LABORATORY MANUAL

SIGNAL & SYSTEM LAB

B.Tech. (Electrical Engineering), 5^{th} Semester

Department of Electrical Engineering
Veer Surendra Sai University of Technology,
BURLA

VISION

To be recognized as a centre of excellence in education and research in the field of Electrical Engineering by producing innovative, creative and ethical Electrical Engineering professionals for socio-economic development of society in order to meet the global challenges.

MISSION

Electrical Engineering Department of VSSUT Burla strives to impart quality education to the students with enhancement of their skills to make them globally competitive through:

M1. Maintaining state of the art research facilities to provide enabling environment to create, analyze, apply and disseminate knowledge.

M2. Fortifying collaboration with world class R&D organizations, educational institutions, industry and alumni for excellence in teaching, research and consultancy practices to fulfill 'Make in India' policy of the Government.

M3. Providing the students with academic environment of excellence, leadership, ethical guidelines and lifelong learning needed for a long productive career.

Program Educational Objectives

The program educational objectives of B.Tech. in Electrical Engineering program of VSSUT Burla are to prepare its graduates:

1. To have basic and advanced knowledge in Electrical Engineering with specialized knowledge in design and commissioning of electrical systems/renewable energy systems comprising of generation, transmission and distribution to become eminent, excellent and skilful engineers.

2. To succeed in getting engineering position with electrical design, manufacturing industries or in software and hardware industries, in private or government sectors, at Indian and in Multinational organizations.

3. To have a well-rounded education that includes excellent communication skills, working effectively on team-based projects, ethical and social responsibility.

4. To have the ability to pursue study in specific area of interest and be able to become successful entrepreneur.

5. To have broad knowledge serving as foundation for lifelong learning in multidisciplinary areas to enable career and professional growth in top academic, industrial and government/corporate organizations.

LIST OF EXPERIMENTS

1) Generation of square wave, triangular, exponential, sinusoidal, step, impulse, and ramp function.
2) Verification of time shifting, time scaling, reflection operations on square wave, triangular, exponential, sinusoidal, step, impulse, and ramp function.
3) To evaluate the convolution of finite discrete time signals and to verify the commutative, associative, distributive, and identity property.
4) To evaluate convolution integral of a given signal.
5) To evaluate discrete time Fourier transform (DTFT) of a signal.
6) To compute frequency response of LTI system from impulse response.
7) To compute frequency response of LTI system by difference equation.
8) Generation of Amplitude Modulation (AM) wave and analysing as frequency content.
9) Determination of frequency response from poles and zeroes.
10) Pole-zero plots on z-plane and determination of magnitude response.
11) To find the impulse response of a system described by Z transform.
12) Implementation of interpolation and decimation concept.

COURSE OUTCOMES:

CO1: Describe various elementary signals and verify its independent variable properties.

CO2: Express the concept of convolution of LTI system.

CO3: Describe the basics of modulation and frequency response of LTI system.

CO4: Apply the knowledge how to use the Laplace Transform for representing signal.

CO5: Apply the knowledge how to use the Z Transform for representing signal Course

EXPERIMENT-1

Aim of experiment:-

Generation of square wave, triangular, exponential, sinusoidal, step, impulse, and ramp function.

Software used: MATLAB R2023a

Theory:

A signal is defined as a physical quantity that varies with time, space or any other independent variable. Three signals include step, impulse, ramp , sinusoidal , and exponential function.

Square wave:Square wave is a non-sinusoidal periodic waveform in which the amplitude alternates at a steady frequency between fixed maximum and minimum value with same duration of maximum and minimum.

$$\pi(t) = \begin{cases} 1, for\ |t| < 0.5 \\ 0, otherwise \end{cases}$$

Triangular wave: Triangular wave is a non-sinusoidal waveform named for its triangular shape . It is perodic , piecewise , linear , continuous , real function.

$\Delta_a(t)$= { 1-| t | / a , | t |<a

{ 0 , | t |>a

Exponential signal: It is written as $x(t)=A\ e^{at}$ where A and a are real . Depending on value of a , we get different signals. If 'a' is +ve , x(t) is growing exponential and if 'a' is -ve , x(t) is decaying.

Sinusoidal signal: It is a mathematical curve that describes a smooth repetitive oscillation. y(t)=sin(t). It occurs in pure and applied mathematics , physics and signal processing.

Unit step function: It is defined as u(t)={1 , t $\geqslant$0

{0 , t < 0

u(n)={1 , n $\geqslant$0

{0 , n < 0

Unit Ramp function:It is defined as r(t) = { t , t $\geqslant$ 0

{0 , t<0

r(n) = { n , n $\geqslant$ 0

{0 , n<0

Unit impulse function:It is defined as δ(t)= { 1 , t = 0

{ 0 , t ≠ 0

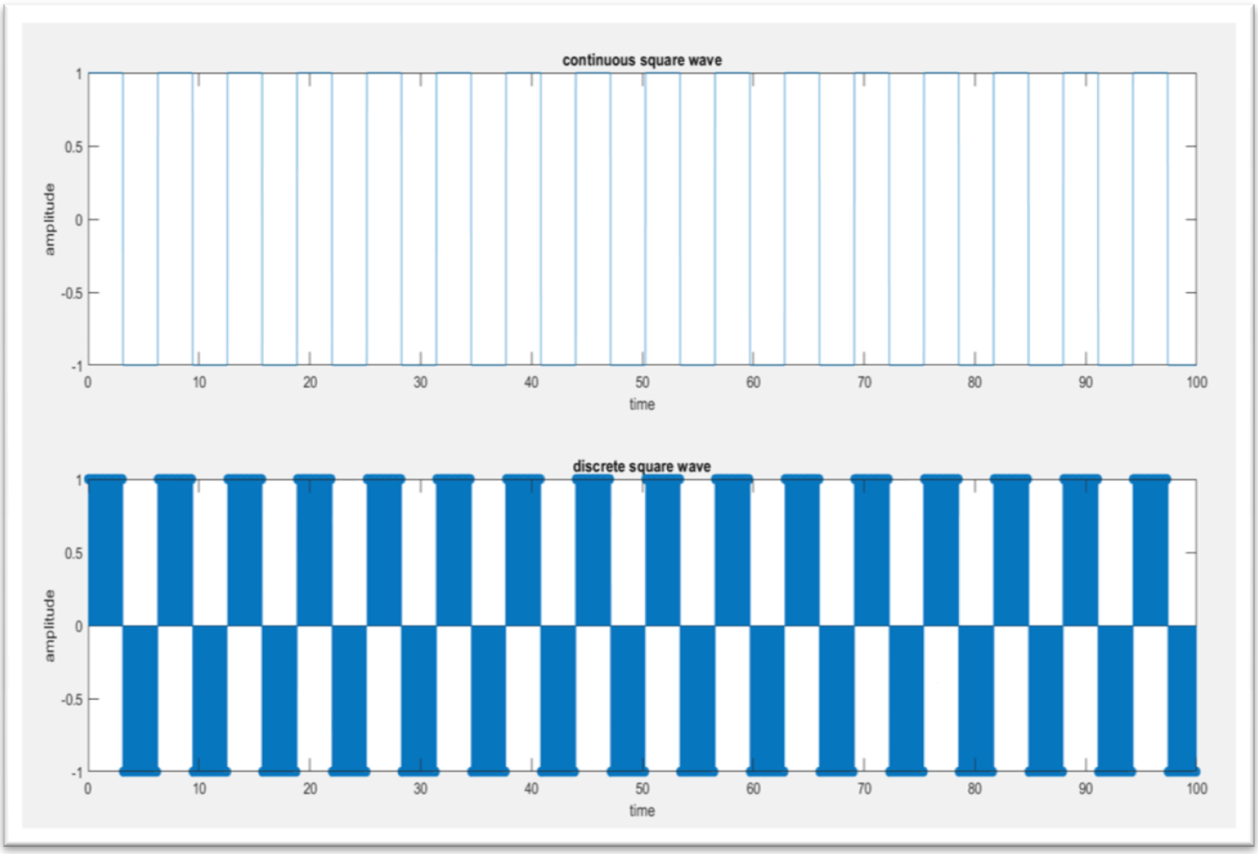
$$\delta(n) = \{\ 1\ ,\ n = 0$$

$$\{\ 0\ ,\ t \neq 0$$

Code:-

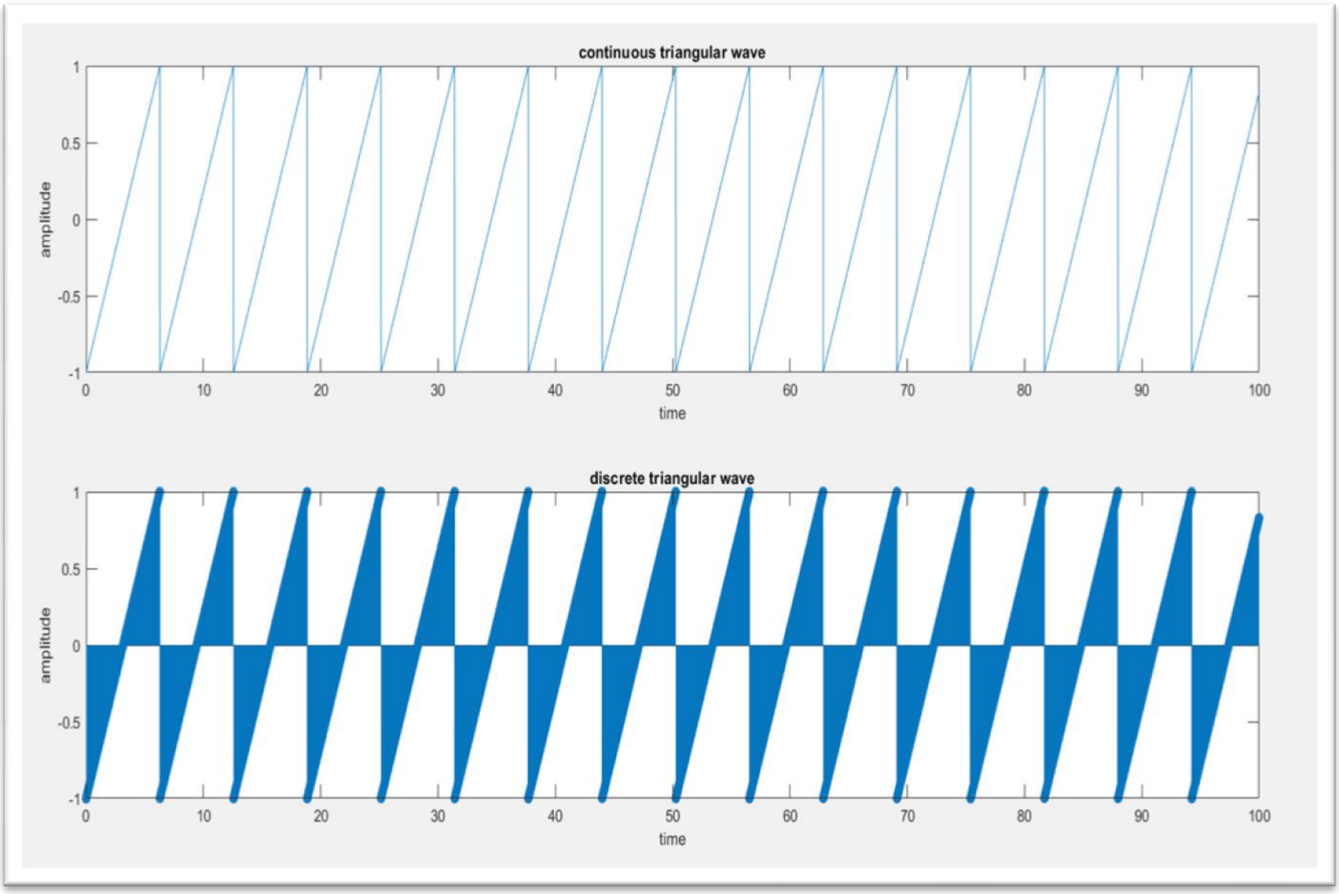
1) Square wave:

```
t=0:0.01:100;
y=square(t);
subplot(2,1,1);
plot(t,y);
xlabel('time');
ylabel('amplitude');
title('continuous square wave');
subplot(2,1,2);
stem(t,y);
xlabel('time');
ylabel('amplitude');
title('discrete square wave');
```



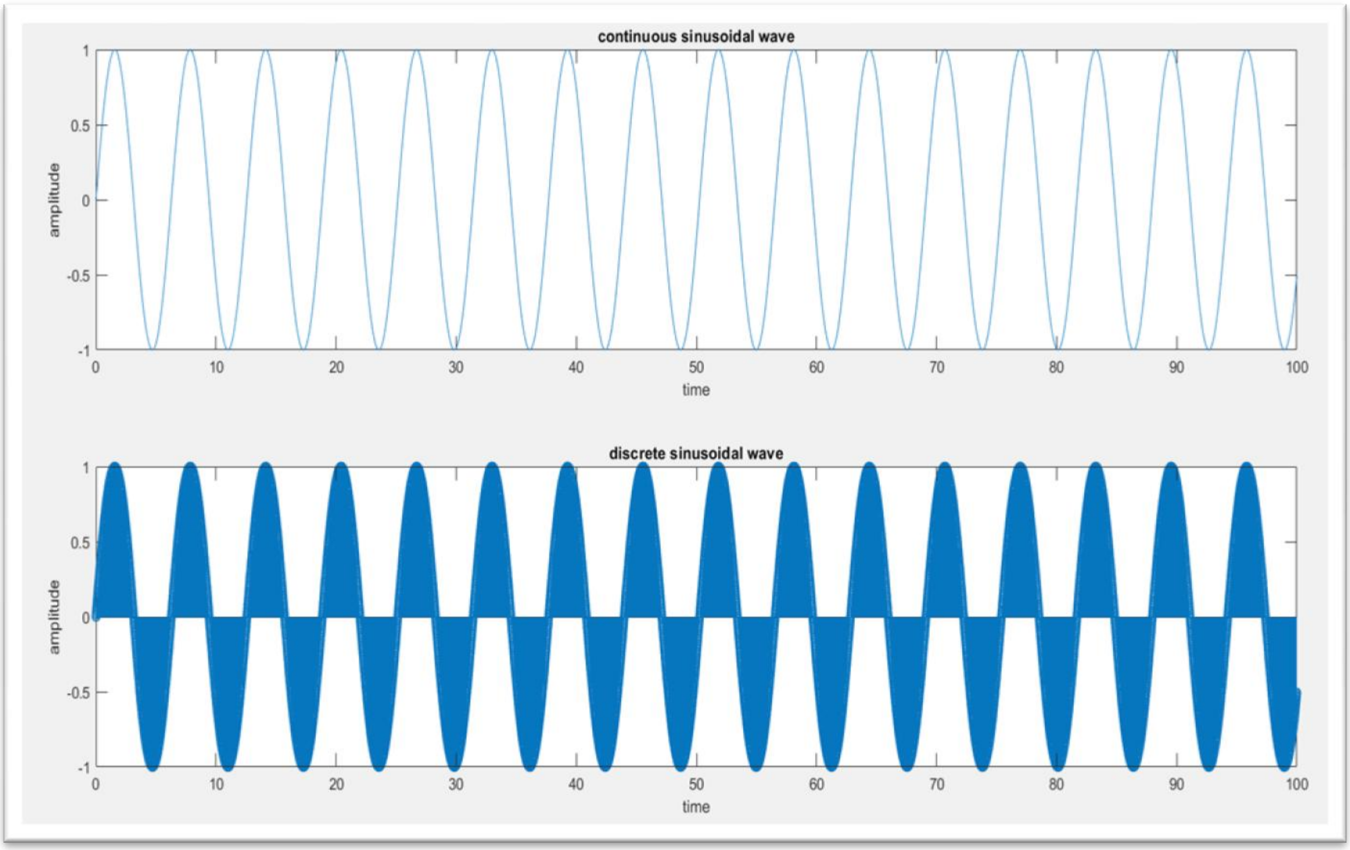
2) Triangular wave:

```
t=0:0.01:100;
y=sawtooth(t);
subplot(2,1,1);
plot(t,y);
xlabel('time');
ylabel('amplitude');
title('continuous triangular wave');
subplot(2,1,2);
stem(t,y);
xlabel('time');
ylabel('amplitude');
title('discrete triangular wave');
```



