2. Generation of Signals

Aim: To generate various signals

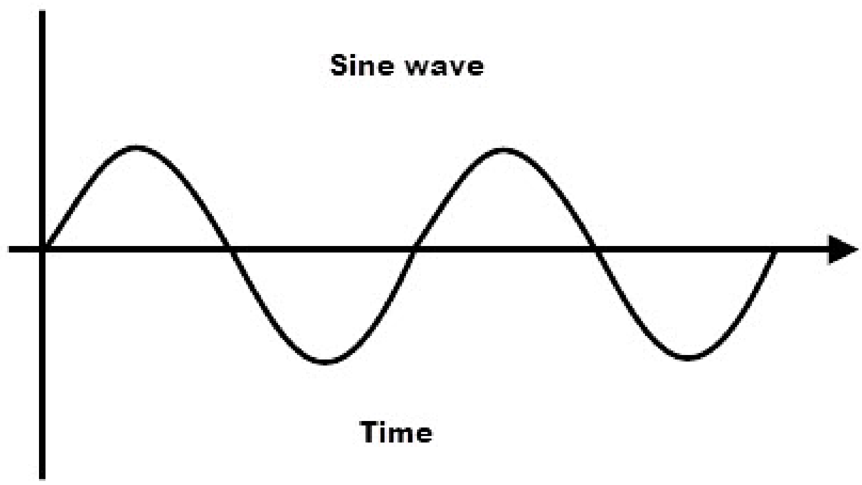
Apparatus: Computer, MATLAB software

Theory:

1. GENERATION OF SINUSOIDAL SIGNAL

Sinusoidal Signals are periodic functions which are based on the sine or cosine function from trigonometry

Model Graph:

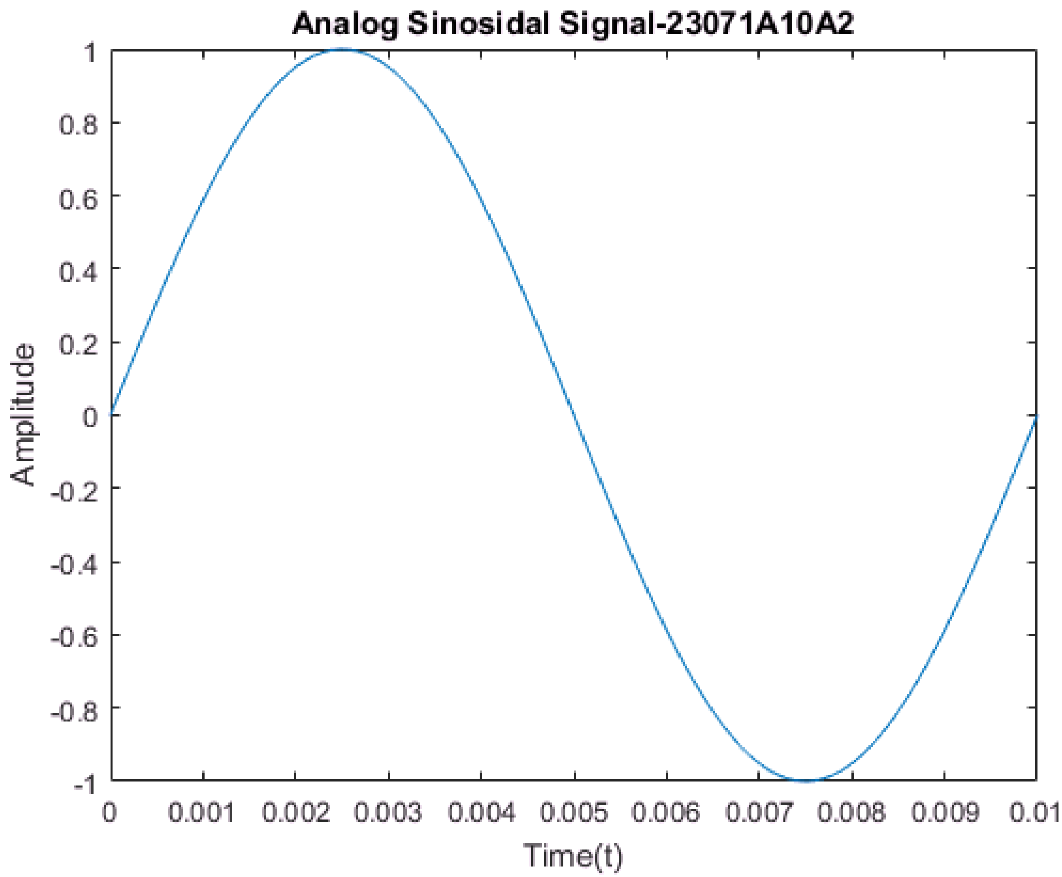


Mathematical Expression:

$y(t)=A \sin(\omega t+ \theta)$

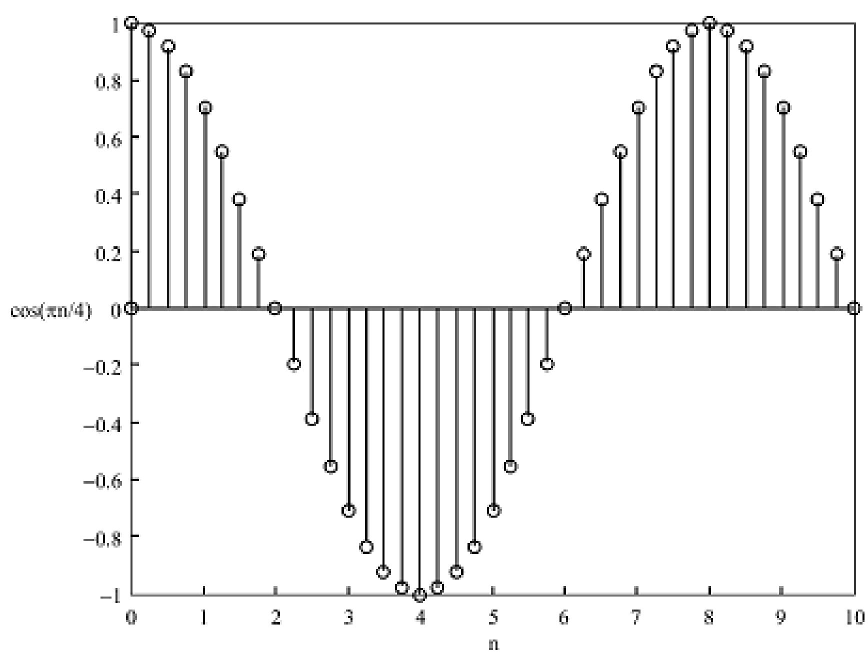
CODE:

```
f=100;
T=1/f;
t=0:T/f:T;
f=100;
T=1/f;
t=0:T/f:T;
x=sin(2*pi*f*t);
plot(t,x)
xlabel('Time(t)')
ylabel('Amplitude')
title('Analog Sinosidal Signal-23071A10A2')
```



2. GENERATION OF DISCRETE SINUSOIDAL SIGNAL

Model Graph:

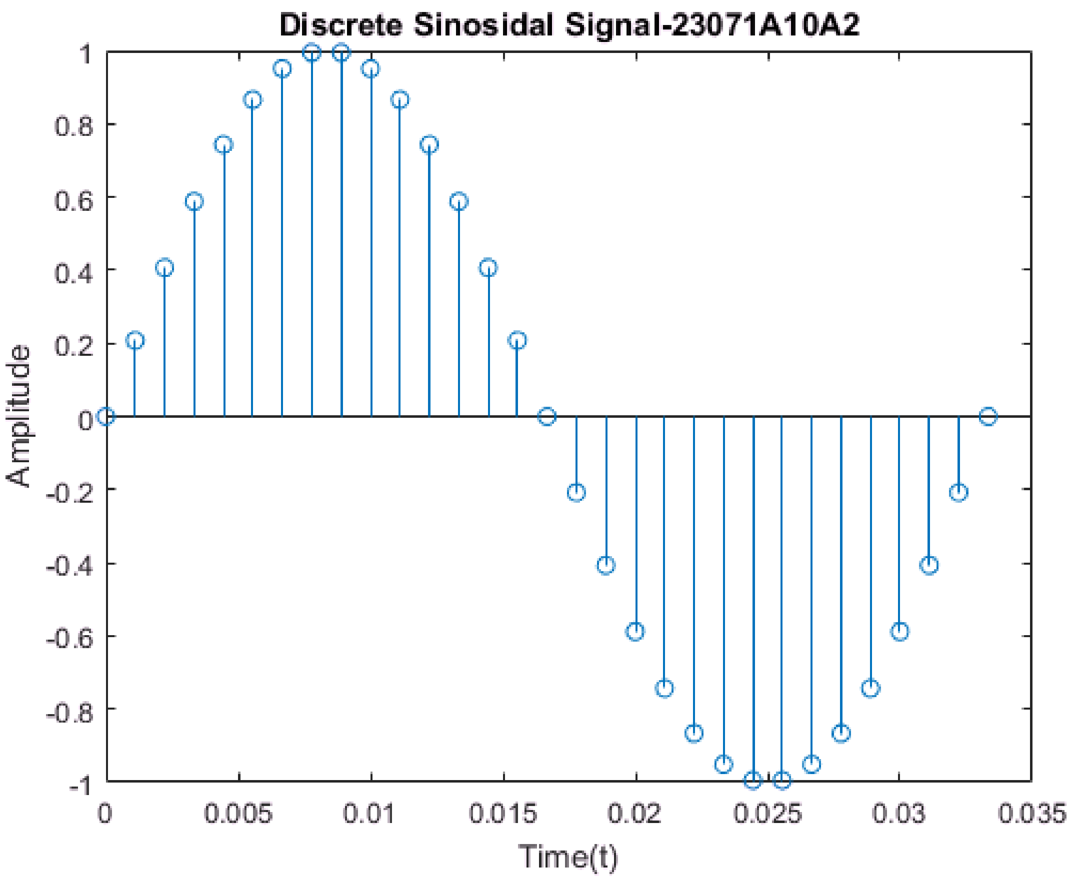


Mathematical Expression:

y(t)=A sin(2πfnt+ θ)

CODE:

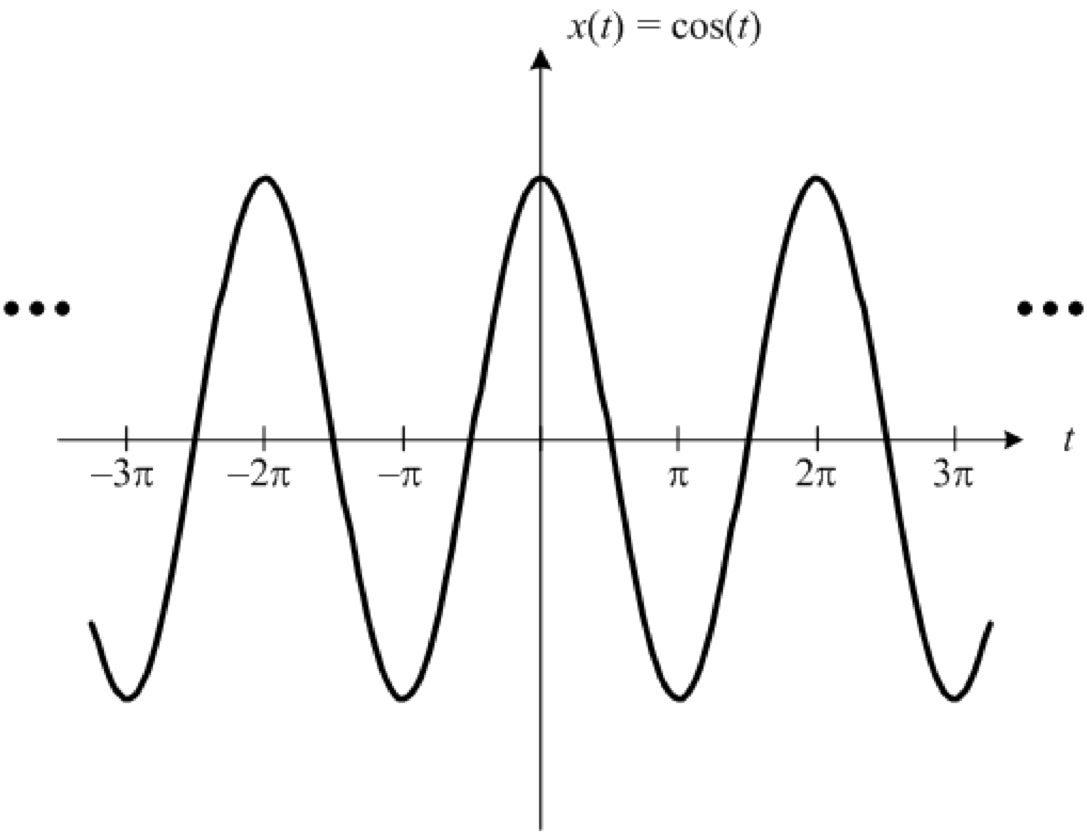
```
f=30;
T=1/f;
t=0:T/f:T;
T=1/f;
t=0:T/f:T;
x=sin(2*pi*f*t);
stem(t,x)
xlabel('Time(t)')
ylabel('Amplitude')
title('Discrete Sinosidal Signal-23071A10A2')
```



3. GENERATION OF COSINE SIGNAL

Cosinusoidal means when angle with respect to reference is 0, the maximum value or peak is obtained.
Cosinusoidal means when angle with respect to reference is 0, the maximum value or peak is obtained.

