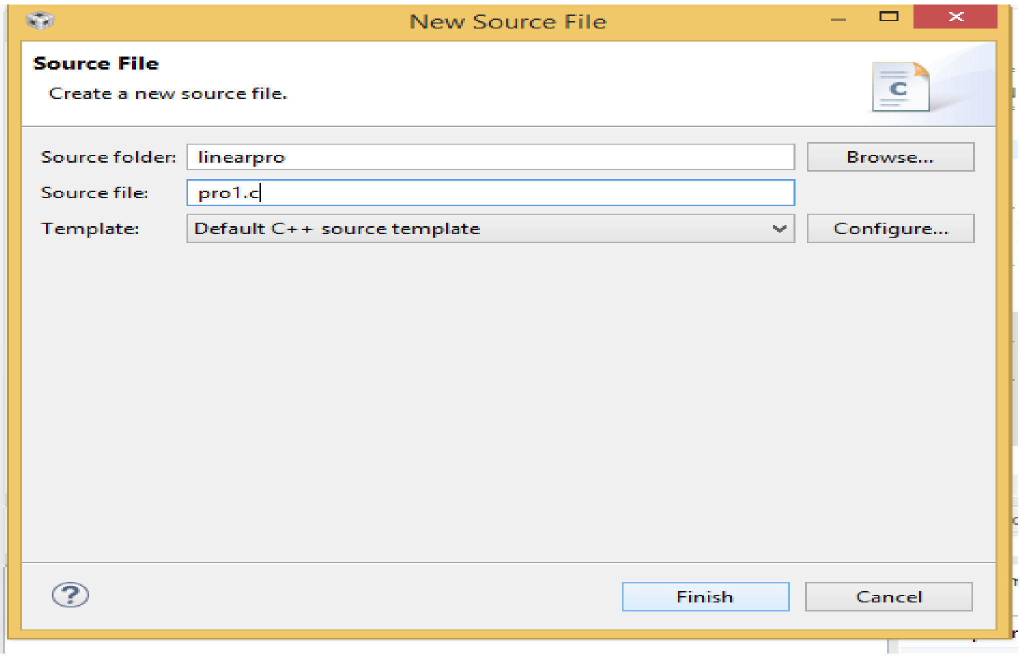
1. Open CCS Version 6.1.1.
2. Go to File menu and select new CCS project.
3. Create new project with following specifications

   Project → New CCS Project
   Target: C671X Floating-point DSP TMS320C671X
   Connection: Spectrum Digital DSK-EVM-eZdsp onboard USB Emulator



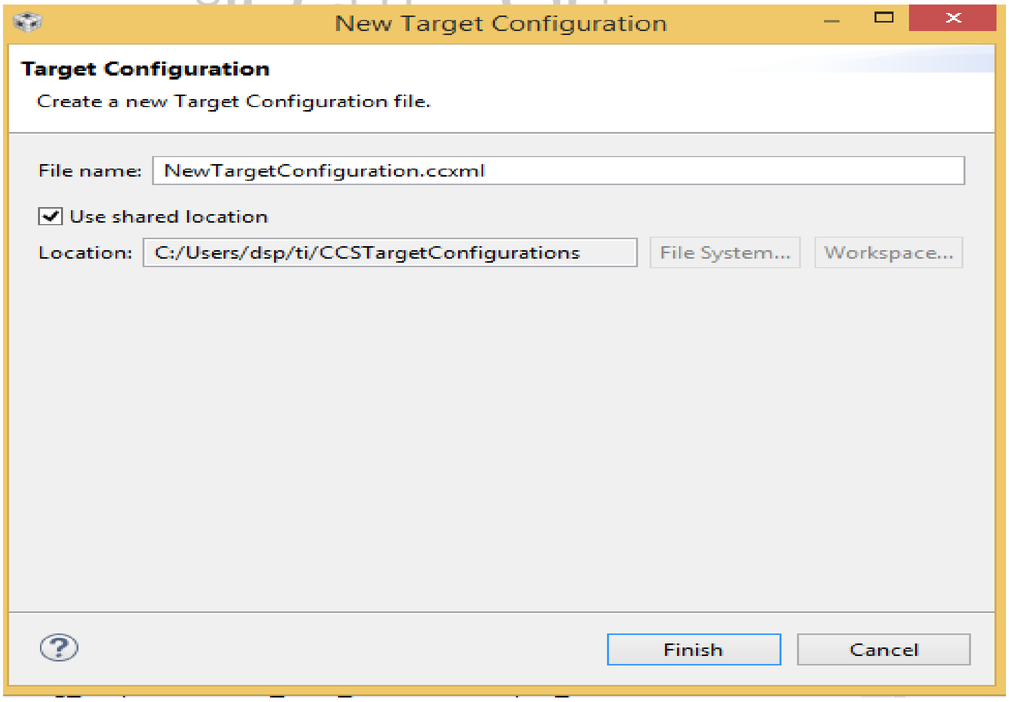
4. Select empty project with main.c and finish.
5. Create a new Source file

File → New →Browse the Source folder→ give file name & Save (Eg: sum.c)

