DEPARTMENT OF
ELECTRONICS & COMMUNICATION ENGINEERING

DIGITAL SIGNAL PROCESSING LAB MANUAL

PVP20 REGULATIONS

III B.TECH II SEM

PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY
(Autonomous, Accredited by NBA & NAAC, an ISO 9001:2015 certified institution)
(Sponsored by Siddhartha Academy of General & Technical Education)
VIJAYAWADA – 520 007

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

DIGITAL SIGNAL PROCESSING LAB

LIST OF EXPERIMENTS

Part – A: (Using MATLAB)

1. Frequency response of a system described by a difference equation
   (First order and Second order Systems)
2. Implementation of discrete time systems in time domain
   (First order and Second order Systems)
3. DFT & IDFT of the given sequences
   (4-Point or 8-point sequences)
4. Properties of DFT (Linearity, Time reversal etc.)
5. Fast Fourier Transform (4-Point or 8-point sequences)
6. Design of IIR Low Pass filter using Butterworth and Chebyshev Approximations
   (For the given specifications)
7. Design of IIR High Pass filter using Butterworth and Chebyshev approximations
   (For the given specifications)
8. Design of FIR Low Pass filters using window technique
   (For the given specifications)
9. Design of FIR High Pass filter using window technique
   (For the given specifications)
10. Implementation of Interpolation and Decimation. (Factor 2 or 3)

Part – B: (Using Code Composer Studio)

1. Linear Convolution.
2. Circular Convolution.
3. Generation of Sine wave & Square wave.

Part – C: Additional Programs

1. Conversion of CD data to DVD data.
2. M-Point Moving Average Filter Design

Expt. No: 1	FREQUENCY RESPONSE OF A SYSTEM
Dt:	

Aim: To plot & observe the frequency response of first order and second order discrete-time 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 b.
Step : 2 Get/Read the coefficients of y(n) to a.
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:

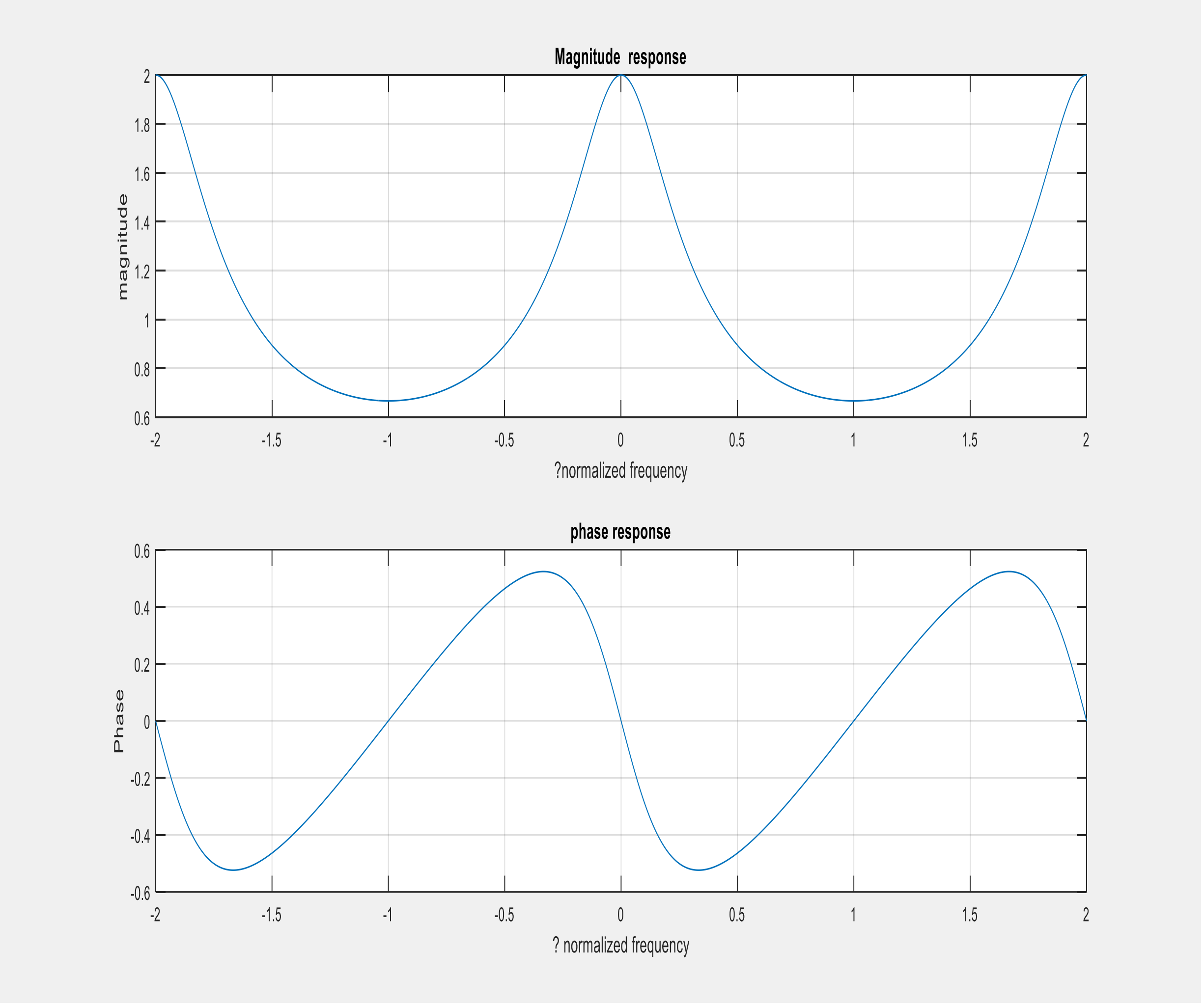
1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type ‘edit’ in the MATLAB prompt ‘>>’ that appears in the Command window.
3. Write the program in the ‘Edit’ window and save it in ‘M-file’.
4. Run the program.
5. Enter the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

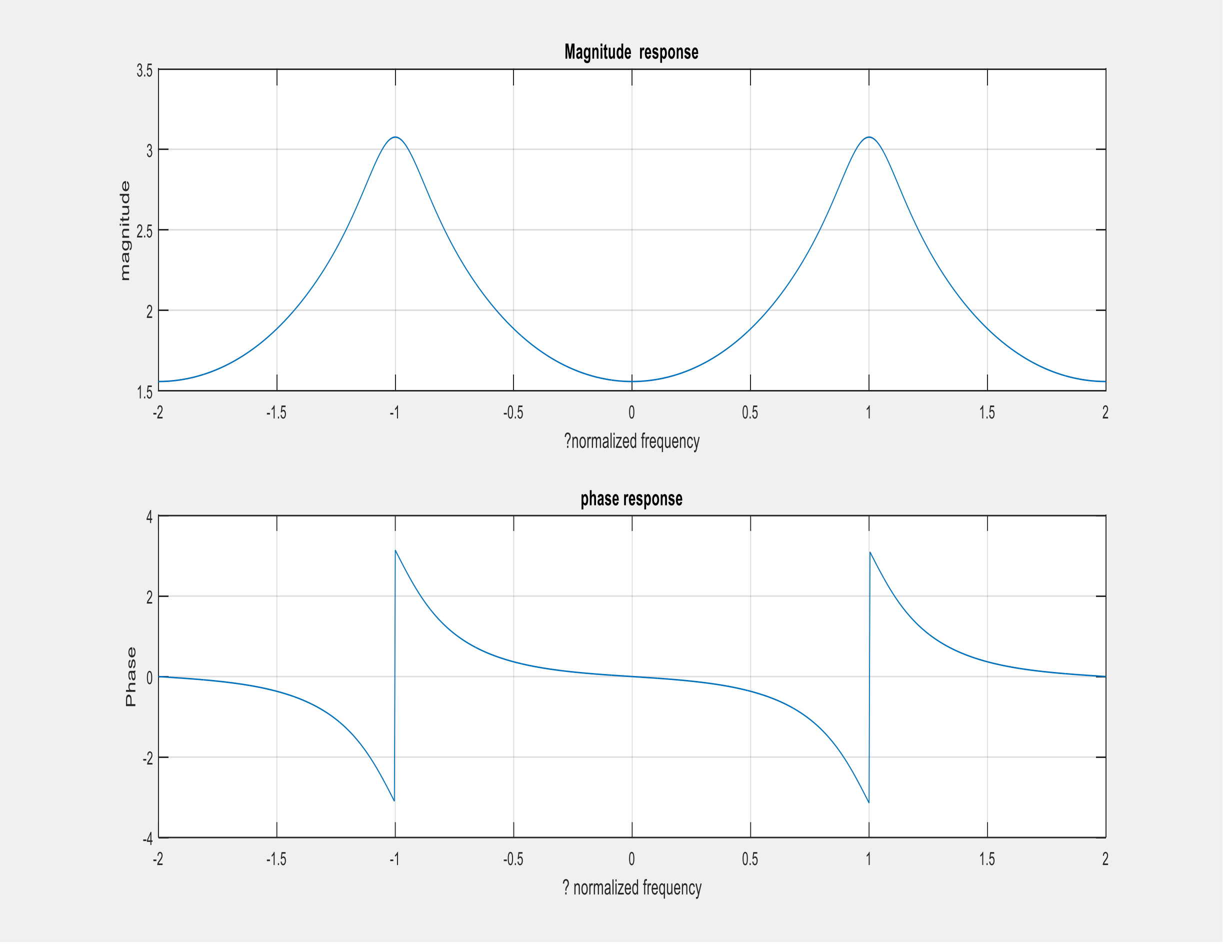


Input 2:

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 systems are observed and plotted using MATLAB.

Expt. No: 2	IMPLEMENTAION OF DISCRETE TIME SYSTEMS IN TIME DOMAIN
Dt:	

Aim: To implement the following discrete time systems described by difference equation in time domain using MATLAB.

a) y(n) - 0.8 y(n-1) = x(n)

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

Equipment Required:

PC loaded with MATLAB software

Algorithm:

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

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

Step : 3 Find the impulse response of the system by using the coefficients b and a.

