

# Mathematical Operations on Basic Discrete Time Signals with MATLAB Programming

N. D. Narkhede<sup>1</sup>, Dr. J. N. Salunke<sup>2</sup>, V. N. Shah<sup>3</sup>

1. Associate Professor, J. T. Mahajan college of Engineering, Faizpur, Jalgaon (India)

2. Director of School of Mathematical Sciences, SRTM University Nanded (India)

3. Assistant Professor, SIER, Agaskhind, Nashik

**Abstract:** This paper deals with the simulation of some of the discrete time signals using MATLAB and discusses some of their mathematical operations and properties. Basic discrete time signals with MATLAB simulation is discussed in introduction along with the time domain and frequency domain analysis of discrete time signals. The frequency domain analysis is discussed with help of two important transform techniques i.e. Z-transform and Discrete Fourier transform. It is thus important to learn first how to generate signals in time domain and perform basic 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 its usage in solving simple digital signal processing (DSP) problems.

**Keywords:** DTFT, Z-transform, MATLAB, DSP Simulation, Commands

\*\*\*\*\*

## I INTRODUCTION

The area of digital signal processing has advanced rapidly over the last few 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 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 effective low-cost software and hardware. New technologies and applications in various fields are now taking advantage of DSP algorithms. Digital Signal Processing 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 understanding of 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

```
clc;
clear all;
close all;
n=-15:15;
u=[zeros(1,15), ones(1,16)];
stem(n,u)
ylim([0, 1.5])
xlabel('time');ylabel('Amplitude')
title('Step signal')
```

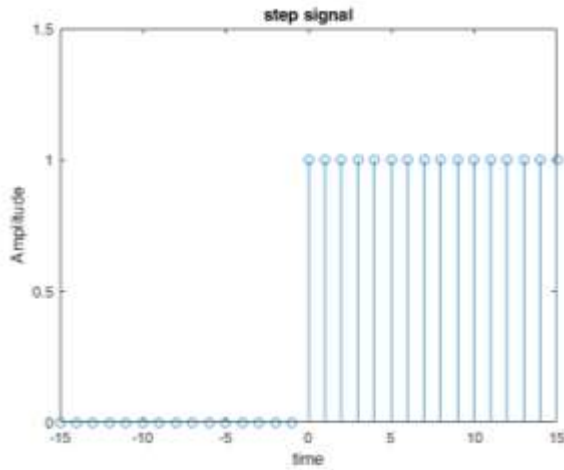


Fig. 1. Unit step signal MATLAB plot

```
clc;
clear all;
close all;
```

```
d=zeros(1,15),1,zeros(1,15)];
stem(n,d)
ylim([0, 1.5])
xlabel('time');
ylabel('Amplitude')
title('Delta signal')
```

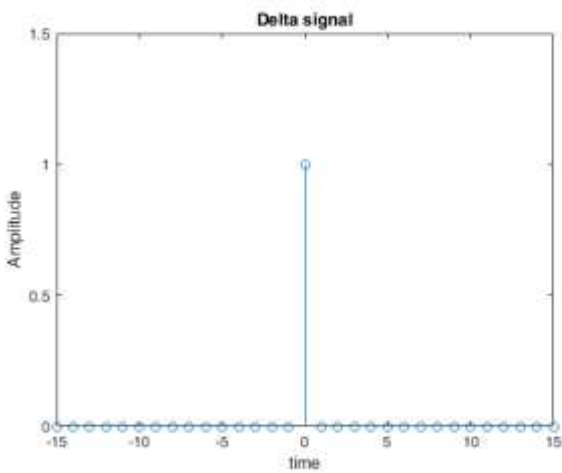


Fig. 2. Delta signal MATLAB plot

```
clc;
clear all;
close all;
```

```
n=0:5;
stem(n)
xlabel('time');
ylabel('Amplitude')
title('Ramp signal')
```