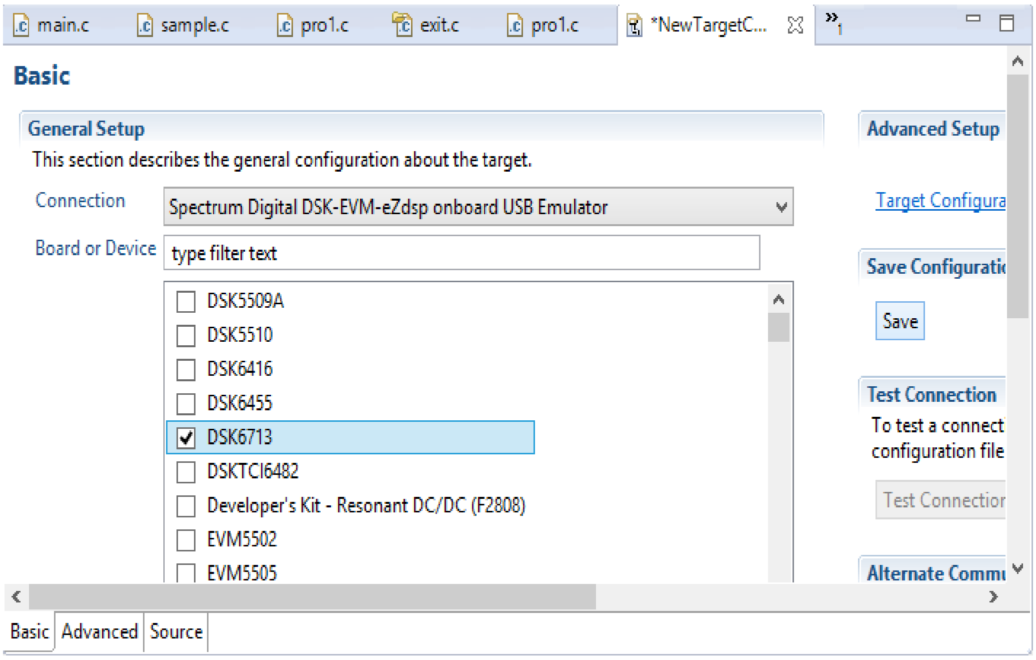


6. Write C program and save it.
7. Create New Target configuration file with extension .ccxml

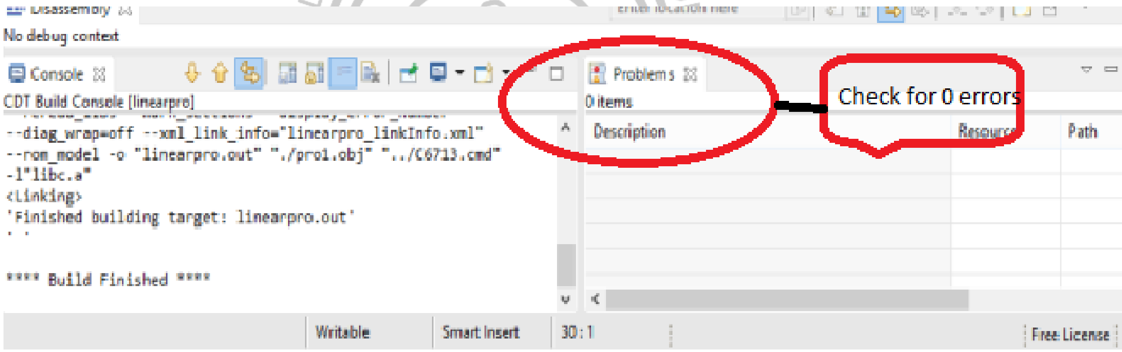
File → New → Target configuration file



Connection: Spectrum Digital DSK - EVM-eZdsp onboard USB Emulator



8. Build the project.



9. Launch the target configuration.

   Select Target configuration →Right click→ Launch target configuration

10. Click on Connect Target on menu bar.

11. Load the program

    Run→ Load Programme →Browse .out file

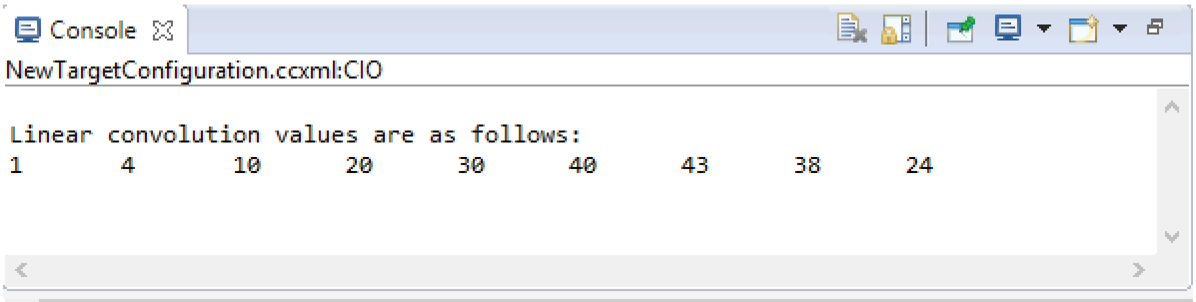
12. Run the program.

13. Observe the output in the console window (to observe the graph, go to expressions, right click on the expression and select the graph).

Input:

enter the length of first sequence: 6
enter the length of second sequence: 4
enter the samples of first sequence x(n): {1,2,3,4,5,6}
enter the samples of second sequence h(n): {1,2,3,4}

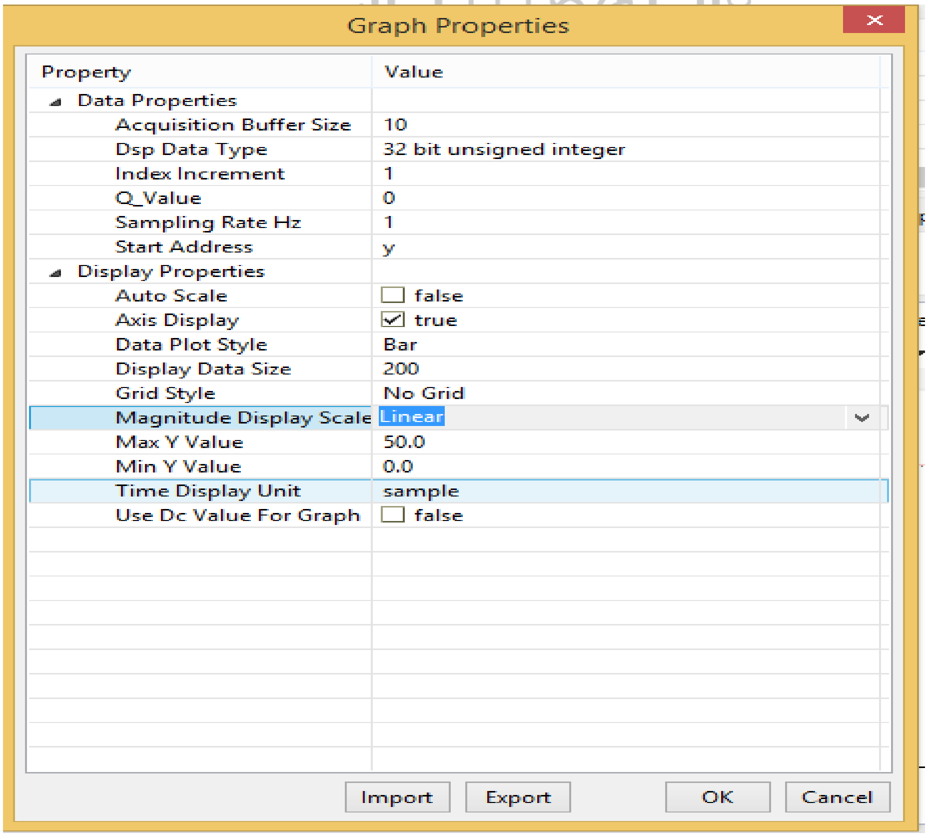
Output:

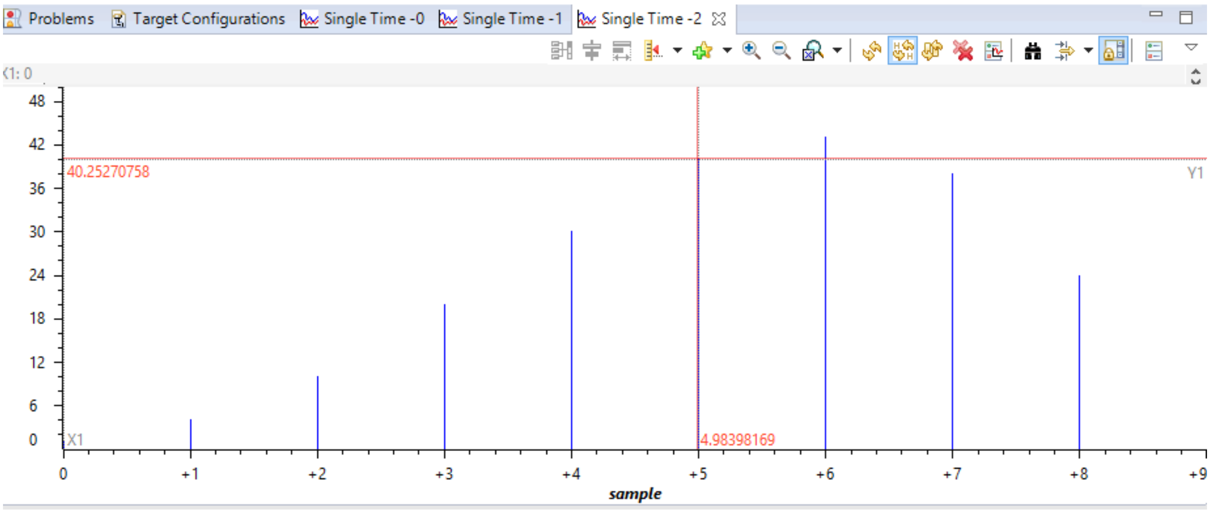


To view output graphically,

Select Tools → graph → Single Time.

Graph settings & graph as follows.





Result: Hence linear convolution of given sequences is performed and output is observed using TMS320C6713 DSP starter kit.

12. CIRCULAR CONVOLUTION

Aim: To perform circular convolution of given sequences using TMS320C6713 DSP Processor.

.

Equipment Required:

PC loaded with Code Composer Studio software, TMS320C6713 DSP Starter kit

Algorithm:

Step : 1 Get/Read the length of input sequence to N1.

Step : 2 Get/Read the length of impulse response to N2.

Step : 3 Get/Read the Input Signal Samples to x[n].

Step : 4 Get/Read the Impulse Response samples to h[n]

Step : 5 Calculate the length of output sequence N using

N=N1+N2-1

Step : 6 If N1 > N2, pad N1-N2 zeros to h[n]

else pad N2-N1 zeros to x[n].

Step : 7 Perform Circular convolution using the formula

$y[n] = \sum_{m=0}^{N-1} x(m) * (h(n-m))N$

Step : 8 Display the values of y[n].

Step : 9 Plot the graphs for x[n], h[n] and y[n].

Procedure:

1. Open Code Composer Studio 6.1.1
2. Create new project with following specifications

   Project → New CCS Project

   Target: C671X Floating -point DSP TMS320C671X

   Connection: Spectrum Digital DSK - EVM-eZdsp onboard USB Emulator
3. Select empty project with main.c and finish.
4. Create a Source file

   File → New →Browse the Source folder→ give file name & Save (Eg: sum.c)
5. Write C program and save it.
6. Create New Target configuration file with extension .ccxml

   File → New → Target configuration file

   Connection: Spectrum Digital DSK - EVM-eZdsp onboard USB Emulator

Device type: DSK6713

Save

7. Build the project.
8. Launch the target configuration.

   Select Target configuration →Right click→ Launch target configuration
9. Click on Connect Target on menu bar.
10. Load the program

    Run→ Load Programme →Browse .out file
11. Run the program.
12. Observe the output in the console window