Step : 4 Plot the impulse response of the system

Procedure:

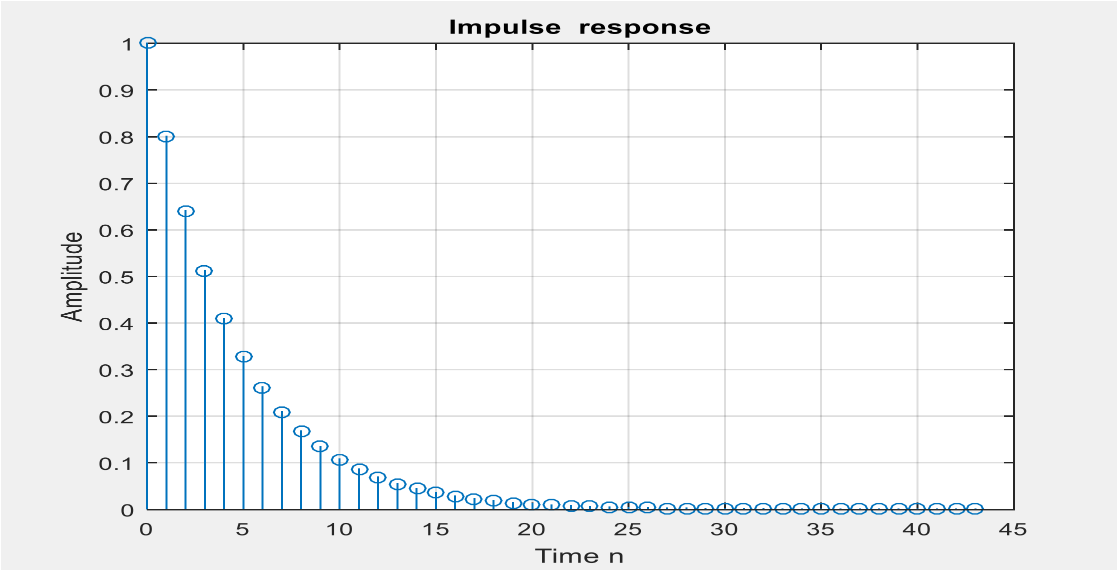
1. Click on the MATLAB icon on the desktop (or go to Start – All programs and click on MATLAB) to get into the Command Window.
2. Type 'edit' in the MATLAB prompt '>>' that appears in the Command window.
3. Write the program in the 'Edit' window and save it in 'M-file'.
4. Run the program.
5. Enter the 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, -0.8]

Output 1:

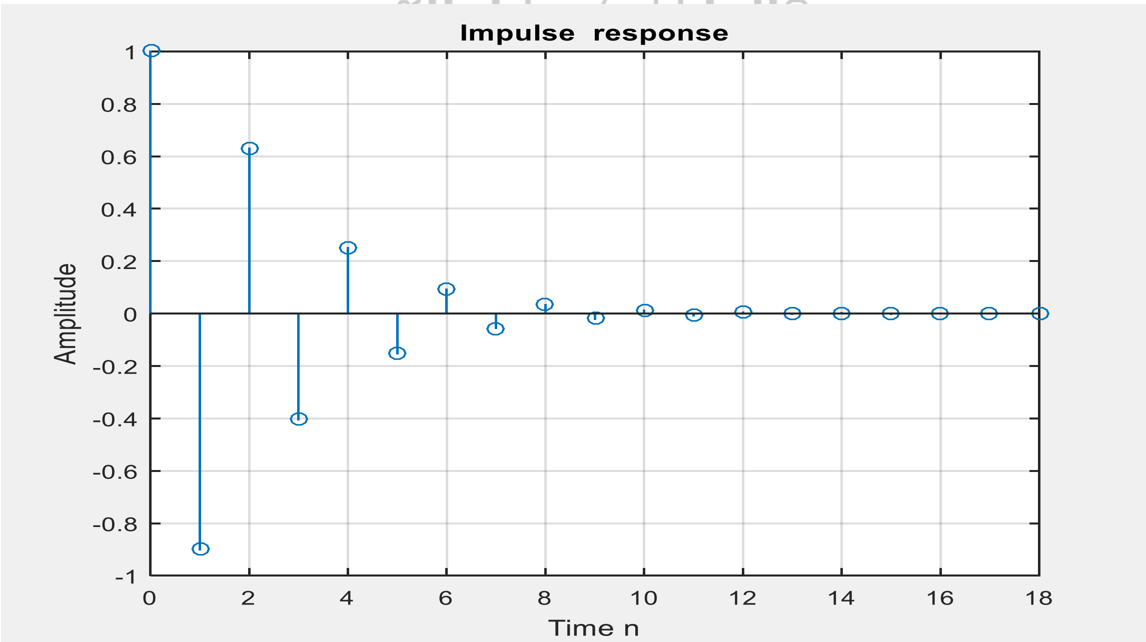


Input 2:

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

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

Output 2:



Result: Hence given discrete time system is implemented and its impulse response is observed using MATLAB.

Expt. No: 3	DFT and IDFT OF GIVEN SEQUENCE
Dt:	

Aim: To determine and plot the DFT and 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

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

Hint: Use two for loops for the above expression

Step : 5 Find the IDFT of the sequence xk using

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

Hint: 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.

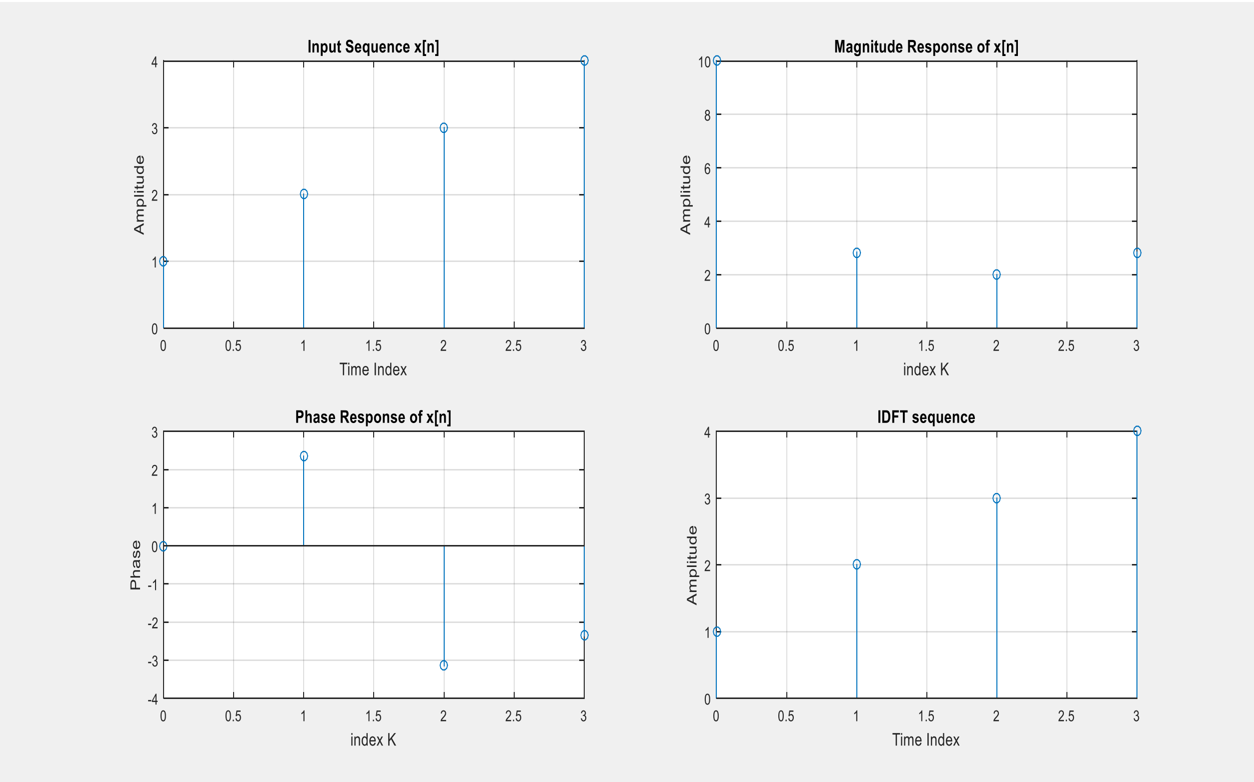
Input:
Enter the sequence x(n): [1 2 3 4]

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

X(K) = (10, -2+j2, -2, -2-j2)



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

Expt. No: 4	PROPERTIES OF DFT
Dt:	

Aim: To verify the following properties of DFT using MATLAB.

a) Linearity Property b) Circular Convolution Property

Equipment Required:

PC loaded with MATLAB software

Algorithm:

Linearity Property:

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

Step : 2 Get/Read the length of sequence x1 to N1.

Step : 3 Get/Read the samples of sequence x2[n] to x2.

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

Step : 5 Get/Read the value of ‘a’ to a.

Step : 6 Get/Read the value of ‘b’ to b.

Step : 7 Find the maximum value of (N1, N2) and store it in N.

Step : 8 If N1 > N2, append N1-N2 zeros to x2.

Step : 9 If N2 > N1, append N2-N1 zeros to x1.

Step : 10 Determine DFT of x1 and store it in X1.

Step : 11 Determine DFT of x2 and store it in X2.

Step : 12 Determine DFT of [a*x1 + b*x2] and store it in X.

Step : 13 Determine a*X1 + b* X2 and store it in X3.

Step : 14 Display X and X3.

Step : 15 Verify and compare X and X3

Circular Convolution Property:

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

Step : 2 Get/Read the length of sequence x1 to N1.

Step : 3 Get/Read the samples of sequence x2[n] to x2.

