ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

MATLAB Simulation for Digital Signal processing

V. K. Singh

Assistant Professor, Department of Electronics & Communication Engineering, BTKIT Dwarahat, Almora, India

Abstract: This paper presents the MATLAB simulation of discrete time signal and also discuss about of their mathematical operations and properties. In this paper we start with basic DSP signals and their MATLAB simulation coding. This paper also explain time domain and frequency domain analysis of discrete time signal for frequency domain analysis two important transform technique discuss like Z-transform and Fourier transform. It is thus important to learn first how to generate in the time domain some basic discrete-time signals in MATLAB and perform elementary operations on them, which are the main objectives of this paper. A secondary objective is to learn the application of some basic MATLAB commands and how to apply them in simple digital signal processing problems.

Keywords: DSP, MATLAB, Simulation, Commands.

I. INTRODUCTION

The area of digital signal processing has advanced rapidly over the last four decades. This advancement is attributed to the progress in digital computer technology and integrated circuit fabrications. There are many reasons why the digital processing of an analog signal is preferred over the signal processing directly in analog domain. During the past several decades the field of digital signal processing (DSP) has grown to be important, both theoretically and technologically. A major reason for its success in industry is the development and use of low-cost software and hardware. New technologies and applications in various fields are now taking advantage of DSP algorithms. Digital Signal Processing (DSP) is concerned with the representation, transformation and manipulation of signals on a computer.

Digital Signal Processing is one of the most powerful technologies that will shape science and engineering in the twenty-first century. Revolutionary changes have already been made in a broad range of fields: communications, medical imaging, radar & sonar, high fidelity music reproduction, and oil prospecting, to name just a few. Each of these areas has developed a deep DSP technology, with its own algorithms, mathematics, and specialized techniques.
Since DSP applications are primarily algorithms that are implemented either on a DSP processor or in software, a fair amount of programming is required. Using interactive software, such as MATLAB, it is now possible to place more emphasis on learning new and difficult concepts than on programming algorithms. MATLAB is an interactive, matrix-based system for scientific and engineering numeric computation and visualization. Its strength lies in the fact that complex numerical problems can be solved easily and in a fraction of the time required by a programming language such as FORTRAN or C. It is also powerful in the sense that, with its relatively simple programming capability, MATLAB can be easily extended to create new commands and functions.

II. GENERATION, ANALYSIS AND PLOTS OF DISCRETE TIME SIGNALS

A. Unit Step Signal:

MATLAB SCRIPT:

```
n=-49:49;
u=[zeros(1,49),ones(1,50)];
stem(n,u)
```

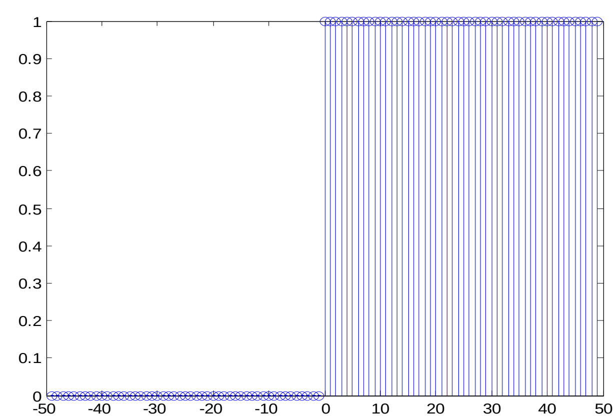

Fig. 1. Unit step signal MATLAB plot

B. Impulse Signal:

MATLAB SCRIPT:

```
n=-49:49;
delta=[zeros(1,49),1,zeros(1,49)];
```

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

stem(n,delta)

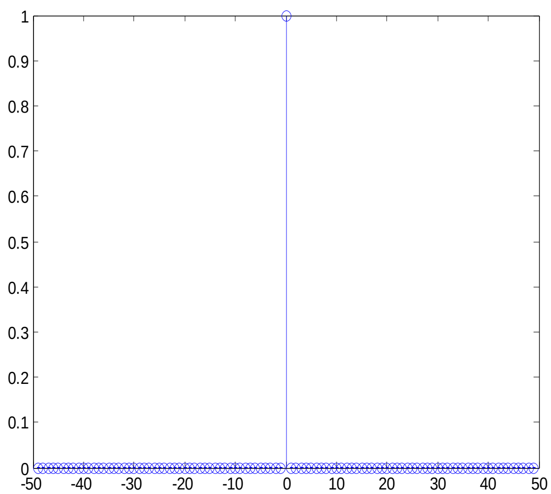

Fig. 2. Impulse signal MATLAB plot

C. Ramp Signal

MATLAB SCRIPT:
n=0:10;
r=n;
stem(n,r)