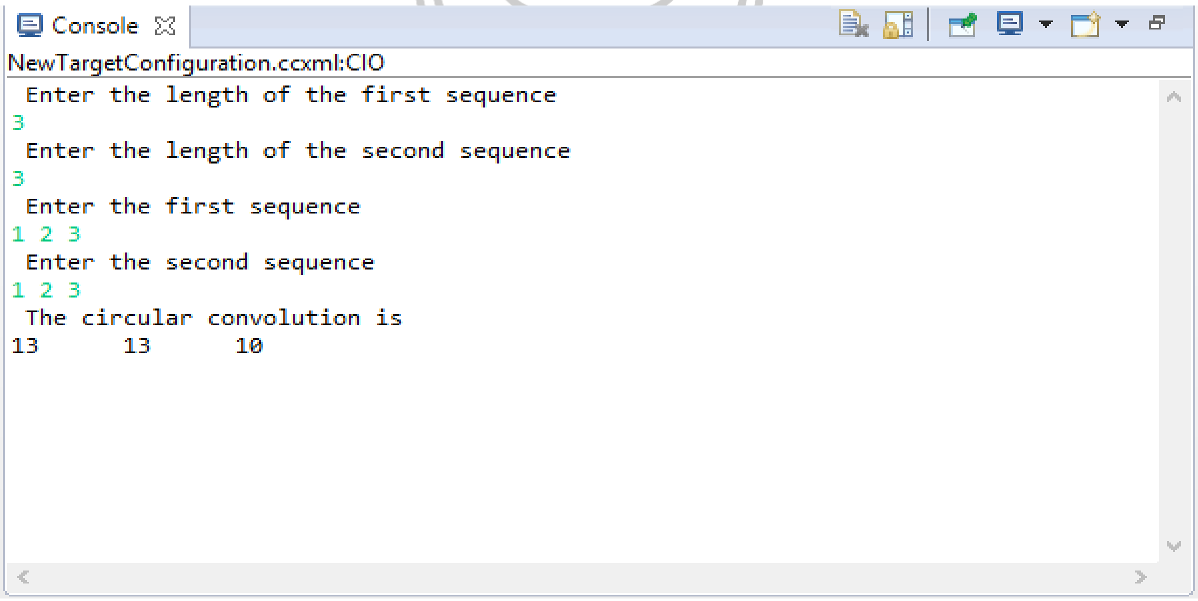
    (to observe the graph, go to expressions, right click on the expression and select the graph).

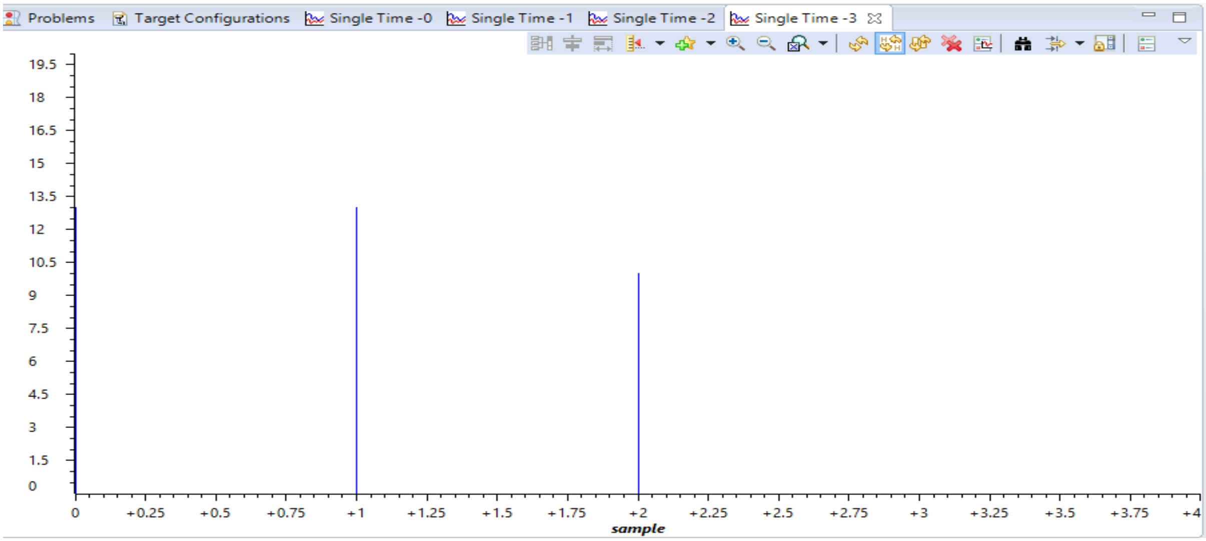
Input:

Enter the length of first sequence: 3
Enter the length of second sequence: 3
Enter the samples of first sequence:{1 2 3}
Enter the samples of second sequence:{1 2 3}

Output:

Circular convolution of x(n) and h(n) is:
13 13 10





Result: Hence circular convolution of given sequences is performed and output is observed using TMS320C6713 DSP starter kit.

13. GENERATION OF SINE WAVE AND SQUARE WAVE

Aim: To generate sine and square waves using TMS320C6713 DSP Processor.

Equipment Required:

PC loaded with Code Composer Studio software, TMS320C6713 DSP Starter kit

Algorithm for Sine wave generation

Step : 1 Get/Read the frequency of the sinusoidal signal to F.

Step : 2 Get/Read the number of samples to N.

Step : 3 Define time scale for a sine signal (Ex: 0<t<100)

Step : 4 Generate sinusoidal signal using

y[t]=sin(2*pi*f*t/N);

Step : 5 Plot the graph of 'y'.

Algorithm for Square wave generation

Step : 1 Get/Read the time period of square signal to t.

Step : 2 Define time scale i for a square signal. (Ex: 0<i<100)

Step : 3 Generate square wave using

```
if((i*2)/t)%2==0
   y[i]= 1 ;
else
   y[i]= -1;
```

Step : 4 Plot the graph of y.

Procedure:

1. Open Code Composer Studio 6.1.1
2. Create new project with following specifications

   Project → New CCS Project

   Target: C671X Floating -point DSP TMS320C671X

   Connection: Spectrum Digital DSK - EVM-eZdsp onboard USB Emulator
3. Select empty project with main.c and finish.
4. Create a Source file

   File → New →Browse the Source folder→ give file name & Save (Eg: sum.c)
5. Write C program and save it.