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

Step : 5 Find the maximum value of (N1, N2) and store it in N.

Step : 6 If N1 > N2, append N1-N2 zeros to x2.

Step : 7 If N2 > N1, append N2-N1 zeros to x1.

Step : 8 Determine N-point circular convolution of x1 and x2 and store it in x.

Step : 9 Determine DFT of x and store it in X.

Step : 10 Determine DFT of x1 and store it in X1.

Step : 11 Determine DFT of x2 and store it in X2.

Step : 12 Multiply X1 and X2 and store it in X3.

Step : 13 Display X and X3.

Step : 14 Verify and compare X and X3

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:

Case1:

Enter your choice: 1

Enter the input sequence x1: [1 2 3 4]

Enter the input sequence x2: [1 2 1 1]

Enter the value of constant a: 2

Enter the value of constant b: 3

Output1:

DFT[ax1(n)+bx2(n)] =

35.0000 + 0.0000i -4.0000 + 1.0000i -7.0000 - 0.0000i -4.0000 - 1.0000i

aX1(k)+bX2(k) =

35.0000 + 0.0000i -4.0000 + 1.0000i -7.0000 - 0.0000i -4.0000 - 1.0000i

Input:

Case2:

Enter your choice: 2

Enter the sequence x1: [1 2 3 4]

Enter the sequence x2: [1 2 1 1]

Output2:

DFT(x1(n)*x2(n)) =

50.0000 + 0.0000i   2.0000 + 2.0000i   2.0000 + 0.0000i   2.0000 - 2.0000i

X1(k).X2(k) =

50.0000 + 0.0000i   2.0000 + 2.0000i   2.0000 + 0.0000i   2.0000 - 2.0000i

Result: Hence the Linearity and Circular Convolution properties of DFT are verified using MATLAB.

Expt. No: 5	FAST FOURIER TRANSFORM
Dt:	

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 FFT 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 Display Y.

Step : 8 Determine the magnitude of Y and store it in my.

Step : 9 Determine the phase of Y and store it in py.

Step : 10 Plot the Graphs of 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 FFT: 8

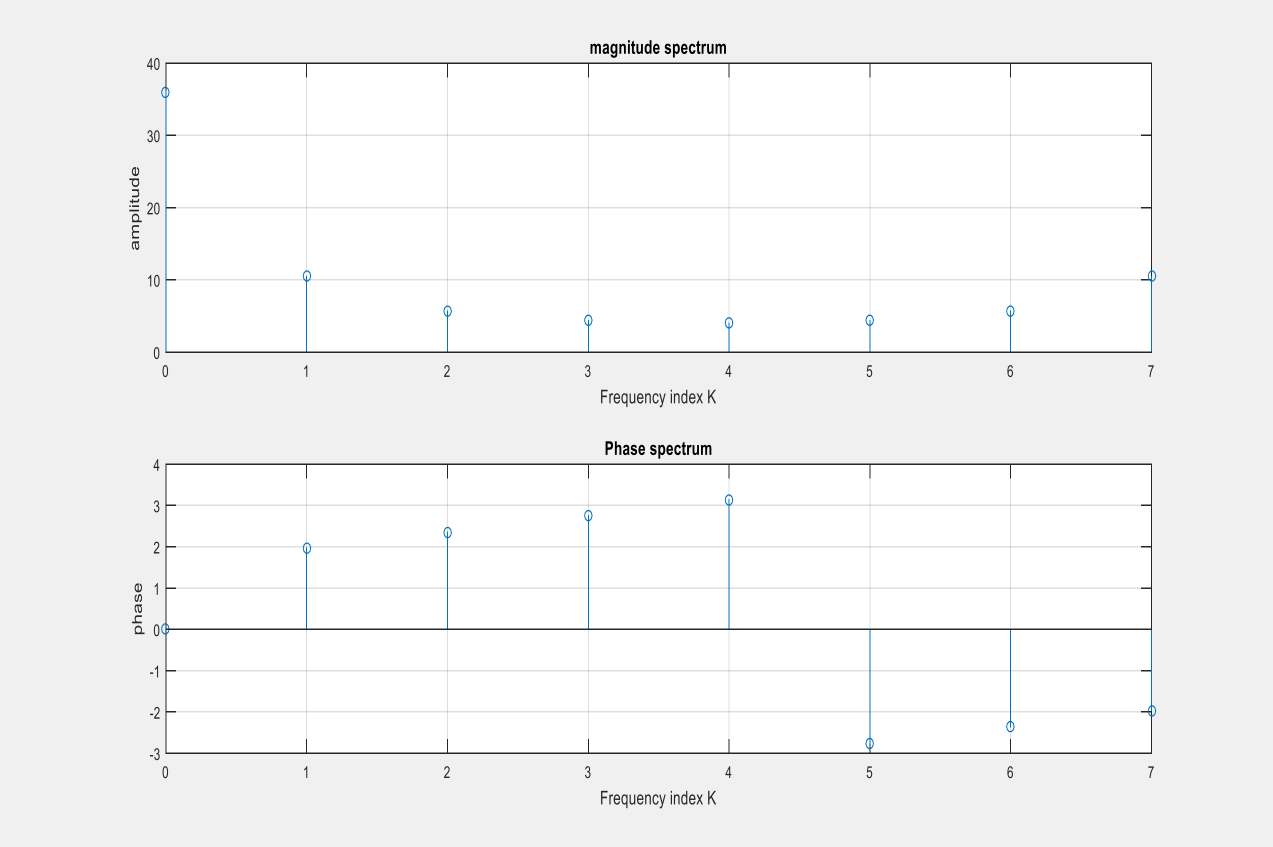
enter the samples of sequence: [1 2 3 4 5 6 7 8]

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

Expt. No: 6	DESIGN OF IIR LOW PASS DIGITAL FILTER
Dt:	

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

w1=2*fp/f; w2=2*fs/f;

Step : 7 Define the frequency vector w from 0 to 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 'n'.

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 in 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

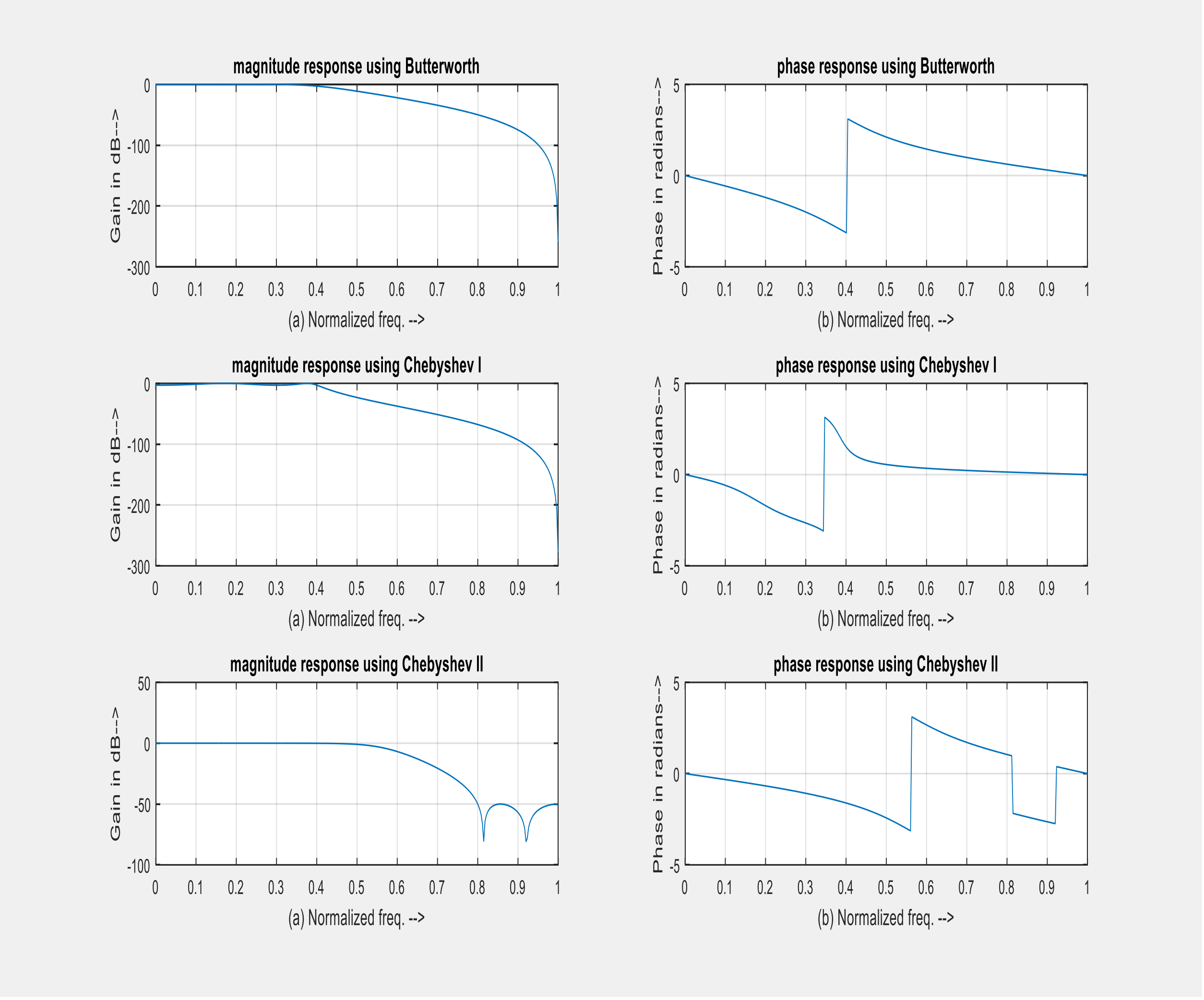
enter the stopband ripple: 50

enter the passband frequency: 1000

enter the stopband frequency: 2000

enter the sampling frequency: 5000

Output:



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

Expt. No: 7	DESIGN OF IIR HIGH PASS DIGITAL FILTER
Dt:	

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

w1=2*fp/f; w2=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 'n'.