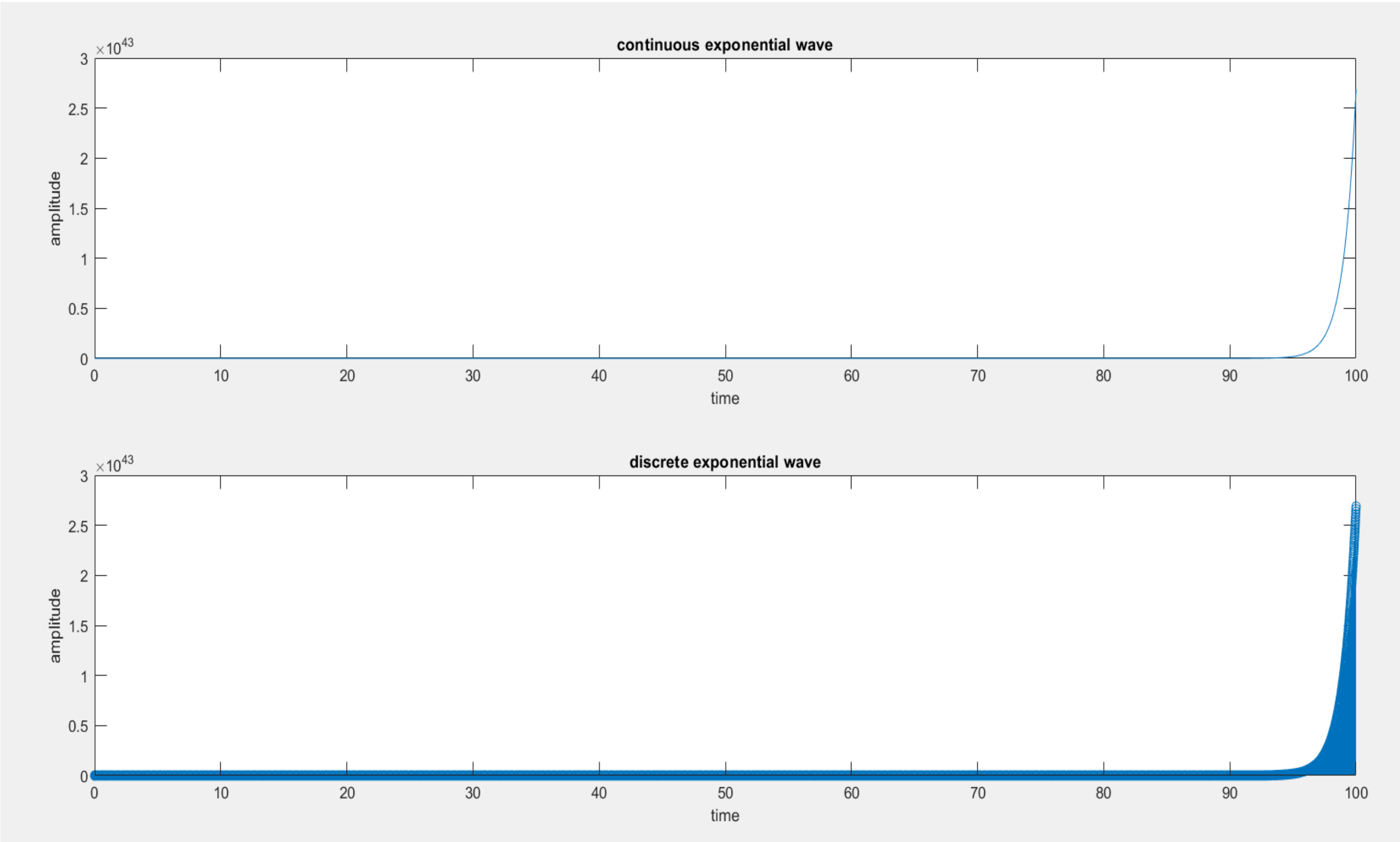
3) Sinusoidal wave

```
t=0:0.01:100;
y=sin(t);
subplot(2,1,1);
plot(t,y);
xlabel('time');
ylabel('amplitude');
title('continuous sinusoidal wave');
subplot(2,1,2);
stem(t,y);
xlabel('time');
ylabel('amplitude');
title('discrete sinusoidal wave');
```



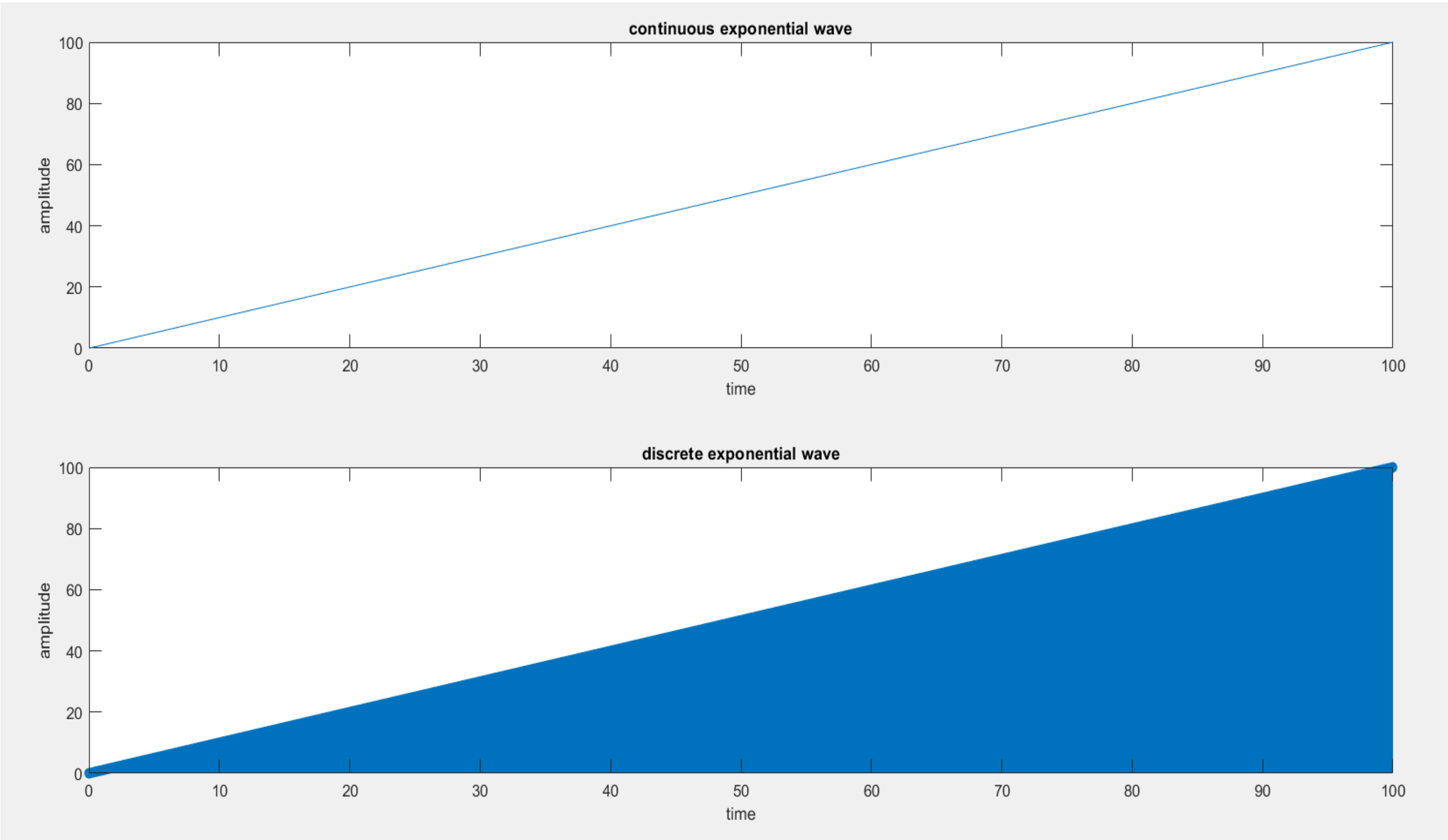
4) <u>Exponential wave:</u>

```
t=0:0.01:100;
y=exp(t);
subplot(2,1,1);
plot(t,y);
xlabel('time');
ylabel('amplitude');
title('continuous exponential wave');
subplot(2,1,2);
stem(t,y);
xlabel('time');
ylabel('amplitude');
title('discrete exponential wave');
```



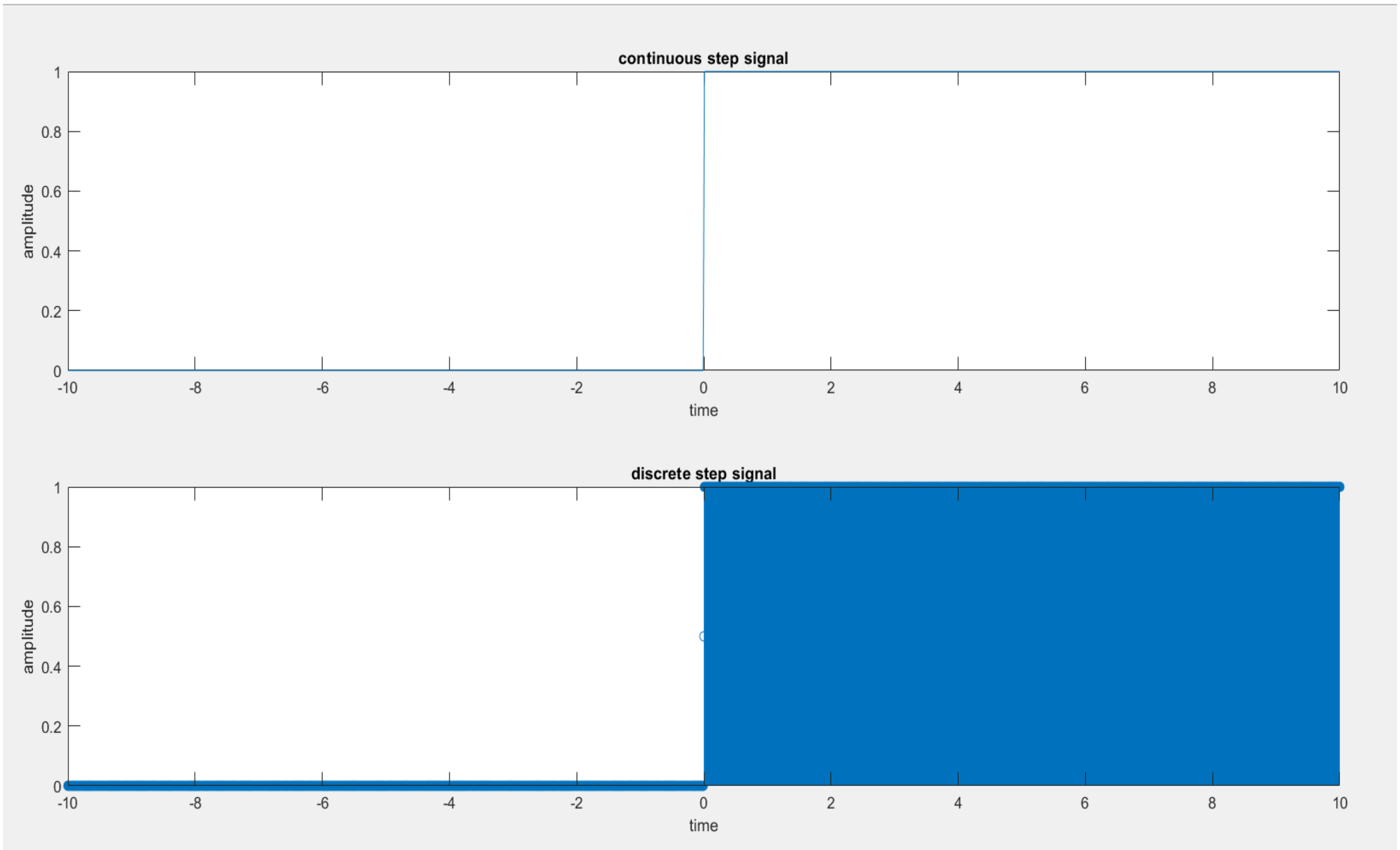
5) <u>Ramp function:</u>

```
t=0:0.01:100;
r=t;
subplot(2,1,1);
plot(t,r);
xlabel('time');
ylabel('amplitude');
title('continuous exponential wave');
subplot(2,1,2);
stem(t,r);
xlabel('time');
ylabel('amplitude');
title('discrete exponential wave');
```



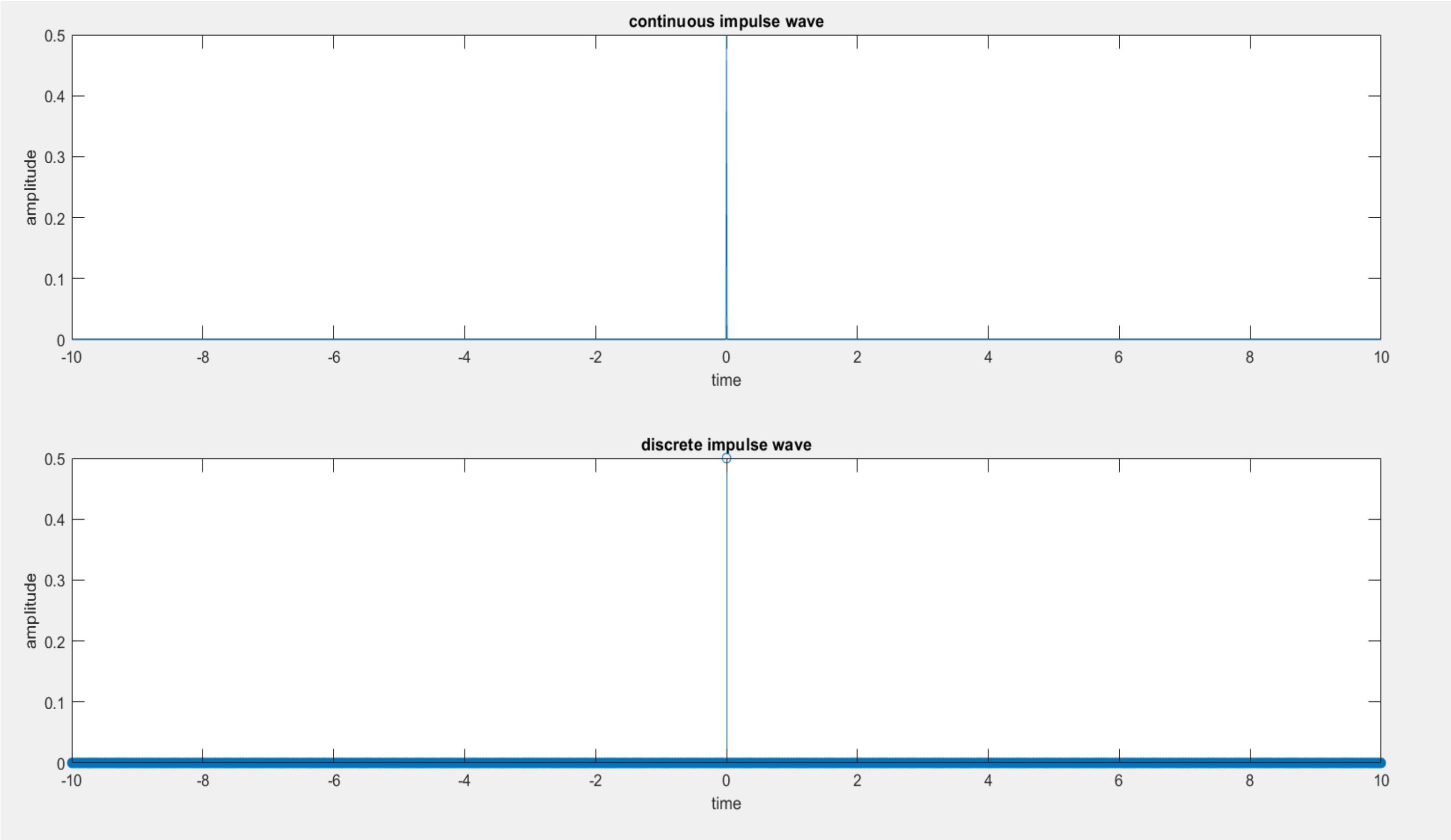
6) <u>Unit step function:</u>

```
t=-10:0.01:10
u=0.5*(1+sign(t))
subplot(2,1,1)
plot(t,u)
xlabel('time')
ylabel('amplitude')
title('continuous step signal')
subplot(2,1,2)
stem(t,u)
xlabel('time')
ylabel('amplitude')
title('discrete step signal')
```



7) <u>Impulse function:</u>

```
t=-10:0.01:10
u=0.5*(1+sign(t))
u1=0.5*(1+sign(t-0.0001))
x=u-u1
subplot(2,1,1)
plot(t,x)
xlabel('time')
ylabel('amplitude')
title('continuous impulse wave')
subplot(2,1,2)
stem(t,x)
xlabel('time')
ylabel('amplitude')
title('discrete impulse wave')
```



Conclusion:

Continuous and discrete time signals for different functions(square ,triangular ,sinusoidal, exponential , step ,ramp, impulse function) are generated.

EXPERIMENT-2

Aim of experiment :-

Verification of time shifting, time scaling, reflection operations on square wave, triangular, exponential, sinusoidal, step, impulse, and ramp function.

