**Batch: B1 Roll No.: 16010121045**

**Experiment No. 2**

**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. 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.

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

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.

**Discrete time signals types:**Plotting various signals using matlab   
  
Code:  
t = -5:0.1:5;

u = @(t) (t >= 0);

r = @(t) t .\* u(t);

delta = @(t) t == 0;

expo = @(t) exp(t);

sinusoidal = @(t) sin(t);

cosinusoidal = @(t) cos(t);

subplot(3, 2, 1);

plot(t, u(t));

title('Unit Step');

subplot(3, 2, 2);

plot(t, r(t));

title('Unit Ramp');

subplot(3, 2, 3);

stem(t, delta(t));

title('Unit Impulse');

subplot(3, 2, 4);

plot(t, expo(t));

title('Exponential');

subplot(3, 2, 5);

plot(t, sinusoidal(t));

title('Sine');

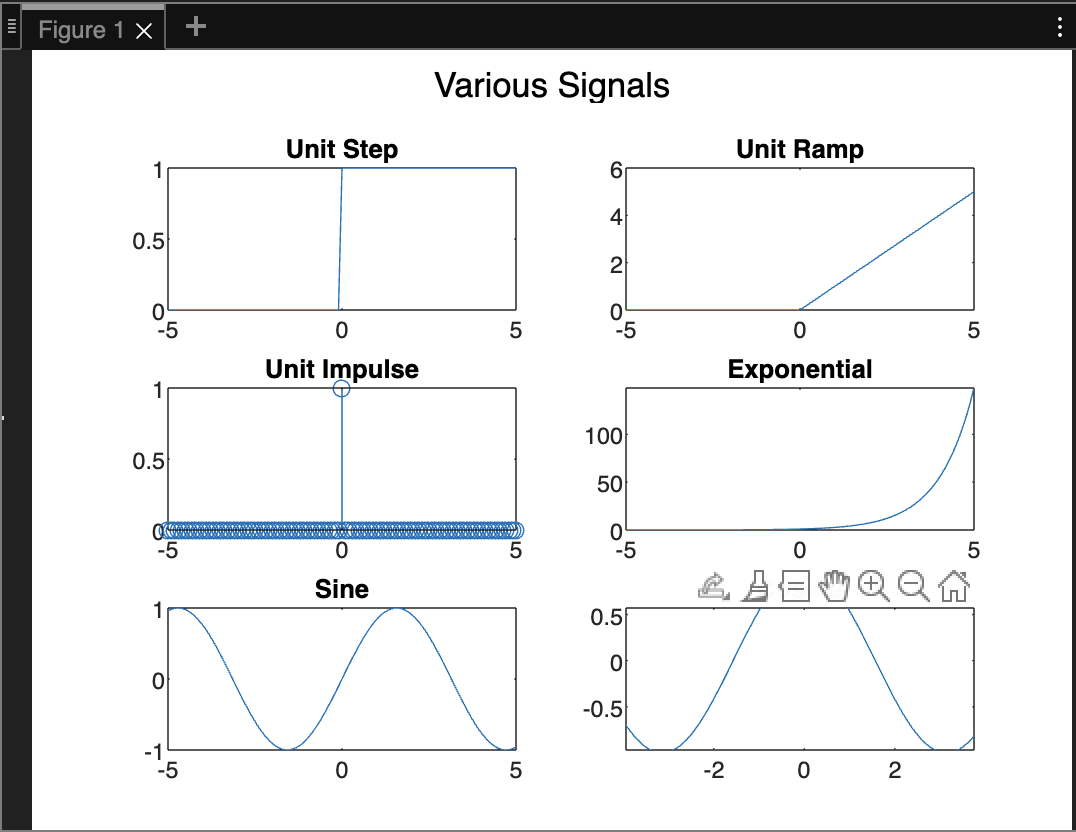
subplot(3, 2, 6);

plot(t, cosinusoidal(t));

title('Cosine');

sgtitle('Various Signals');