Model Graph:

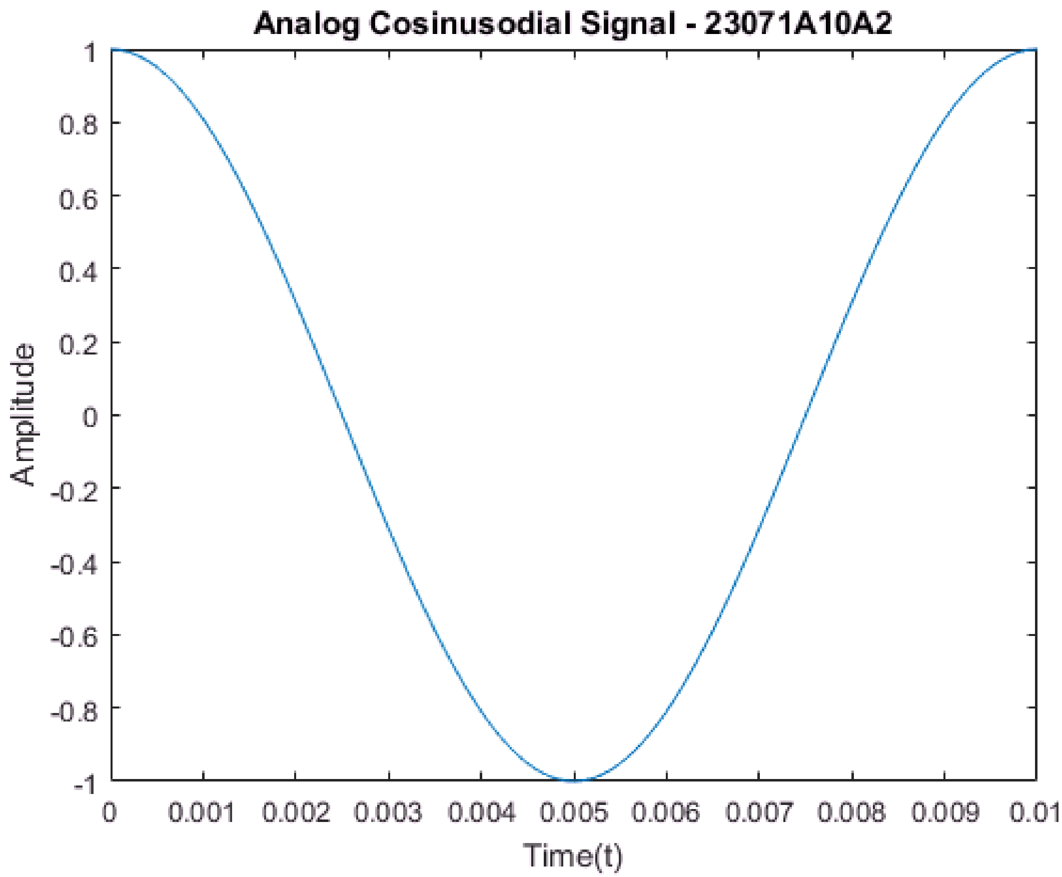


Mathematical Expression:

y = a cos(bx + c)

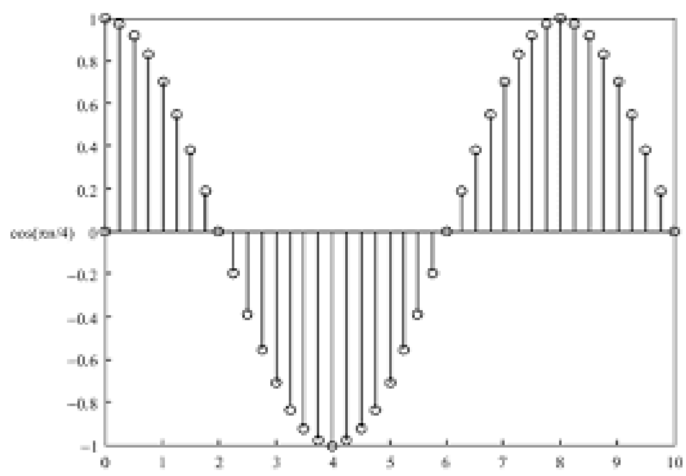
CODE:

```
f=100;
T=1/f;
t=0:T/f:T;
x=cos(2*pi*f*t);
plot(t,x)
xlabel('Time(t)')
ylabel('Amplitude')
title('Analog Cosinusodial Signal - 23071A10A2')
```



4. GENERATION OF DISCRETE COSINE SIGNAL

Model Graph:



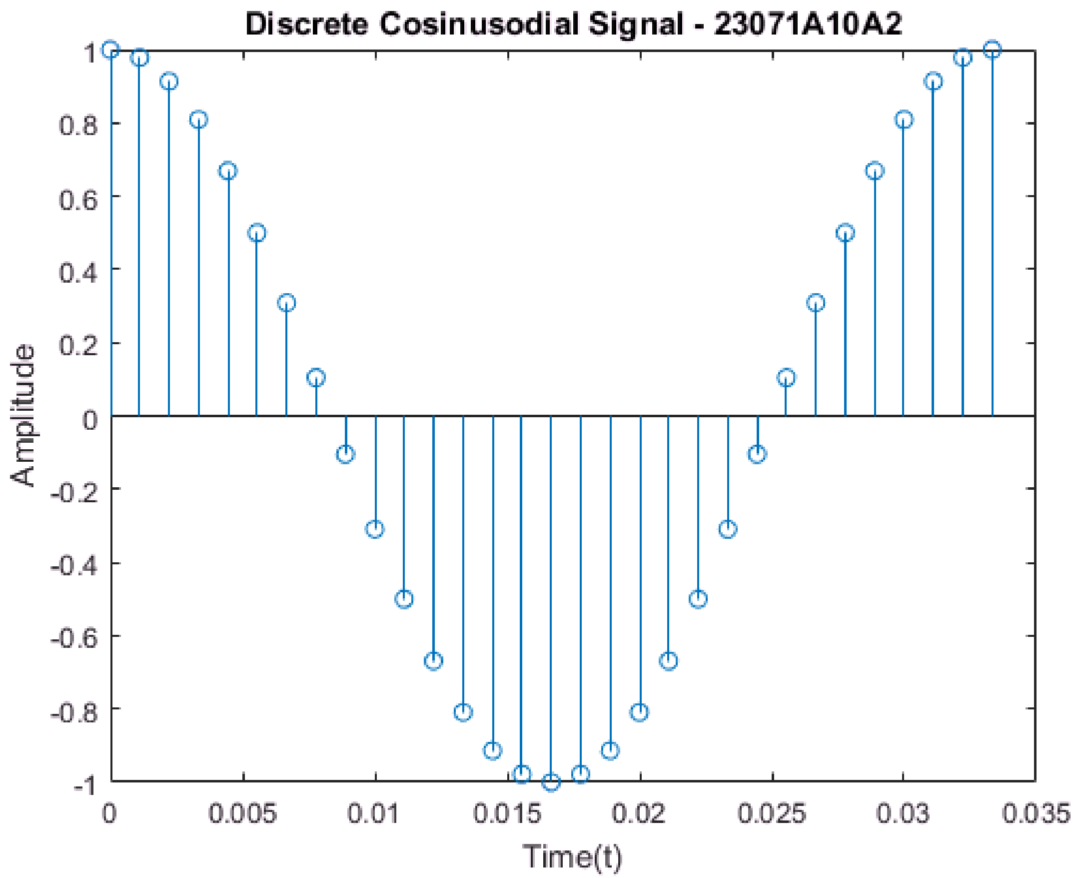
Mathematical Expression:

y(t)=A cos(2πfnt+ θ)

CODE:

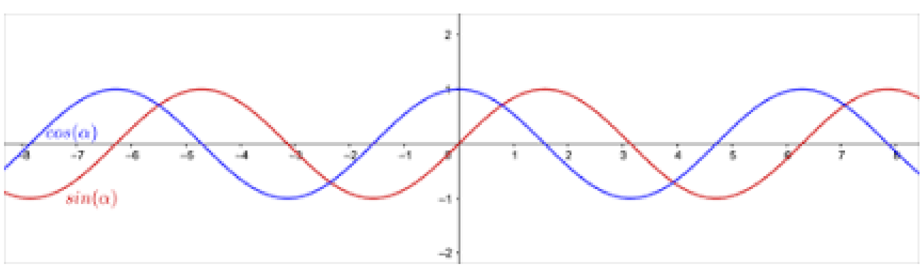
```
f=30;
T=1/f;
t=0:T/f:T;
x=cos(2*pi*f*t);
```

```
stem(t,x)
xlabel('Time(t)')
ylabel('Amplitude')
title('Discrete Cosinusodial Signal - 23071A10A2')
```



5. GENERATION TWO SIGNALS ON THE SAME AXIS

<u>Model Graph</u>:



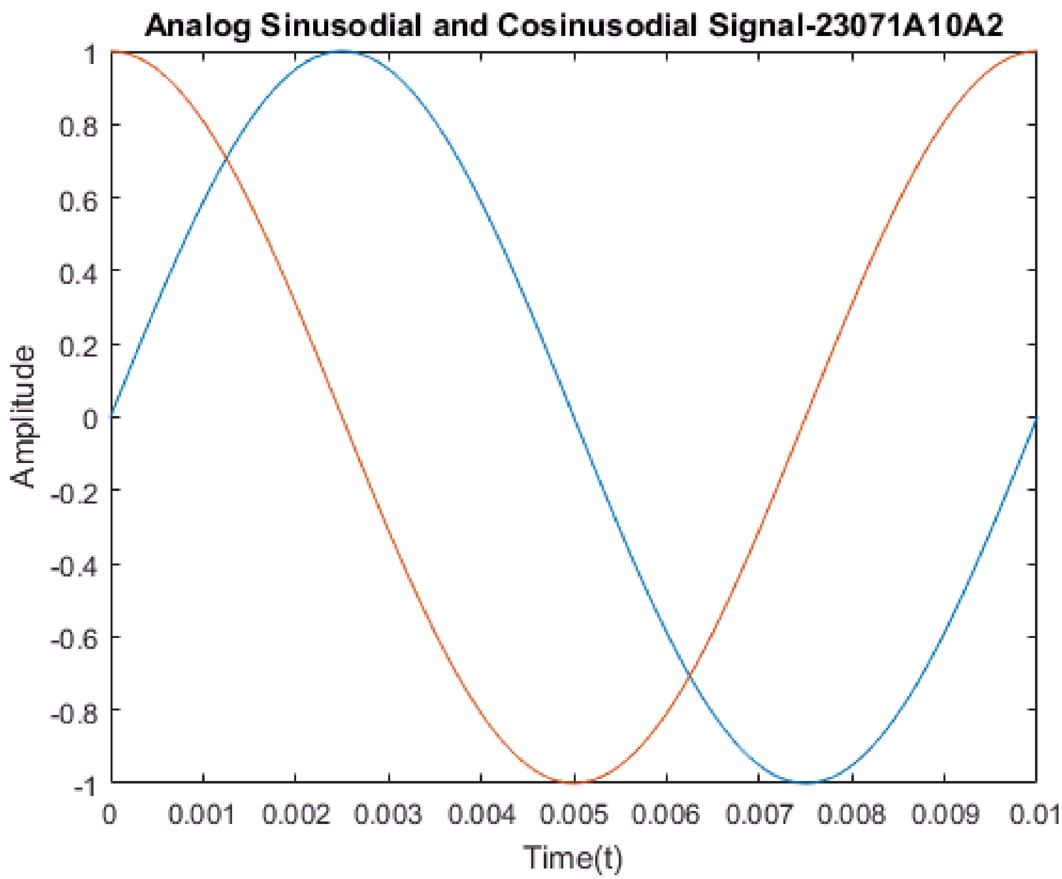
<u>Mathematical Expression</u>:

y = A sin (x)+Bcos (x)

<u>CODE</u>:

```
f=100;
T=1/f;
t=0:T/f:T;
x=sin(2*pi*f*t);
y=cos(2*pi*f*t);
plot(t,x,t,y)
```

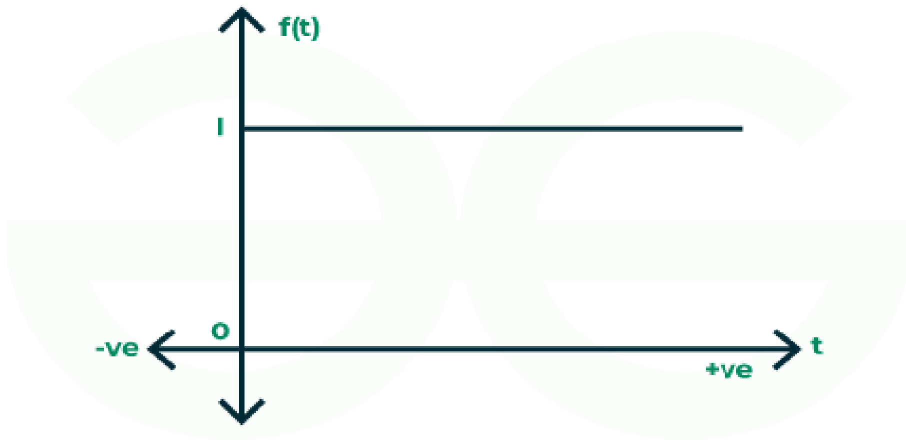
```
xlabel('Time(t)')
ylabel('Amplitude')
title('Analog Sinusodial and Cosinusodial Signal-23071A10A2')
```



6. GENERATION OF UNIT STEP SIGNAL

A unit step function is a mathematical function that switches on at time t=0 and is represented by the values 1 for t≥0 and 0 for t<0. It is commonly used in computer science to represent an idealized switch and can be multiplied by other functions to switch them on at t=0.

Model Graph:



Unit Step Signal in control system

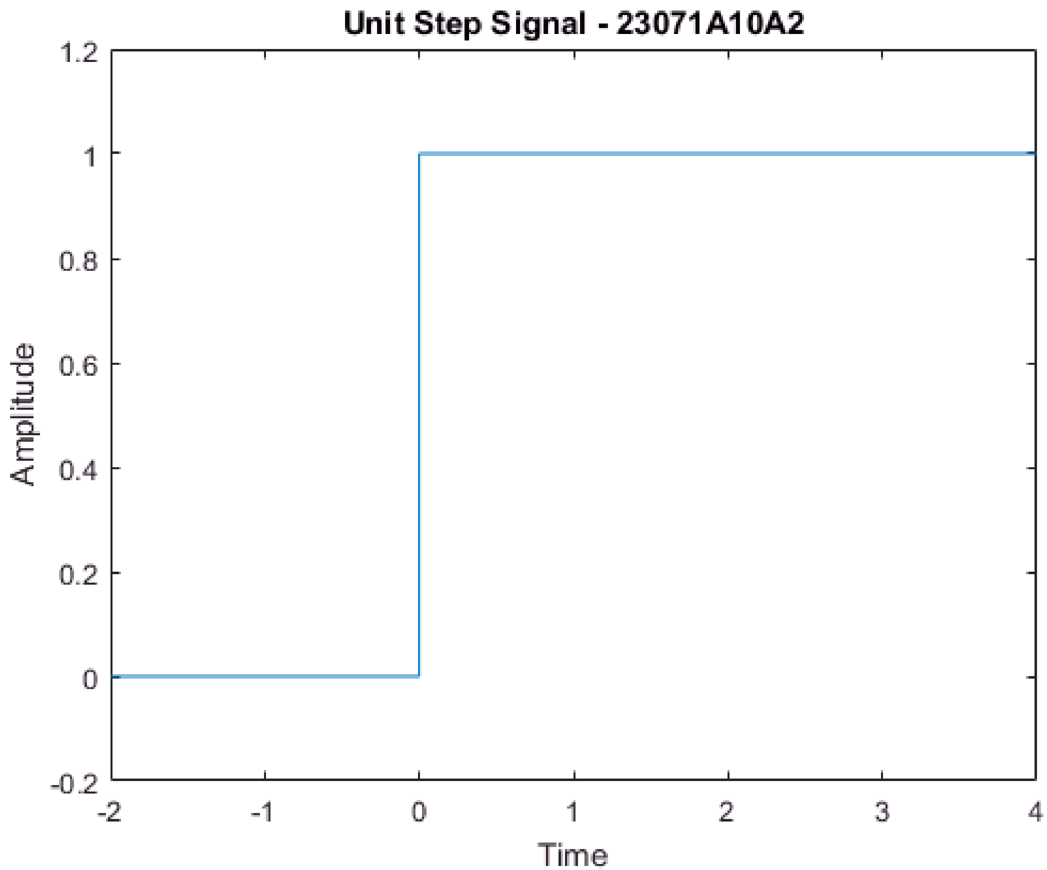
Mathematical Expression:

u(t) = 1 for 0<=t<=1

u(t) =0 for t<0

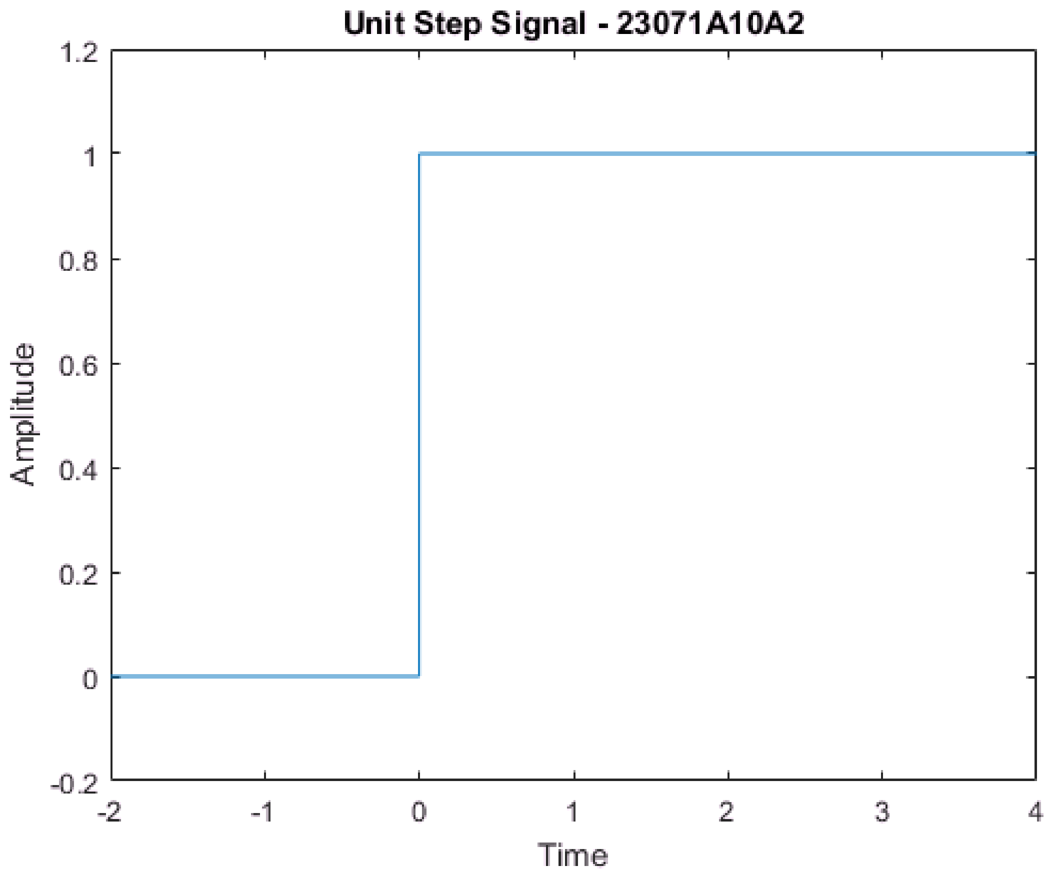
CODE-1:

```
t = -2:0.001:4;
x=zeros(1,length(t));
for i =1:length(t);
    if t(i) >= 0;
        x(i) = 1;
    else
        x(i) = 0;
    end
end

plot(t,x)
axis([-2 4 -0.2 1.2])
xlabel('Time')
ylabel('Amplitude')
title('Unit Step Signal - 23071A10A2')
```



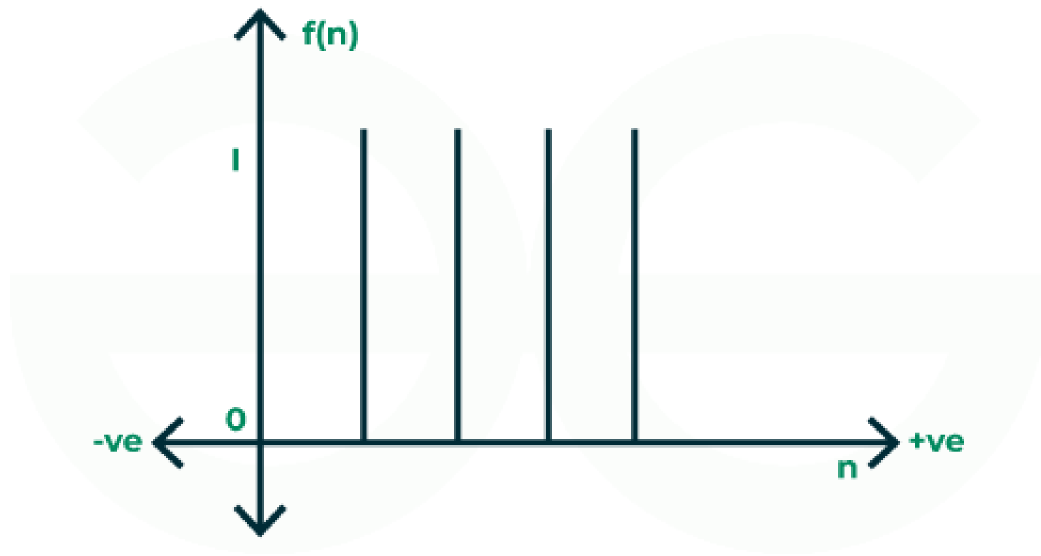
CODE-2:

```
 t = -2:0.001:4;
x=zeros(1,length(t));
x = (t>=0);

plot(t,x)
axis([-2 4 -0.2 1.2])
xlabel('Time')
ylabel('Amplitude')
title('Unit Step Signal - 23071A10A2')
```



7. Discrete Unit Step Function

Model Graph:



Unit Step Signal in control system

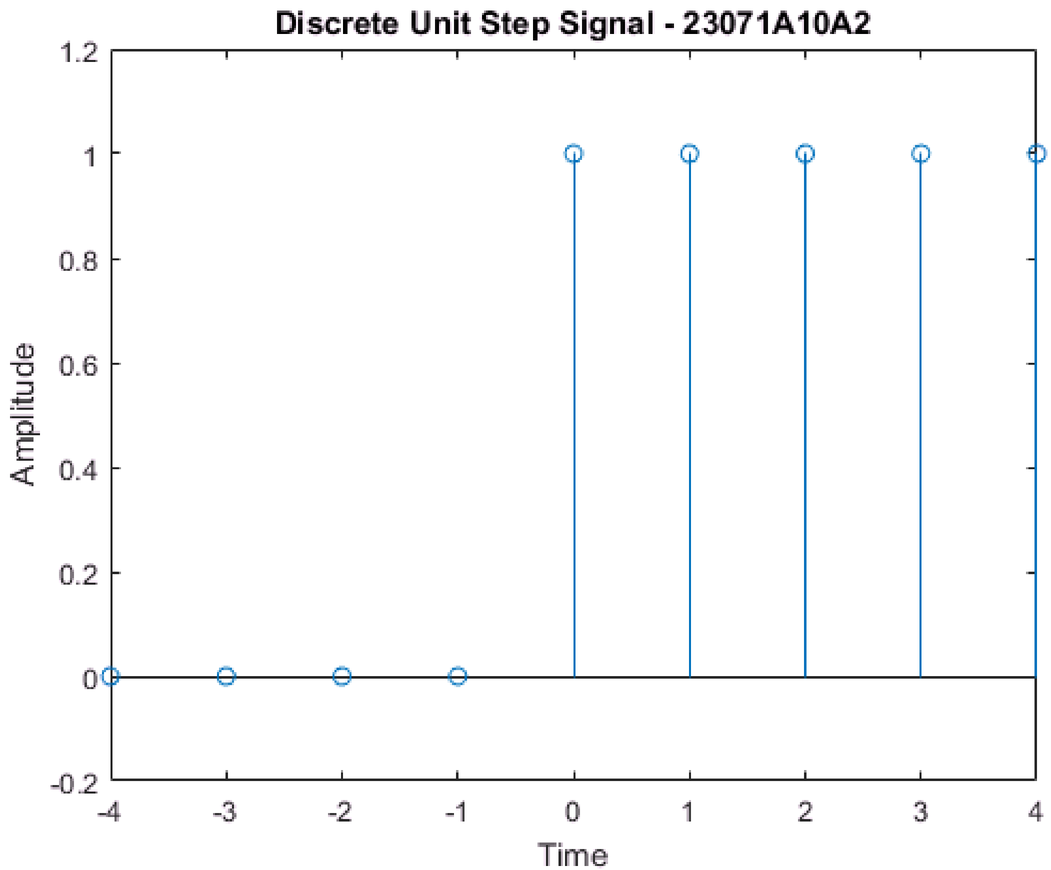
Mathematical Expression:

u(t) = 1 for t>=0

u(t) 0 for t<0

CODE:

```
    t = -4:1:4;
x=zeros(1,length(t));
for i =1:length(t);
    if t(i) >= 0;
        x(i) = 1;
    else
        x(i) = 0;
    end
end

stem(t,x)
axis([-4 4 -0.2 1.2])
xlabel('Time')
ylabel('Amplitude')
title('Discrete Unit Step Signal - 23071A10A2')
```



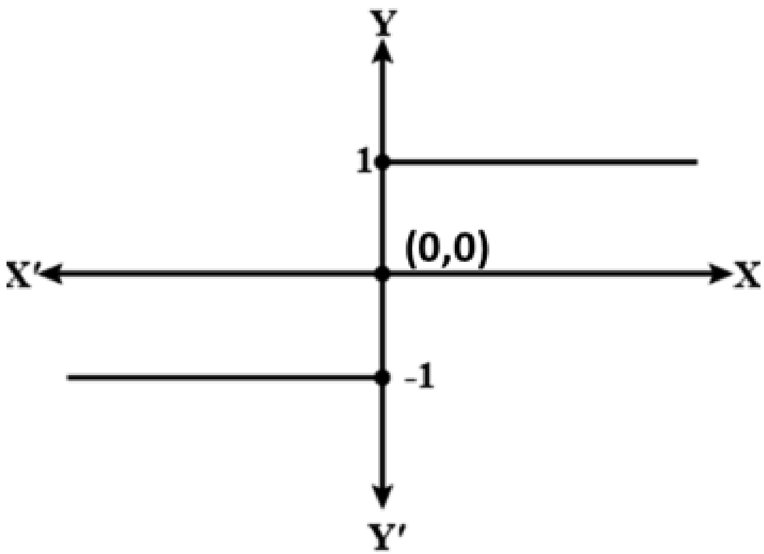
8. GENERATION OF SIGNUM FUNCTION

The signum function simply gives the sign for the given values of x. For x value greater than zero, the value of the output is +1, for x value lesser than zero, the value of the output is -1, and for x value equal to zero, the output is equal to zero.

Model Graph:

$$Sgn(x) = \begin{cases} -1\ , & if\ x < 0 \\ 0, & if, x = 0 \\ 1, & if, x > 0 \end{cases}$$

Graph of Signum Function



Mathematical Expression:

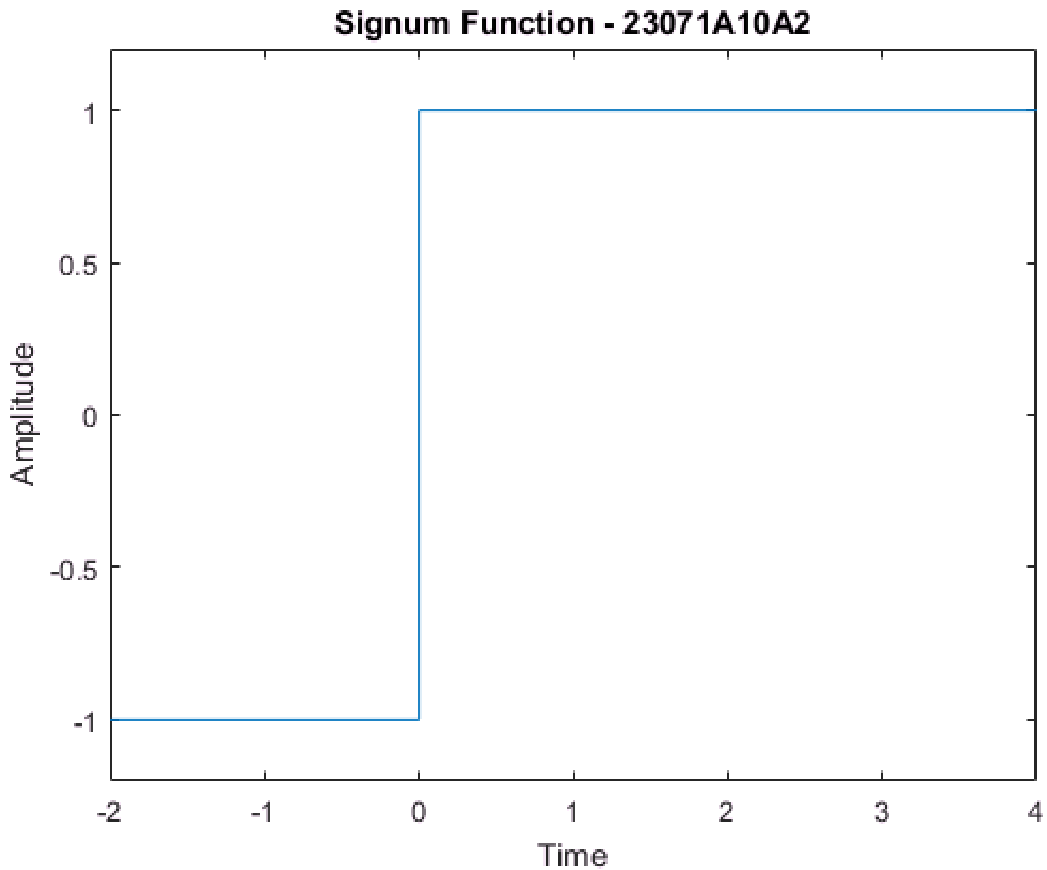
sgn(t) = 1 for t>0

sgn(t) = 0 for t=0

sgn(t) = -1 for t<0

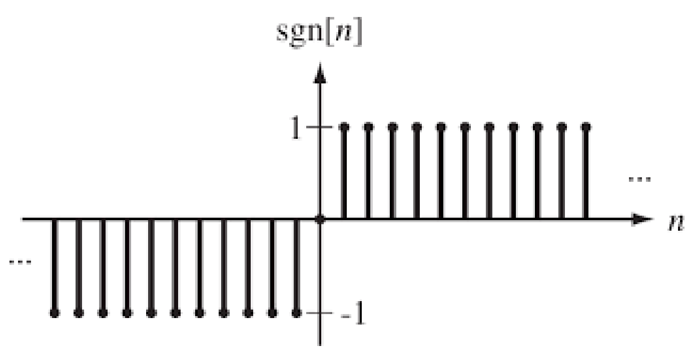
CODE :

```
t = -2:0.001:4;
   x=zeros(1,length(t));
   for i =1:length(t);
       if t(i) >= 0;
           x(i) = 1;
       else
           x(i) = -1;
       end
   end

   plot(t,x)
   axis([-2 4 -1.2 1.2])
   xlabel('Time')
   ylabel('Amplitude')
   title('Signum Function - 23071A10A2')
```