Software used: MATLAB R2023a

Theory:

Time shifting, time scaling, reflection operations are basic set of operations in signals.
Time shifting :-`Consider a signal x(t) . The time shifting of x(t) may delay or advance the signal m time . Mathematically this can be represented as y(t) = x(t-T) . If T is +VE , the shifting delays the signal and if T is -VE , the shifting advances the signal x(t). Similarly time shifting of a discrete time signal is represented by y(n)=x(n-k).

Time scaling:- Consider a signal x(t) . The time scaling of x(t) can be accomplished by replacing 't' by 'at' in signal x(t). [y(t) = x (at)]

If a=2, x(t) is compressed in time by factor 2

If a=1/2, x(t) is expanded in time by factor 2

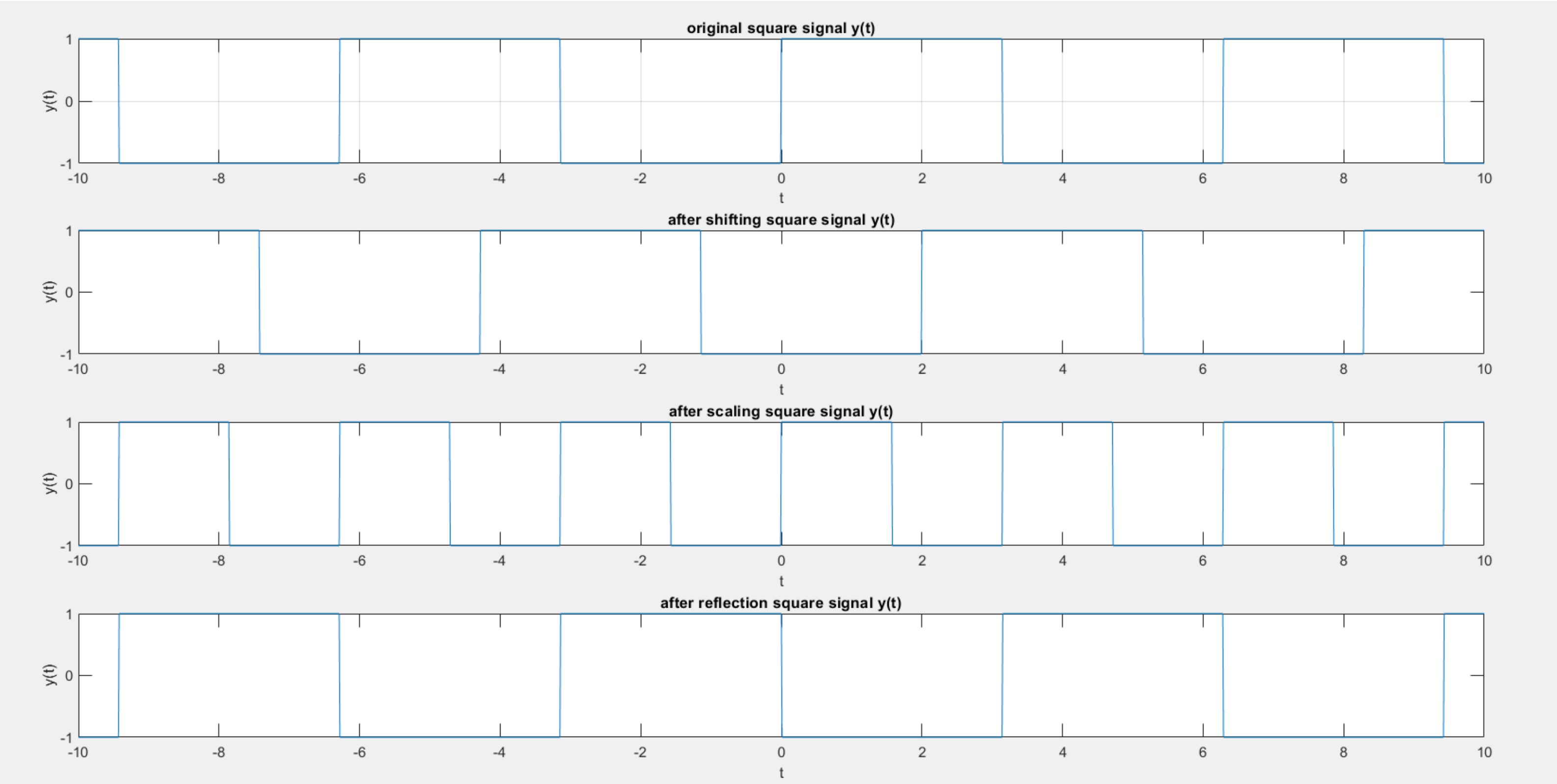
Reflection:- The time reversal of signal x(t) can be obtained by folding the signal about t=0 . It is denoted by x(-t). The time reversal of a discrete signal x(n) can be obtained by folding the sequence x(n) about n=0.

CODES:-

1)Square

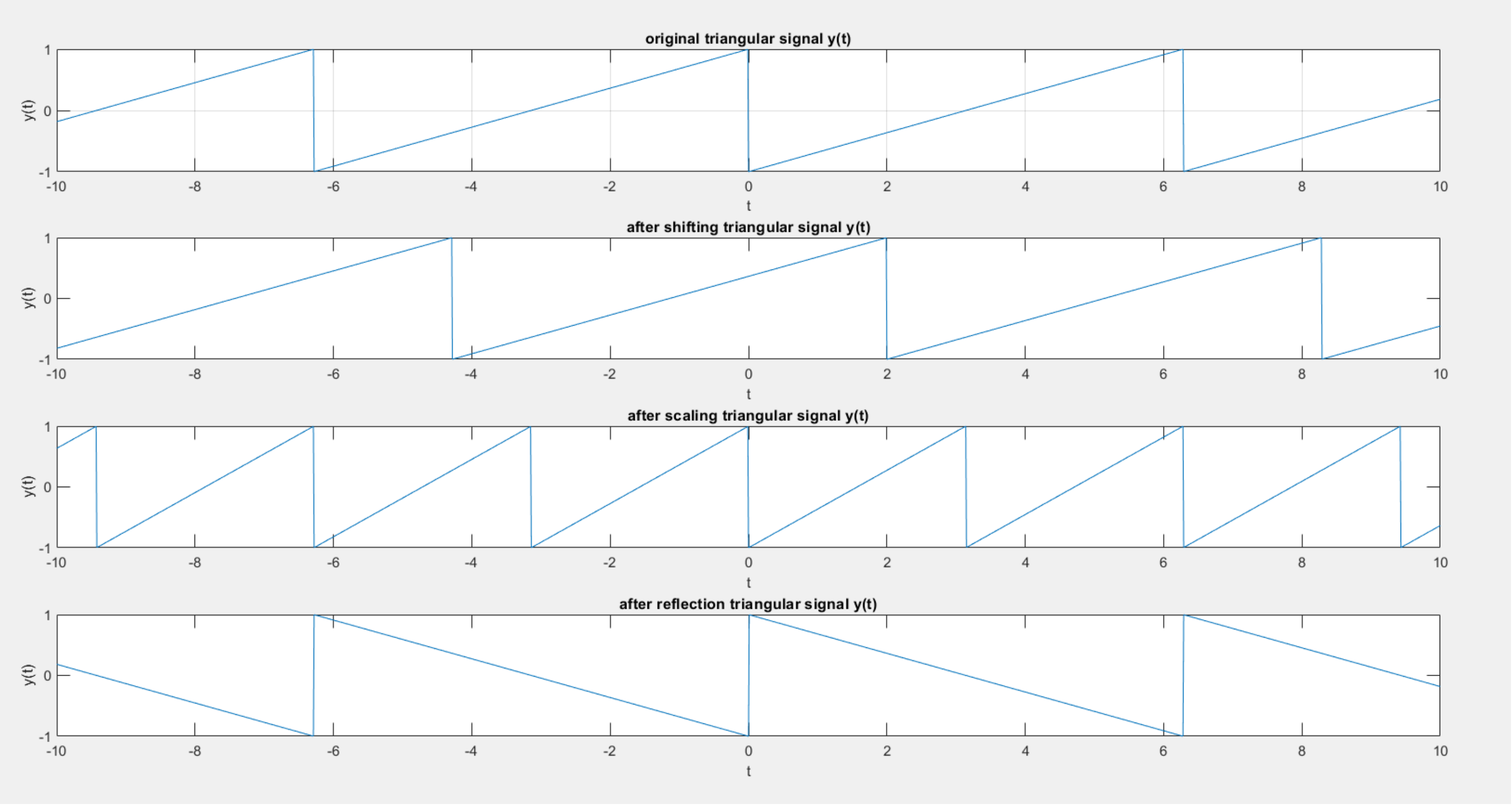
```
t=-10:0.01:10;
y=square(t);
y1=square(t-2);
y2=square(2*t);
y3=square(-t);
subplot(4,1,1);
plot(t,y);
grid on
xlabel('t');
ylabel('y(t)');
title('original square signal y(t)' );
subplot(4,1,2);
plot(t,y1);
xlabel('t');
ylabel('y(t)');
title('after shifting square signal y(t)' );
subplot(4,1,3);
```

```
plot(t,y2);
xlabel('t');
ylabel('y(t)');
title('after scaling square signal y(t)' );
subplot(4,1,4);
plot(t,y3);
xlabel('t');
ylabel('y(t)');
title('after reflection square signal y(t)' );
```



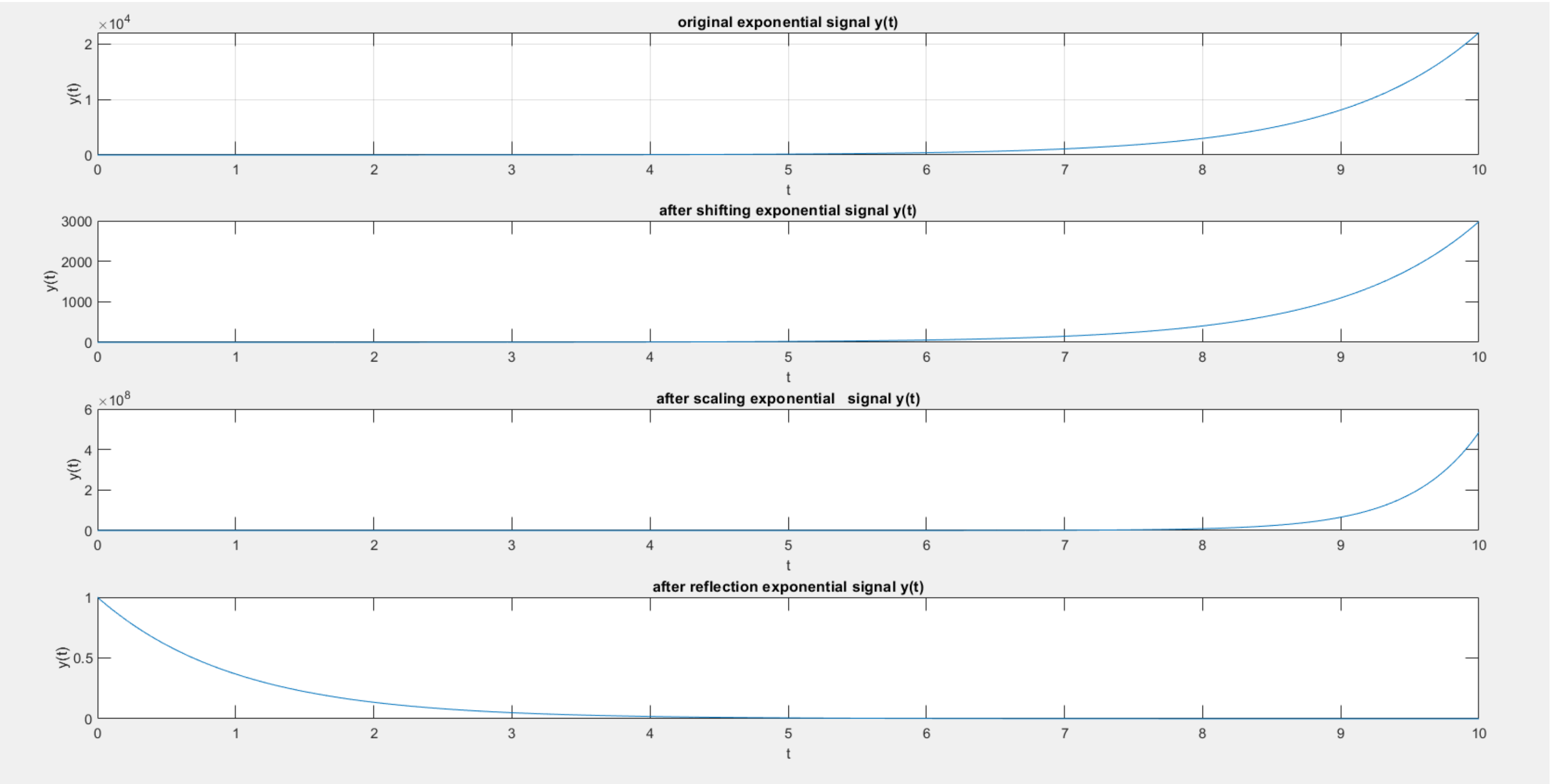
2)Triangular

```
t=-10:0.01:10;
y=sawtooth(t);
y1=sawtooth(t-2);
y2=sawtooth(2*t);
y3=sawtooth(-t);
subplot(4,1,1);
plot(t,y);
grid on
xlabel('t');
ylabel('y(t)');
title('original triangular signal y(t)' );
subplot(4,1,2);
plot(t,y1);
xlabel('t');
ylabel('y(t)');
title('after shifting triangular signal y(t)' );
subplot(4,1,3);
plot(t,y2);
xlabel('t');
ylabel('y(t)');
title('after scaling triangular signal y(t)' );
subplot(4,1,4);
plot(t,y3);
xlabel('t');
ylabel('y(t)');
title('after reflection triangular signal y(t)' );
```

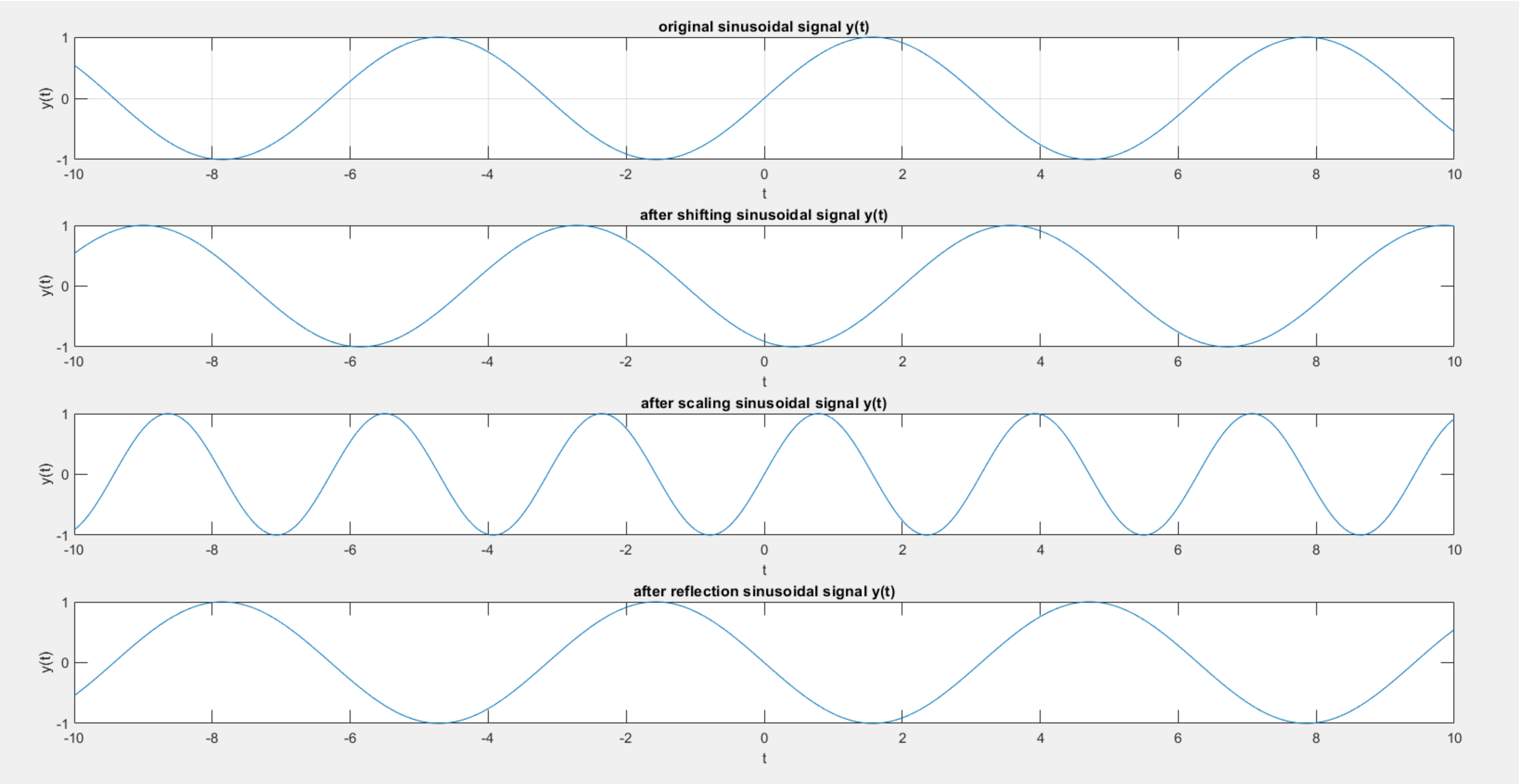


3)Exponential

```
t=0:0.01:10;
y=exp(t);
y1=exp(t-2);
y2=exp(2*t);
y3=exp(-t);
subplot(4,1,1);
plot(t,y);
grid on
xlabel('t');
ylabel('y(t)');
title('original exponential signal y(t)' );
subplot(4,1,2);
plot(t,y1);
xlabel('t');
ylabel('y(t)');
title('after shifting exponential signal y(t)' );
subplot(4,1,3);
plot(t,y2);
xlabel('t');
ylabel('y(t)');
title('after scaling exponential   signal y(t)' );
subplot(4,1,4);
plot(t,y3);
xlabel('t');
ylabel('y(t)');
title('after reflection exponential signal y(t)' );
```