Output:



**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.
6. Folding

Code:

t = -5:0.1:5;

signal1 = sin(t);

signal2 = 0.5 \* cos(t);

addition\_result = signal1 + signal2;

subtraction\_result = signal1 - signal2;

multiplication\_result = signal1 .\* signal2;

upscaling\_factor = 2;

downscaling\_factor = 0.5;

upscaled\_signal = upscaling\_factor \* signal1;

downscaled\_signal = downscaling\_factor \* signal1;

shift\_amount = 2;

advanced\_signal = circshift(signal1, shift\_amount);

delayed\_signal = circshift(signal1, -shift\_amount);

folded\_signal = fliplr(signal1);

subplot(3, 3, 1);

plot(t, signal1);

title('Signal 1');

subplot(3, 3, 2);

plot(t, signal2);

title('Signal 2');

subplot(3, 3, 3);

plot(t, addition\_result);

title('Addition');

subplot(3, 3, 4);

plot(t, subtraction\_result);

title('Subtraction');

subplot(3, 3, 5);

plot(t, multiplication\_result);

title('Multiplication');

subplot(3, 3, 6);

plot(t, upscaled\_signal);

title('Upscaling');

subplot(3, 3, 7);

plot(t, downscaled\_signal);

title('Downscaling');

subplot(3, 3, 8);

plot(t, advanced\_signal);

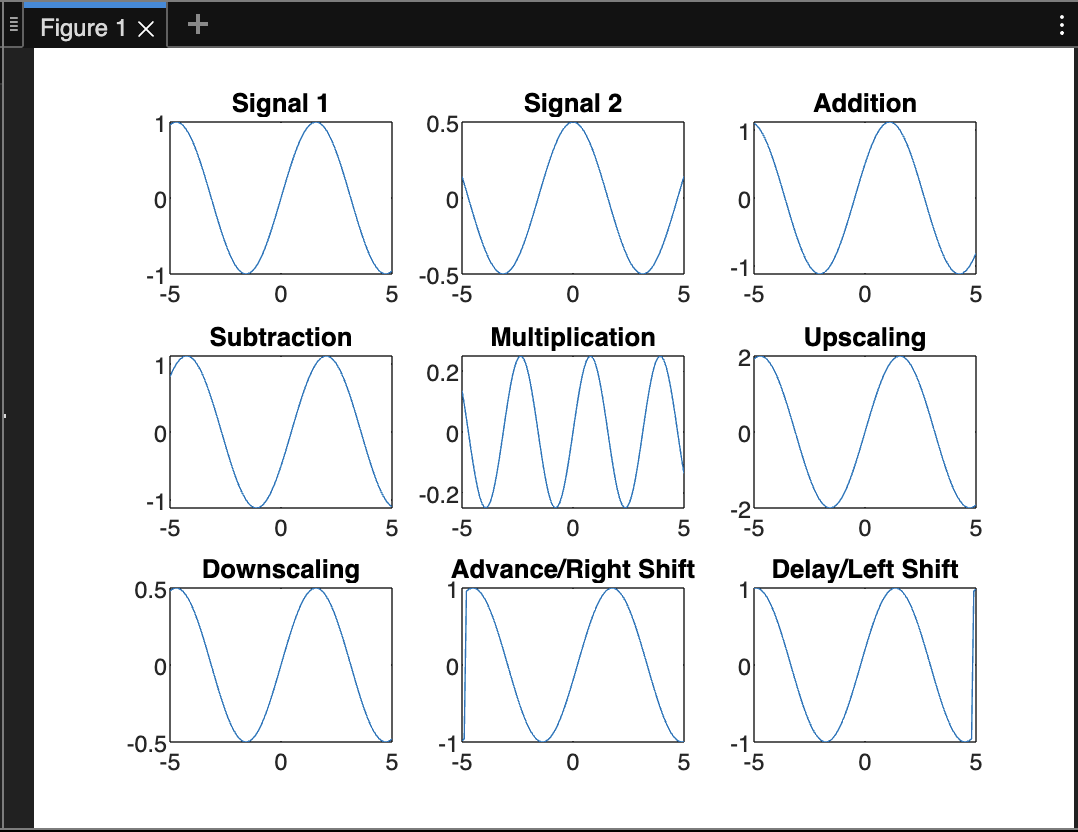
title('Advance/Right Shift');

subplot(3, 3, 9);

plot(t, delayed\_signal);

title('Delay/Left Shift');

