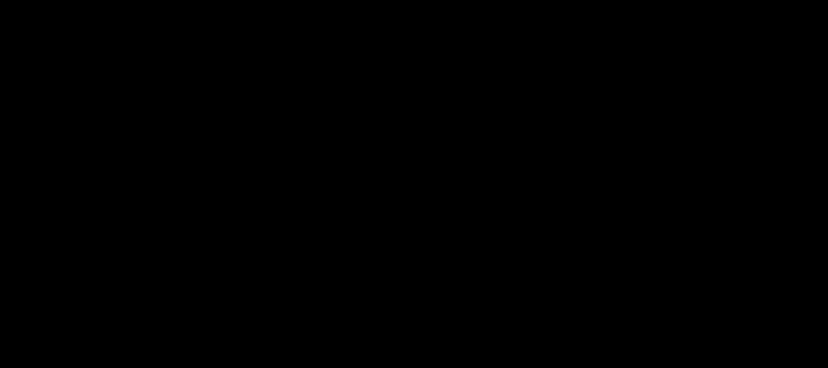


**K. J. Somaiya College of Engineering, Mumbai-77**

**(Autonomous College Affiliated to University of Mumbai)**



**Batch:\_ A1** **Roll No. 1911004**



**Experiment No. 2**



**Grade: AA / AB / BB / BC / CC / CD /DD**



**Signature of the Staff In-charge with date**



**Title:** Represent discrete time signals and perform different operations on them.

**Objective:** To familiarize the beginner to MATLAB by introducing the basic featuresand commands of the program.

**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
| **CO** | **Outcome** |
|  |  |
| **CO1** | Identify various discrete time signals and systems and perform signal manipulation |
|
|  |
|  |  |

**Books/ Journals/ Websites referred:**

1. http://www.mathworks.com/support/
2. www.math.mtu.edu/~msgocken/intro/intro.html
3. www.mccormick.northwestern.edu/docs/efirst/matlab.pdf
4. A.Nagoor Kani “Digital Signal Processing”, 2nd Edition, TMH Education.

**Pre Lab/ Prior Concepts:**

Using MATLAB we can easily generate all basic functions such as unit step, ramp, growing and decaying exponential, etc. The various signals plotted in this program are Step signal,



Dept. of Computer Engg. DSIP Lab Sem VI Jan-Apr 2022 Page 1

Ramp signal, Exponential signal etc

**1. Unit Step Signal**

The step signal is defined as

U[n] = k ; if n>=0

= 0 ; otherwise

When k=1 it is called as unit step signal.

**2. Ramp Signal**

The ramp signal is defined as r[n] = n ; if n>=0

* + 0 ; otherwise

1. **Exponential Signal**

The exponential signal is defined as

X[n] =a^n

When ‘a’ is greater than 1 it is **increasing** exponential

When ‘a’ is less than 1 it is **decaying** exponential.

**4. Impulse Signal**

The impulse signal is defined as d[n] = k ; if n=0

= 0 ; otherwise

When k=1 it is called as unit impulse





**K. J. Somaiya College of Engineering, Mumbai-77**

**(Autonomous College Affiliated to University of Mumbai)**

The functions used in this program are:

1. **Ones**

This function is used to create an array of all ones Syntax: Y=ones (m, n)

**Description:**

Y=ones (n) returns an n-by-n matrix of 1’s.

An error message appears if n is not a scalar.

Y=ones (m, n) or Y=ones([m n]) returns an m-by-n matrix of ones.

**b. Zeros**

This function is used to create an array of all zeros

Syntax: Y=zeros(m,n)

**Description:**

Y=zeros(n) returns an n-by-n matrix of 0’s.

An error message appears if n is not a scalar.

Y=zeros (m,n) or Y=ones([m n]) returns an m-by-n matrix of Zeros.

**c. EXP**

This function is used to plot exponential signals

Syntax: Y=exp(X)

**Description:**

The exp function is an elementary function that operates element-wise on arrays. Its domain includes complex numbers.

Y=exp(X) returns the exponential for each element of X. For complex, it returns the complex exponential.