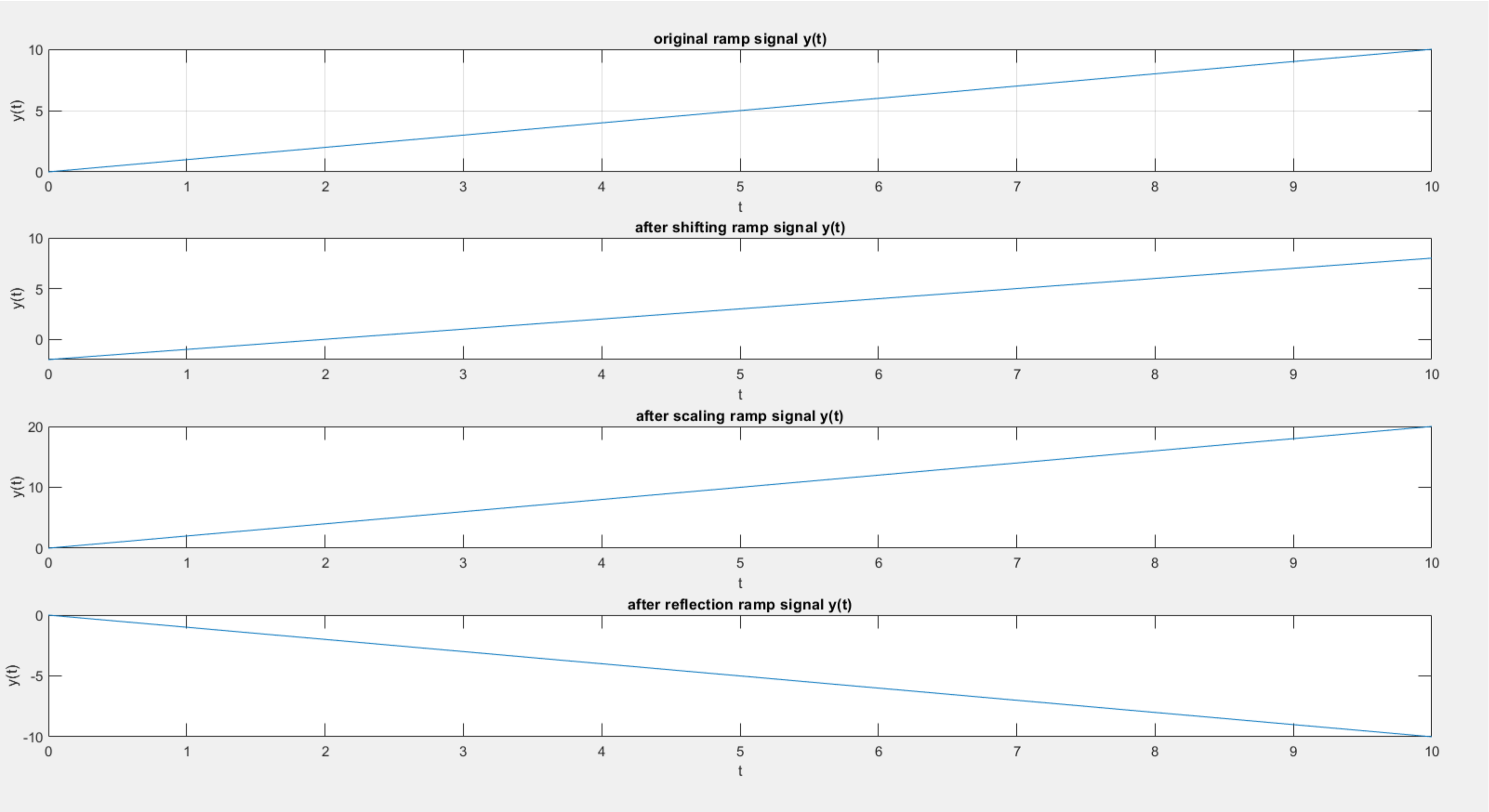


```
4)Sinusoidal
t=-10:0.01:10;
y=sin(t);
y1=sin(t-2);
y2=sin(2*t);
y3=sin(-t);
subplot(4,1,1);
plot(t,y);
grid on
xlabel('t');
ylabel('y(t)');
title('original sinusoidal signal y(t)' );
subplot(4,1,2);
plot(t,y1);
xlabel('t');
ylabel('y(t)');
title('after shifting sinusoidal signal y(t)' );
subplot(4,1,3);
plot(t,y2);
xlabel('t');
ylabel('y(t)');
title('after scaling sinusoidal signal y(t)' );
subplot(4,1,4);
plot(t,y3);
xlabel('t');
ylabel('y(t)');
title('after reflection sinusoidal signal y(t)' );
```

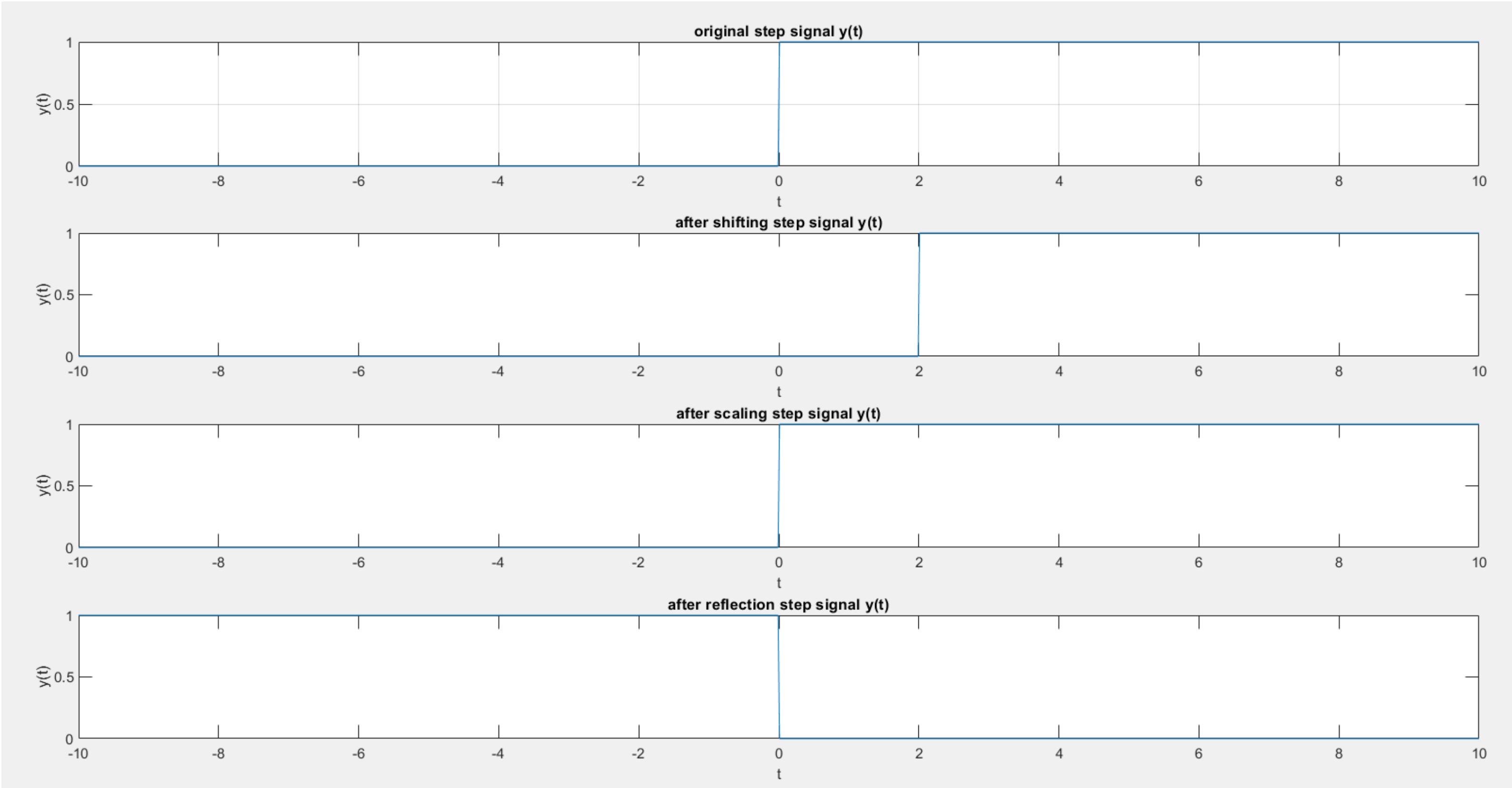


5)Ramp

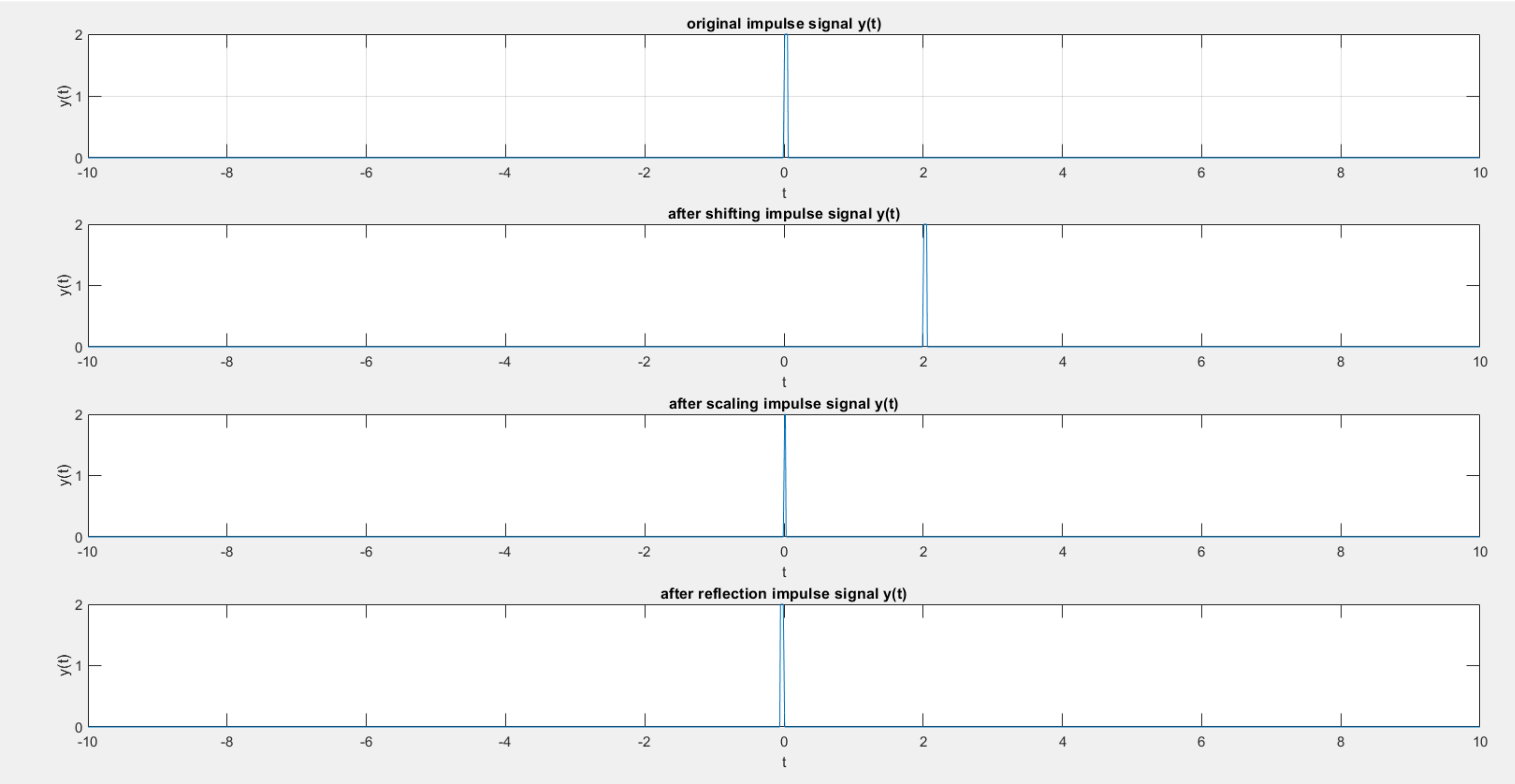
```
t=0:0.1:10;
y=t;
y1=t-2;
y2=2*t;
y3=-t;
subplot(4,1,1);
plot(t,y);
grid on
xlabel('t');
ylabel('y(t)');
title('original ramp signal y(t)' );
subplot(4,1,2);
plot(t,y1);
xlabel('t');
ylabel('y(t)');
title('after shifting ramp signal y(t)' );
subplot(4,1,3);
plot(t,y2);
xlabel('t');
ylabel('y(t)');
title('after scaling ramp signal y(t)' );
subplot(4,1,4);
plot(t,y3);
xlabel('t');
ylabel('y(t)');
title('after reflection ramp signal y(t)' );
```



```
6)Unit step
t=-10:0.01:10;
y=0.5*(1+sign(t));
y1=0.5*(1+sign(t-2));
y2=0.5*(1+sign(2*t));
y3=0.5*(1+sign(-t));
subplot(4,1,1);
plot(t,y);
grid on
xlabel('t');
ylabel('y(t)');
title('original step signal y(t)' );
subplot(4,1,2);
plot(t,y1);
xlabel('t');
ylabel('y(t)');
title('after shifting step signal y(t)' );
subplot(4,1,3);
plot(t,y2);
xlabel('t');
ylabel('y(t)');
title('after scaling step signal y(t)' );
subplot(4,1,4);
plot(t,y3);
xlabel('t');
ylabel('y(t)');
title('after reflection step signal y(t)' );
```



```
7)Impulse
t=-10:0.01:10;
y=sign(t)-sign(t-0.05);
y1=sign(t-2)-sign(t-2-0.05);
y2=sign(2*t)-sign(2*t-0.05);
y3=sign(-t)-sign(-t-0.05);
subplot(4,1,1);
plot(t,y);
grid on
xlabel('t');
ylabel('y(t)');
title('original impulse signal y(t)' );
subplot(4,1,2);
plot(t,y1);
xlabel('t');
ylabel('y(t)');
title('after shifting impulse signal y(t)' );
subplot(4,1,3);
plot(t,y2);
xlabel('t');
ylabel('y(t)');
title('after scaling impulse signal y(t)' );
subplot(4,1,4);
plot(t,y3);
xlabel('t');
ylabel('y(t)');
title('after reflection impulse signal y(t)' );
```



<u>Conclusion:-</u>Time shifting, time scaling, reflection operations are verified on square , triangular , exponential , sinusoidal , step , impulse and ramp function.

EXPERIMENT-3

Aim of experiment :-

To evaluate the convolution of finite discrete time signals and to verify the commutative, associative, distributive, and identity property.

