Experiment No. 1

Aim: Write a MATLAB Program to generate Discrete Time Signal and plot them.

Tool: Matlab

Theory:

A discrete-time signal is a sequence of values that correspond to particular instants in time. The time instants at which the signal is defined are the signal's sample times, and the associated signal values are the signal's samples.

Impluse function :-The discrete time unit impulse function, also known as the unit sample function, is of great importance to the study of signals and systems. The function takes a value of one at time n=0 and a value of zero elsewhere. It has several important properties that will appear again when studying systems.

Step Function :- In discrete time the unit impulse is the first difference of the unit step, and the unit step is the running sum of the unit impulse. Correspondingly, in continuous time the unit im- pulse is the derivative of the unit step, and the unit step is the running integral of the impulse.

Ramp Function :- The discrete time unit ramp signal is that function which starts from n = 0 and increases linearly. It is denoted by r(n). It is signal whose amplitude varies linearly with time n. mathematically, the discrete time unit ramp sequence is defined as − r(n)= n for n≥0 ; 0 for n<0.

Exponential Function :- Discrete time complex exponentials are signals of great importance to the study of signals and systems. They can be related to sinusoids through Euler's formula, which identifies the real and imaginary parts of complex.

Sine Function :- a discrete-time sinusoid is periodic if its radian frequency Ω is a rational multiple of π. Otherwise, the discrete-time sinusoid is non-periodic. The fundamental period is 12 which corresponds to k = 1 envelope cycles.

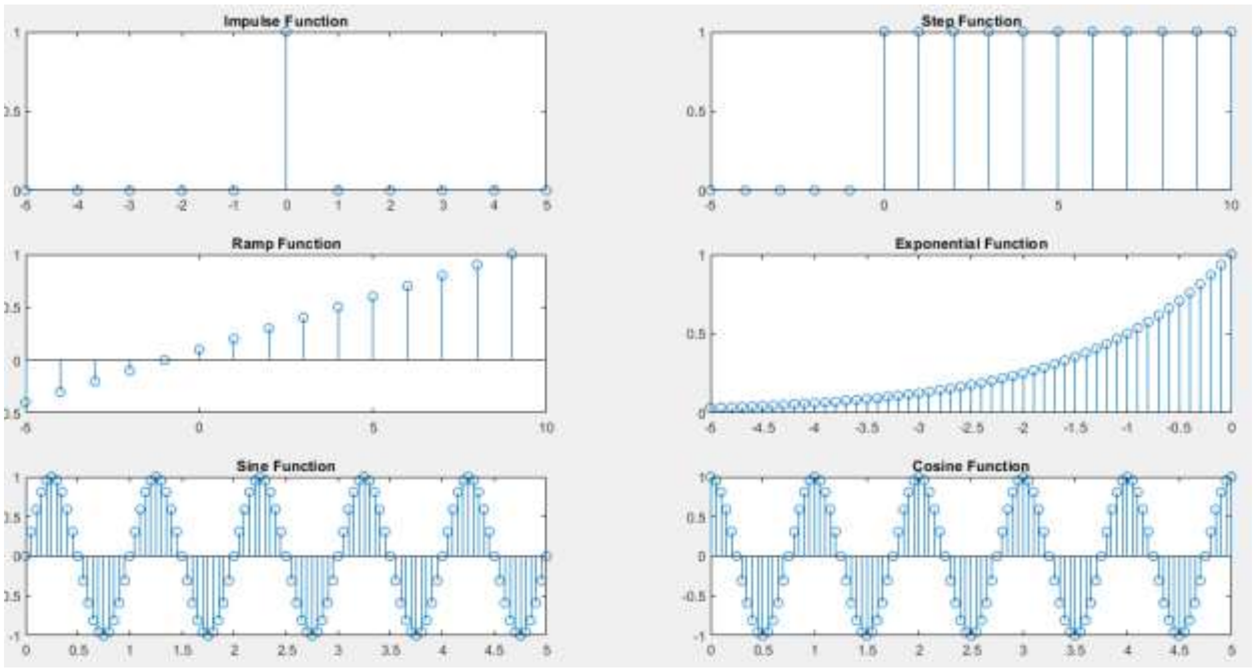
Cosine Function :- A discrete-time signal is periodic if there is a non-zero integer p ∈ Discrete Time such that for all n ∈ Discrete Time, x(n + p) = x(n). x(n) = Cos(2π f n).