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 in 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

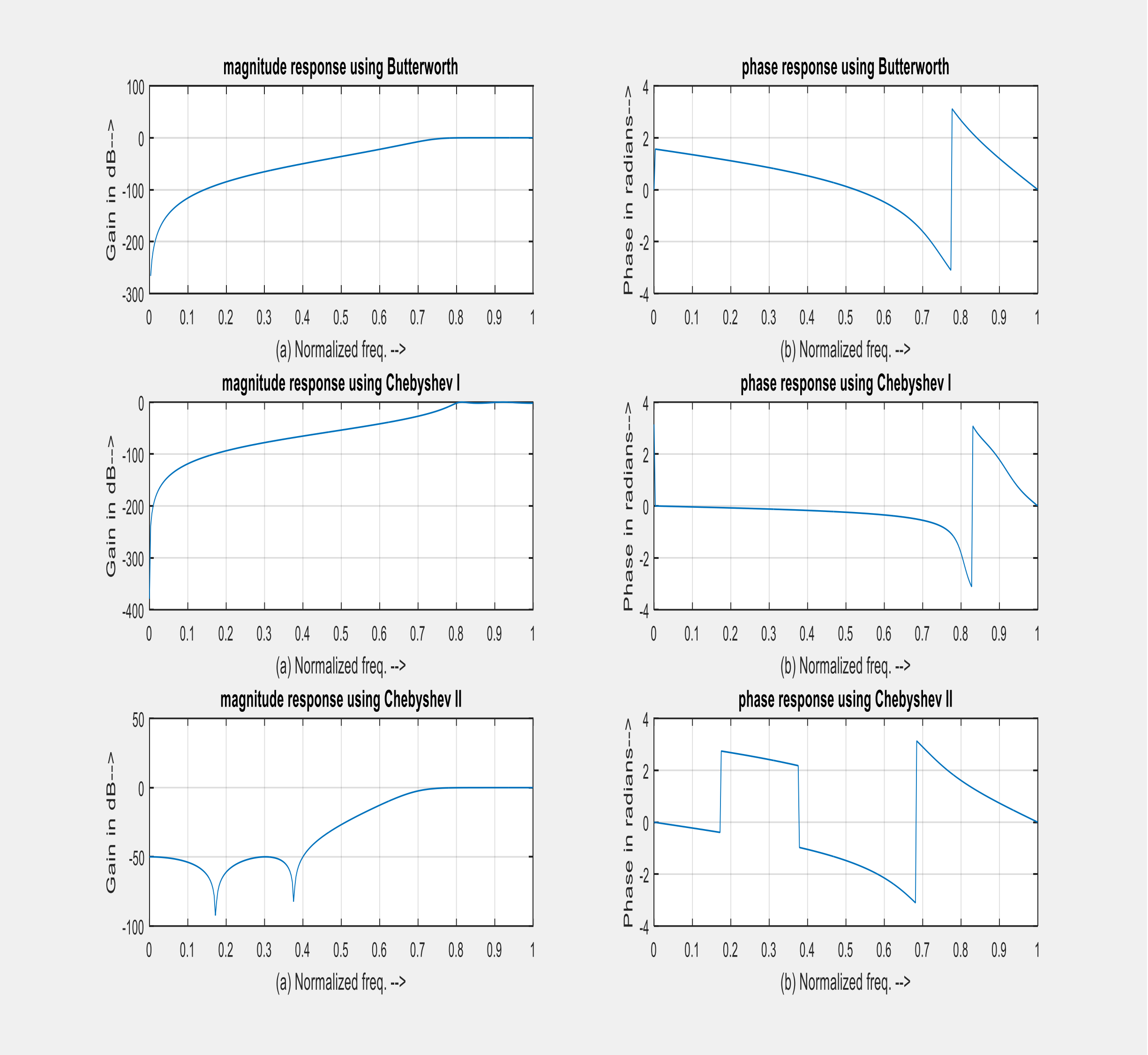
enter the stopband ripple: 50

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.

Expt. No: 8	DESIGN OF FIR LOW PASS DIGITAL FILTER
Dt:	

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 Determine the order 'n' of the filter.

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

Step : 7 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 : 8 Normalize the pass band and stop band frequencies using

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

Step : 9 Define the frequency vector w from 0 to π.

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 in 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

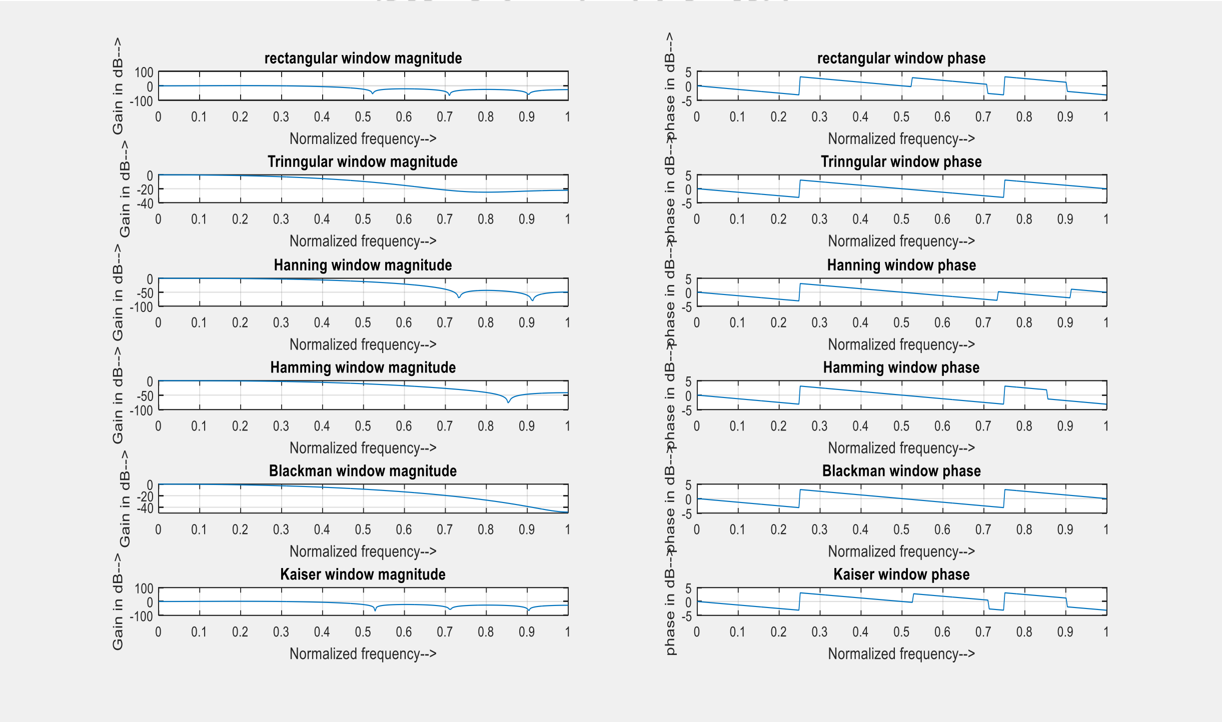
enter the stopband ripple: 0.01

enter passband frequency: 1000

enter stopband frequency: 2000

enter sampling frequency: 5000

Output:



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

Expt. No: 9	DESIGN OF FIR HIGH PASS DIGITAL FILTER
Dt:	

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 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 : 7 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 : 8 Normalize the pass band and stop band frequencies using

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

Step : 9 Define the frequency vector w from 0 to π.

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 in 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

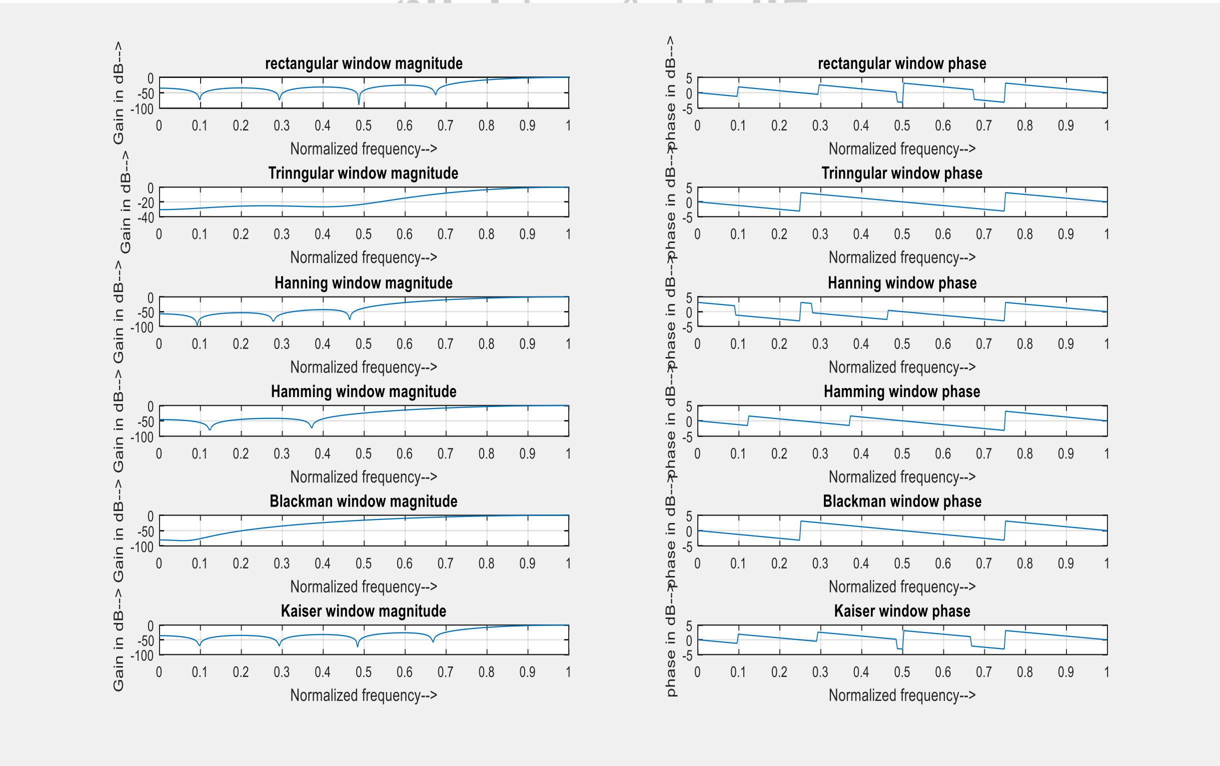
enter the stopband ripple: 0.01

enter passband frequency: 2000

enter stopband frequency: 1000

enter sampling frequency: 5000

Output:



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

Expt. No: 10	IMPLEMENTATION OF INTERPOLATION AND DECIMATION
Dt:	

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 frequency 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 original signal x using

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

Step : 9 Perform interpolation on x using interpolation factor 'i'.