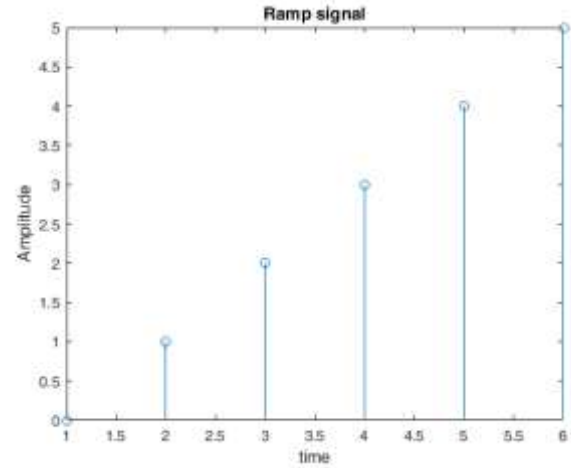


Fig. 3. Ramp signal MATLAB plot

```
clc;
clear all;
close all;
```

```
n=-15:15;
f=1/20;
s=sin(2*pi*f*n);
stem(n,s)
xlabel('time');
ylabel('Amplitude')
title('Sinusoidal signal')
```

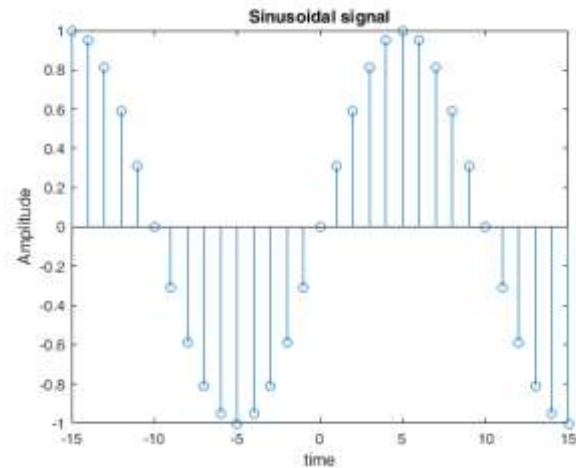


Fig. 4. Sinusoidal signal MATLAB plot

```
clc;
clear all;
close all;
```

```
n=0:15;
x=0.75.^n;
stem(n,x);xlabel('time');
ylabel('Amplitude');title('Exponential signal')
```

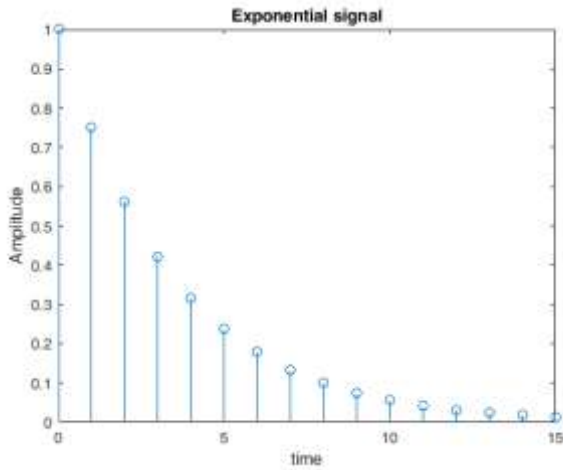


Fig. 5. Exponential signal MATLAB plot

```
clc;
clear all;
close all;
n=-5:5;
y1=[1,1,-1,-2,1,2,3,4,5,1,2];
y2=[1,0,1,2,1,2,-3,-2,-5,3,2];
y=y1+y2;
stem(n,y)
ylim([0 5]); xlabel('time');ylabel('Amplitude')
title('Addition of two sequences')
```