Software used: MATLAB R2023a

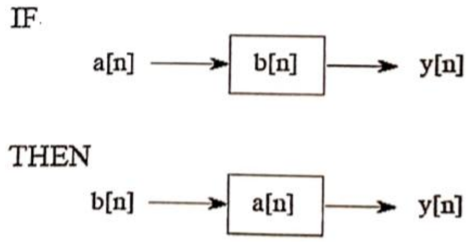
Theory:

Convolution is a mathematical way of combining two signals to form a third signal. It is the single most important technique in signal processing. Convolution is represented by the symbol *, and can be written as **y[n]=x[n] *h[n]**. Convolution sum provides a concise, mathematical way to express the output of an LTI system based on an arbitrary discrete-time input signal and the system's response. The convolution sum is expressed as

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

PROPERTY:

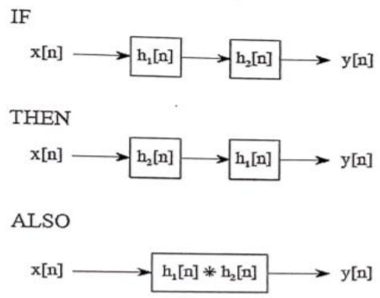
1. **COMMUTATIVE PROPERTY**: The commutative property states that the order in which two sequences are convolved is not important. From a system's point of view, this property states that a system with a unit sample response h(n) and input x(n) behaves in exactly the same way as a system with unit sample response x(n) and an input h(n).. Mathematically, the commutative property is



a(n)*b(n) = b(n)*a(n)

2 **ASSOCIATIVE PROPERTY:** The associative property states that if two systems with unit sample responses h1(n) and h2(n) are connected in cascade, an equivalent system is one that has a unit sample response equal to the convolution of h1(n)and h2(n). Mathematically, the associative property is

{x(n)*h1(n)}*h2(n)=x(n)*{h1(n)*h2(n)}



3 **DISTRIBUTIVE PROPERTY:** Distributive property asserts that if two systems with unit sample responses h1 (n) and h2(n) are connected in parallel, an equivalent system is one that has a unit sample response equal to the sum of h1(n) and h2(n) Mathematically, the distributive property is

X(n)*{h1(n)+h2(n)} = x(n)*h1(n)+x(n)*h2(n)

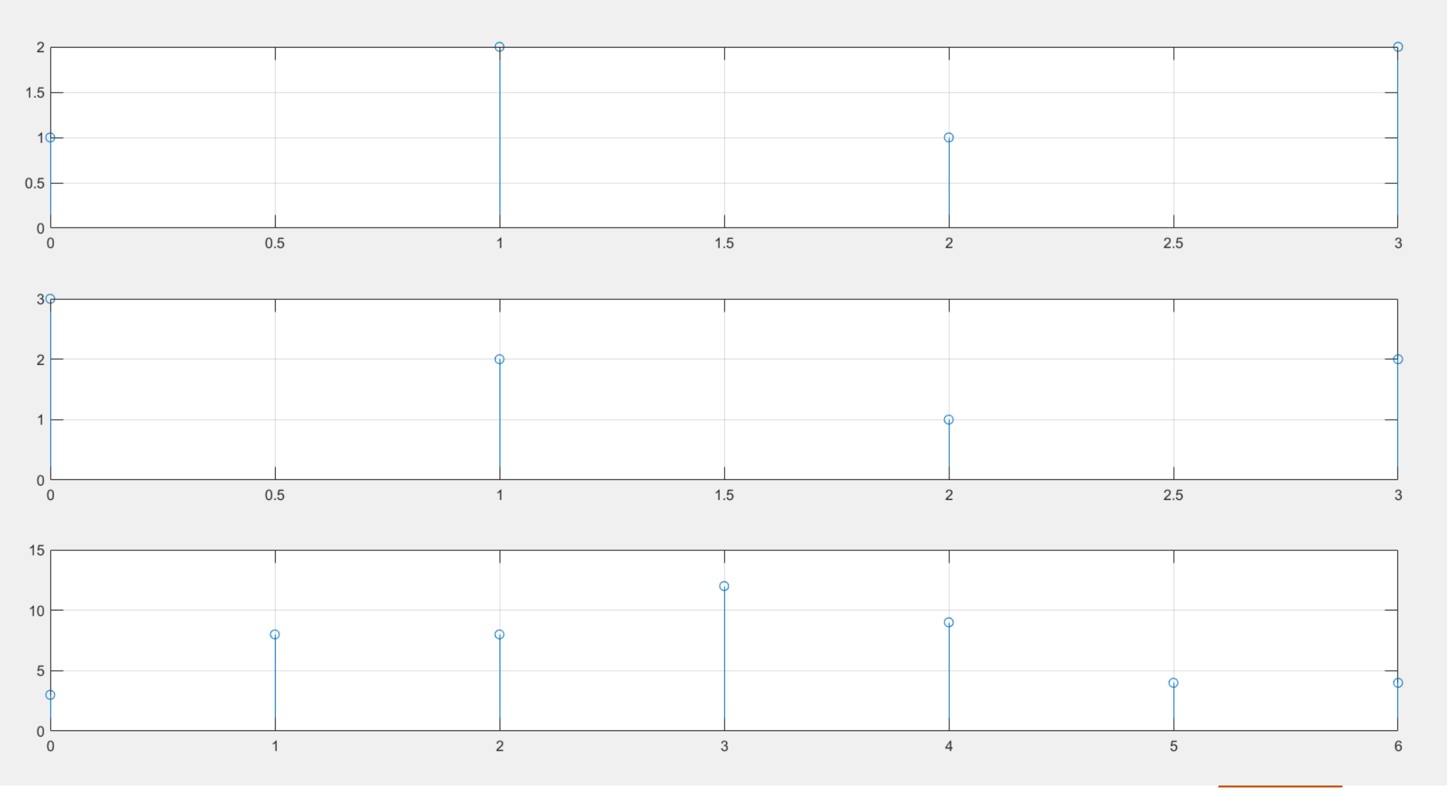
4 **IDENTITY PROPERTY:** The identity property states that if a function convoluted with delta function d(n), it results the function itself. This is the goal of systems at transmit or store signals. Mathematically, the identity property is

x(n)*d(n)=x(n)

CODES:

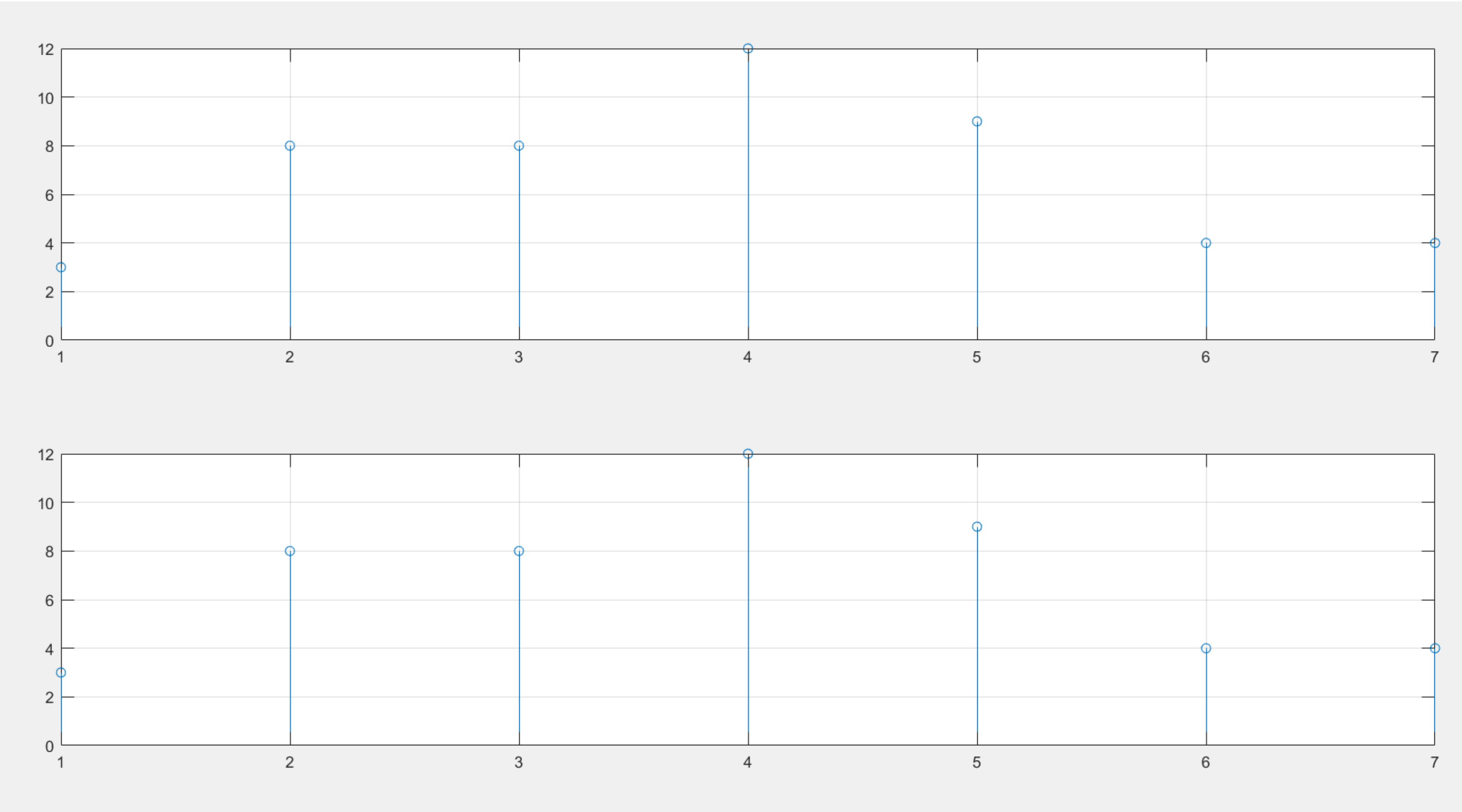
1)Convolution Sum

```
n=0:1:3
x=[1 2 1 2]
h=[3 2 1 2]
y=conv(x,h)
subplot(3,1,1)
stem(n,x)
grid on
hold on
subplot(3,1,2)
stem(n,h)
grid on
hold on
n1=0:1:6
subplot(3,1,3)
stem(n1,y)
grid on
hold on
```



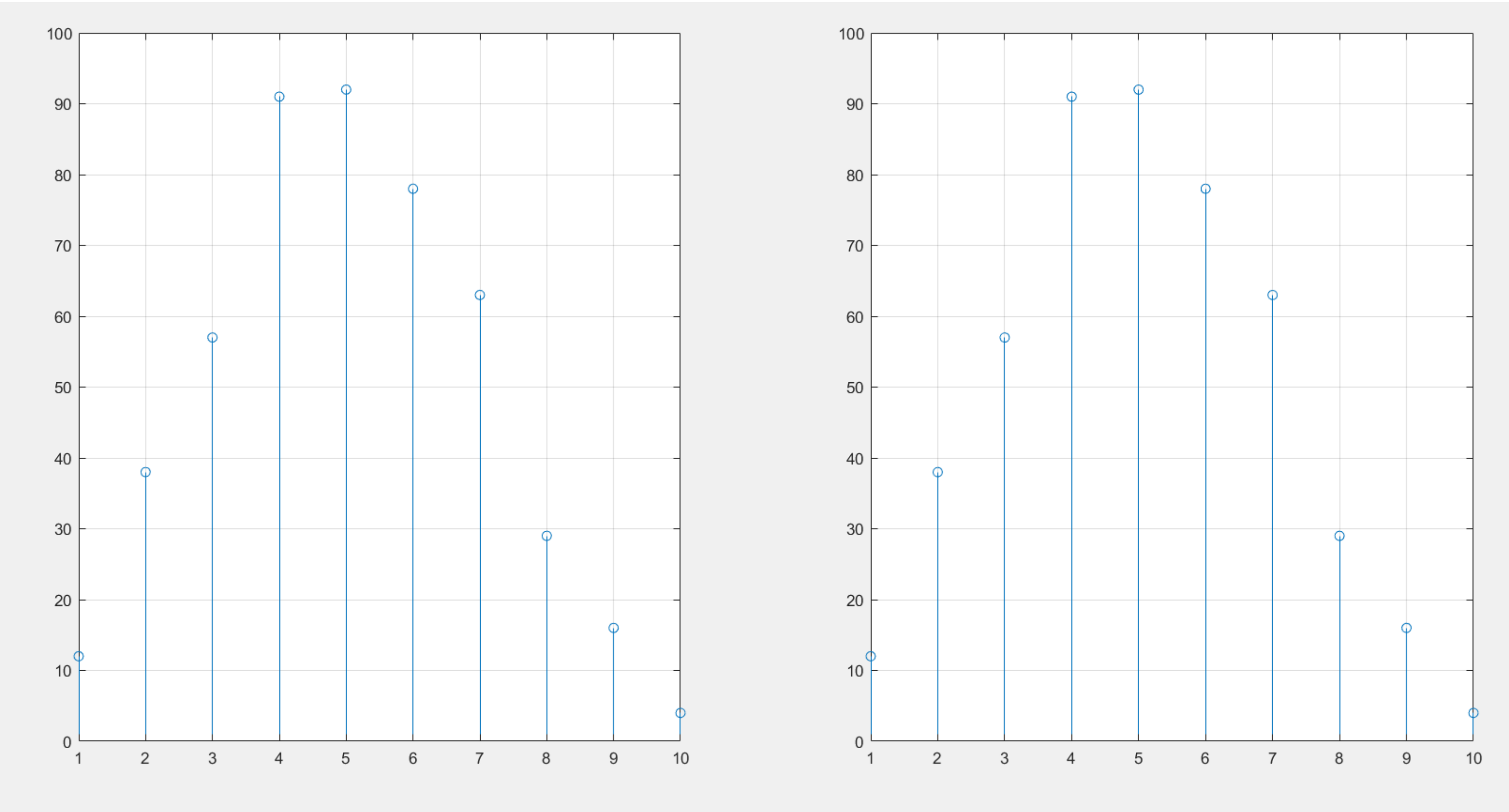
2)Commutative

```
n=0:1:3
x=[1 2 1 2]
h=[3 2 1 2]
y=conv(x,h)
subplot(3,1,1)
stem(n,x)
grid on
hold on
subplot(3,1,2)
stem(n,h)
grid on
hold on
n1=0:1:6
subplot(3,1,3)
stem(n1,y)
grid on
hold on
```



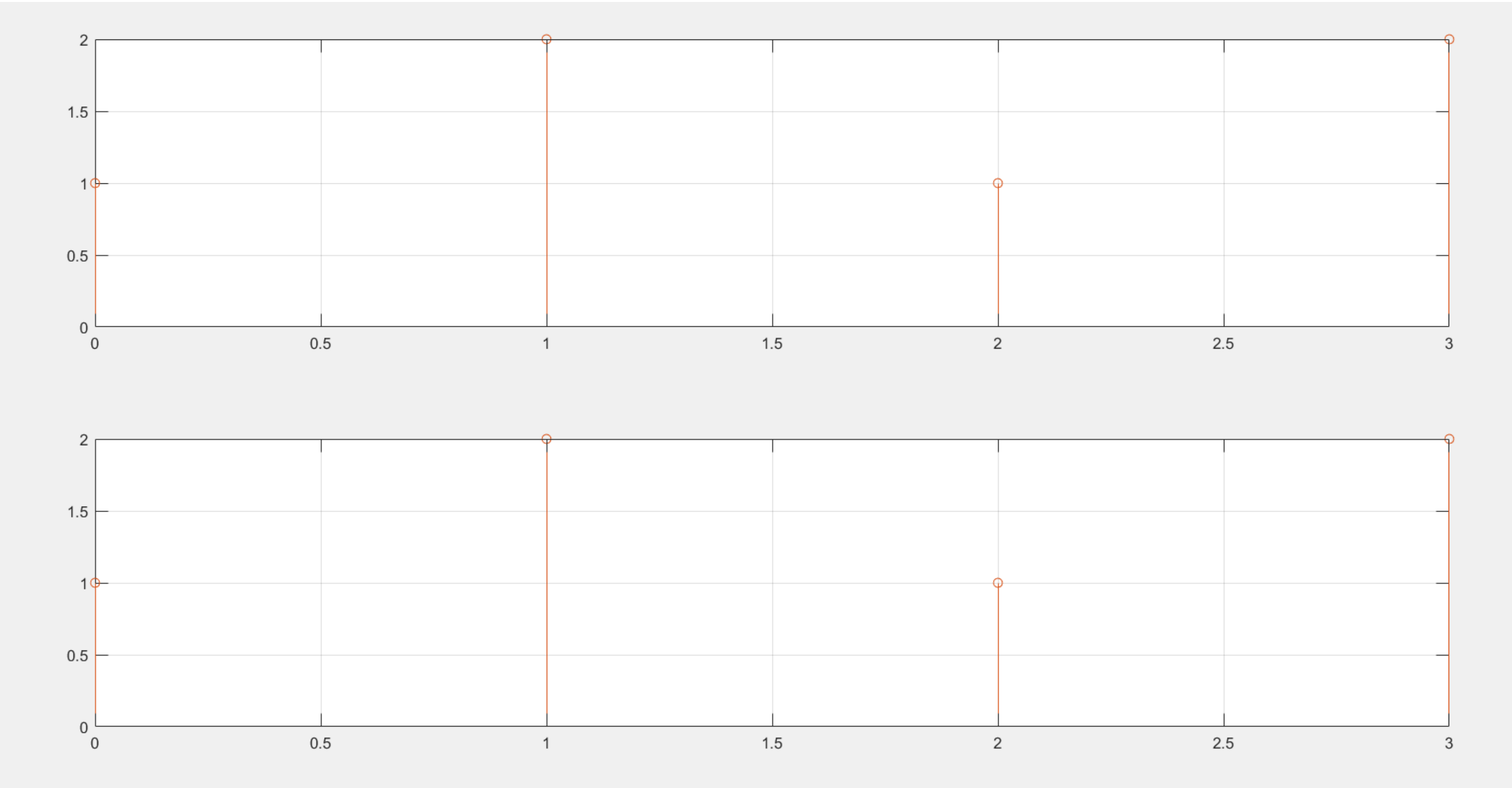
3)Associative

```
x=[1 2 1 2]
h=[3 2 1 2]
y=[4 2 3 1]
a=conv(x,h)
b=conv(a,y)
c=conv(h,y)
d=conv(x,c)
subplot(1,2,1)
stem(b)
grid on
hold on
subplot(1,2,2)
stem(d)
grid on
hold on
```