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 frequency 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'.

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

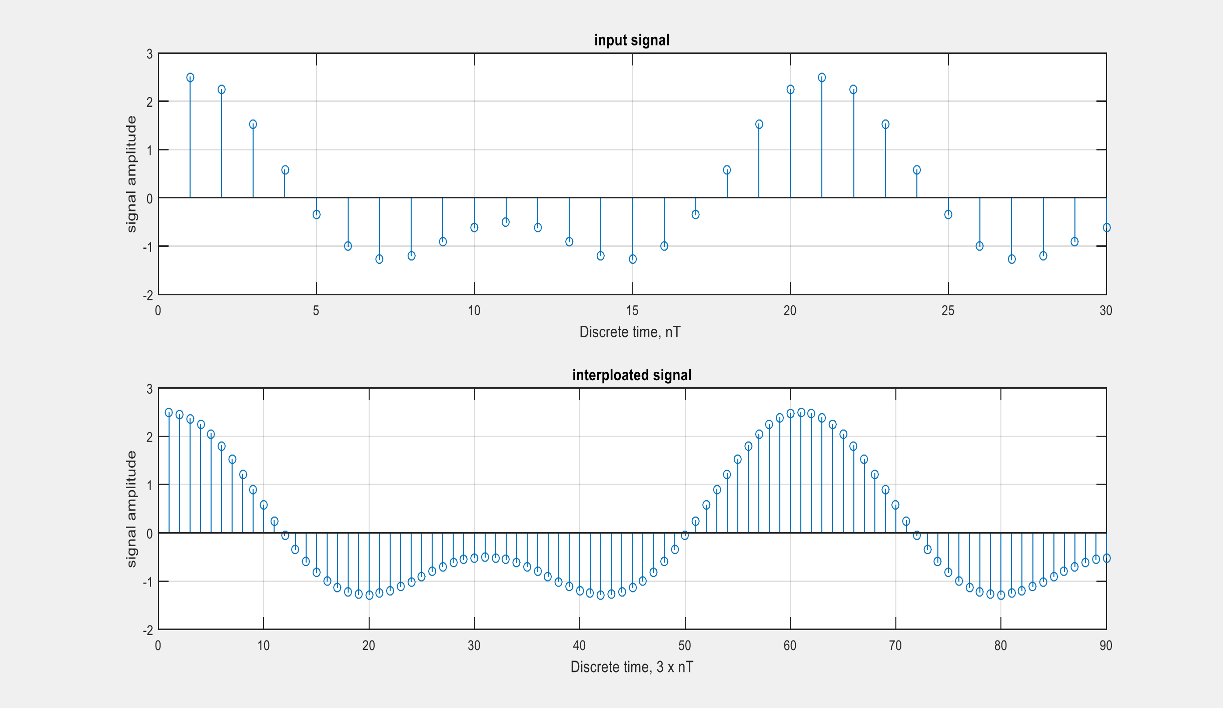
Enter the amplitude of second Sinusoidal: 1

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

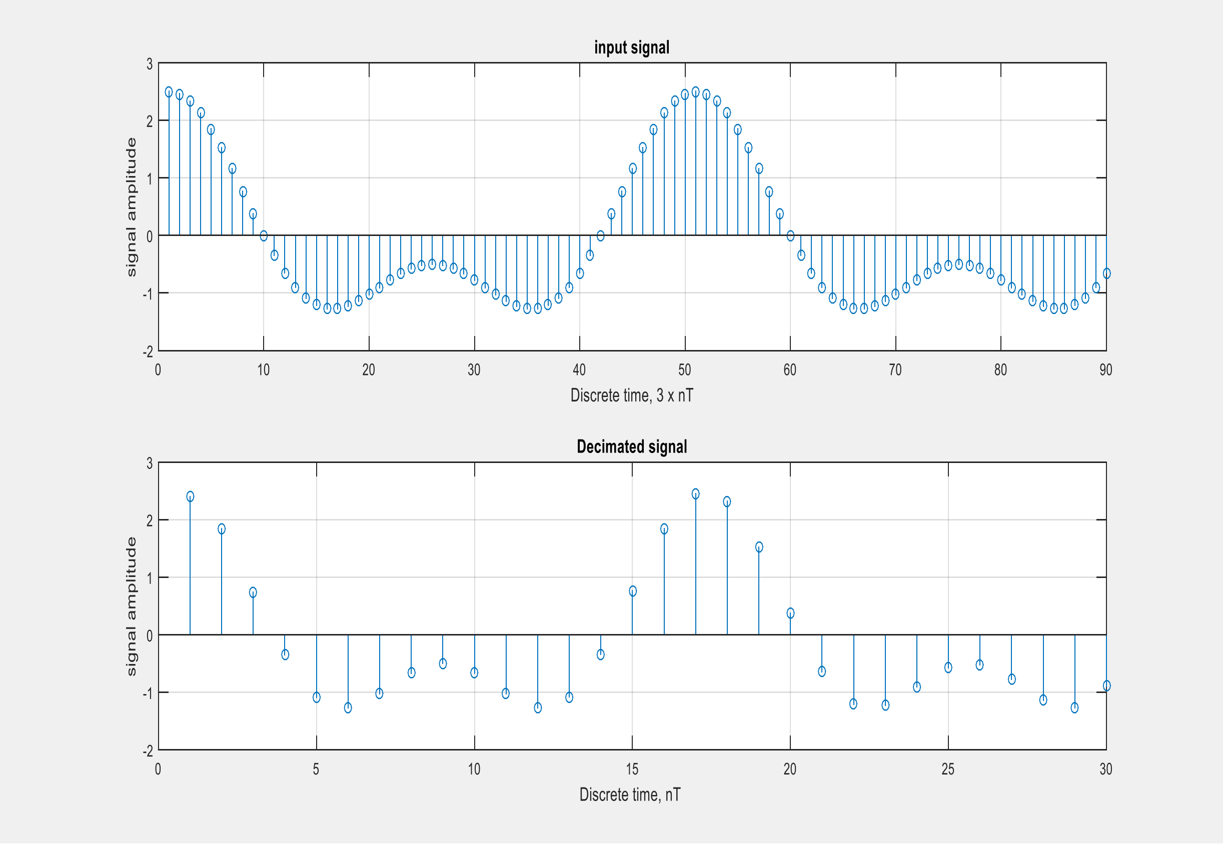
Enter the amplitude of second Sinusoidal: 1

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.

Expt. No: 11	LINEAR CONVOLUTION
Dt:	

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].

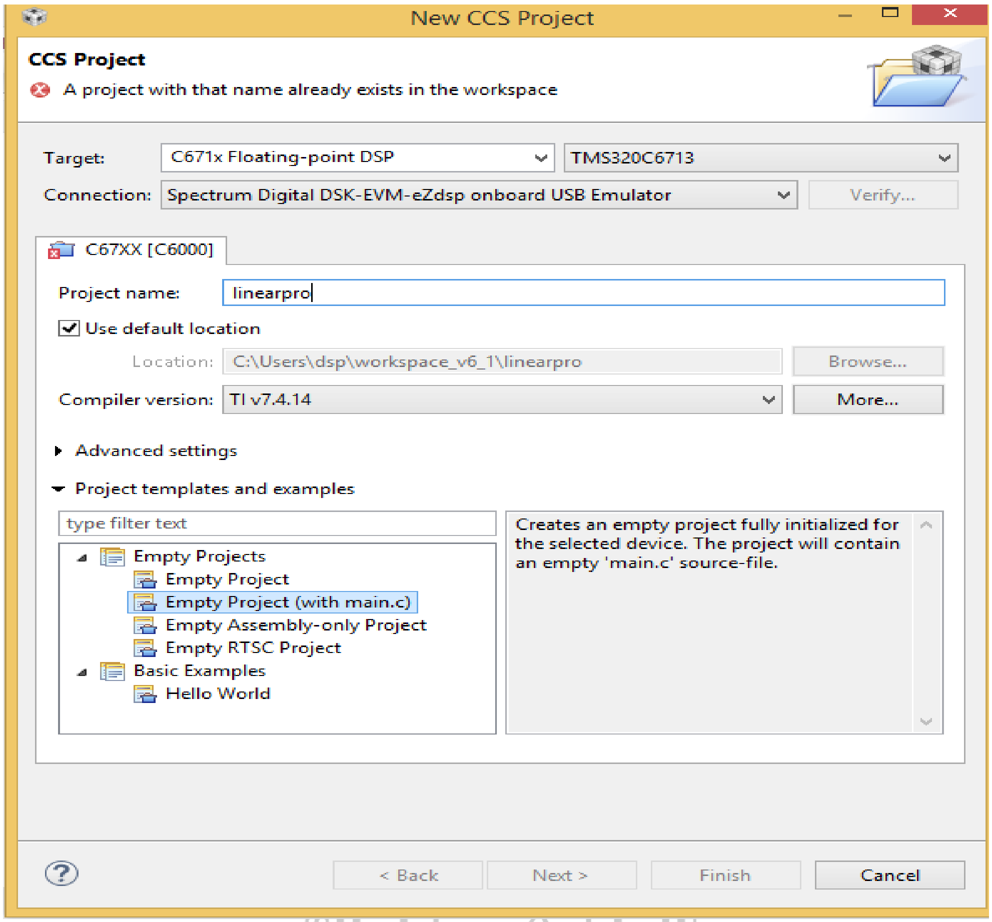
Procedure:

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

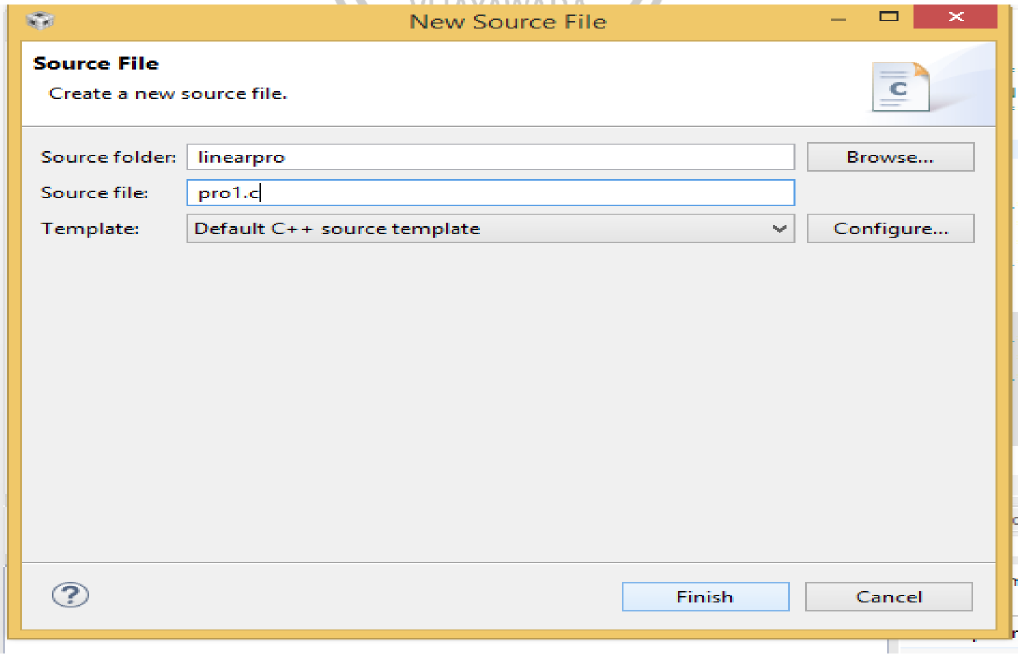
Target: C671X Floating-point DSP TMS320C671X

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



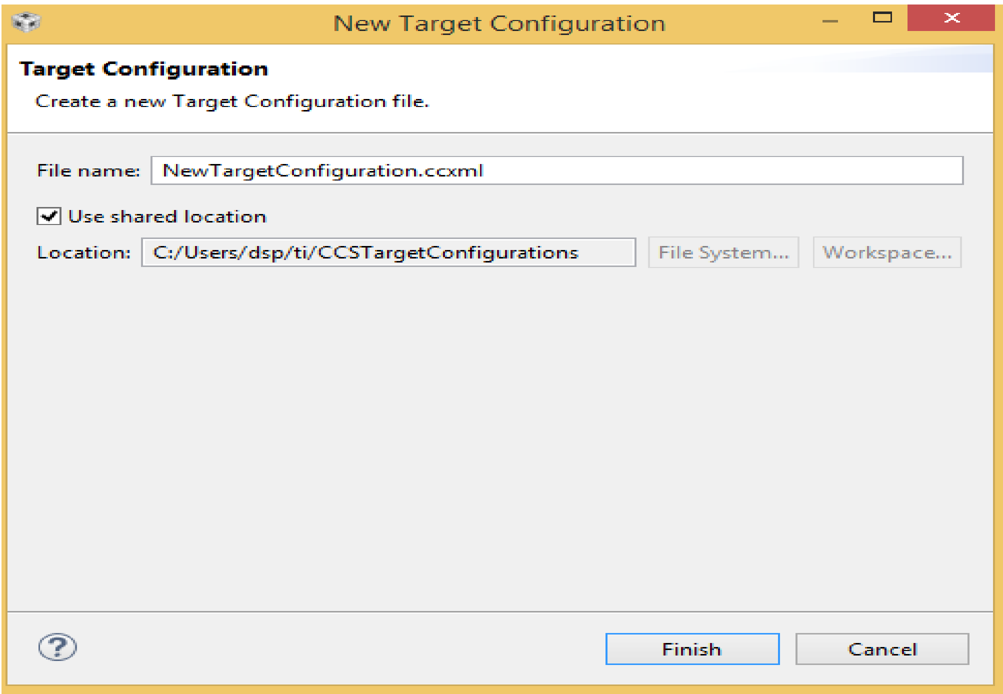
4. Select empty project with main.c and finish.
5. Create a new Source file

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

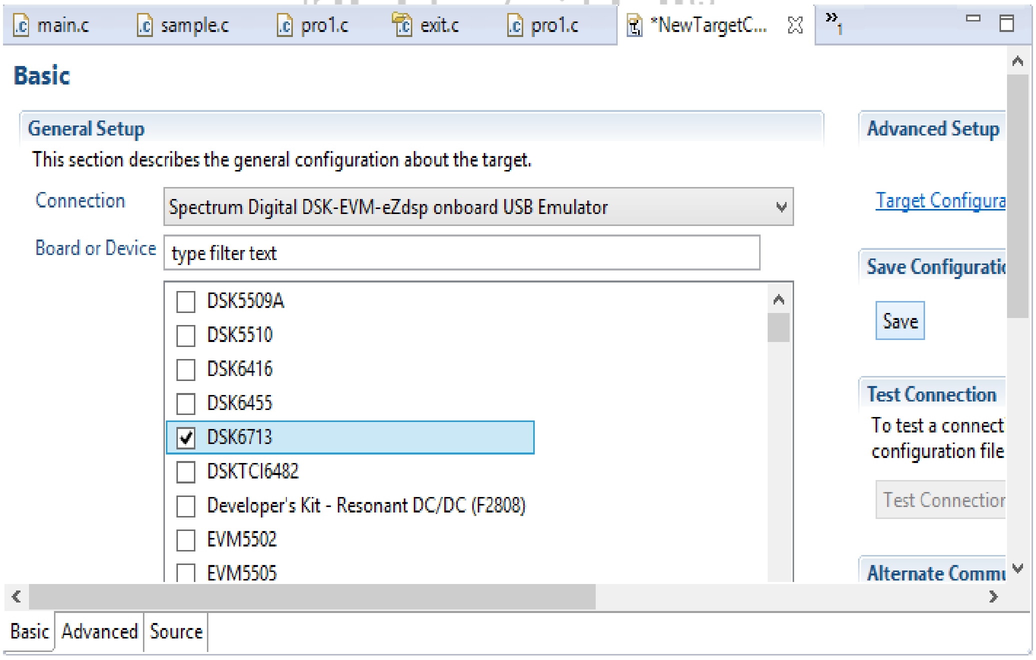


6. Write C program and save it.
7. Create New Target configuration file with extension .ccxml

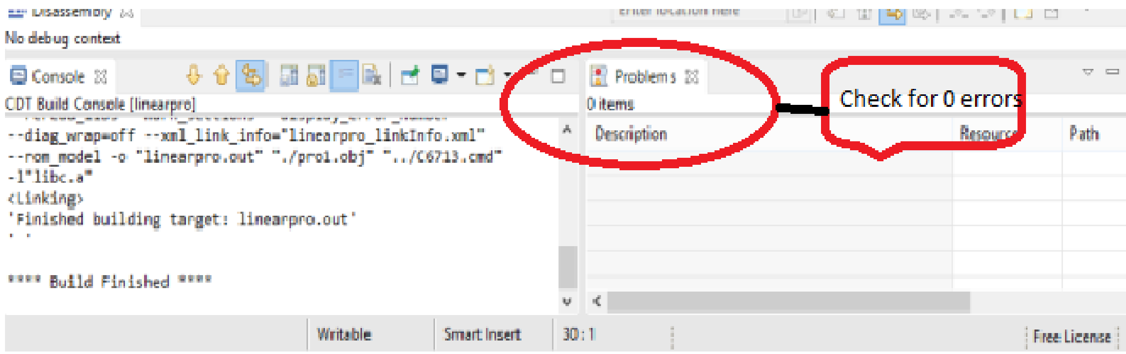
File → New → Target configuration file



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

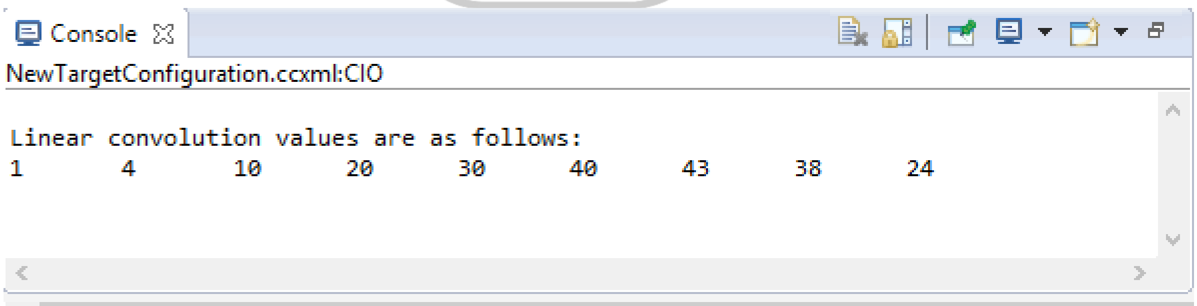
    Run→ Load Programme →Browse .out file