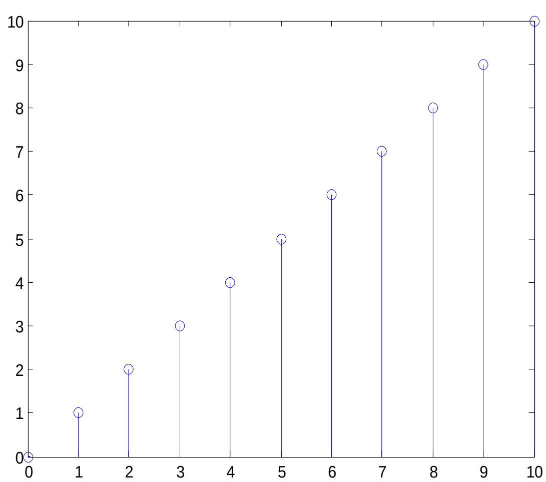

Fig. 3. Ramp signal MATLAB plot

D. Sin Signal

MATLAB SCRIPT:
n=-18:18;
f=1/12;
stem(n,sin(2*pi*f*n))

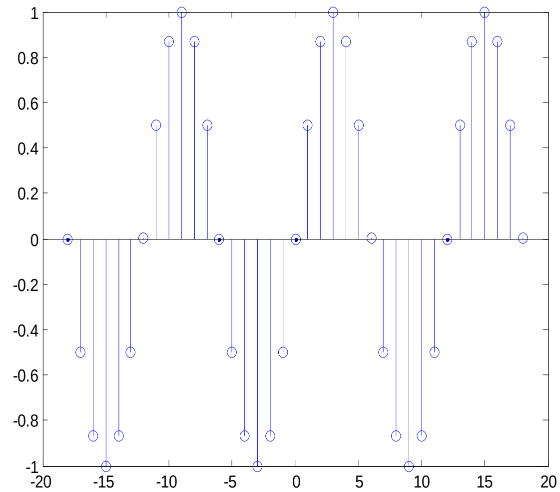

Fig. 4. Sin signal MATLAB plot

E. Exponential Signal

MATLAB SCRIPT
c=1; alpha=0.8; n=-10:10;
x=c*alpha.^n;
stem(n,x)

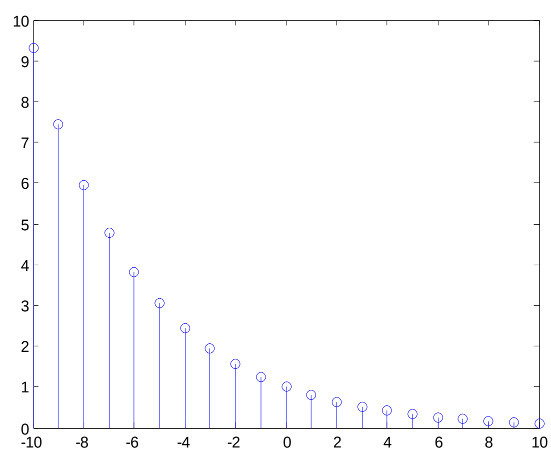

Fig. 5. Exponential signal MATLAB plot

III. IMPLEMENTATION OF OPERATIONS ON SEQUENCES (ADDITION, MULTIPLICATION, FOLDING)

A. Addition of two sequences

MATLAB SCRIPT:
n=-3:3;y1=[1,3,5,6,7,9,2];y2=[2,4,7,2,9,5,6];
y3=y1+y2;
stem(n,y3)

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

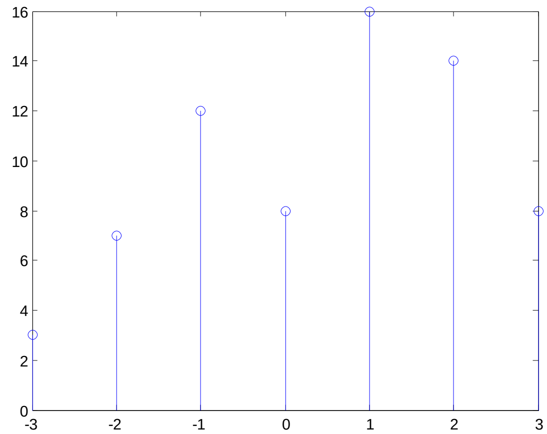

Fig. 6. Addition of two sequences MATLAB plot

B. Multiplication of two sequences

MATLAB SCRIPT:

```
n=-3:3;
y1=[1,3,5,6,7,9,2];
y2=[2,4,7,2,9,5,6];
y3=y1.*y2;
stem(n,y3)
```