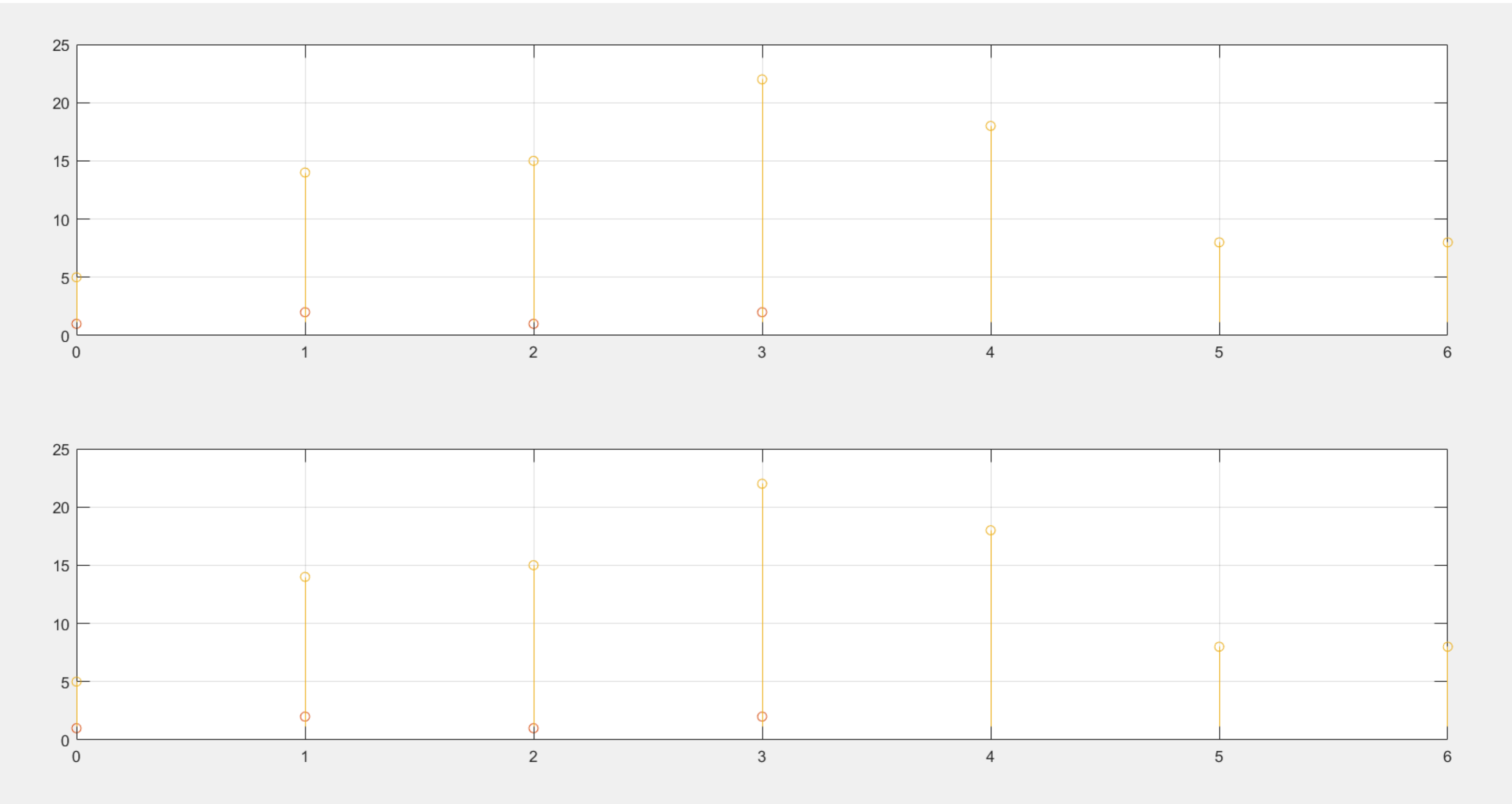
4)Identitive

```
n=0:1:3
x=[1,2,1,2]
h=[1]
y=conv(x,h)
subplot(2,1,1)
stem(n,x)
grid on
hold on
subplot(2,1,2)
stem(n,y)
grid on
hold on
```



5)Distributive

```
n=0:1:3;
x=[1,2,1,2];
h1=[3,2,1,2];
h2=[2,2,1,2]
u1=conv(x,h1)
u2=conv(x,h2)
y1=u1+u2
u3=h1+h2
y=conv(x,u3)
n1=0:1:6
subplot(2,1,1)
stem(n1,y1)
grid on
hold on
subplot(2,1,2)
stem(n1,y)
grid on
hold on
```



CONCLUSION:

The convolution of finite duration discrete time signals was evaluated and properties like commutative, associative, distributive and identity were verified for the signal.

EXPERIMENT-4

Aim of experiment :- To evaluate convolution integral of a given signal.

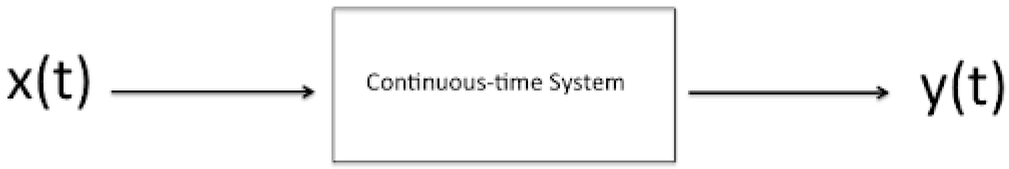
Software used: MATLAB R2023a

Theory:

LTI signals can be characterized by it's impulse response, defined as the output of an LTI system due to a unit impulse signal input applied at time t=0. Given the impulse response, the output due to an arbitrary output signal is determined by expressing the input as a weighted superposition of time shifted impulse. This weighted super position is termed as convolution integral for continuous time signal.

Convolution, one of the most important concepts in electrical engineering, can be used to determine the output a system produces for a given input signal. It can be shown that a linear time invariant system is completely characterized by its impulse response. convolution can be used to determine a linear time invariant system's output from knowledge of the input and the impulse response.

CONVOLUTION OF CONTIUOUS TIME:



Convolution integral of x(t) & h(t):-

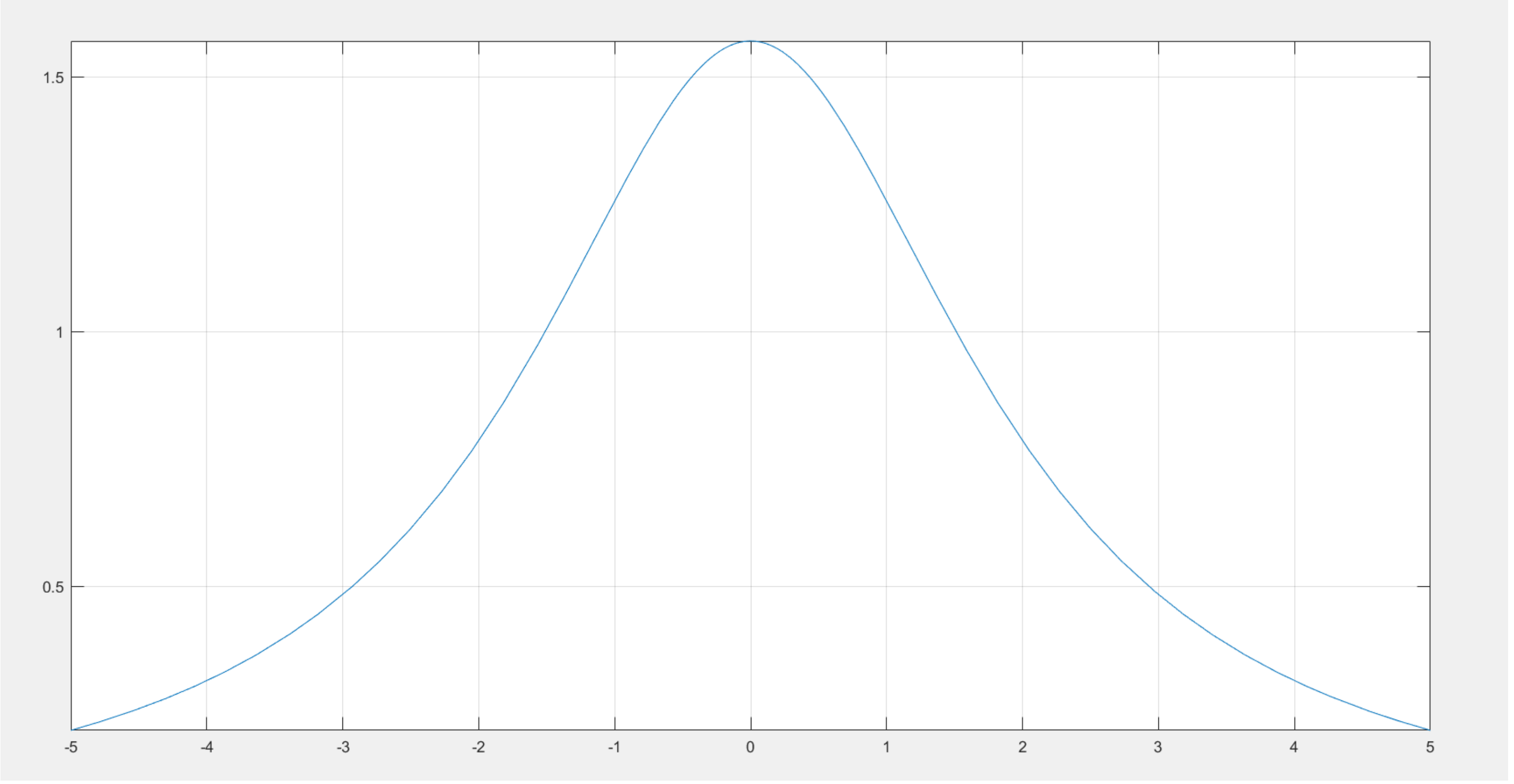
$$y(t)=x(t)*h(t)$$

$$y(t) = \int_{-\infty}^{+\infty} x(\tau)\, h(t-\tau)\, d\tau$$

Codes:-

```
symst tau real,
x=1/(1+(t*t))
f=subs(x,tau)*subs(x,t-tau)
y=int(f,tau,-inf,inf)
y1=simplify(y)
disp('the output of continuous integral is')
```

```
disp(y1)
title('input continuous time signal')
fplot(y1)
grid on
hold on
```



CONCLUSION:

Thus evaluation of convolution of continuous time signals was done using matlab.

EXPERIMENT-5

Aim of experiment :- To evaluate discrete time fourier transform (DTFT) of a signal.

Software used: MATLAB R2023a

Theory:

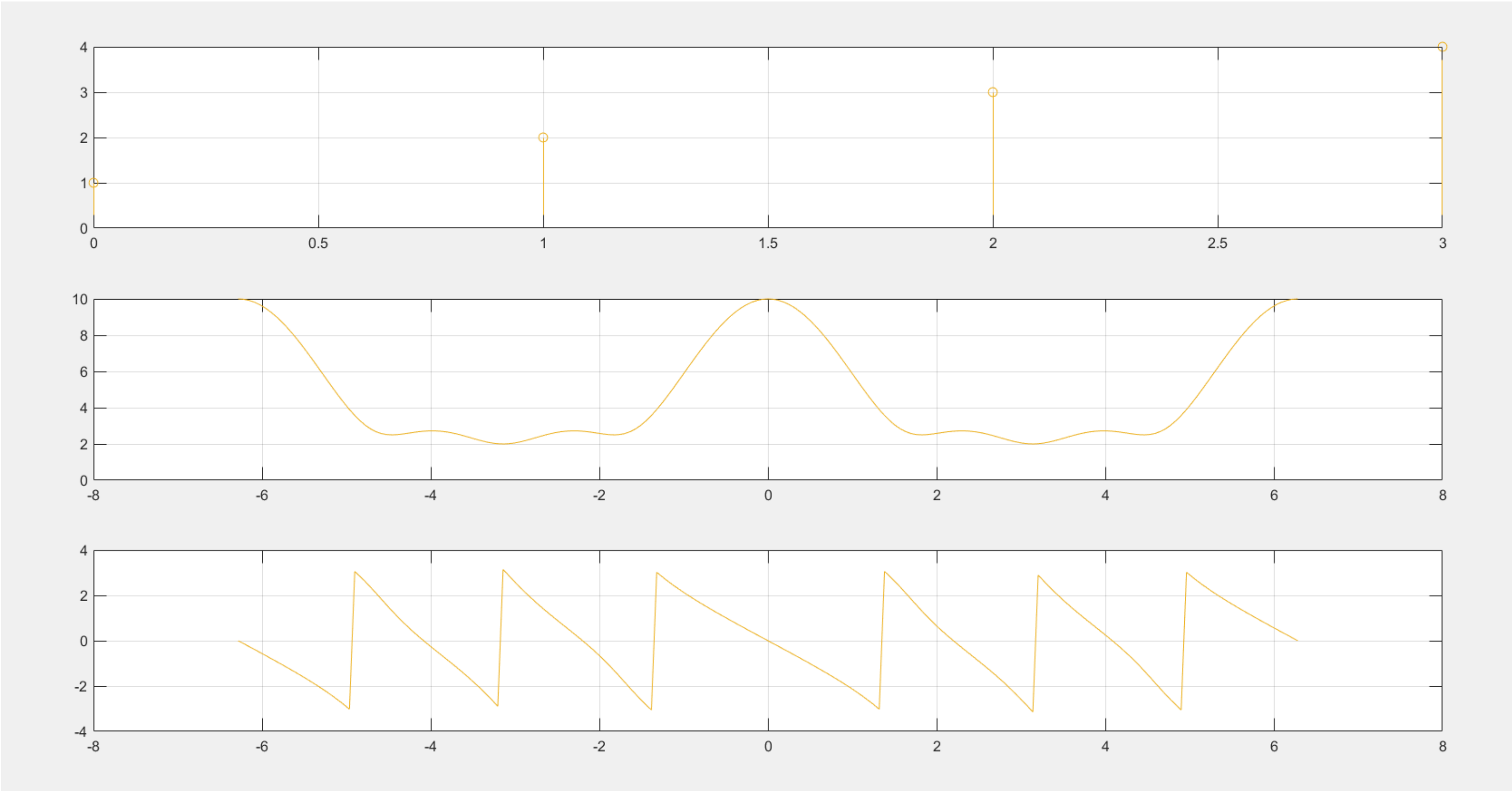
The discrete-time Fourier transform (DTFT) is a form of Fourier analysis that is applicable to the uniformly-spaced samples of a continuous function. The term discrete-time refers to the fact that the transform operates on discrete data (samples) whose interval often has units of time. From only the samples, it produces a function of frequency that is a periodic summation of the continuous Fourier transform of the original continuous function. Under certain theoretical conditions, described by the sampling theorem, the original continuous function can be recovered perfectly from the DTFT and thus from the original discrete samples. The DTFT itself is a continuous function of frequency, but discrete samples of it can be readily calculated via the discrete Fourier transform (DFT), which is by far the most common method of modern Fourier analysis.

The discrete-time Fourier transform of a discrete set of real or complex numbers x[n], for all integer n, is a Fourier series, which produces a periodic function of a frequency variable. When the frequency variable, w, has normalized unit of radians/sample, the periodicity is 2π, and the Fourier series is:

$$X(\omega) = \sum x[n]e^{-i\omega n}$$

CODES:-

```
n=0:1:3
x=[1 2 3 4]
w=-2*pi:2*pi/100:2*pi
b=exp(-j).^(n'*w)
X=x*b
subplot(3,1,1)
stem(n,x)
grid on
hold on
subplot(3,1,2)
plot(w,abs(X))
grid on
hold on
subplot(3,1,3)
plot(w,angle(X))
grid on
hold on
```



Conclusion:-

From the above experiment DTFT of a given sequence were evaluated and graph were plotted using MATLAB.