9. Discrete Signum Fuinction:

<u>Model Graph</u>:



<u>Mathematical Expression</u>:

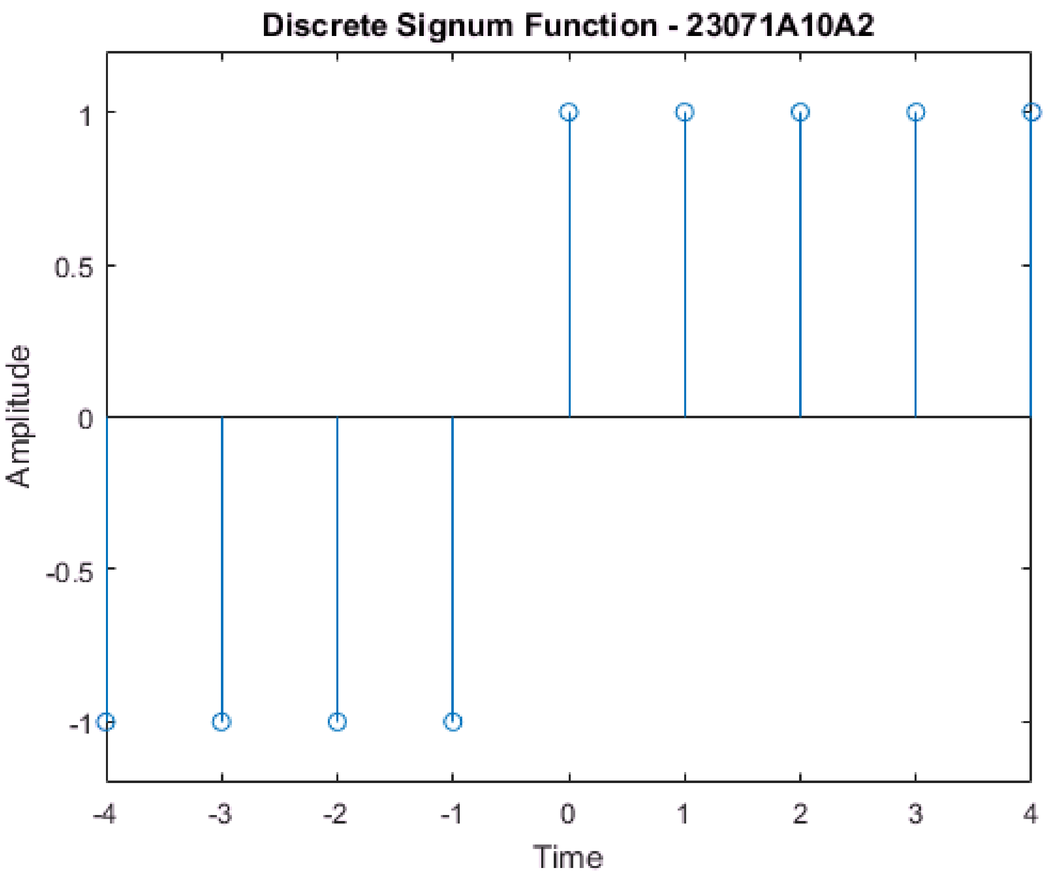
sgn(t) = 1 for t>0

sgn(t) = 0 for t=0

sgn(t) = -1 for t<0

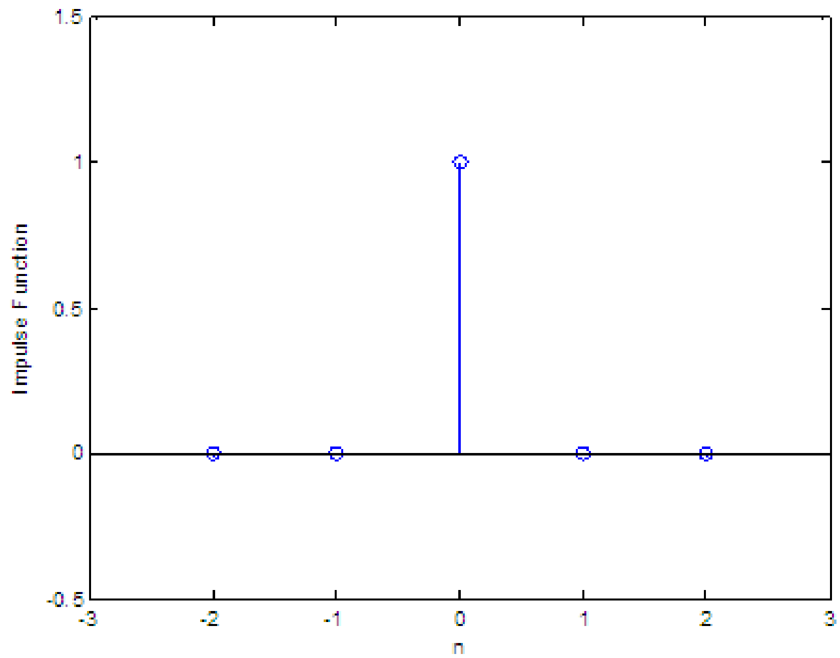
```
 t = -4:1:4;
x=zeros(1,length(t));
for i =1:length(t);
    if t(i) >= 0;
        x(i) = 1;
    else
```

```
            x(i) = -1;
        end
    end

    stem(t,x)
    axis([-4 4 -1.2 1.2])
    xlabel('Time')
    ylabel('Amplitude')
    title('Discrete Signum Function - 23071A10A2')
```



10. GENERATION OF IMPULSE FUNCTION:

Model Graph:

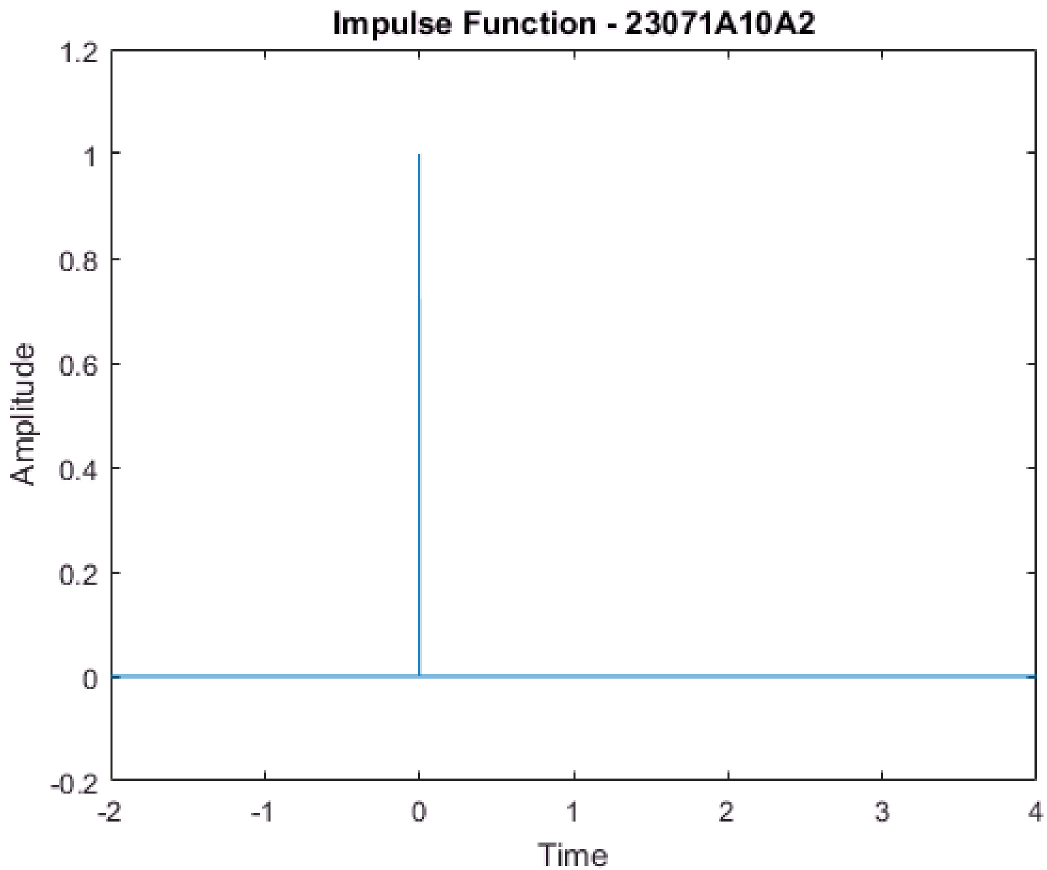


<u>Mathematical Expression</u>:

f(t) = 1 for t=0

<u>CODE-1</u>:

```
t = -2:0.001:4;
   x=zeros(1,length(t));
   for i =1:length(t);
       if t(i) == 0;
           x(i) = 1;
       else
           x(i) = 0;
       end
   end

   plot(t,x)
   axis([-2 4 -0.2 1.2])
   xlabel('Time')
   ylabel('Amplitude')
   title('Impulse Function - 23071A10A2')
```

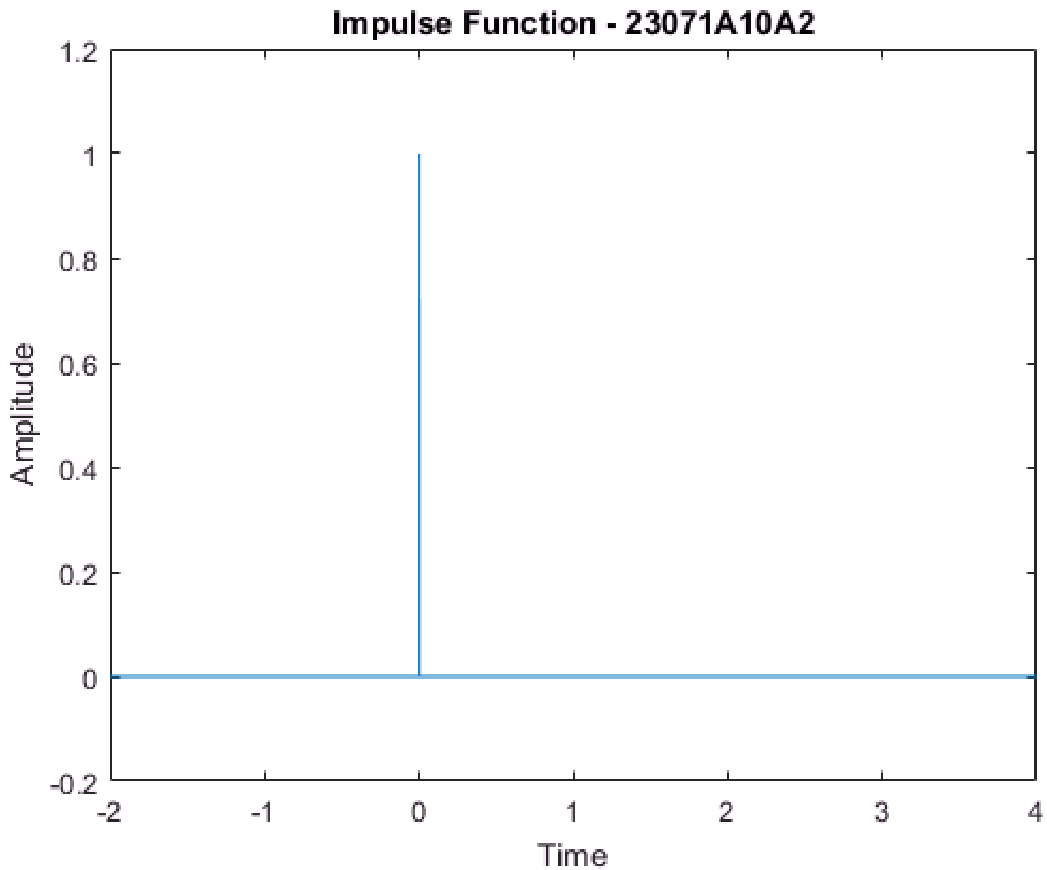


CODE-2:

```
 t = -2:0.001:4;
x=zeros(1,length(t));
x=(t==0)
```

```
x =
  Columns 1 through 1666
    ⋮
```

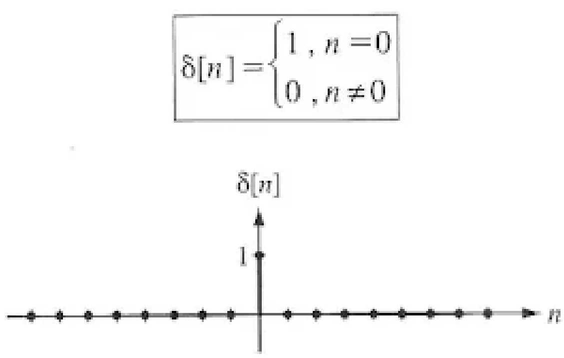
```
plot(t,x)
axis([-2 4 -0.2 1.2])
xlabel('Time')
ylabel('Amplitude')
title('Impulse Function - 23071A10A2')
```



11. Discrete Impulse Function:

The unit impulse function or Dirac delta function, denoted δ(t), is usually taken to mean a rectangular pulse of unit area, and in the limit the width of the pulse tends to zero whilst its magnitude tends to infinity.