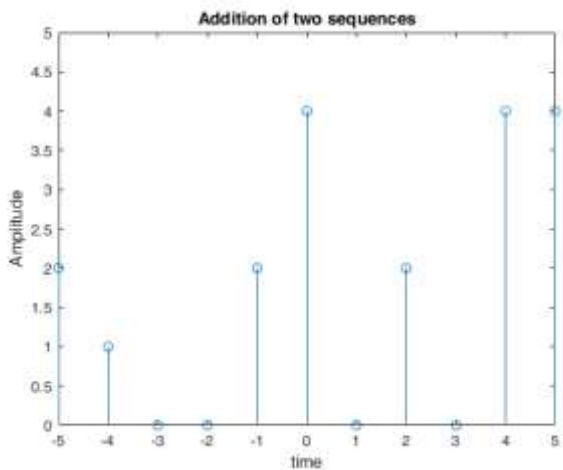


Fig. 5. Addition of two signal

```
clc;
clear all;
close all;
n=-5:5;
y1=[1,1,-1,-2,1,2,3,4,5,1,2];
y2=[1,0,1,2,1,2,-3,-2,-5,3,2];
y=y1.*y2;
stem(n,y)
ylim([0 5])
xlabel('time');ylabel('Amplitude')
title('Multiplication of two sequences')
```

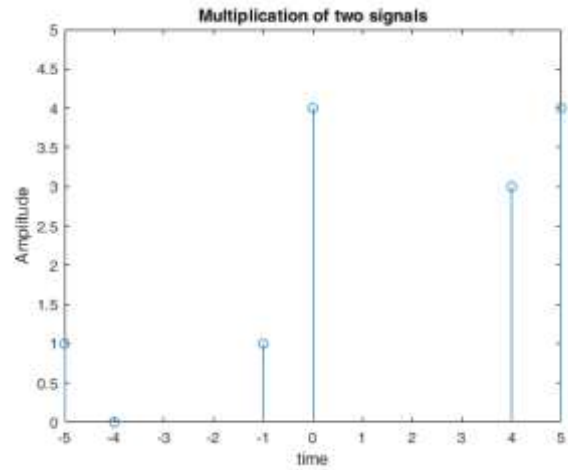


Fig. 6. Multiplication of two signal

```
clc;clear all;close all;
n=-5:5;
y1=[1,1,-1,-2,1,2,3,4,5,1,2];
y2=fliplr(y1);
subplot(211); stem(n,y1); ylim([0 5])
xlabel('time');ylabel('Amplitude'); title('Original Sequence')
subplot(212); stem(n,y2);ylim([0 5])
xlabel('time');ylabel('Amplitude');title('Folding operation of Sequence')
```

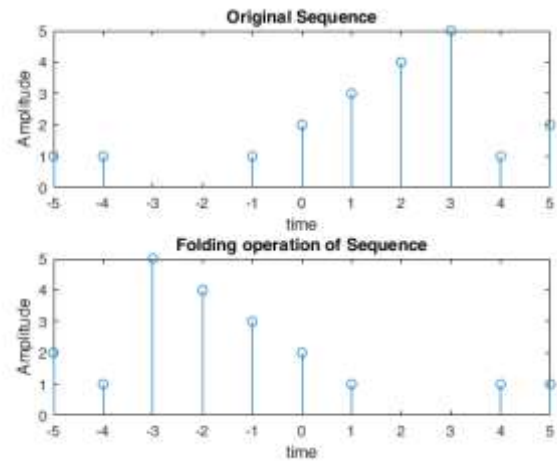


Fig. 7. Folding of a signal

```
clc;
clear all;close all;
n=0:13;
x=[1 1 zeros(1,12)];
a=[1];
b=[1, -0.5, 0.03];
y=filter(a,b,x);
stem(n,y)
xlabel('time');ylabel('Amplitude'); title('Impuse respopnse of differential equation')
```

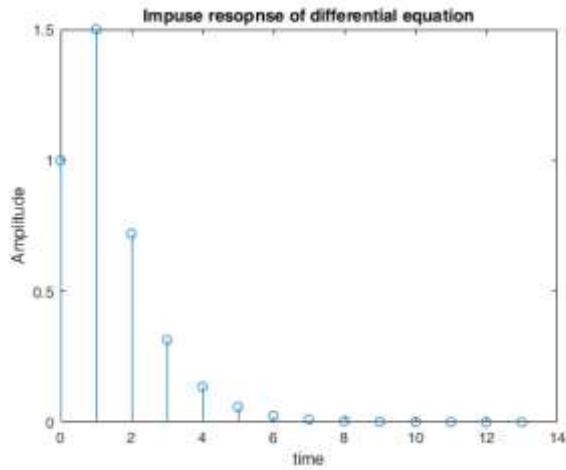


Fig. 8. Impulse Response of differential equation

```
clc;
clear all;close all;
n=0:13;
x=[ones(1,12)];
a=[0 1];
b=[1, -0.7, 3];
y0=[1 1];
c=filtic(a,b,y0);
y=filter(a,b,x,c);
stem(y);xlabel('time');
ylabel('Amplitude')
title('Unit step response')
```