EXPERIMENT-6

Aim of experiment :- To compute frequency response of LTI system from impulse response.

Software used: MATLAB R2023a

Theory:

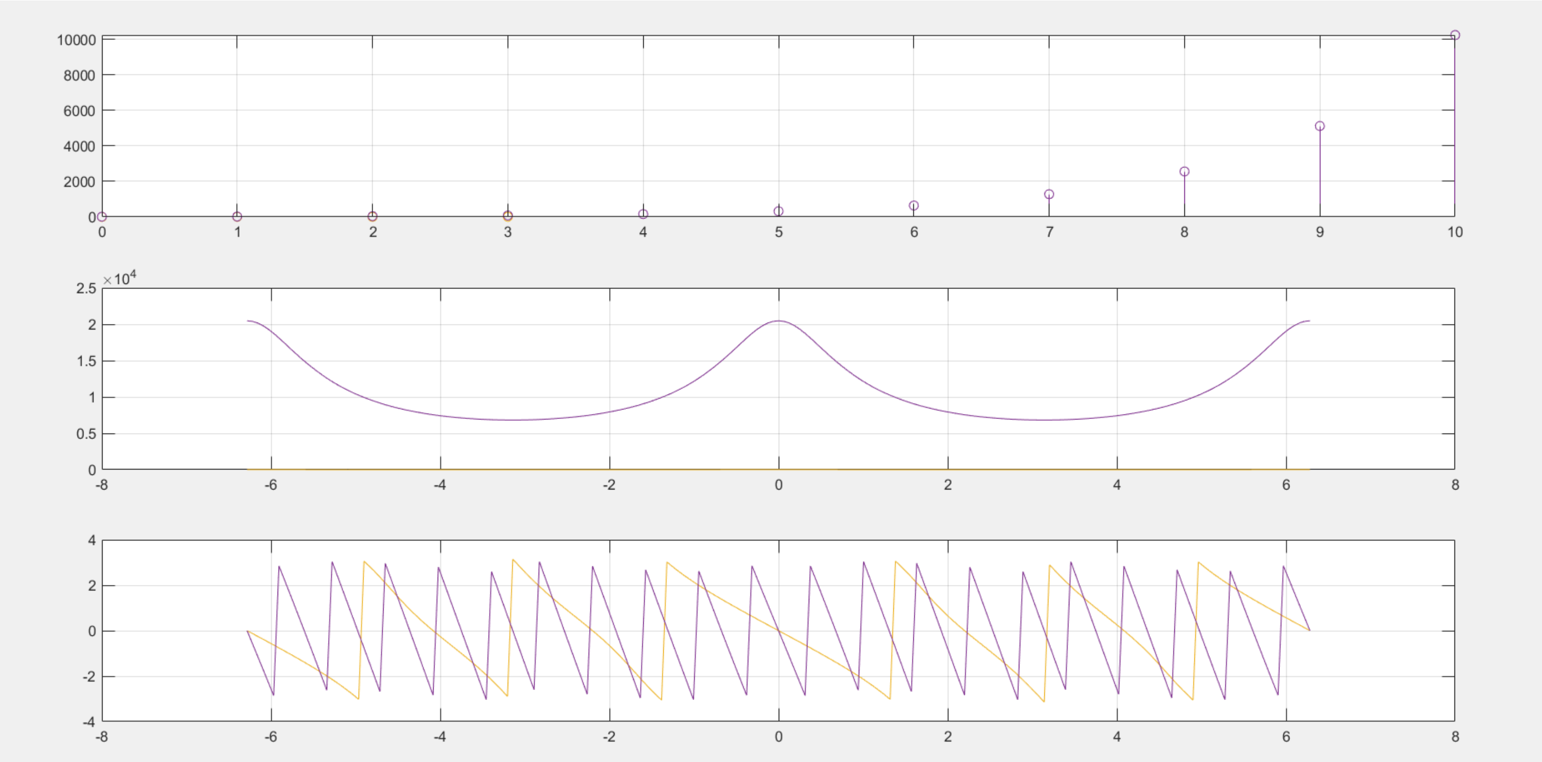
The given equation is the convolution theorem for discrete-time LTI systems. That is, for any signal that is input to an LTI system, the system's output is equal to the discrete convolution of the input signal and the system's impulse response.

$h(n)= \; 10*(2)^n, \; 0<n<10$

$\qquad\qquad 0, \text{ otherwise}$

CODES:-

```
n=0:1:10
h=10*2.^n
w=-2*pi:2*pi/100:2*pi
b=exp(-j).^(n'*w)
H=h*b
subplot(3,1,1)
stem(n,h)
grid on
hold on
subplot(3,1,2)
plot(w,abs(H))
grid on
hold on
subplot(3,1,3)
plot(w,angle(H))
grid on
hold on
```



Conclusion:-

From the above experiment frequency response of LTI system from impulse response was founded and graph was plotted by using MATLAB.

EXPERIMENT-7

Aim of experiment :- To compute frequency response of LTI system by difference equation.

Software used: MATLAB R2023a

Theory:

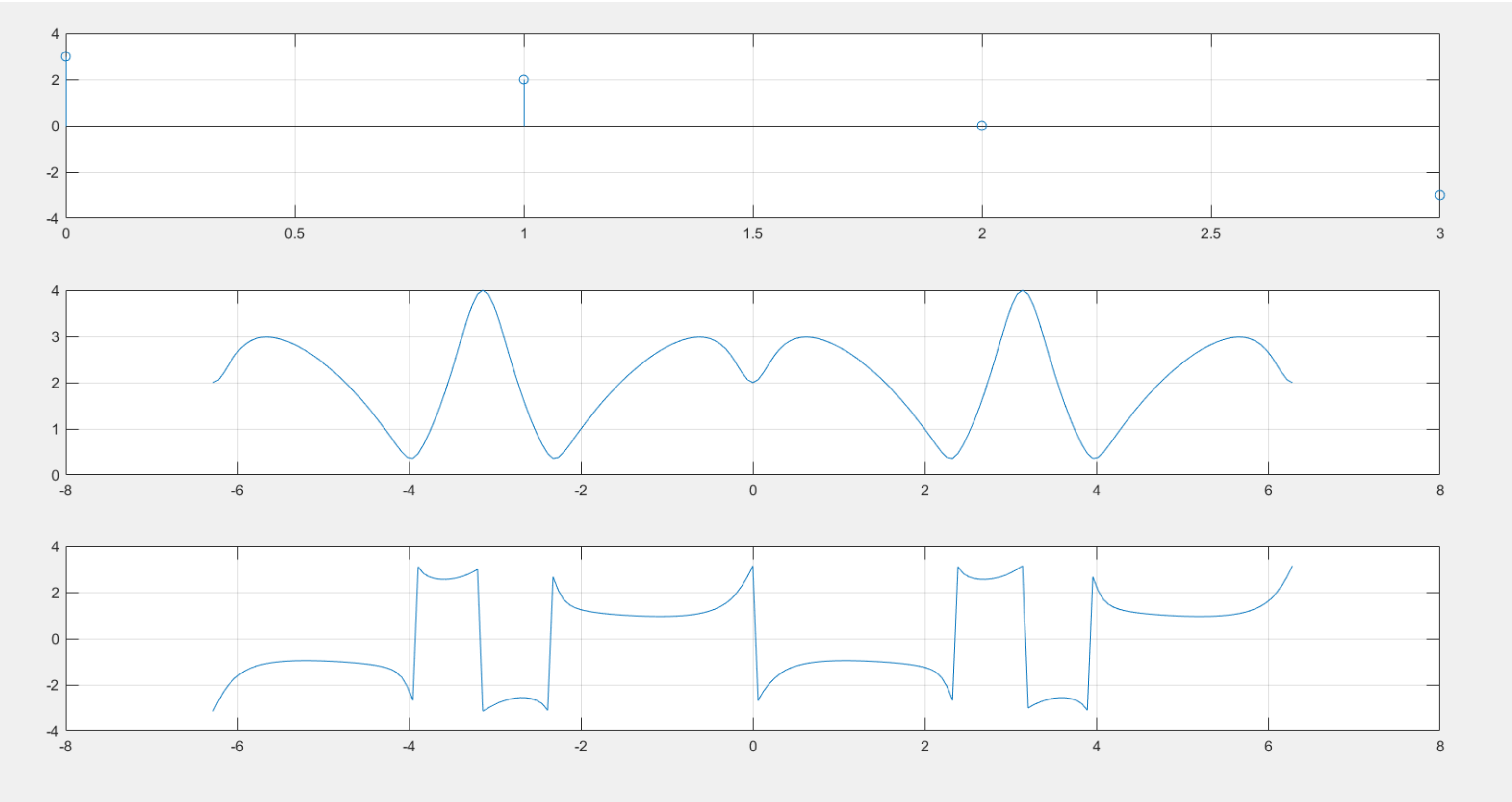
LTI systems are analysed using differential equations for a given input, the output of the system can be obtained by solving the differential equation. The solution of the differential equation consists of 2 parts: (i)zero state response & (ii)zero input response.

$$a_N d^N y(t)/dt^N + a_{N-1} d^{N-1} y(t)/dt^{N-1} + \ldots\ldots\ldots\ldots\ldots\ldots + a_1 dy(t)/dt + a_0 y(t) = b_M d^M x(t)/dt^M + B_{M-1} d^{M-1} x(t)/dt^{M-1} + \ldots\ldots\ldots + b_0 x(t)$$

The response of LTI systems having differential equations can be found out by H(jw)=y(jw)./x(jw)

CODES:-

```
n=0:1:3
y=[3 2 0 -3]
w=-2*pi:2*pi/100:2*pi
b=exp(-j).^(n'*w)
Y=y*b
n1=0:1:2
x=[1 0 -2]
b1=exp(-j).^(n1'*w)
X=x*b1
H=Y./X
subplot(3,1,1)
stem(n,y)
grid on
hold on
subplot(3,1,2)
plot(w,abs(H))
grid on
hold on
subplot(3,1,3)
plot(w,angle(H))
grid on
hold on
```



Conclusion:-

The frequency response of LTI system by differential equation was found successfully using MATLAB.

EXPERIMENT-8

Aim of experiment :- Generation of Amplitude Modulation (AM) wave and analysing as frequency content.

Software used: MATLAB R2023a

Theory: Modulation is defined as the process by which some characteristics of a carrier wave is varied in accordance with message signal. Message signal is known as modulating signal & the result of modulation is called modulated wave.

Amplitude modulation is modulation in which carrier amplitude varied with message signal.

Consider a sinusoidal carrier wave.

$C(t)= A_c \cos [w_c t]$ where w_c is carrier frequency

Consider a message signal m(t).

Modulated signal, $s(t)= A_c [1+k\, m(t)] \cos (w_c t)$

If $|k\, m(t)|= 1$, it is 100% modulation. If $|k\, m(t)|> 1$, it is over modulation. If $|k\, m(t)|< 1$, it is under modulation.

CODES:-

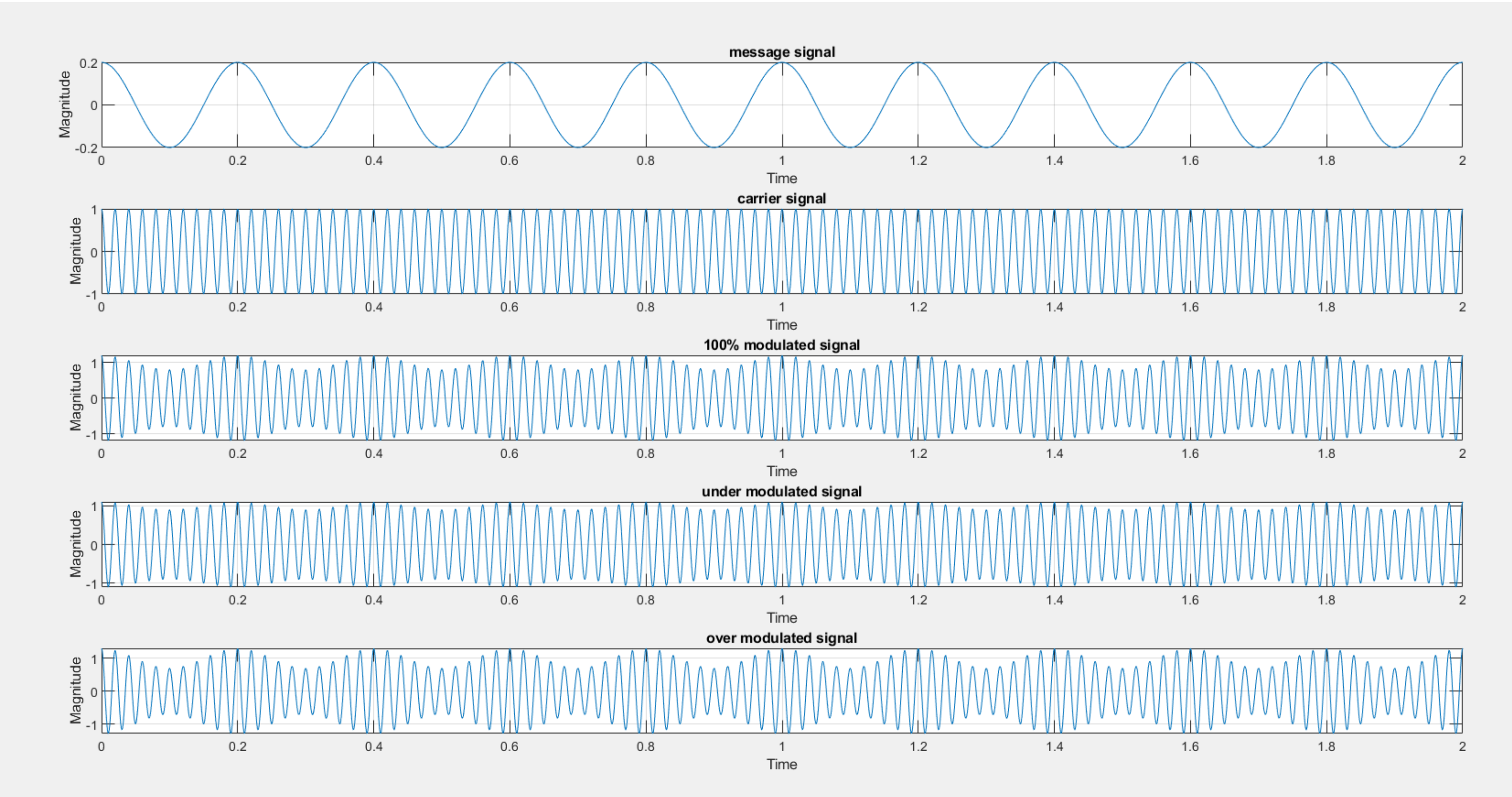
```
t=0:0.001:2
w1=100*pi
w2=10*pi
A1=1
A2=0.2
C=A1*cos(w1*t)
M=A2*cos(w2*t)

K=1
S=A1*(1+K*M).*cos(w1*t)
subplot(5,1,1)
plot(t,M)
xlabel('Time')
ylabel('Magnitude')
title('message signal')
grid on
hold on
subplot(5,1,2)
plot(t,C)
xlabel('Time')
ylabel('Magnitude')
title('carrier signal')
grid on
hold on
subplot(5,1,3)
plot(t,S)
xlabel('Time')
ylabel('Magnitude')
title('100% modulated signal')
grid on
```

```
hold on

K1=0.5
S1=A1*(1+K1*M).*cos(w1*t)
subplot(5,1,4)
plot(t,S1)
xlabel('Time')
ylabel('Magnitude')
title('under modulated signal')
grid on
hold on

K2=1.5
S2=A1*(1+K2*M).*cos(w1*t)
subplot(5,1,5)
plot(t,S2)
xlabel('Time')
ylabel('Magnitude')
title('over modulated signal')
grid on
hold on
```



Conclusion:-

Generated amplitude modulated wave for K=1, 1.5 & 0.5 is 100 is 100% modulation , over modulation and under modulation respectively and we analysed its frequency content.

EXPERIMENT-9

Aim of experiment :- Determination of frequency response from poles and zeroes.