<u>Model Graph</u>:



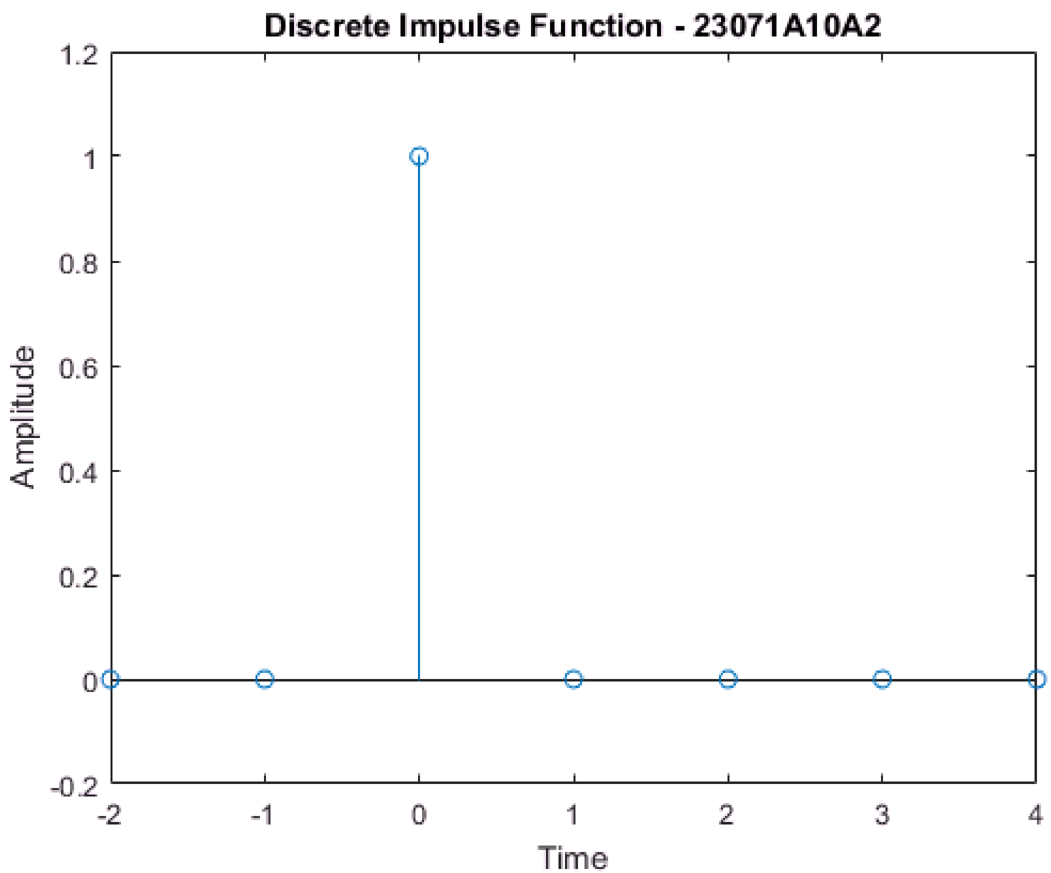
<u>Mathematical Expression</u>:

f(t) = 1 for t=0

<u>CODE</u>:

```
t = -2:1:4;
   x=zeros(1,length(t));
   for i =1:length(t);
```

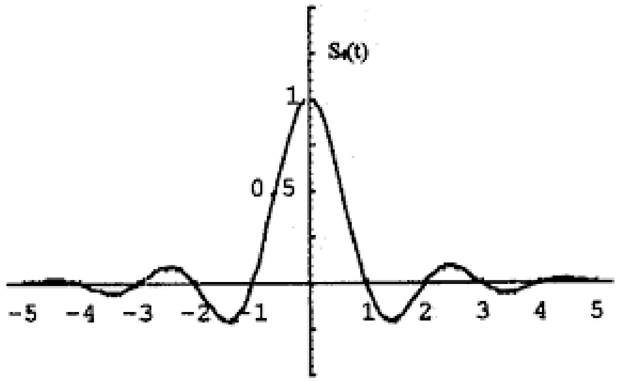
```
        if t(i) == 0;
            x(i) = 1;
        else
            x(i) = 0;
        end
    end

    stem(t,x)
    axis([-2 4 -0.2 1.2])
    xlabel('Time')
    ylabel('Amplitude')
    title('Discrete Impulse Function - 23071A10A2')
```



12. GENERATION OF SAMPLING OR SINC FUNCTION:

A 'Sampling Function' is a bandlimited function used in the context of the sampling theorem, typically defined within a frequency interval. It is a key component in the Fourier expansion of signal functions in the field of Computer Science.

Model Graph:

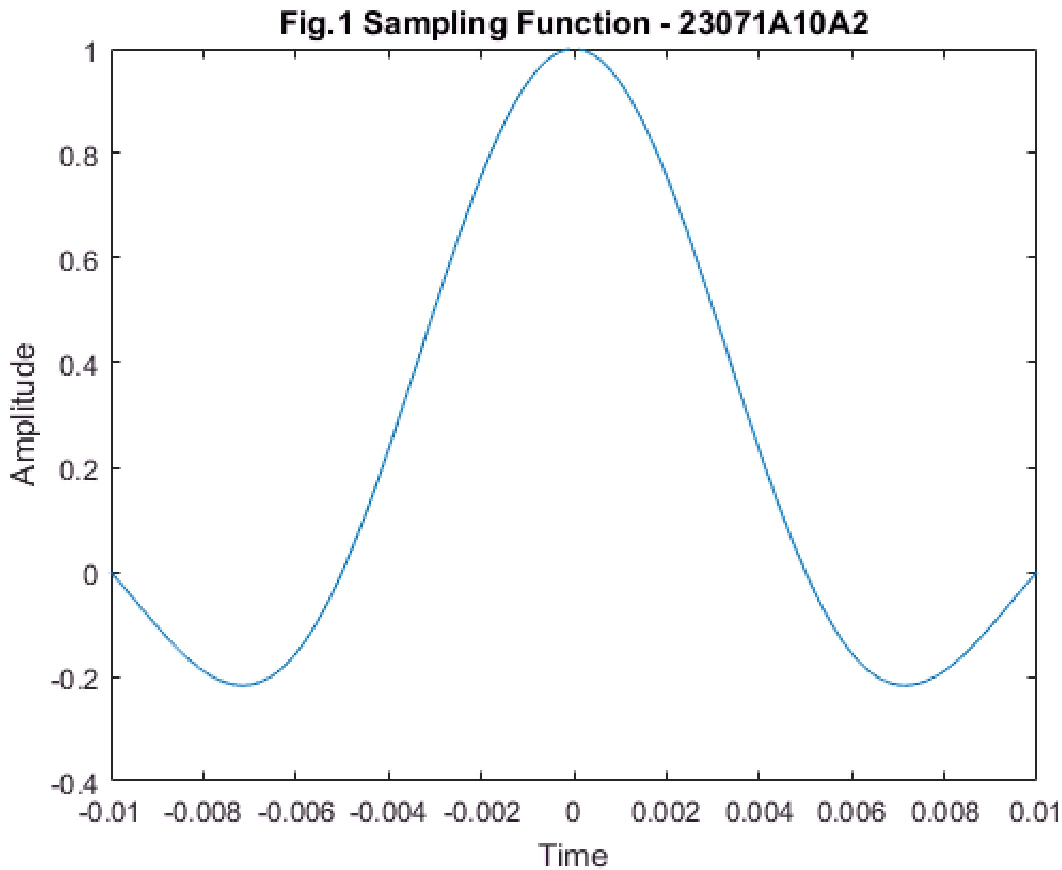


Mathematical Expression:

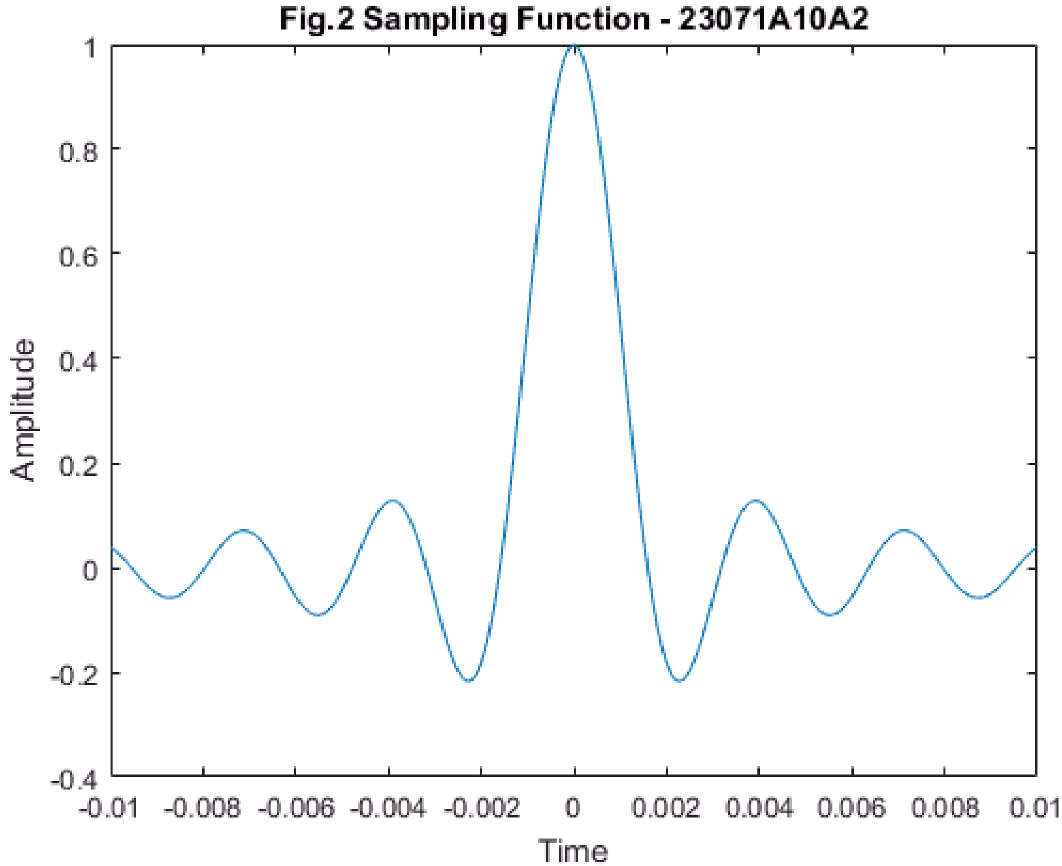
sa(t) or sinc(t) = sin(t) / t ; for all t values

CODE:

```
f = 100;
T = 1/f;
t = -T:T/f:T;
x1 = sin(2*pi*f*t);
x2 = 2*pi*f*t;
x3 = x1./x2;
y = sinc(2*pi*f*t);

plot(t,x3)
xlabel('Time')
ylabel('Amplitude')
title('Fig.1 Sampling Function - 23071A10A2')
```

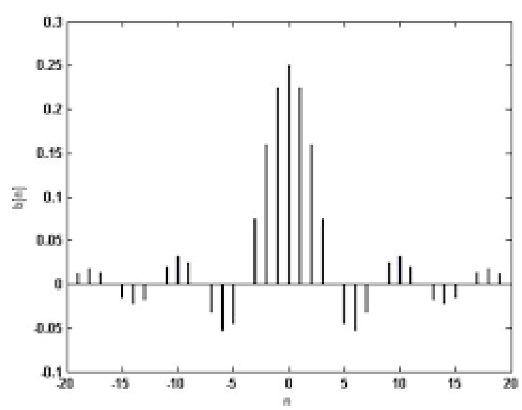



```
plot(t,y)
xlabel('Time')
ylabel('Amplitude')
title('Fig.2 Sampling Function - 23071A10A2')
```



13. DISCRETE SINC FUNCTION:

<u>Model Graph</u>:



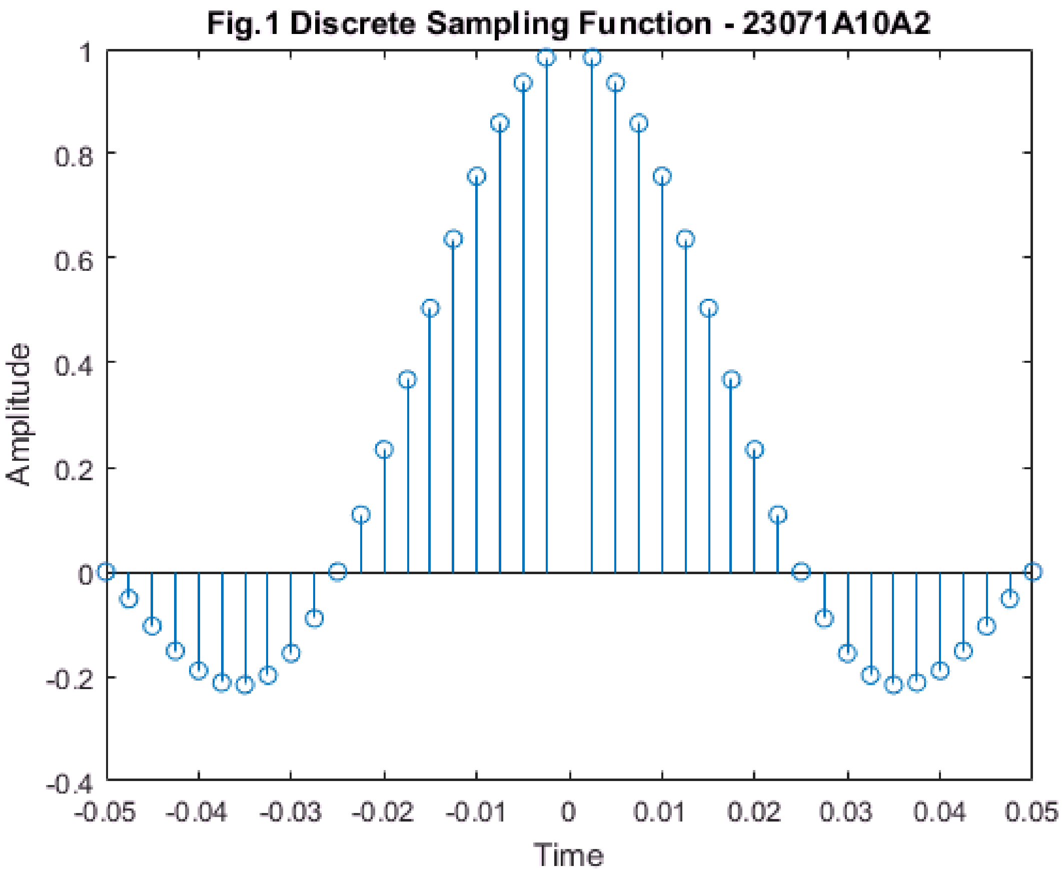
<u>Mathematical Expression</u>:

sa(t) or sinc(t) = sin(t) / t ; for all t values

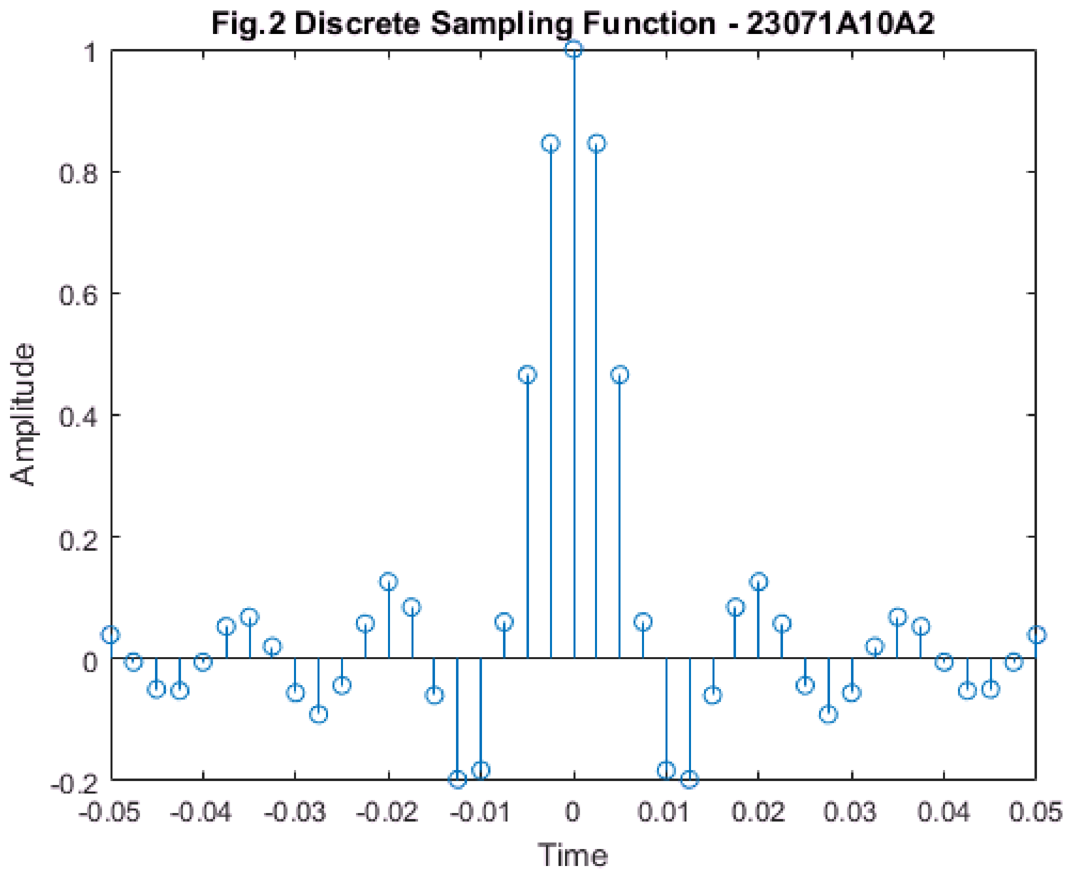
<u>CODE</u>:

```
f = 20;
T = 1/f;
t = -T:T/f:T;
```

```
x1 = sin(2*pi*f*t);
x2 = 2*pi*f*t;
x3 = x1./x2;
y = sinc(2*pi*f*t);

stem(t,x3)
xlabel('Time')
ylabel('Amplitude')
title('Fig.1 Discrete Sampling Function - 23071A10A2')
```



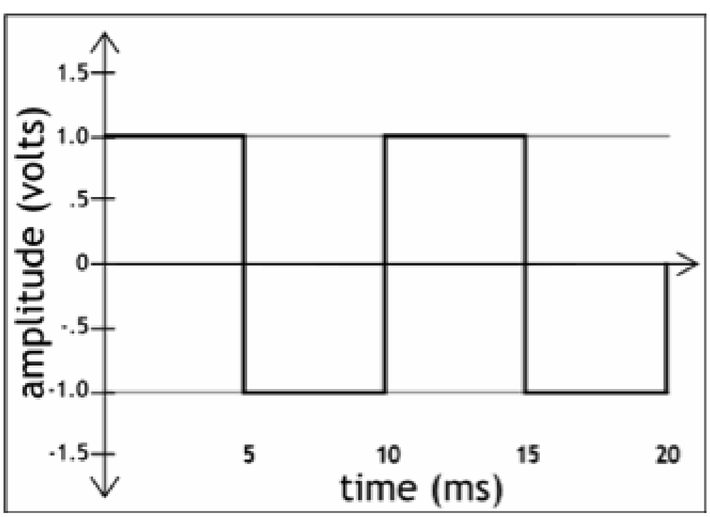

```
stem(t,y)
xlabel('Time')
ylabel('Amplitude')
title('Fig.2 Discrete Sampling Function - 23071A10A2')
```



14. SQUARE OR RECTANGLE SIGNAL:

A square wave is a type of waveform that alternates between a high and a low state in a very abrupt, straight-line manner, resembling the outline of a square if graphed. It differs from the smooth oscillations of a sine wave because it transitions sharply and instantly from one value to another. This waveform is commonly used in digital electronics and signal processing as it clearly represents binary conditions: on or off, high or low, 1 or 0. Square waves are essential in clocks and timing circuits where precise, clear transitions are needed to maintain the timing and operation of digital systems.

Model Graph:



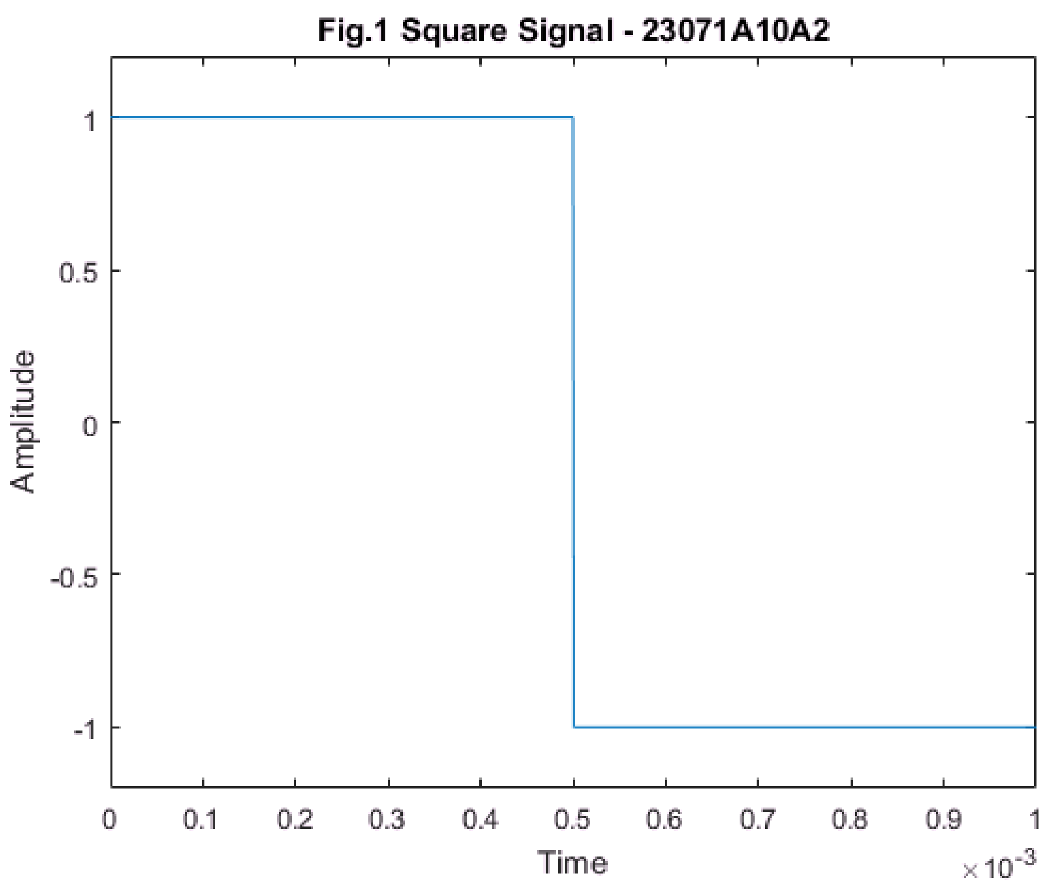
Mathematical Expression:

R(t) = 1; 0<=t<=T

R(t) =-1; For t<0

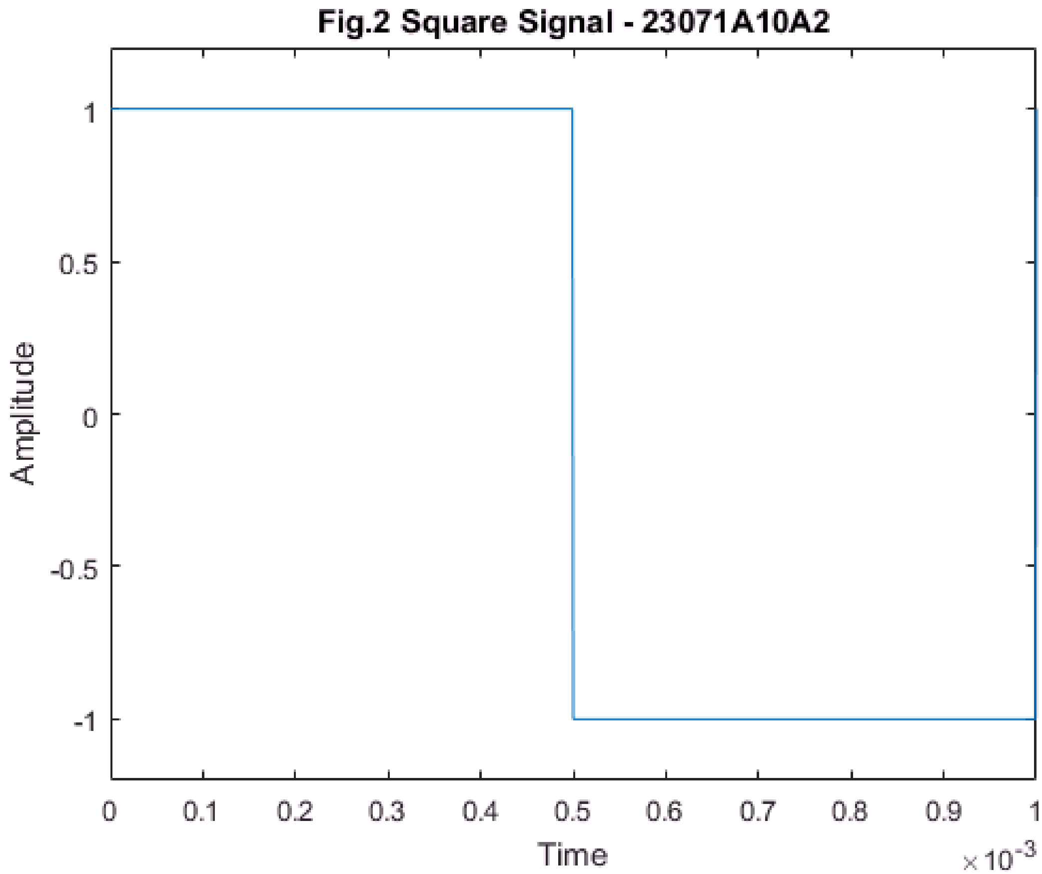
CODE:

```
clear all
f = 1000;
T = 1/f;
ts = 0:T/f:T;
for i = 1:length(ts);
    if ts(i) >= 0 && ts(i) <= 0.5*T;
        xs(i) = 1;
    else
        xs(i) = -1;
    end
end
ys = square(2*pi*f*ts);

plot(ts,xs)
axis([0 T -1.2 1.2])
xlabel('Time')
ylabel('Amplitude')
title('Fig.1 Square Signal - 23071A10A2')
```



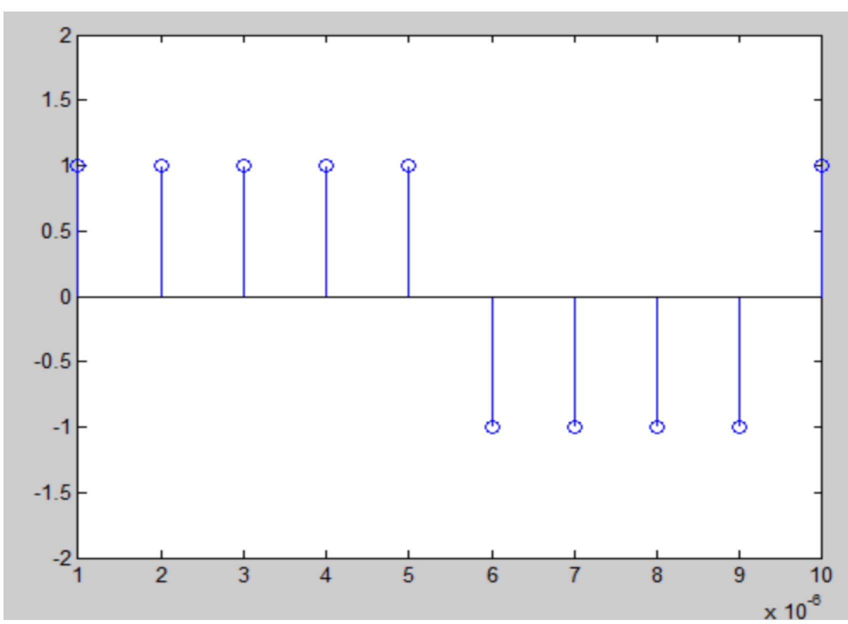

```
plot(ts,ys)
axis([0 T -1.2 1.2])
xlabel('Time')
```

```
ylabel('Amplitude')
title('Fig.2 Square Signal - 23071A10A2')
```



15. DISCRETE SQUARE OR RECTANGLE SIGNAL:

<u>Model Graph</u>:



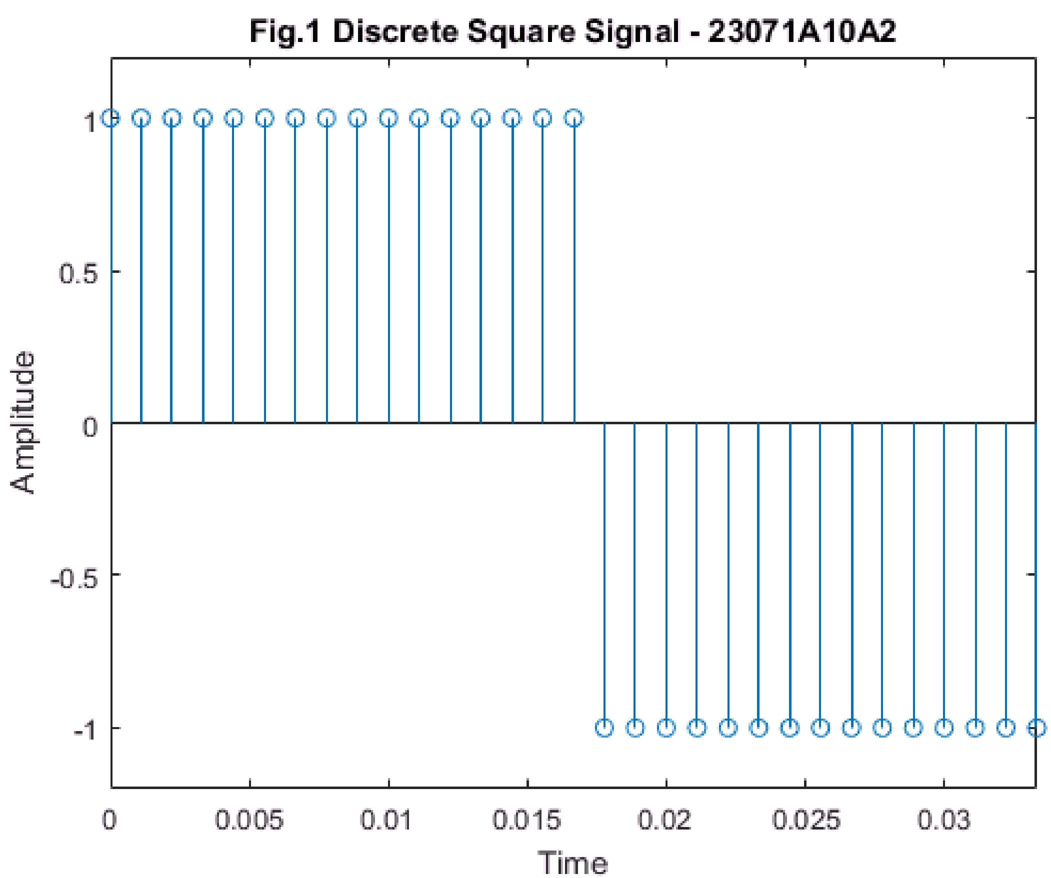
<u>Mathematical Expression</u>:

R(t) = 1; For 0<=t<=T

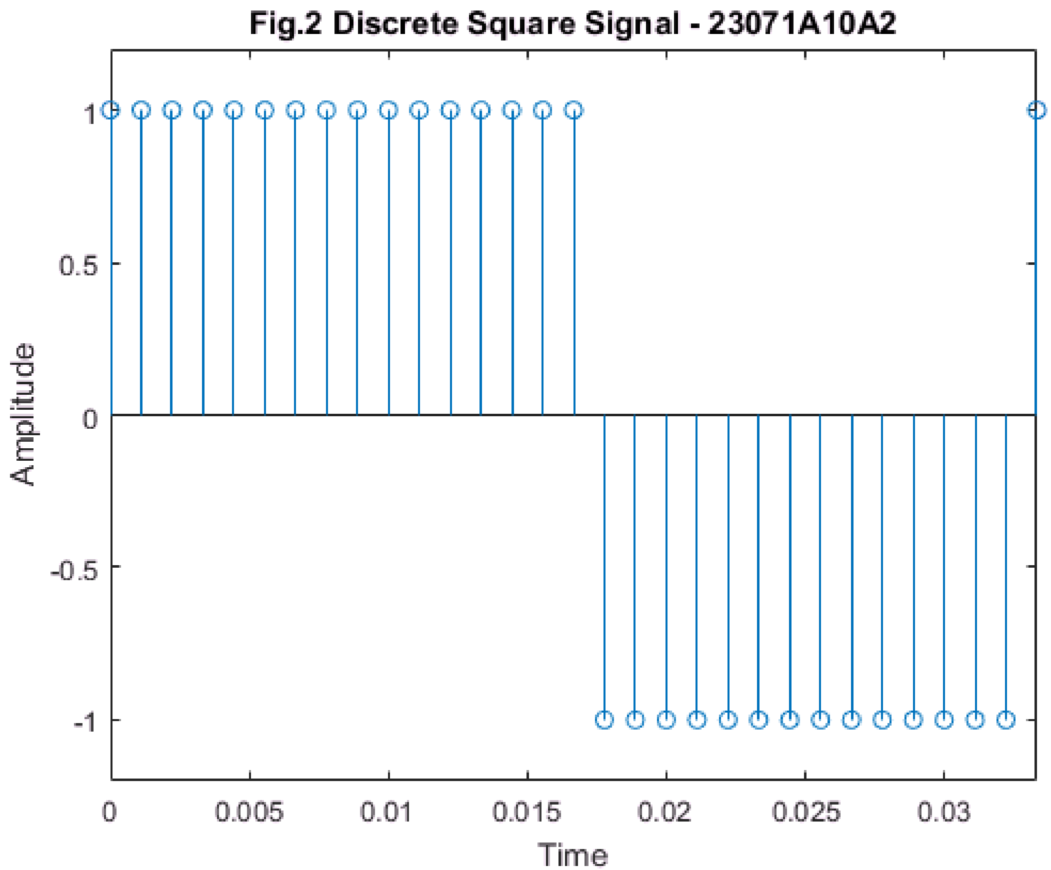
R(t) =-1; For t<0

CODE:

```
f = 30;
T = 1/f;
ts = 0:T/f:T;
for i = 1:length(ts);
    if ts(i) >= 0 && ts(i) <= 0.5*T;
        x(i) = 1;
    else
        x(i) = -1;
    end
end
ys = square(2*pi*f*ts);
stem(ts,x)
axis([0 T -1.2 1.2])
xlabel('Time')
ylabel('Amplitude')
title('Fig.1 Discrete Square Signal - 23071A10A2')
```



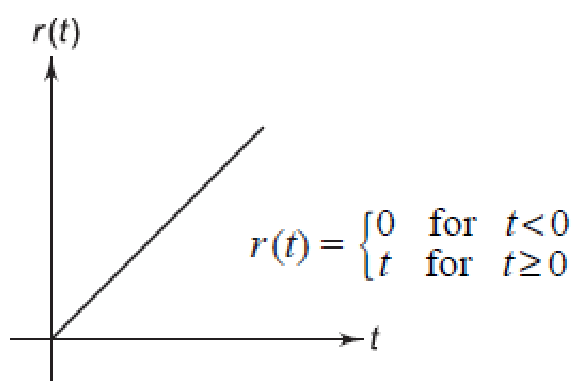

```
stem(ts,ys)
axis([0 T -1.2 1.2])
xlabel('Time')
ylabel('Amplitude')
title('Fig.2 Discrete Square Signal - 23071A10A2')
```



16. RAMP SIGNAL:

A ramp signal begins at an initial value and changes over time according to the slope you specify. The current value depends on the current simulation time.

<u>Model Graph</u>:



Ramp function

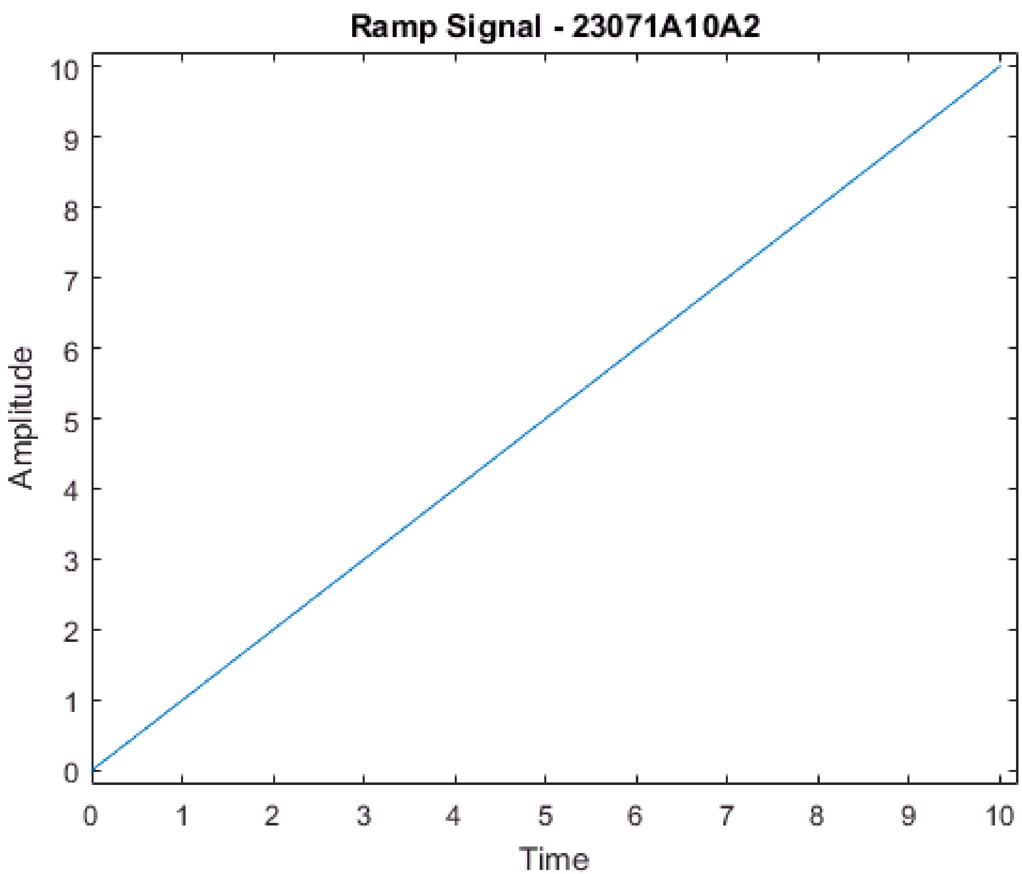
<u>Mathematical Expression</u>:

r(t) = 1; For t>=0

r(t) = 0; For t<0

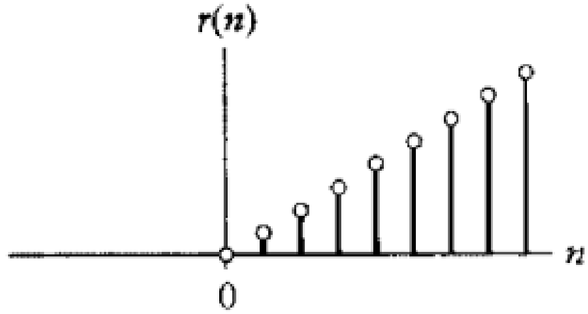
CODE:

```
clear all
t = 0:1:10;
    for i =1:length(t);
        if t(i) >= 0;
            xr(i) = t(i);
        else
            xr(i) = 0;
        end
    end
    plot(t,xr)
    axis([0 10.2 -0.2 10.2])
    xlabel('Time')
    ylabel('Amplitude')
    title('Ramp Signal - 23071A10A2')
```



17. DISCRETE RAMP SIGNAL:

Model Graph:



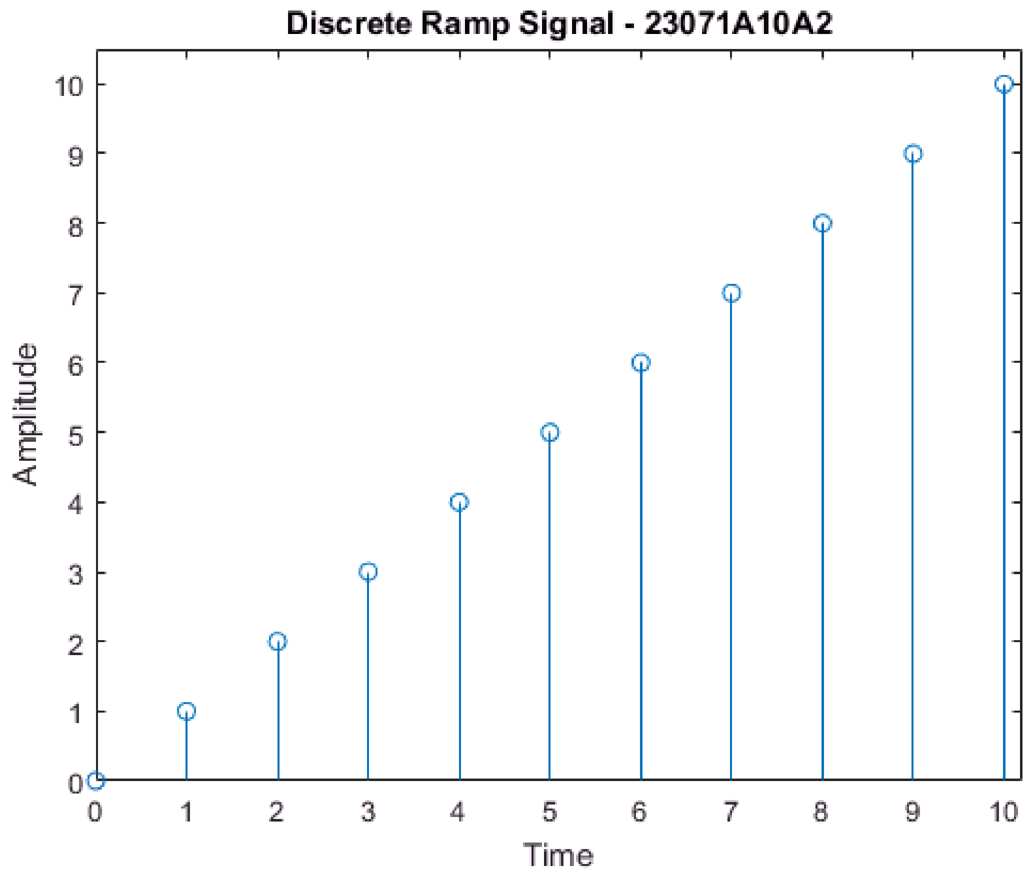
Mathematical Expression:

r(t) = 1; For t>=0

r(t) = 0; For t<0

CODE:

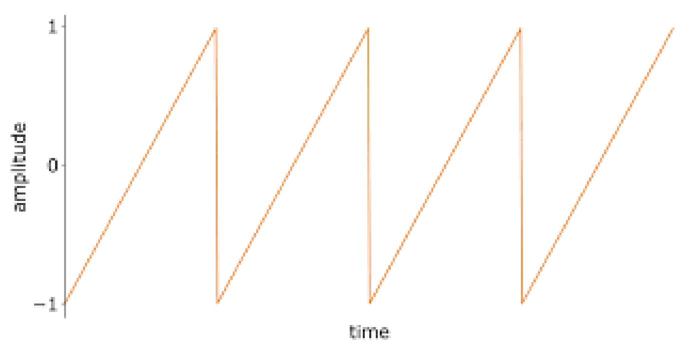
```
    clear all
t = 0:1:10;
    for i =1:length(t);
        if t(i) >= 0;
            xr(i) = t(i);
        else
            xr(i) = 0;
        end
    end
    stem(t,xr)
    axis([0 10.2 0 10.5])
    xlabel('Time')
    ylabel('Amplitude')
    title('Discrete Ramp Signal - 23071A10A2')
```



18. SAWTOOTH SIGNAL:

sawtooth is similar to the sine function but creates a sawtooth wave with peaks of –1 and 1. The sawtooth wave is defined to be –1 at multiples of 2π and to increase linearly with time with a slope of 1/π at all other times.