6. Create New Target configuration file with extension .ccxml

   File → New → Target configuration file

   Connection: Spectrum Digital DSK - EVM-eZdsp onboard USB Emulator

   Device type: DSK6713

   Save

7. Build the project.
8. Launch the target configuration.

   Select Target configuration →Right click→ Launch target configuration

9. Click on Connect Target on menu bar.
10. Load the program

    Run→ Load Programme →Browse .out file

11. Run the program.
12. Observe the output in the console window

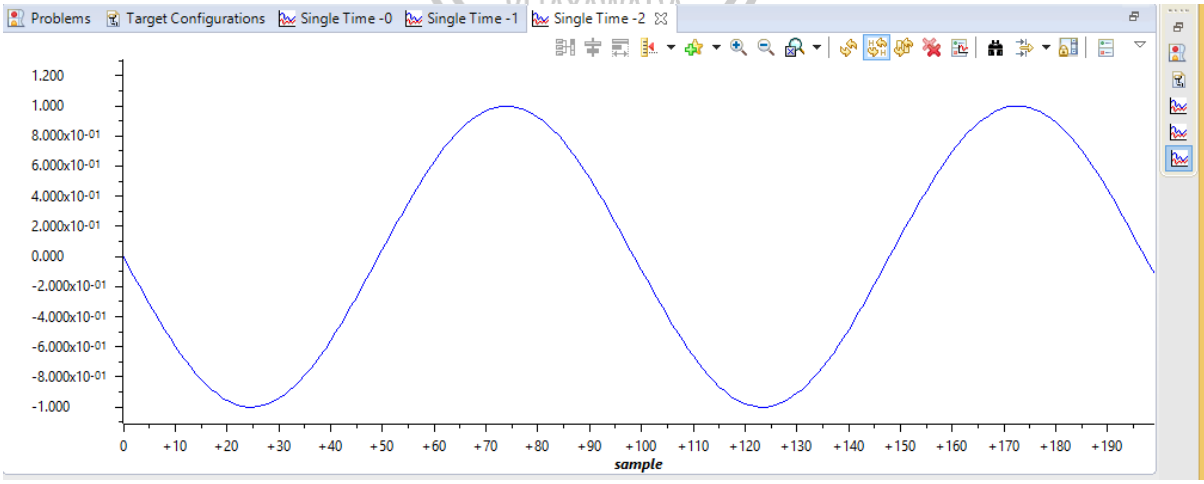
    (to observe the graph, go to expressions, right click on the expression and select the graph).

Input:

enter the frequency: 10

enter the no. of samples: 512

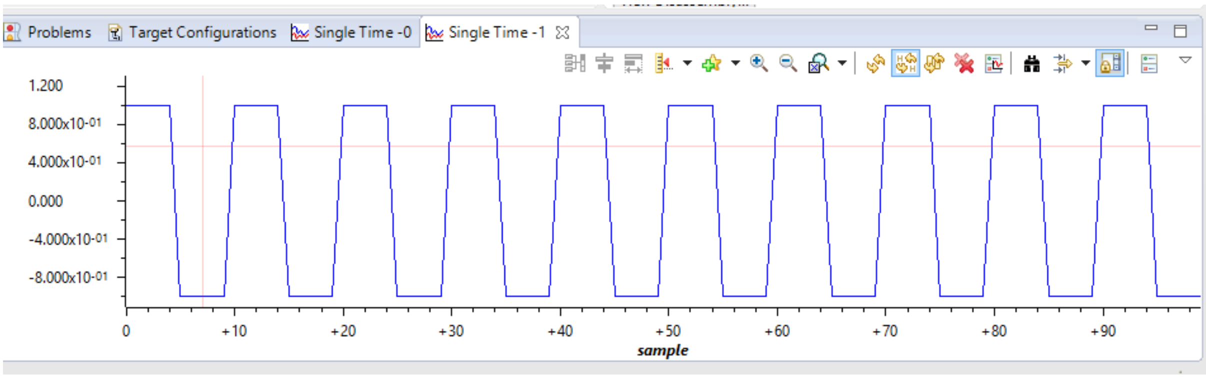
Output:



Input:

enter the time period: 10

Output:



Result: Hence sine wave and square waves are generated and observed using TMS320C6713 DSP starter kit.

14. IMPLEMENTATION OF INTERPOLATION AND DECIMATION

Aim: To implement decimation and interpolation on a given signal/sequence.

Equipment Required:

PC loaded with MATLAB software

Algorithm for Interpolation:

Step : 1 Get/Read the amplitude of first sinusoidal signal to a1.

Step : 2 Get/Read the frequency of first sinusoidal signal to f1.

Step : 3 Get/Read the amplitude of second sinusoidal signal to a2.

Step : 4 Get/Read the frequency of second sinusoidal signal to f2.

Step : 5 Get/Read the sampling to Fs.

Step : 6 Get/Read the up sampling factor to ‘i’.

Step : 7 Define time vector ‘t’ from 0:1/Fs:1.

Step : 8 Generate the signal ‘x’ using

$x = a1\cos(2\pi f1t) + a2\cos(2\pi f2t)$

Step : 9 Perform interpolation on ‘x’ using interpolation factor ‘i’ and store it in ‘y’.

Step : 10 Plot the graphs of original signal ‘x’ and interpolated signal ‘y’.

Algorithm for Decimation:

Step : 1 Get/Read the amplitude of first sinusoidal signal to a1.

Step : 2 Get/Read the frequency of first sinusoidal signal to f1.

Step : 3 Get/Read the amplitude of second sinusoidal signal to a2.

Step : 4 Get/Read the frequency of second sinusoidal signal to f2.

Step : 5 Get/Read the sampling to Fs.

Step : 6 Get/Read the down sampling factor to ‘d’.

Step : 7 Define time vector ‘t’ from 0:1/Fs:1.

Step : 8 Generate the original signal ‘x’ using

$x = a1\cos(2\pi f1t) + a2\cos(2\pi f2t)$

Step : 9 Perform decimation on ‘x’ using decimation factor ‘d’ and store it in ‘y’.

Step : 10 Plot the graphs of original signal ‘x’ and decimated signal ‘y’.