Software used: MATLAB R2023a

Theory:

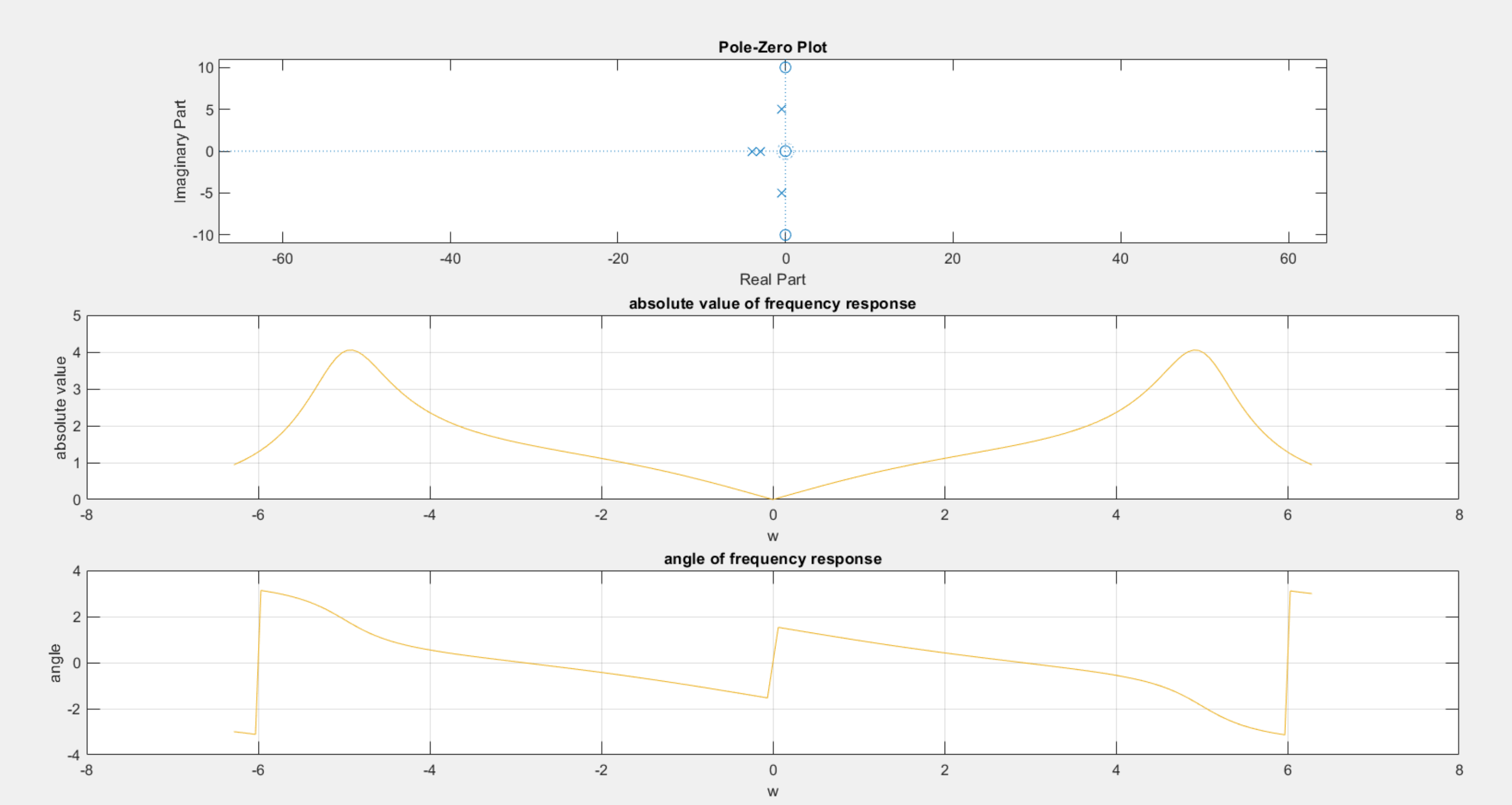
The transfer function of a system is the ratio of 2 polynomials. The roots of numerator polynomial are called zeros of the transfer function & the roots of the denominator polynomial are called the poles of the transfer function.

H(s)= y(s)/x(s) where H(s) is the frequency response of the signal.

The part of the total response which is due to initial condition is known as natural response. On the other hand, the force of response is the response due to input alone.

CODES:-

```
k=2
z=[0 10j -10j]'
p=[-0.5+5j -0.5-5j -3 -4]
[num,den]=zp2tf(z,p,k)
w=-2*pi:(2*pi/100):2*pi
h=freqs(num,den,w)
subplot(3,1,1)
zplane(num,den)
subplot(3,1,2)
plot(w,abs(h))
title('absolute value of frequency response')
xlabel('w')
ylabel('absolute value')
grid on
hold on
subplot(3,1,3)
plot(w,angle(h))
title('angle of frequency response')
xlabel('w')
ylabel('angle')
grid on
hold on
```



CONCLUSION:-

From the above experiment the frequency response comprising of magnitude response and phase response was plotted for thesystem from its given poles and zeros.

EXPERIMENT-10

Aim of experiment :- Pole-zero plot on z-plane and determination of magnitude response.

Software used: MATLAB R2023a

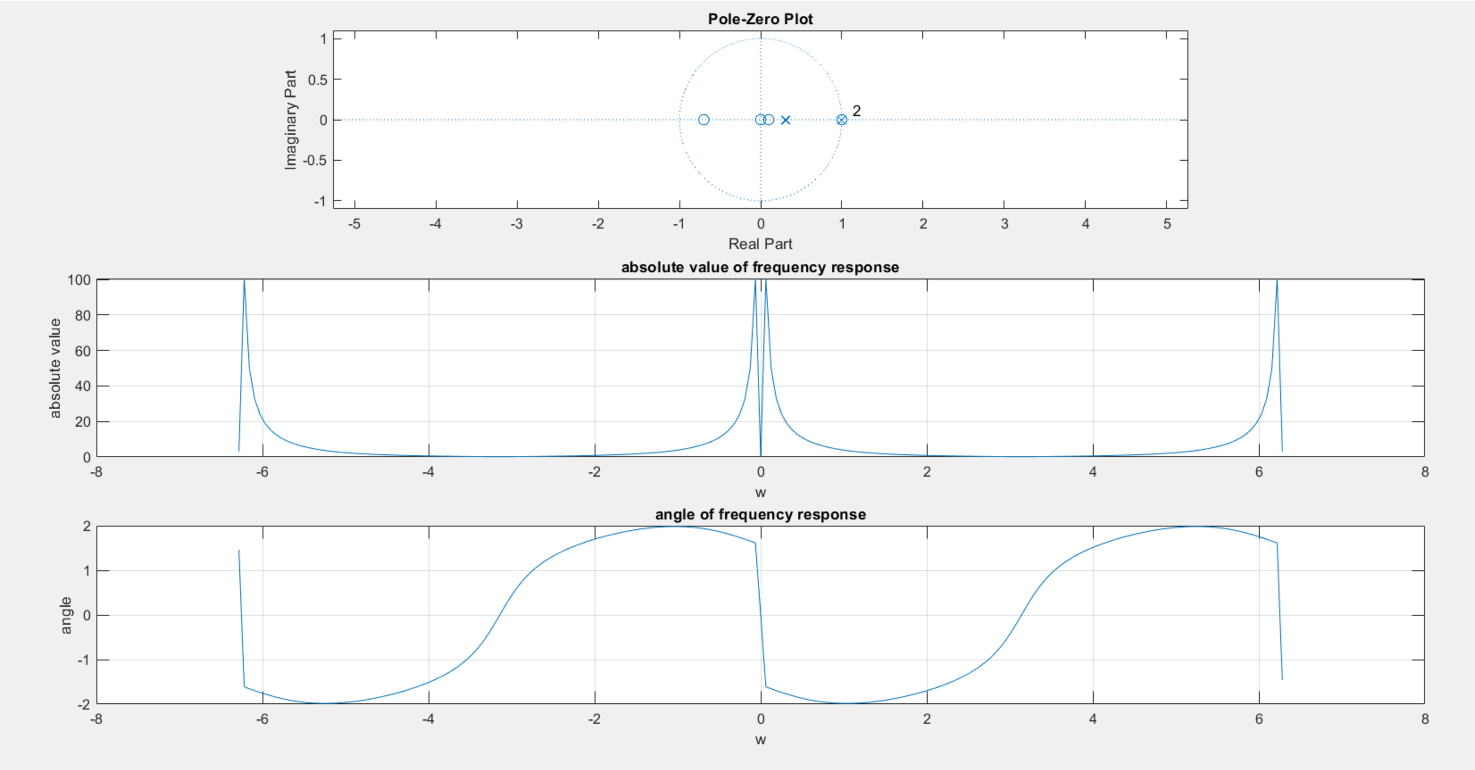
Theory:

The transfer function of a system is the ratio of 2 polynomials. The roots of numerator polynomial are called zeros of transfer function & roots of denominator polynomial are called poles.

The function 'freq z' helps us to find the frequency response of the system.

CODES:-

```
[num,den]=zp2tf(z,p,k)
w=-2*pi:(2*pi/100):2*pi
h=freqz(num,den,w)
subplot(3,1,1)
zplane(num,den)
subplot(3,1,2)
plot(w,abs(h))
title('absolute value of frequency response')
xlabel('w')
ylabel('absolute value')
grid on
hold on
subplot(3,1,3)
plot(w,angle(h))
title('angle of frequency response')
xlabel('w')
ylabel('angle')
grid on
hold on
```



Conclusion:-

From the above experiment the pole-zero plot in Z plane, magnitude response and phase response were plotted for the system.

EXPERIMENT-11

Aim of experiment :- To find the impulse response of a system described by Z transform.

Software used: MATLAB R2023a

Theory:

The Z Transform is used to represent sampled signals and Linear Time Invariant (LTI) systems, such as filters, in a way similar to the Laplace transform representing continuous-time signals.

Signal representation

The Z Transform is used to represent sampled signals in a way similar to the Laplace transform representing continuous-time signals. A sampled signal is given by the sum of its samples, each one delayed by a different multiple of the sampling period Ts. The Laplace transform represents a delay of one sampling period {\displaystyle T_{s}} by:

$z^{-1} \stackrel{\text{def}}{=} e^{sT_s}$

With this, the Z-transform can be represented as

X(z) = Z{x[n]= $\sum x[n]z^{-n}$

where the x[n]{\displaystyle x[n]} are the consecutive values of the sampled signal.

Linear time invariant systems

Continuous-time Linear Time Invariant (LTI) systems can be represented by a transfer function which is a fraction of two polynomials of the complex variable {\displaystyle s}. Their frequency response is estimated by taking {\displaystyle s=j\omega }, this is by estimating the transfer function along the imaginary axis. In order to ensure stability, the poles of the transfer function (the roots of the denominator polynomial) must be on the left half plane of {\displaystyles}s. Discrete-time LTI systems can be represented by the fraction of two polynomials of the complex variable {\displaystyle z}z:

H(s)=num(s)din(s)

From the definition:

$z \stackrel{\text{def}}{=} e^{sT_s}$

we find that their frequency response be estimated by taking {\displaystyle z=e^{j\omega T_{s}}} z $=e^{jwTs}$this is by estimating the transfer function around the unit circle.

In order to ensure stability, the poles of the transfer function (the roots of the denominator polynomial) must be inside the unit circle.

QUESTION :-

$B(z) = 1 - 10z^{-1} + 4z^{-2} + 4z^{-3}$

$A(Z) = 2 – 2z^{-1} – 1z^{-2}$

H(z) = B(z) / A(z)

Find z-transform

CODES:-

```
b=[1 -10 4 4]
a=[2 -2 -1]
[r,p,k]=residue(b,a)
```

OUTCOME:-

```
r = -1.9189
-0.3311
p =1.3660
   -0.3660
k =0.5000
 -4.5000
```

Conclusion:-

From this experiment, the theory of Z transform function of a system and analyzing its frequency content was properly studied and poles and zeros determining the magnitude were successfully obtained using MATLAB.

EXPERIMENT-12

Aim of experiment :- Implementation of interpolation and decimation concept.

Software used: MATLAB R2023a

Theory:

A digital signal processing system that uses signal with different frequencies is probably performing with multirate digital processing. Multirate digital signal processing after uses sample rate conversion to convert from one sampling frequency to another sampling frequency. Sampling rate conversion uses decimation to decrease the sampling rate, interpolation to increase the sampling rate. The sampling rate conversion is done in digital domain and uses combination of decimation and interpolation.

DECIMATION: Decimation removes samples from a signal, therefore it can only down sample the signal by integer factor:

f_s / f_s^{new}=D>1 where, f_s = Old sampling rate

f_s^{new} = New sampling rate

The sampling theorem states that the highest frequency in a signal should be less than half of the sampling frequency.

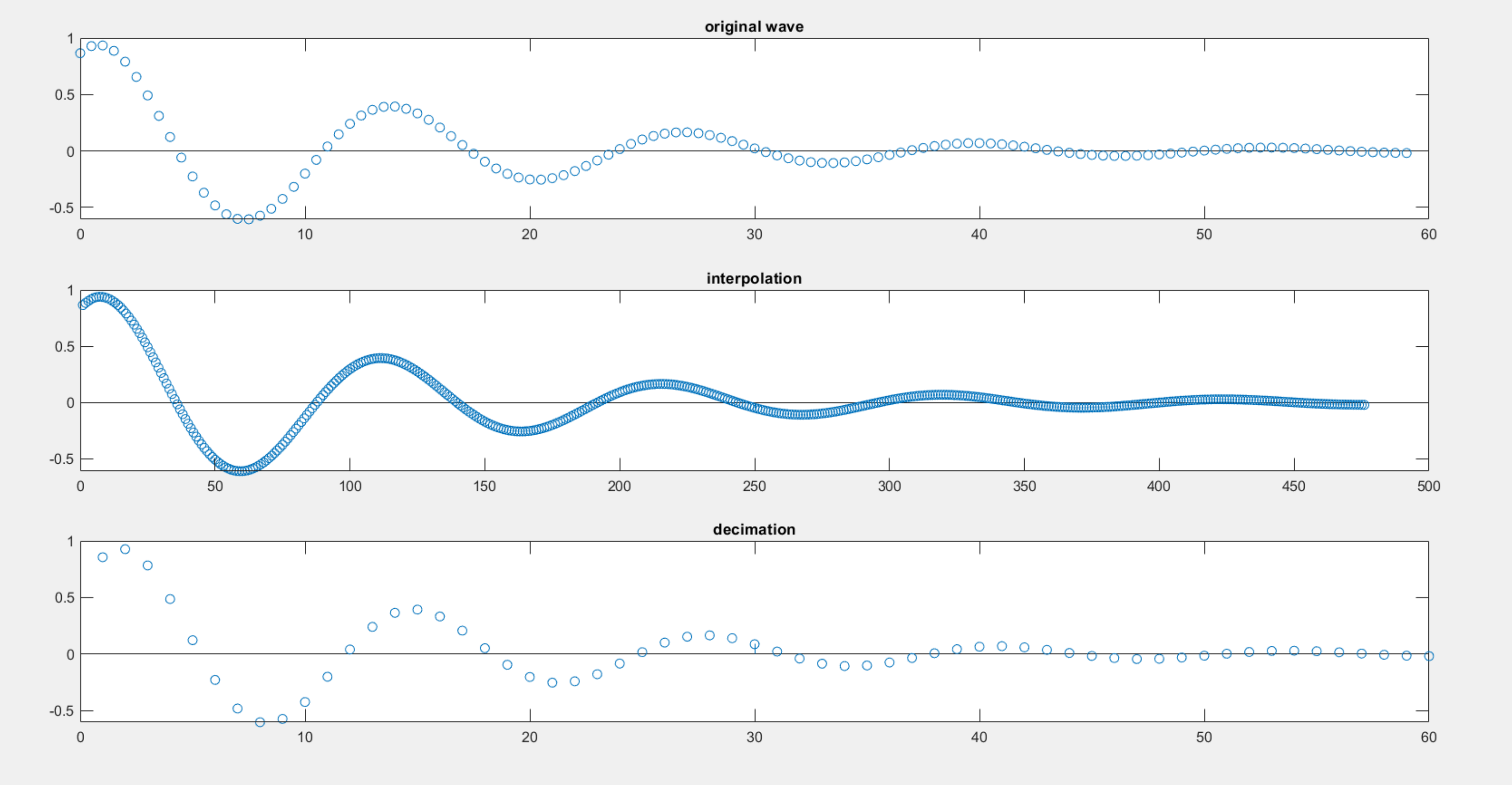
INTERPOLATION: Interpolation increases the sampling frequency by estimating the value of signal between the samples. The new sampling frequency is greater than the old sampling frequency, $f_s^{new}>f_s$. If common interpolation approach is zero filling based on interpolation:

Decimation, f_s / f_s^{new} = D... for down sampling.

Interpolation, f_s^{new} / fs =D... for up sampling.

CODES:-

```
n=0:0.5:59
x=exp(-n/15).*sin((2*pi*n)/13 + (pi/3))
y1=interp(x,4)
y2=decimate(x,2)
subplot(3,1,1)
stem(n,x)
title('original wave')
subplot(3,1,2)
stem(y1)
title('interpolation')
subplot(3,1,3)
stem(y2)
title('decimation')
```



Conclusion:-

From the above experiment, the decimation and interpolation process were implemented to obtain the desired sampling frequency.