<u>Model Graph</u>:



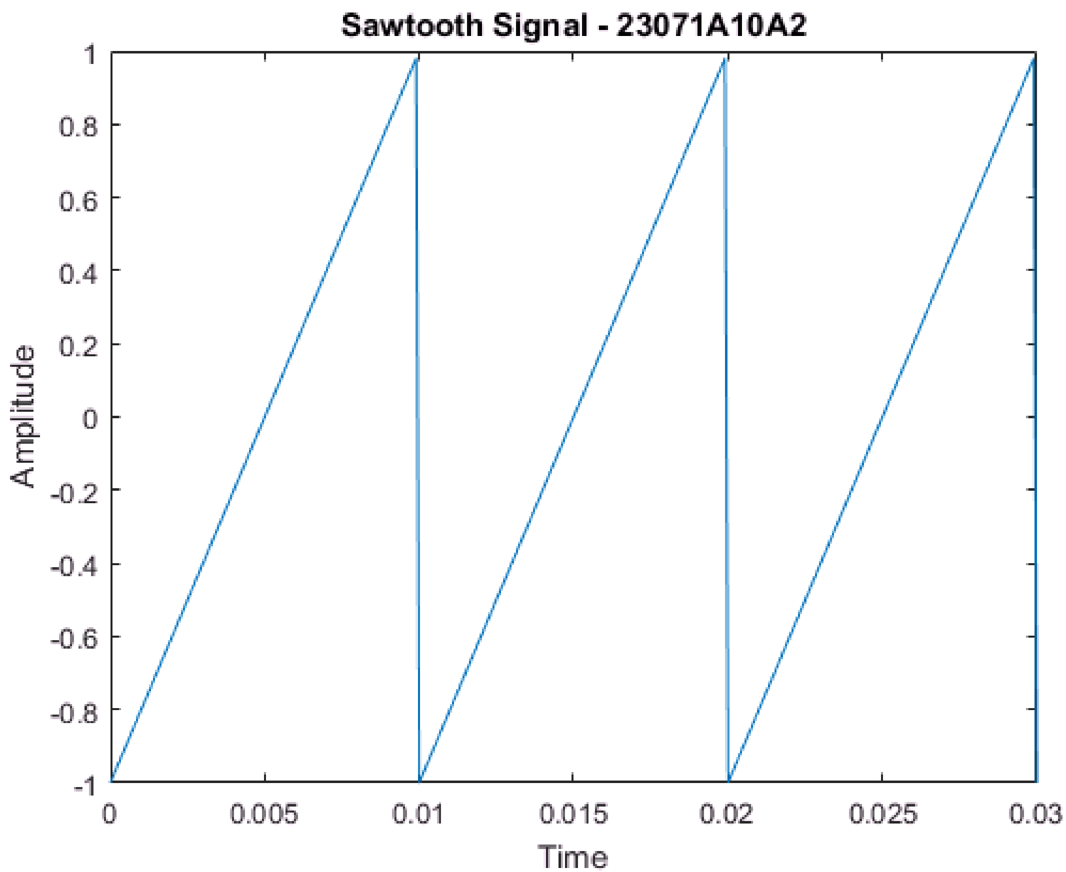
<u>Mathematical Expression</u>:

S(x)=Afrac(x/T+phi),

<u>CODE</u>:

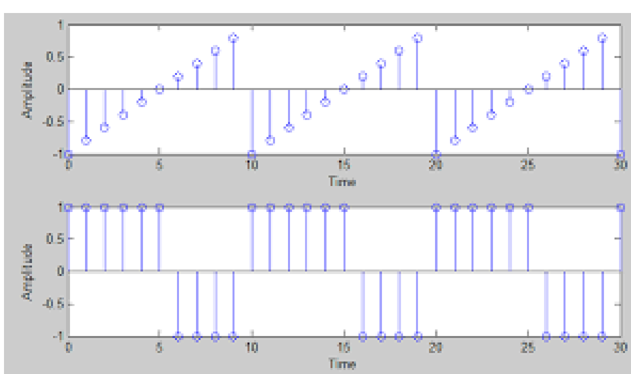
```
f = 100;
T = 1/f;
```

```
t = 0:T/f:3*T;
x = sawtooth(2*pi*f*t);
plot(t,x)
xlabel('Time')
ylabel('Amplitude')
title('Sawtooth Signal - 23071A10A2')
```



19. DISCRETE SAWTOOTH SIGNAL:

Model Graph:

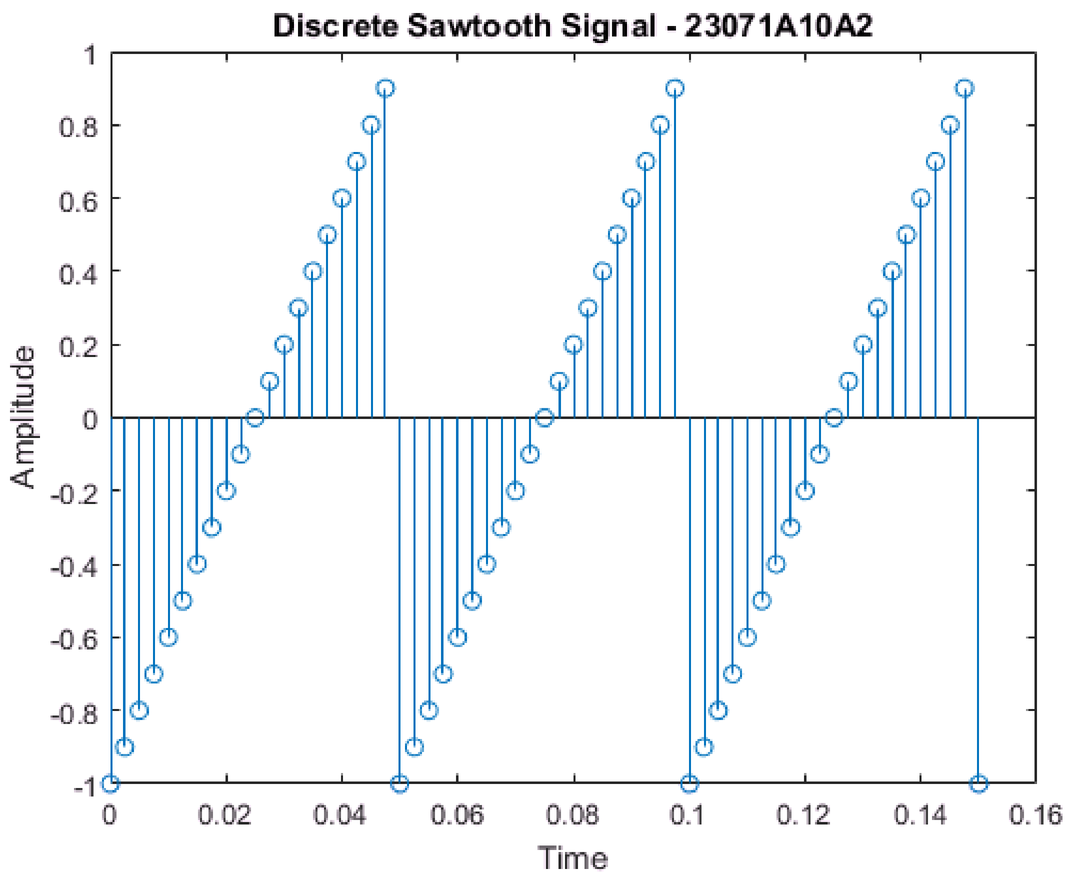


Mathematical Expression:

S(x)=Afrac(x/T+phi),

CODE:

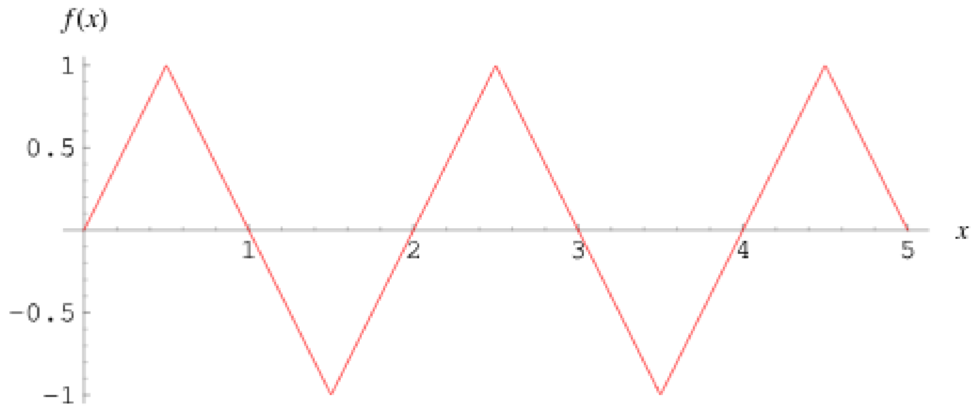
```
f = 20;
T = 1/f;
t = 0:T/f:3*T;
x = sawtooth(2*pi*f*t);
stem(t,x)
xlabel('Time')
ylabel('Amplitude')
title('Discrete Sawtooth Signal - 23071A10A2')
```



20. GENERATION OF TRIANGULAR SIGNAL:

A **triangular function** (also known as a **triangle function**, **hat function**, or **tent function**) is a function whose graph takes the shape of a triangle. Often this is an isosceles triangle of height 1 and base 2 in which case it is referred to as *the* triangular function. Triangular functions are useful in signal processing and *communication systems engineering* as representations of idealized signals, and the triangular function specifically as an integral transform kernel function from which more realistic signals can be derived, for example in kernel density estimation.

Model Graph:

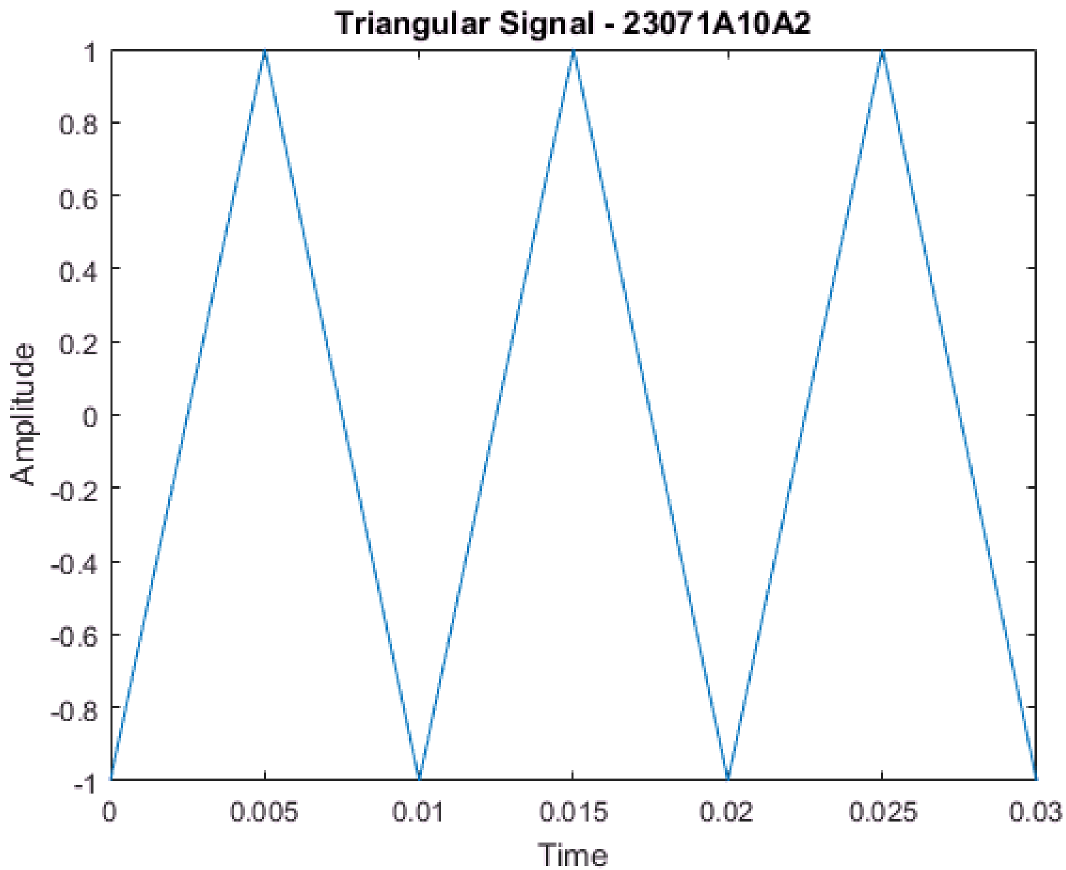


<u>Mathematical Expression</u>:

$$\operatorname{tri}(x) = \Lambda(x) \overset{\text{def}}{=} \max\big(1 - |x|, 0\big) = \begin{cases} 1 - |x|, & |x| < 1; \\ 0 & \text{otherwise.} \end{cases}$$

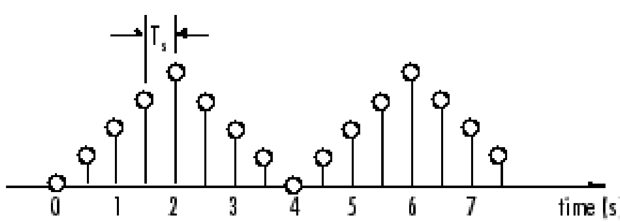
<u>CODE</u>:

```
f = 100;
T = 1/f;
t = 0:T/f:3*T;
x = sawtooth(2*pi*f*t,0.5);
plot(t,x)
xlabel('Time')
ylabel('Amplitude')
title('Triangular Signal - 23071A10A2')
```



21. DISCRETE TRIANGULAR SIGNAL:

Model Graph:



Mathematical Expression:

$$\mathrm{tri}(x) = \Lambda(x) \overset{\mathrm{def}}{=} \max\left(1 - |x|, 0\right)$$
$$= \begin{cases} 1 - |x|, & |x| < 1; \\ 0 & \text{otherwise.} \end{cases}$$

CODE:

```
f = 20;
T = 1/f;
t = 0:T/f:3*T;
x = sawtooth(2*pi*f*t,0.5);
stem(t,x)
xlabel('Time')
```

```
ylabel('Amplitude')
title('Discrete Triangular Signal - 23071A10A2')
```