OutPut:



**Conclusion:-**

With help of Matlab we were able to represent discrete time signals and perform

different operations on them

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

u = @(n) 1.\*(n>=0);

x = @(n) 8.\*((0.5).^n).\*(u(n+1) - u(n-3));

n = -10:1:10;

X = x(n);

subplot(3,3,2);

stem(n,X)

xlabel("Sample");

ylabel("Amplitude");

title("Original");

Y = x(n-3);

subplot(3,3,4);

stem(n,Y);

xlabel("Sample");

ylabel("Amplitude");

title("Y(n) = [x-3]");

F = x(n+1);

subplot(3,3,5);

stem(n,F);

xlabel("Sample");

ylabel("Amplitude");

title("F(n) = x[n+1]");

G = x(-n+4);

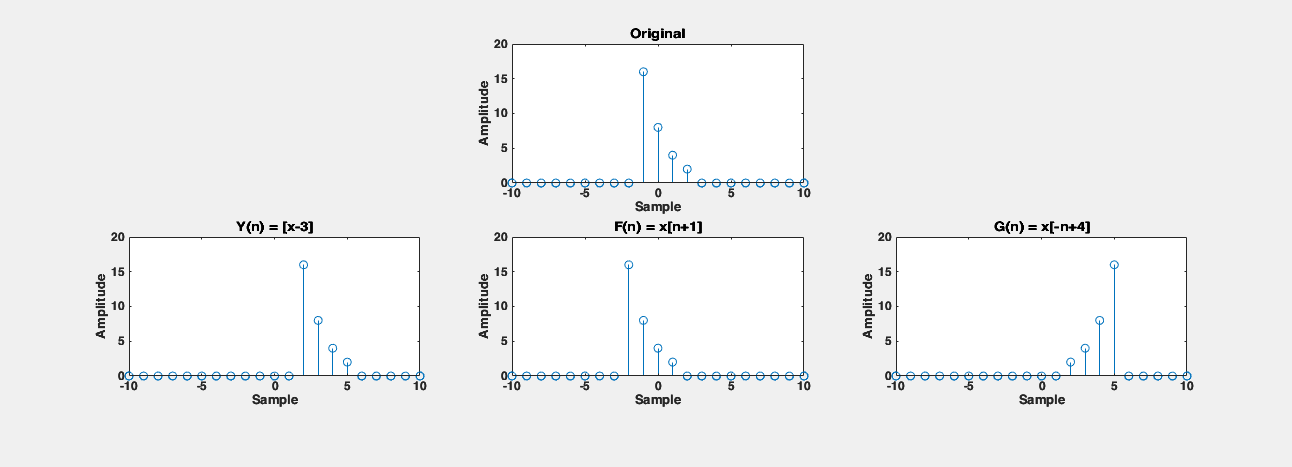
subplot(3,3,6);

stem(n,G);

xlabel("Sample");

ylabel("Amplitude");

title("G(n) = x[-n+4]");

****

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

**Ans: c.** Exponential

1. **For energy signals the energy will be finite and the average power will be** 0
2. **In a signal x(n), if ‘n’ is replaced by ‘n/3’ the it is called** Expansion
3. **The system y(n)=sin[x(n)] is**
   1. **Stable**
   2. **BIBO stable**
   3. **Unstable**
   4. **None of the above**

**Ans:** d. None of the above