**Steps with Syntax for representation of above discrete time signals:**

x = [];

y = [];

p = -3

q = 5

a = [0];

b = [1];

for t=p:1:q

x = cat(2, x, t)

if t == 0

y = cat(2,y,b)

else

y = cat(2,y,a)

end

end

subplot(2,2,1);

stem(x,y);

ylabel('d(n)');

xlabel('unit impulse');

x = []

y = []

for t=p:1:q

x = cat(2, x, t)

if t >= 0

y = cat(2,y,b)

else

y = cat(2,y,a)

end

end

subplot(2,2,2);

stem(x, y);

ylabel('Amplitude');

xlabel('unit step');

x = []

y = []

for t=p:1:q

x = cat(2, x, t)

if t >= 0

y = cat(2,y,[t])

else

y = cat(2,y,a)

end

end

subplot(2,2,3);

stem(x,y);

ylabel('Amplitude');

xlabel('Ramp');

t=p:1:q;

a = 0.5

y2=exp(a\*t);

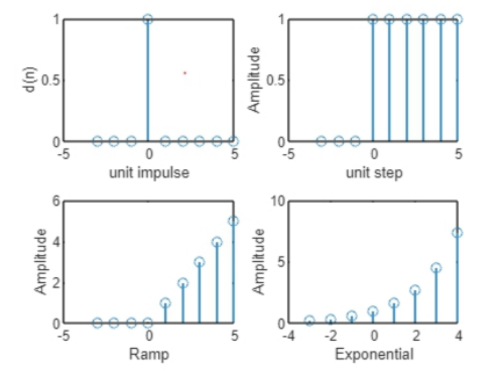
subplot(2,2,4);

stem(t,y2);

xlabel('Exponential');

ylabel('Amplitude');

axis([-4 4 0 10])



**Operations on Signals:**

1. Addition of signals.
2. Subtraction of signals.
3. Multiplication of two signals.
4. Scaling – Upscaling & Downscaling.
5. Shift operation – Advance/Right shift & Delay/Left shift.



**Steps with Syntax for representation of above operations on discrete time signals**

n= input('Enter number of discrete points for both the signal : ');

t = [];

t = (0:n-1);

v1= [];

v2= [];

i=0;

for i=1:n

d=input(['Enter discrete signal 1 value number ' num2str(i) ':']);

v1(end+1)=d;

end

fprintf("Discrete signal 1 is ");

disp(v1);

subplot(4,3,1);

stem(t,v1);

axis([t(1) t(end) -10 10]);

title('Signal 1')

xlabel('t')

ylabel('x(t)')

for i=1:n

d=input(['Enter discrete signal 2 value number ' num2str(i) ':']);

v2(end+1)=d;

end

fprintf("Discrete signal 2 is ");

disp(v2);

subplot(4,3,2);

stem(t,v2);

axis([t(1) t(end) -10 10]);

title('Signal 2')

xlabel('t')

ylabel('x(t)')

delay(v1,t);

advance(v1,t);

upscale(v1,t);

downscale(v1,t);

add(v1,v2,t);

sub(v1,v2,t);

mult(v1,v2,t);

reverse(v1,t);

function delay(v1,t)

sh1=input('Enter shifting factor for delayed signal: ');

t=t+sh1;

subplot(4,3,3);

stem(t,v1);

axis([t(1) t(end) -10 10]);

title('Time delayed signal 1');

xlabel('t');

ylabel('x(t)');

end

function advance(v1,t)

sh2=input('Enter shifting factor for advanced signal: ');

t=t-sh2;

subplot(4,3,4);

stem(t,v1);

axis([t(1) t(end) -10 10]);

title('Time advanced signal 1');

xlabel('t');

ylabel('x(t)');

end

function upscale(v1,t)

a=input('Enter upscaling factor for signal 1: ');

v1=v1\*a;

subplot(4,3,5);

stem(t,v1);

axis([t(1) t(end) -10\*a 10\*a]);

title('Upscaled Signal 1');

xlabel('t');

ylabel('x(t)');

end

function downscale(v1,t)

a=input('Enter downscaling factor for signal 1: ');

v1=v1/a;

subplot(4,3,6);

stem(t,v1);

axis([t(1) t(end) -10\*a 10\*a]);

title('Downscaled Signal 1');

xlabel('t');

ylabel('x(t)');

end

function add(v1,v2,t)

sum = v1 + v2;

fprintf('Resultant signal after addition is: ');

disp(sum);

subplot(4,3,7);

stem(t,sum);

axis([t(1) t(end) -20 20]);

title('Addition Signal');

xlabel('t');

ylabel('x(t)');

end

function sub(v1,v2,t)

diff = v1 - v2;

fprintf('Resultant signal after subtraction is: ');

disp(diff);

subplot(4,3,8);

stem(t,diff);

axis([t(1) t(end) -20 20]);

title('Subtraction Signal');

xlabel('t');

ylabel('x(t)');

end

function mult(v1,v2,t)

mult = v1 .\* v2;

fprintf('Resultant signal after multiplication is: ');

disp(mult);

subplot(4,3,9);

stem(t,mult);

axis([t(1) t(end) -40 40]);

title('Multiplication Signal');

xlabel('t');

ylabel('x(t)');

end

function reverse(v1,t)

v1=fliplr(v1);

t=fliplr(-t);

fprintf('Resultant signal after reversing is: ');

disp(v1);

disp(t);

subplot(4,3,10);

stem(t,v1);