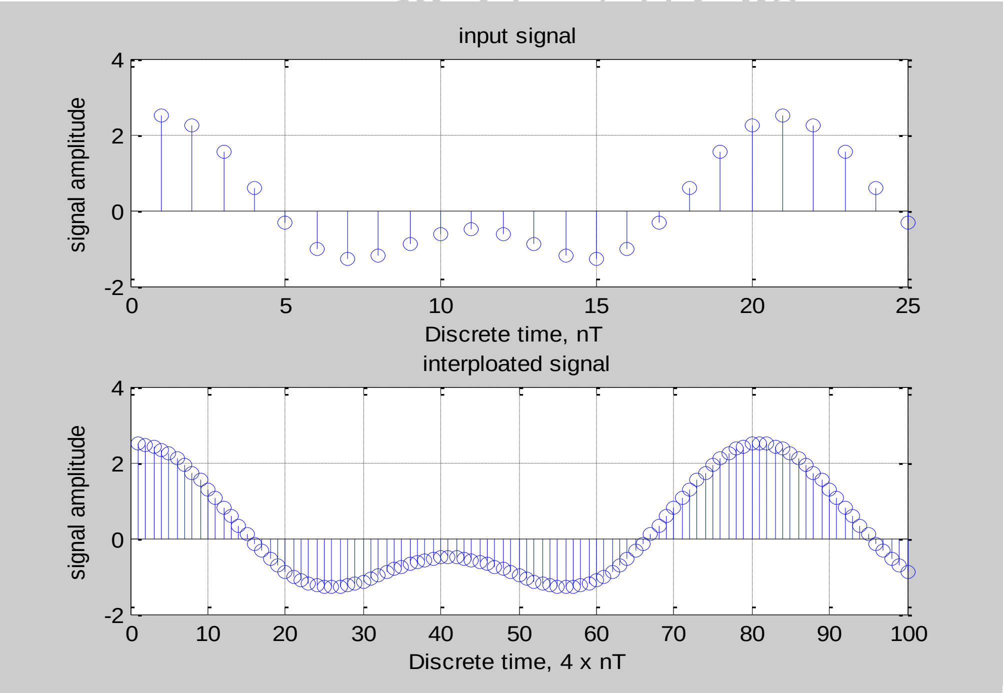
Procedure:

1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Input:

Enter the amplitude of first Sinusoidal: 1.5
Enter the frequency of first Sinusoidal: 50
Enter the amplitude of second Sinusoidal: 1
Enter the frequency of second Sinusoidal: 100
Enter the sampling frequency: 1000
Enter Up sampling factor: 4

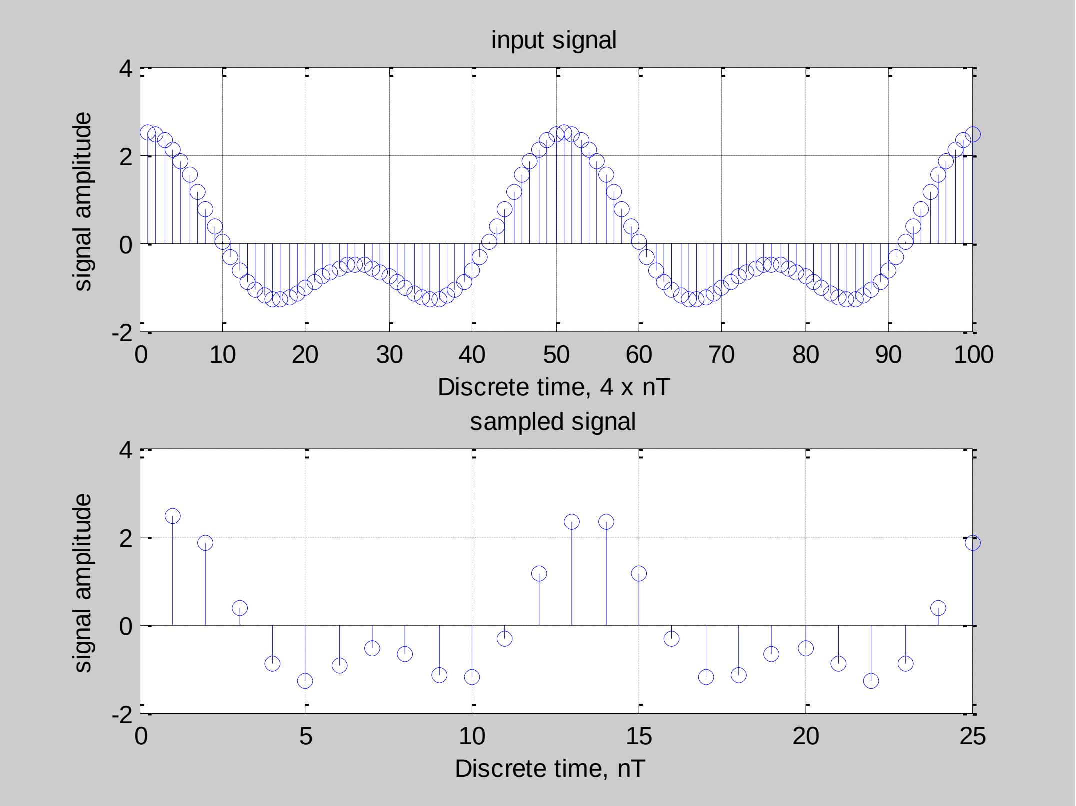
Output:



Input:

Enter the amplitude of first Sinusoidal: 1.5
Enter the frequency of first Sinusoidal: 20
Enter the amplitude of second Sinusoidal: 1
Enter the frequency of second Sinusoidal: 40
Enter the sampling frequency: 1000
Enter Down sampling factor: 4

Output:



Result: Hence decimation and interpolation are implemented on a signal and observed the outputs using MATLAB.

15. CONVERSION OF CD DATA TO DVD DATA

Aim: To convert CD data into DVD data using MATLAB.