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}

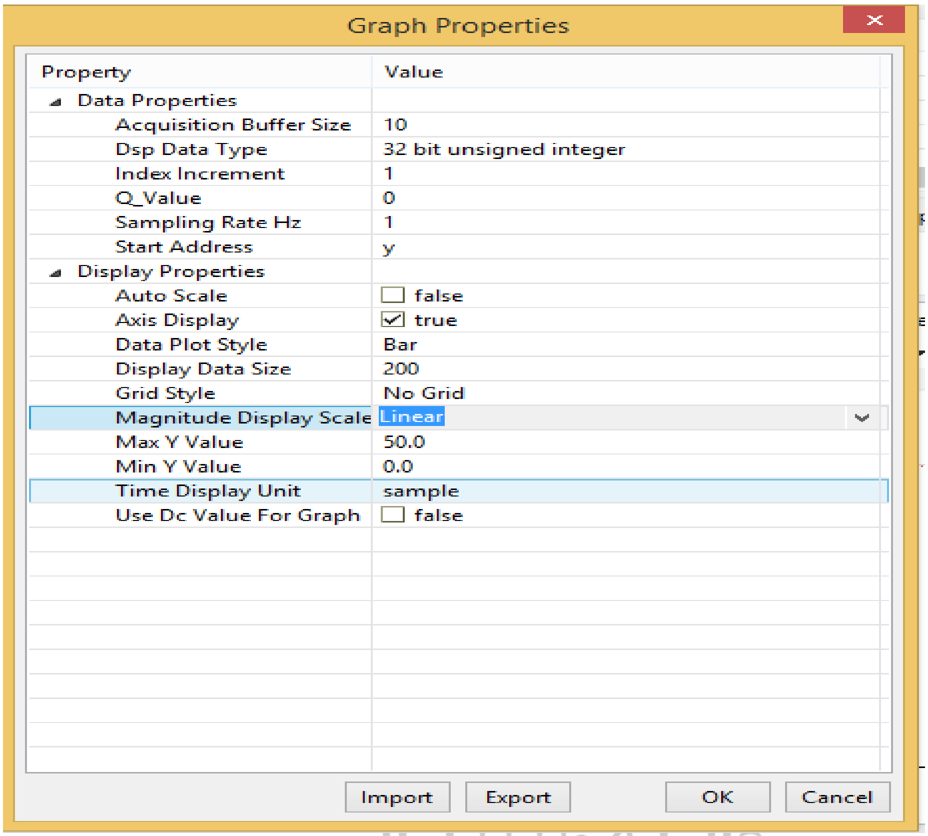
Output:

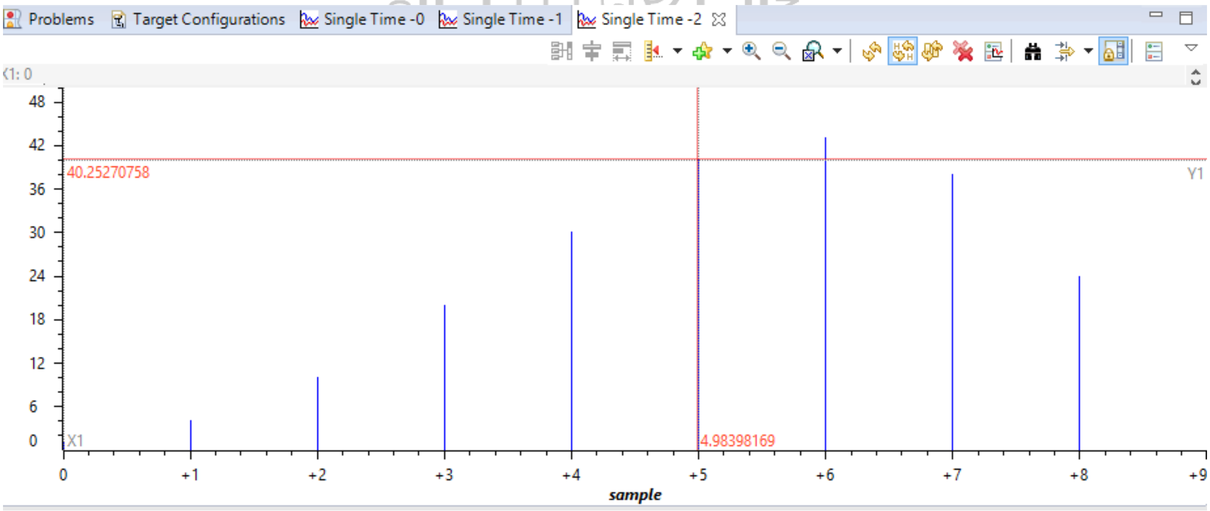


To view output graphically,

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.

Expt. No: 12	CIRCULAR CONVOLUTION
Dt:	

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

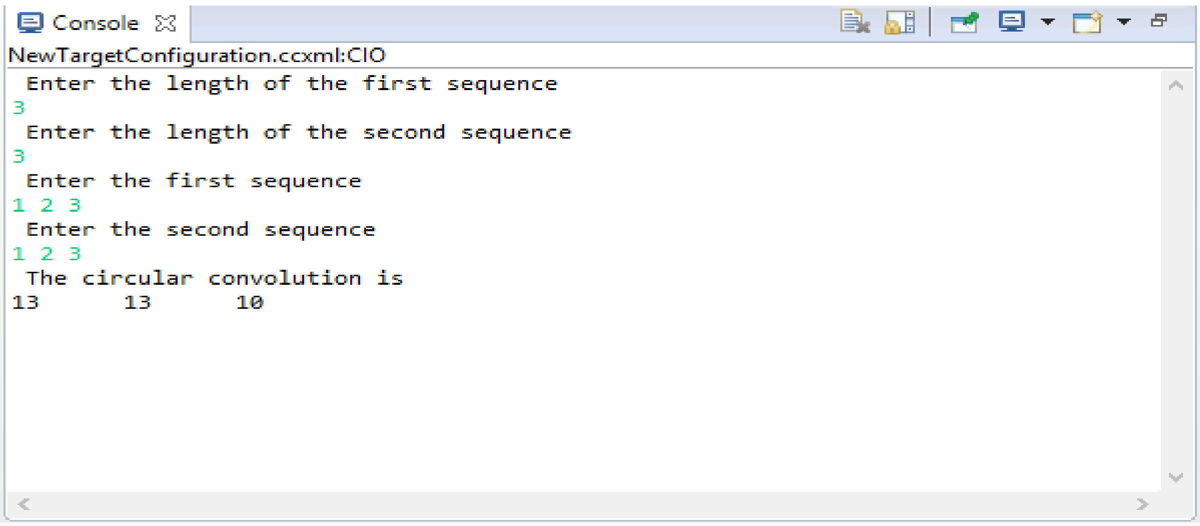
Enter the samples of first sequence:{1 2 3}

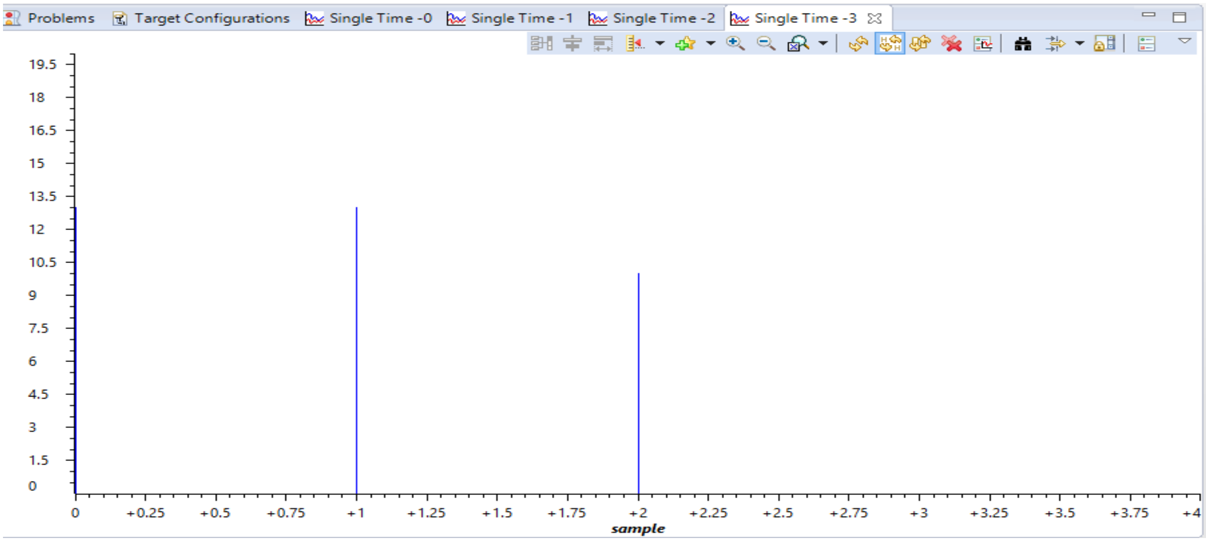
Enter the samples of second sequence:{1 2 3}

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.

Expt. No: 13	GENERATION OF SINE WAVE AND SQUARE WAVE
Dt:	

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.