Code:

```
% Generate Impluse function
n=-5:1:5;
x=[0 0 0 0 0 1 0 0 0 0 0];
subplot(3,2,1);
stem(n,x);
title('Impulse Function');
% Generate Step function
n=-5:1:10;
y=[0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1];
subplot(3,2,2);
stem(n,y);
title('Step Function');
% Generate Ramp function
```

```
n=-5:1:9;
z=[-0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
subplot(3,2,3);
stem(n,z);
title('Ramp Function');
% Generate Exponential function
n=-5:0.1:0;
a=2;
b=a.^n;
subplot(3,2,4);
stem(n,b);
title('Exponential Function');
% Generate Sine function
n=0:0.05:5;
s=sin(2*pi*n);
subplot(3,2,5);
stem(n,s);
title('Sine Function');
% Generate Cosine function
n=0:0.05:5;
c=cos(2*pi*n);
subplot(3,2,6);
stem(n,c);
title('Cosine Function');
```

Result and Observations:



Conclusion: Hence the Discrete Time Signals are generated and implemented.

Experiment No. 2

Aim: Write a MATLAB Program for computing Linear Convolution of Two Sequences.

Tool: Matlab

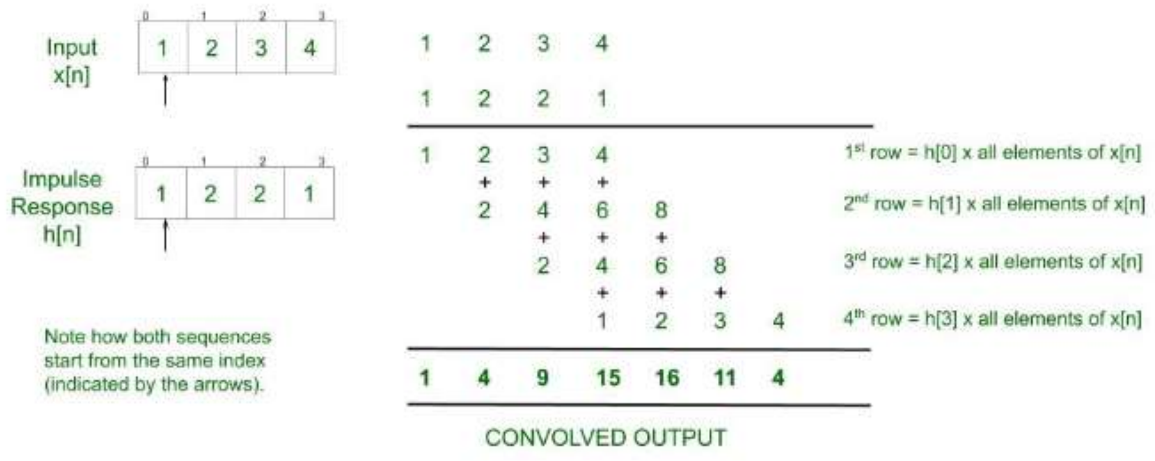
Theory:

Linear Convolution is a means by which one may relate the output and input of an LTI system given the system's impulse response. Clearly, it is required to convolve the input signal with the impulse response of the system. Using the expression earlier, the following equation can be formed-

$$f(n) * h(n) = \sum_{k=-\infty}^{\infty} f(k).h(n-k)$$

The reason why the expression is summed an infinite number of times is just to ensure that the probability of the two functions overlapping is 1. The impulse response is time-shifted endlessly so that during some duration of time, the two functions will certainly overlap. It may seem it would be careless on behalf of the programmer to run an infinite loop – the code may continue to execute for as long as the two functions do not overlap.

The solution lies in the fact LTI systems are being used. Since the functions do not change their values/shape over time (time-invariant), they can simply be slid closer to each other. Remember only the output is required, and it is not important 'when' the output is received. All manual calculations also depend on the same idea.