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Step : 1 Get/Read the frequency of CD signal 44.1 KHz to ‘f'_c’.

Step : 2 Define time vector ‘t’.

Step : 3 Generate the signal ‘x’ using $x = \sin(2\pi f_c t)$.

Step : 4 Plot the original CD signal.

Step : 5 Set interpolation factor as i=13

Step : 6 Set decimation factor as d=5

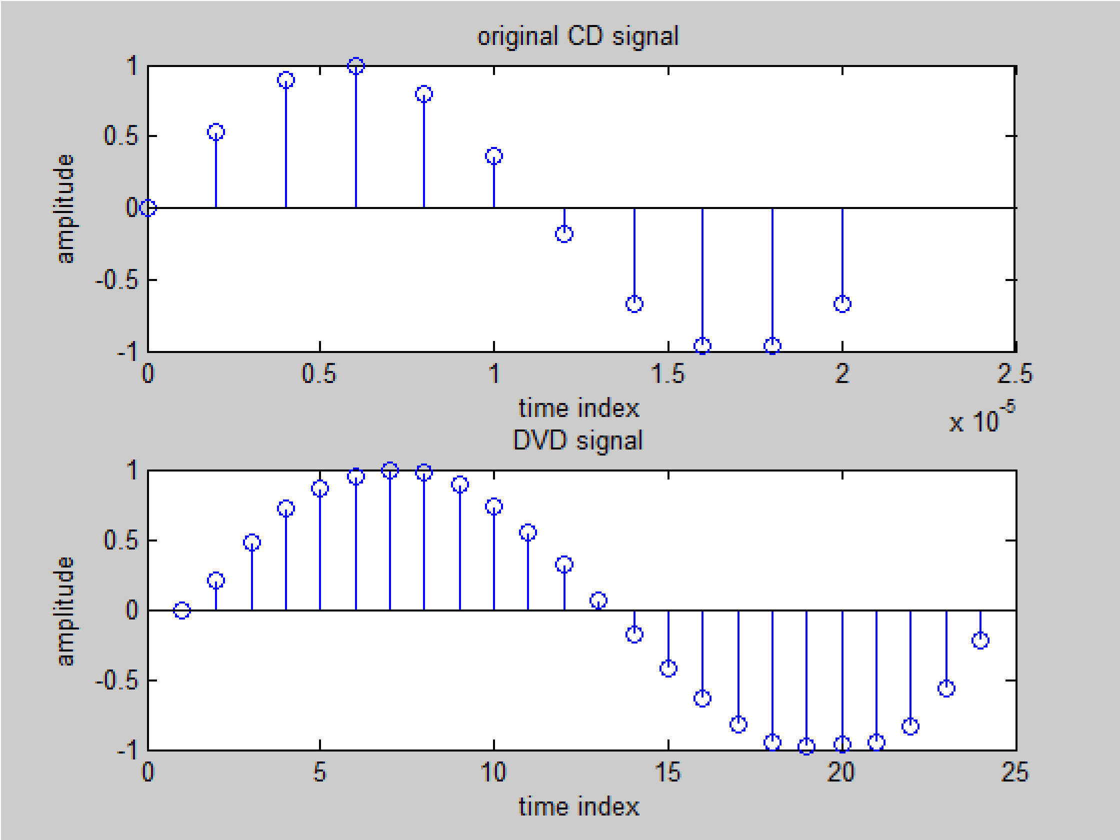
Step : 7 Use resample function to obtain DVD signal ‘y’.

Step : 8 Plot the graphs of CD signal ‘x’ and DVD signal ‘y’.

Procedure:

1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Output:



Result: Hence CD data is converted to DVD data using MATLAB.

16. SUM OF SINUSOIDAL SIGNALS

Aim: To generate and plot the sum of sinusoidal signals using MATLAB.

Equipment Required:

PC Loaded with MATLAB software

Procedure:

1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Algorithm:

Step : 1 Define time vector ‘t’ from $0{:}2\pi$.

Step : 2 Generate the signal ‘y1’ using y1= sin(t).

Step : 3 Generate the signal ‘y2’ using y2= $\frac{1}{3}\sin(3t)$.

Step : 4 Generate the signal ‘y3’ using y3= $\frac{1}{5}\sin(5t)$.

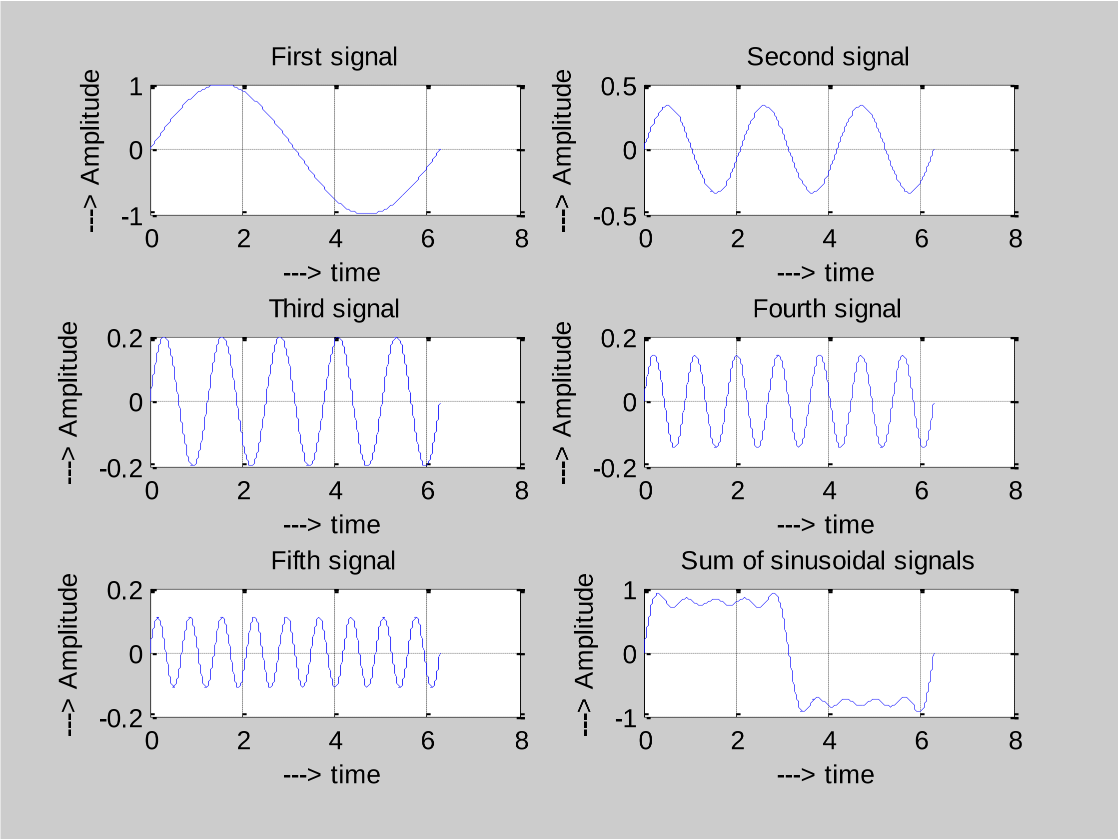
Step : 5 Generate the signal ‘y4’ using y4= $\frac{1}{7}\sin(7t)$.