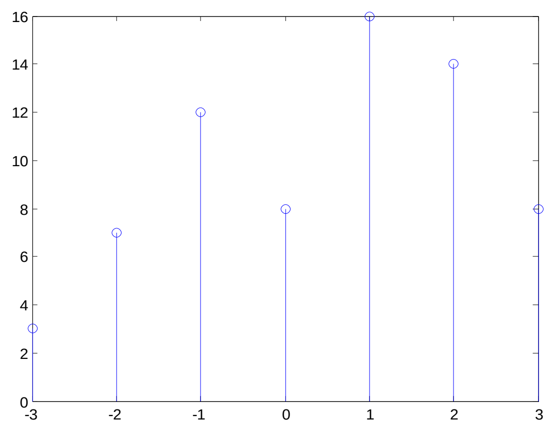

Fig. 7. Multiplication of two sequences MATLAB plot

C. Folding operation on sequence

MATLAB SCRIPT:

```
n=-3:3;
y1=[1,3,5,6,7,9,2];
y2=fliplr(y1); stem(n,y2)
```

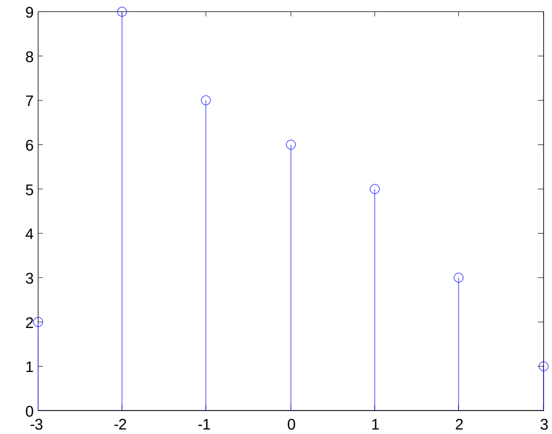

Fig. 8. Folding operation on sequence MATLAB plot

IV. PLOT THE RESPONSE FOR DIFFERENTIAL EQUATION

A. Impulse Response y(n)-0.6y(n-1)+0.08y(n-2)=x(n)

MATLAB SCRIPT:

```
n=0:20;
x=[1,zeros(1,20)];
a=[1];
b=[1 -0.6 0.08];
y=filter(a,b,x);
stem(n,y)
```

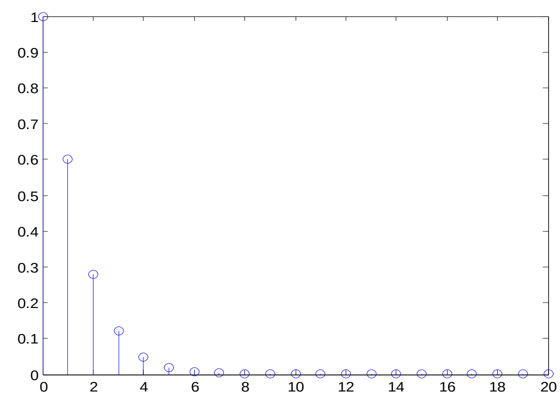

Fig. 9. Impulse Response of differential equation MATLAB plot

B. Unit step Response:

y(n)=0.7y(n-1)+0.12y(n-2)=x(n-1)+x(n-2)

with initial condition y(-1) =1,y(-2)=1

MATLAB SCRIPT:

```
n=0:20;
x=[ones(1,21)];
a=[0 1 1];
b=[1 -0.7 .12];
y0=[1 1];
ic=filtic(a,b,y0);
```

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

```
y=filter(a,b,x,ic);
stem(n,y)
```