Input x[n]
0 1 2 3
1 2 3 4
Impulse Response h[n]
0 1 2 3
1 2 2 1
Note how both sequences start from the same index (indicated by the arrows).
1 2 3 4
1 2 2 1
1st row = h[0] x all elements of x[n]
2nd row = h[1] x all elements of x[n]
3rd row = h[2] x all elements of x[n]
4th row = h[3] x all elements of x[n]
1 4 9 15 16 11 4
CONVOLVED OUTPUT

Explanation: Here's one technique that may be used while calculating the output:

- Take the input signal and impulse response as two separate single-row matrices.
- The first element of the impulse response is multiplied with every element of the input signal. This result is stored.

- The second element of the impulse response is multiplied with every element of the input signal. The result is shifted by one step to the right and stored.
- The above two steps are done for the remaining elements in the impulse response.
- Once all elements have been multiplied, align all the results under one another. Refer to the figure below.
- Vertically, add all the elements in each column.
- The resulting single row matrix is the convolved output.

Input Parameter: Enter First sequence x1[n] : [1 2 3 4 5]

Enter First location Of Sequence x1[n] : 2

Enter Second sequence x2[n] : [5 4 3 2 1]

Enter First location Of Sequence x2[n] : 5

Convoluted Sequence

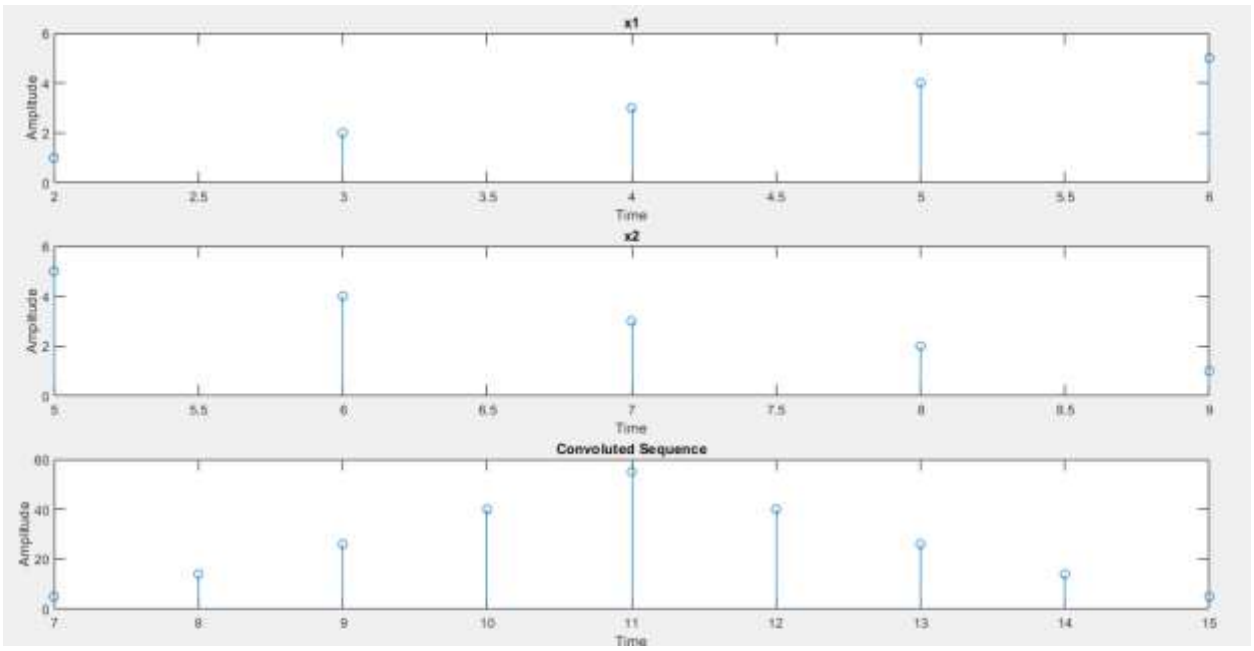
```
  5   14   26   40   55   40   26   14    5
```