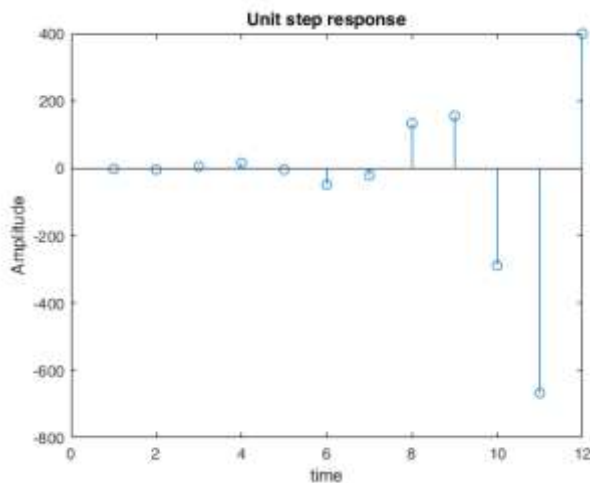


Fig. 9. Unit Step response

```
clc;
clear all;close all;

a=[1];
b=[1 -0.6 -0.4 .8];
zplane(a,b);
```

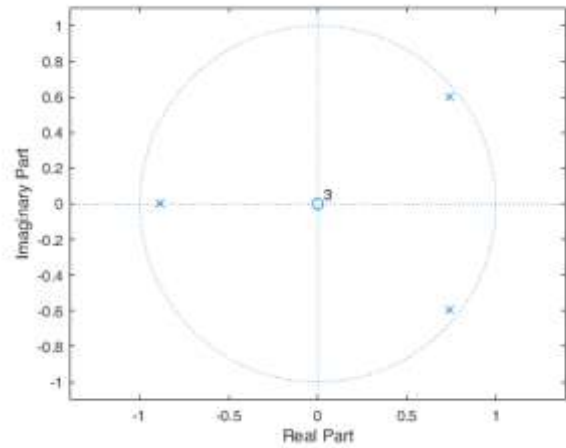


Fig. 10. Pole Zero plot

```
clc;clear;close all;
x1=[1 0.5 2 1 4];
x2=[0.5 1 3 1 1];
x3=conv(x1,x2);
subplot(3,1,1);stem(x1);xlabel('n');ylabel('x1(n)');
title('First Input');
subplot(3,1,2);stem(x2);xlabel('n');ylabel('x2(n)');
title('Second Input');
subplot(3,1,3)
stem(x3);xlabel('n');ylabel('x3(n)');
title('Linear Convolution Output');
```

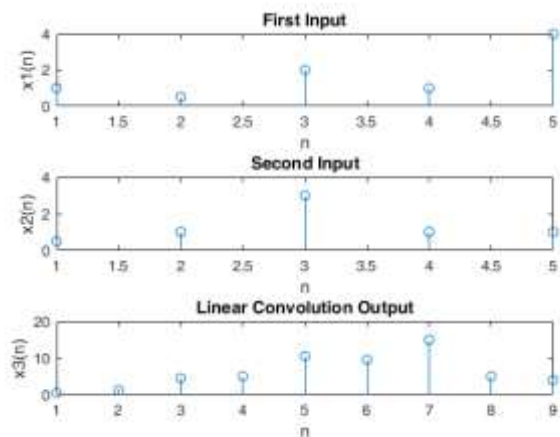


Fig. 11. Linear Convolution

```
clc;clear;close all;
x1=[1 0.5 2 1 4];
x2=[0.5 1 3 1 1];
x3=cconv(x1,x2);
subplot(3,1,1);stem(x1);xlabel('n');ylabel('x1(n)');
title('First Input');
subplot(3,1,2);stem(x2);xlabel('n');ylabel('x2(n)');
title('Second Input');
subplot(3,1,3);stem(x3);xlabel('n');ylabel('x3(n)');
title('Circular Convolution Output');
```