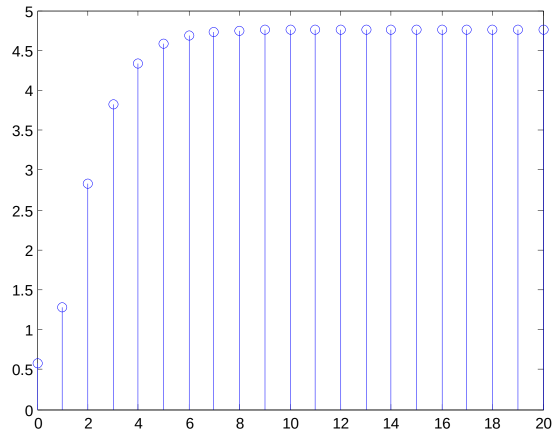

Fig. 9. Unit step Response of differential equation MATLAB plot

C. Pole Zero Plot of y(n)-0.6y(n-1)+0.08y(n-2)=x(n)

MATLAB SCRIPT:

```
a=[1];
b=[1 -0.6 0.08];
zplane(a,b)
```

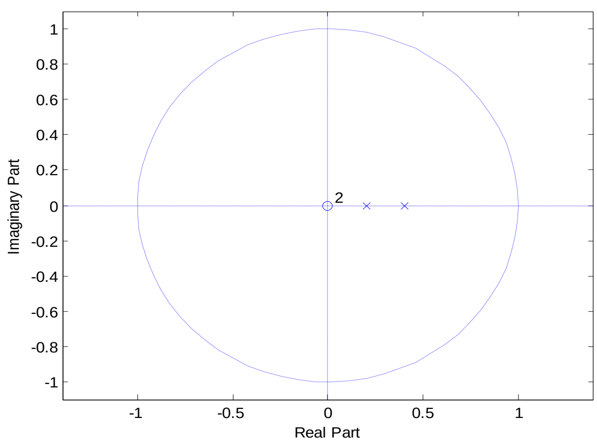

Fig. 10. Pole zeros plot of differential equation

V. LINEAR/CIRCULAR CONVOLUTION OF TWO SEQUENCES.

A. Linear convolution of two sequences

MATLAB SCRIPT:

```
clc;clear all;close all;
x=input('Enter x[n]:');
nx=0:length(x)-1;
h=input('Enter h[n]:');
nh=0:length(h)-1;
z=conv(x,h);
nz=0:length(z)-1;
subplot(3,1,1);
stem(nx,x);
title('Input sequence x[n]');
subplot(3,1,2);
stem(nh,h);
xlabel('Time');ylabel('Amplitude');
title('Impulse response of the system h[n]');
subplot(3,1,3);
stem(nz,z);
xlabel('Time');ylabel('Amplitude');
title('Linear Convolution');
Enter:
```

x[n]:[1 2 3 1];
h[n]:[1 2 1 -1];

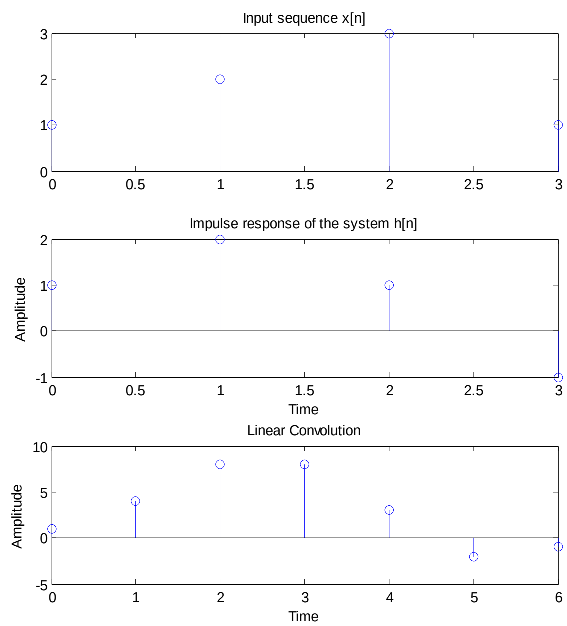

Fig. 11. Linear Convolution of two sequences

B. Circular convolution of two sequences

MATLAB SCRIPT:

```
clc;clear all;close all;
x=input('Enter x[n]:');
h=input('Enter h[n]:');
xl=length(x);
hl=length(h);
m=max(xl,hl);
z=ifft(fft(x,m).*fft(h,m));
zl=length(z);
%Plots
nx=0:xl-1;
nh=0:hl-1;
nz=0:zl-1;
subplot(3,1,1);
stem(nx,x);
```