Code:

```
clear all;
close all;
x1 = input('Enter First sequence x1(n)[] : ');
t1 = input('Enter First location Of Sequence x1 : ');
x2 = input('Enter Second sequence x2(n)[] : ');
t2 = input('Enter First location Of Sequence x2 : ');
l1 = length(x1); %length of sequence x1
l2 = length(x2); %length of sequence x2
ln = l1+l2-1; %length of convoluted sequence
y = conv(x1,x2); % performing convolution using conv() function
a = t1+l1-1;
t = t1:a;
subplot(3,1,1);
stem(t,x1);
xlabel('Time');
ylabel('Amplitude');
title('x1');
a = t2+l2-1;
t = t2:a;
subplot(3,1,2);
stem(t,x2);
xlabel('Time');
ylabel('Amplitude');
title('x2');
tn = t1+t2;
a = tn+ln-1;
t = tn:a;
subplot(3,1,3);
disp('Convoluted Sequence ');
disp(y); % For printing the convoluted output in MATLAB command window
stem(t,y); %  For plotting the convoluted output in MATLAB graph window
```

```
xlabel('Time');
ylabel('Amplitude');
title('Convoluted Sequence');
```

Result and Observations:



Conclusion: Hence the linear convolution is studied.

Experiment No. 3

Aim: Write a MATLAB Program to compute frequency response of first order system h(n)=$(0.8)^n$ U(n)

Tool: Matlab

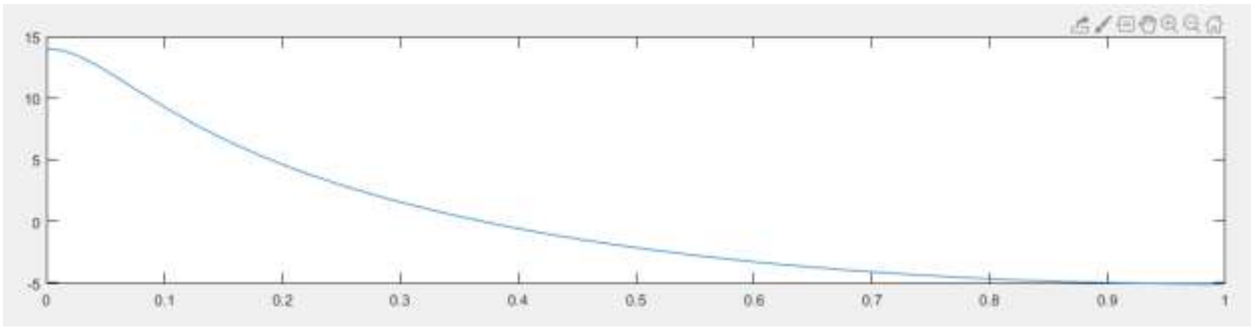
Theory:

A frequency response describes the steady-state response of a system to sinusoidal inputs of varying frequencies and lets control engineers analyze and design control systems in the frequency domain. Using frequency response analysis during the design phase helps the designer to determine whether there will be frequencies that will have excessive vibration or noise. The design can then be changed as needed to improve the vibration and noise characteristics of the vehicle if required to understand why the frequency domain is important consider an acoustic guitar. A first order system or network is one that contains but a single energy storage element such as an inductor or capacitor. Each of these elements, either singly or in combination and in association with resistors, may be arranged in series, parallel, series-parallel or parallel-series.
Whatever the arrangement, the differential equation that governs the behaviour of the network is of first order.
In electronics this stimulus would be an input signal. In the audible range it is usually referred to in connection with electronic amplifiers, microphones and loudspeakers. Radio spectrum frequency response can refer to measurements of coaxial cable, twisted-pair cable, video switching equipment, wireless communications devices, and antenna systems. Infrasonic frequency response measurements include earthquakes and electroencephalography (brain waves).

Code:

```
b=1;
a=[1 -0.8];
[H,w]=freqz(b,a);
subplot(2,1,1);
plot(w/pi,20*log10((abs(H))));
```

Result and Observations:



Conclusion: Hence the frequency response of first order system h(n)=$(0.8)^n$ U(n) is computed.