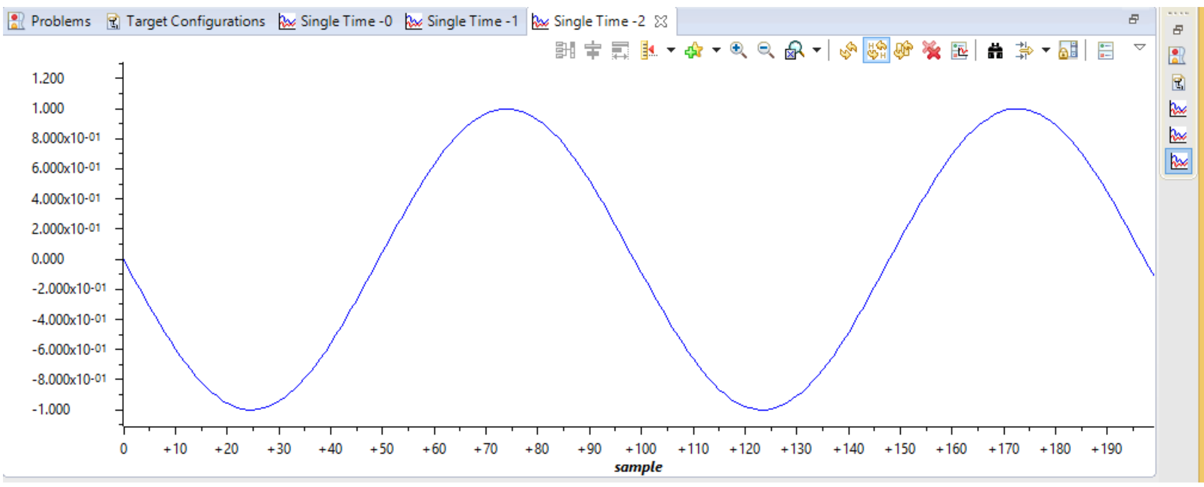
12. Observe the output in the console window

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

Input:

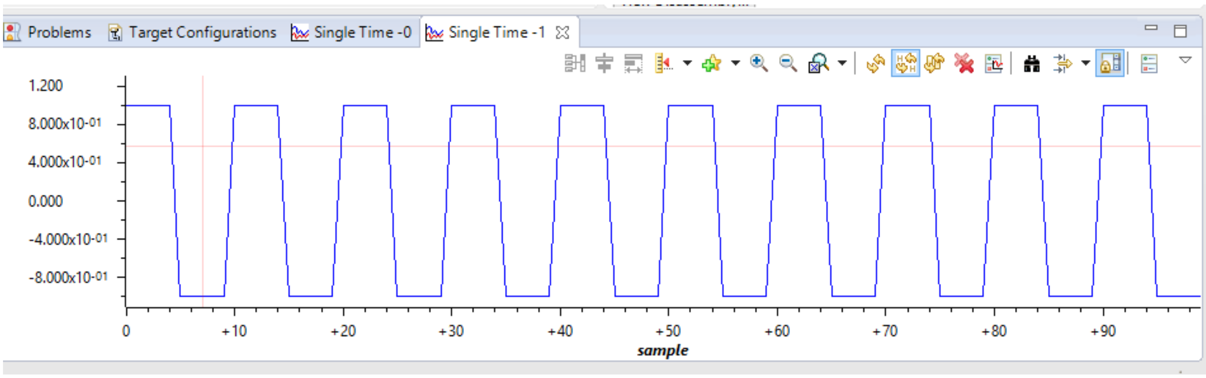
enter the frequency: 10
enter the no. of samples: 512

Output:



Input:
enter the time period: 10

Output:



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

Expt. No: 14	CONVERSION OF CD DATA TO DVD DATA
Dt:	

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 to f_c = 44.1 KHz.

Step : 2 Define time Vector t.

Step : 3 Obtain 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

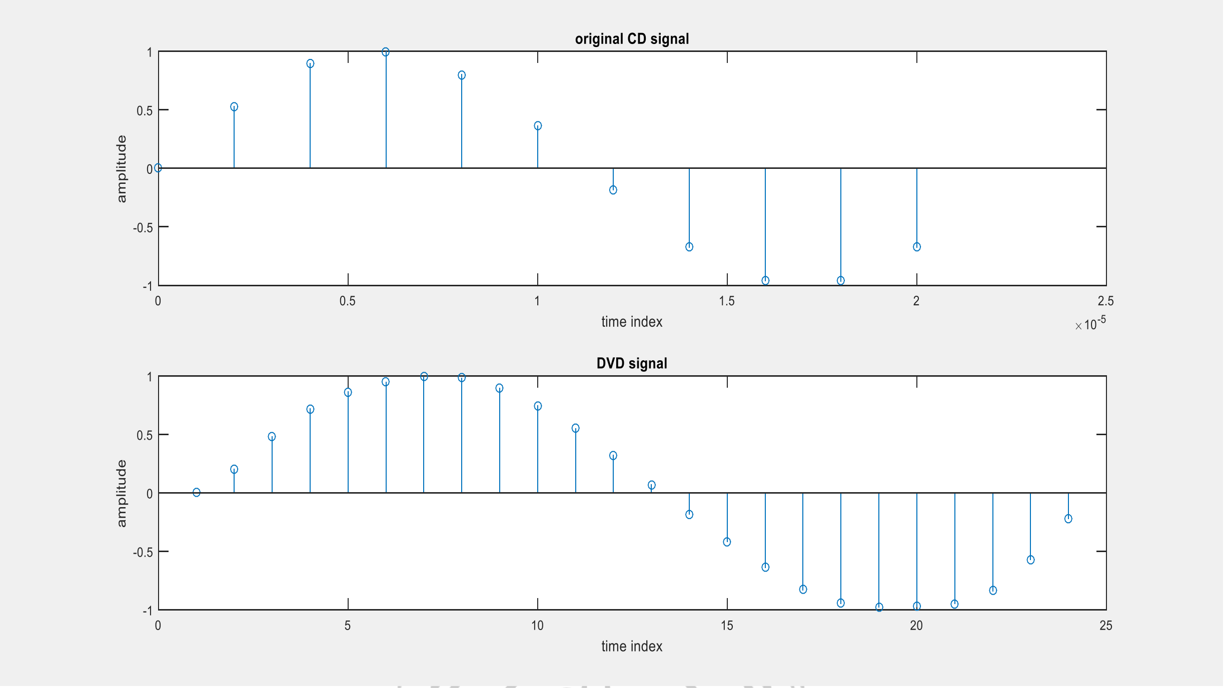
Step : 7 Use resample function to obtain DVD signal 'y'.

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.

Expt. No: 15	DESIGN OF M-POINT MOVING AVERAGE FILTER
Dt:	

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 'x' using x = $2n(0.9)^n$

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

Step : 4 Obtain a corrupted signal 'xd' using xd = x+d.

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

$$xavg = \frac{1}{M}\sum_{i=1}^{M} xd(i)$$

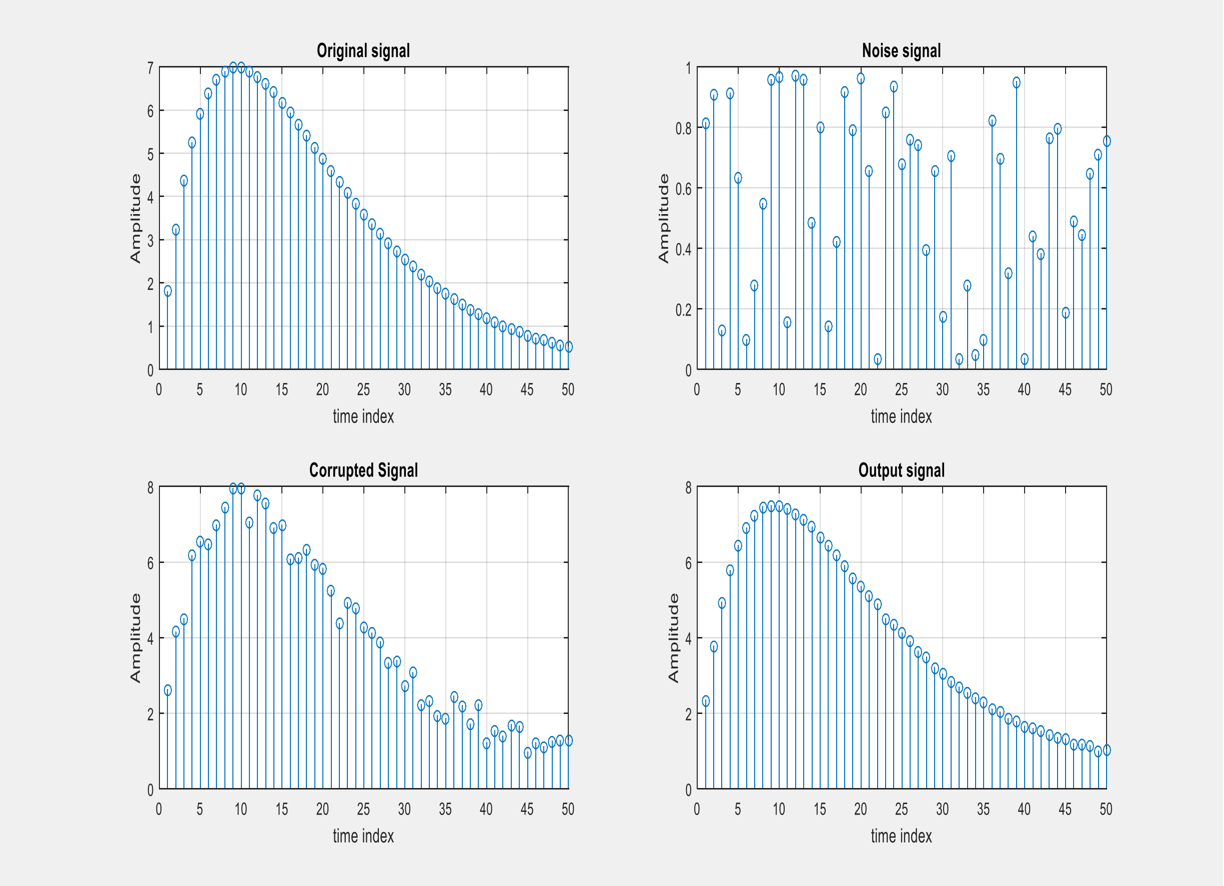
Use For loop to determine the above summation.

Step : 6 Plot the graphs of original signal 'x', noise signal 'd', corrupted signal 'xd' 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.