Step : 6 Generate the signal ‘y5’ using y5= $\frac{1}{9}\sin(9t)$.

Step : 7 Obtain the signal ‘y’ using y= y1+y2+y3+y4+y5.

Step : 8 Plot the graphs of y1, y2, y3, y4, y5 and ‘y’.

Output:



Result: Hence sum of sinusoidal signals is generated and plotted using MATLAB.

DESIGN OF M-POINT MOVING AVERAGE FILTER

Aim: To design and plot the output of M-point Moving Average Filter using MATLAB.

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Step : 1 Define a time vector n from 1:50.

Step : 2 Generate a signal 's' using s = $2n(0.9)^n$.

Step : 3 Generate a random noise signal 'd' with same length as that of 's'.

Step : 4 Obtain a corrupted signal 'p' using p = s+d.

Step : 5 M- point moving average filter is obtained by

$$x_{avg} = \frac{1}{M}\sum_{i=1}^{M} x(i)$$

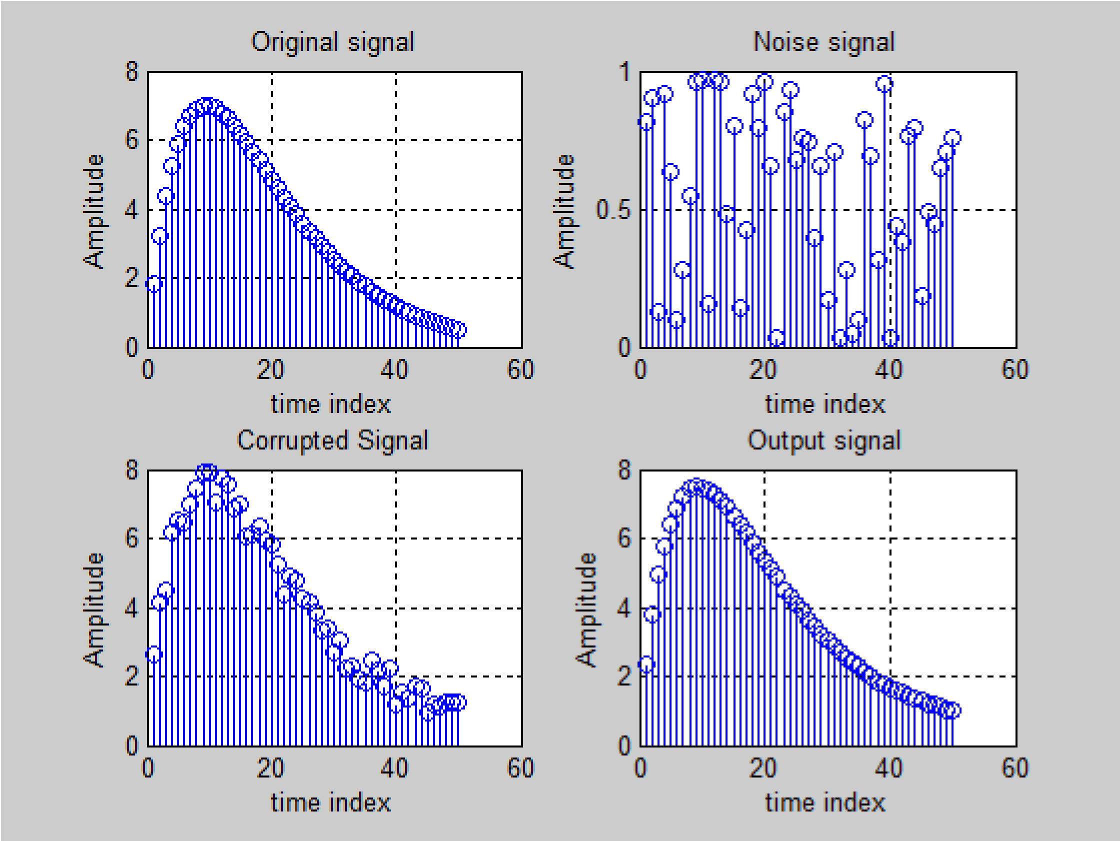
Use For loop to determine the above summation.

Step : 6 Plot the graphs of original signal 's', noise signal 'd', corrupted signal 'p' and filtered signal 'xavg'.

Procedure:

1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type 'edit' in the MATLAB prompt '>>' that appears in the Command window.
3. Write the program in the 'Edit' window and save it in 'M-file'.
4. Run the program.
5. Enter the input in the command window.
6. The result is displayed in the Command window and the graphical output is displayed in the Figure Window.

Output:



Result: Hence M-point Moving average filter is designed and output is observed using MATLAB.