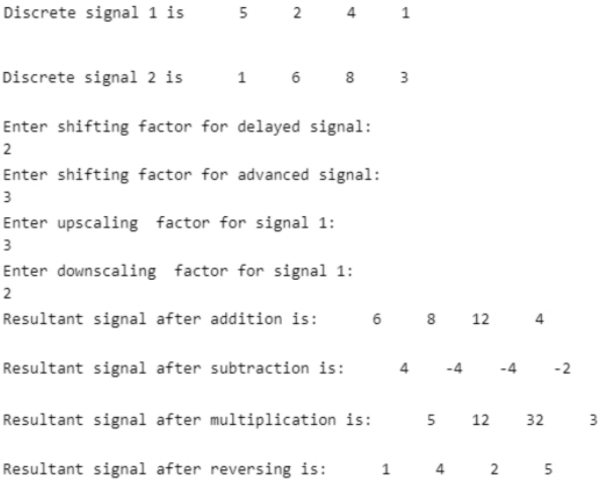
axis([t(1) t(end) -10 10]);

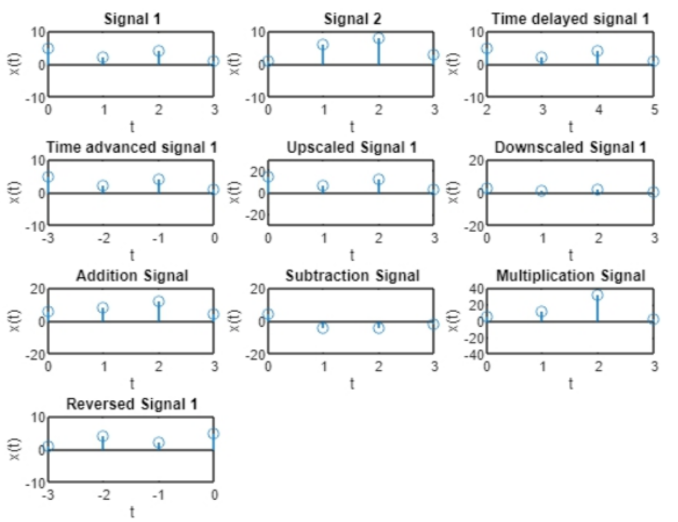
title('Reversed Signal 1');

xlabel('t');

ylabel('x(t)');

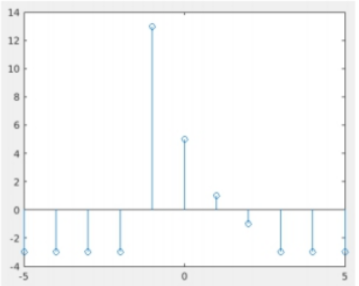
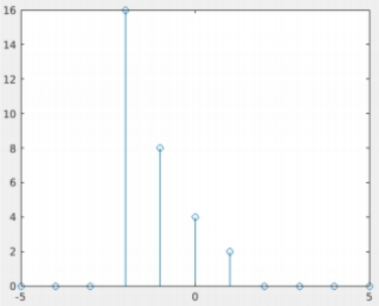
end



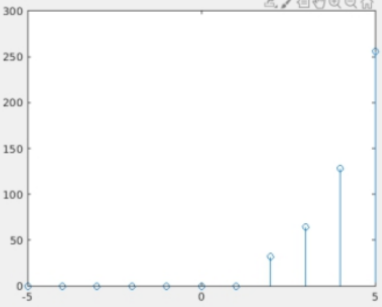


**Conclusion: -** We performed various operations mentioned on the discrete signals.

**Post Lab Questions**

1. Let x(n) = 8(0.5)n (u[n+1] - u[n-3]). Sketch the following signals
2. Y(n) = [x-3] 
3. F(n) = x[n+1]

1. G(n) = x[-n+4]



1. The process of conversion of continuous time signal into discrete time signal is known as quantization.
2. Which of the following is example of deterministic signal?
   1. Step
   2. Ramp
   3. Exponential
   4. All of the above

Answer: d.All of the above

1. For energy signals the energy will be finite and the average power will be zero
2. In a signal x(n), if ‘n’ is replaced by ‘n/3’ the it is called up-sampling.
3. The system y(n)=sin[x(n)] is
   1. Stable
   2. BIBO stable
   3. Unstable
   4. None of the above

Answer: b. BIBO stable