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

```
xlabel('Time');ylabel('Amplitud')
title('Input sequence x[n]');
subplot(3,1,2);
stem(nh,h);
xlabel('Time');ylabel('Amplitude
title('Impulse response of the system h[n]');
subplot(3,1,3);
stem(nz,z);
xlabel('Time');ylabel('Amplitude)
title('Circular Convolution');
```

Enter

x[n]:[2 1 2 1];
h[n]:[1 2 3 4]

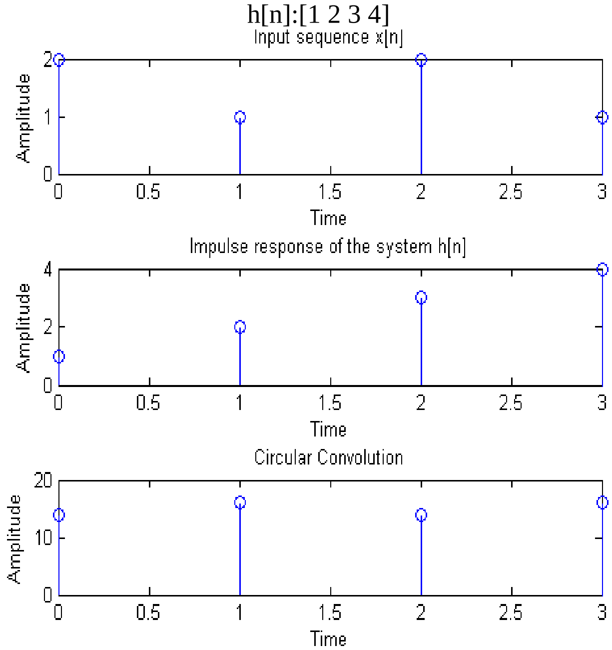

Fig. 12. Circular Convolution of two sequences

VI. COMPUTATION AND PLOT OF N-POINT DFT OF A GIVEN SEQUENCE

MATLAB SCRIPT:

```
clc;clear all;close all;
x=input('Enter x[n]:');
nx=0:length(x)-1;
N=4;
%Compute DFT
n=0:length(x)-1;
for k=0:N-1
w=exp(-j*2*pi*k*n/N);
dot_prod=x.*w;
X(k+1)=sum(dot_prod);
end
%Plot the input
subplot(3,1,1);
stem(nx,x);
xlabel('Time');ylabel('Input x[n]');
title('Input sequence x[n] ');
title('Computation of DFT ');
%Plot the magnitude spectrum
subplot(3,1,2);
stem(abs(X));
xlabel('Time');ylabel('Amplitude');
title('Magnitude Spectrum');
%Plot the phase spectrum
subplot(3,1,3);
stem(angle(X));
xlabel('Time');ylabel('Angle in radian');
title('Phase Spectrum');
```

Enter X[n]: [1 2 3 0]

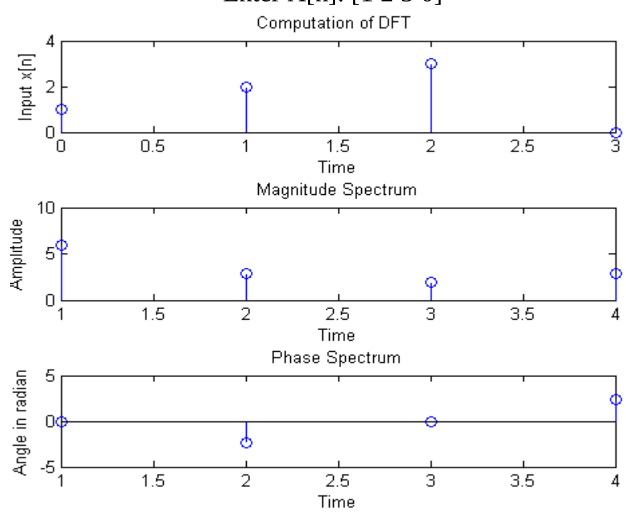

Fig. 13. N-point DFT computation

VII. CONCLUSION

This paper helps lots of beginner those who are interested to learn DSP with MATLAB programming, because this paper start with basic signal programming and goes to various mathematical operations, frequency domain analysis programming. Main features of this paper Linear/ Circular convolution programming you can easily see the difference between these two convolutions and than Fourier transform analysis of discrete time signal.

REFERENCES

[1] V. K. Ingle and J. G. Proakis, Digital Signal Processing using MATLAB, Cengage Learning , 2007.
[2 Online Resources like, nptel.com, etc.