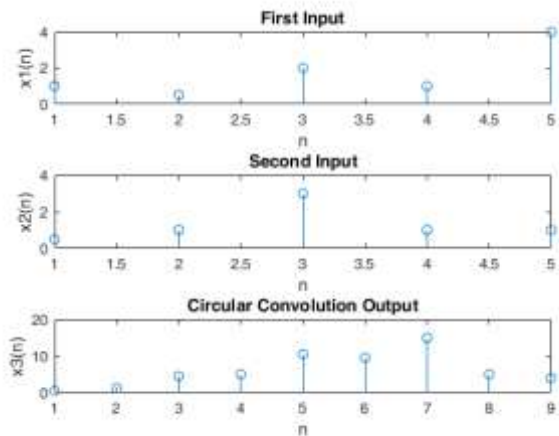


Fig. 12. Circular Convolution

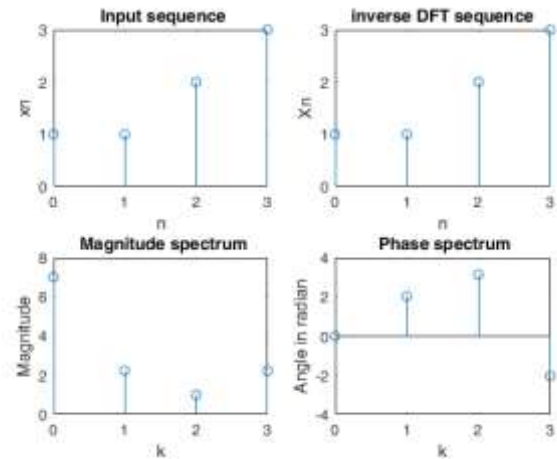


Fig. 13. DFT of a signal

```
clear all
clc
close all
N=4;
xn=[1,1,2,3];
disp('DFT of the sequence xn is,')
Xk=fft(xn,N);
disp('The magnitude sequence is,')
MagXk=abs(Xk);
disp(' The phase sequence is,')
PhaXk=angle(Xk);
disp(' inverse DFT of the sequence Xk is,')
Xn=ifft(Xk);

n= 0:1:N-1;
wk=0:1:N-1;
subplot(2,2,1)
stem(n,xn)
title(' Input sequence')
xlabel('n'); ylabel('xn')
subplot(2,2,2)
stem(n,Xn)
title('inverse DFT sequence')
xlabel('n'); ylabel('Xn')
subplot(2,2,3)
stem(wk,MagXk)
title(' Magnitude spectrum')
xlabel('k'); ylabel('Magnitude')
subplot(2,2,4)
stem(wk,PhaXk)
title(' Phase spectrum')
xlabel('k'); ylabel('Angle in radian')
```

### III CONCLUSION

This paper helps lots of beginner those who are interested to learn DSP with MATLAB programming, because this paper starts with basic signal programming and goes to various mathematical operations and frequency domain analysis programming. The main features of this paper involve Linear/ Circular convolution programming, difference between these two convolutions and then the Discrete Fourier transform analysis of discrete time signals.

### REFERENCES

- [1] Smith, J.O., 2007. *Mathematics of the discrete Fourier transform (DFT): with audio applications*. Julius Smith
- [2] Roberts, M.J., 2012. *Signals and systems: analysis using transform methods and MATLAB*. McGraw-Hill Higher Education
- [3] Kamen, E. and Heck, B.S., 2007. *Fundamentals of Signals and Systems Using the Web and MatLab: AND Mathworks, MATLAB Sim SV 07*. Prentice Hall Press.
- [4] V. K. Ingle and J. G. Proakis, *Digital Signal Processing using MATLAB*, Cengage Learning, 2007.
- [5] Online Resources like, nptel.com, etc.
- [6] Haykin, S. and Van Veen, B., 2007. *Signals and systems*. John Wiley & Sons