Name of Experiment: Characterization of signals by Variable type and Understanding the Basic Signals using MATLAB

Continuous Time Signals

A signal is called continuous-time (or analog) signals if the independent (time) is defined in continuous interval. For $1-D$ signals, time domain of signal is continuous interval of the axis. In other words, for continuous-time signals the independent variable t is continuous. Moreover, the dependent value that usually denotes the amplitude of the signal is also continuous variable. An example of such a signal is speech as a function of time. An analog Signal is expressed by a function $x(t)$, where t takes real values.

Discrete Time Signals

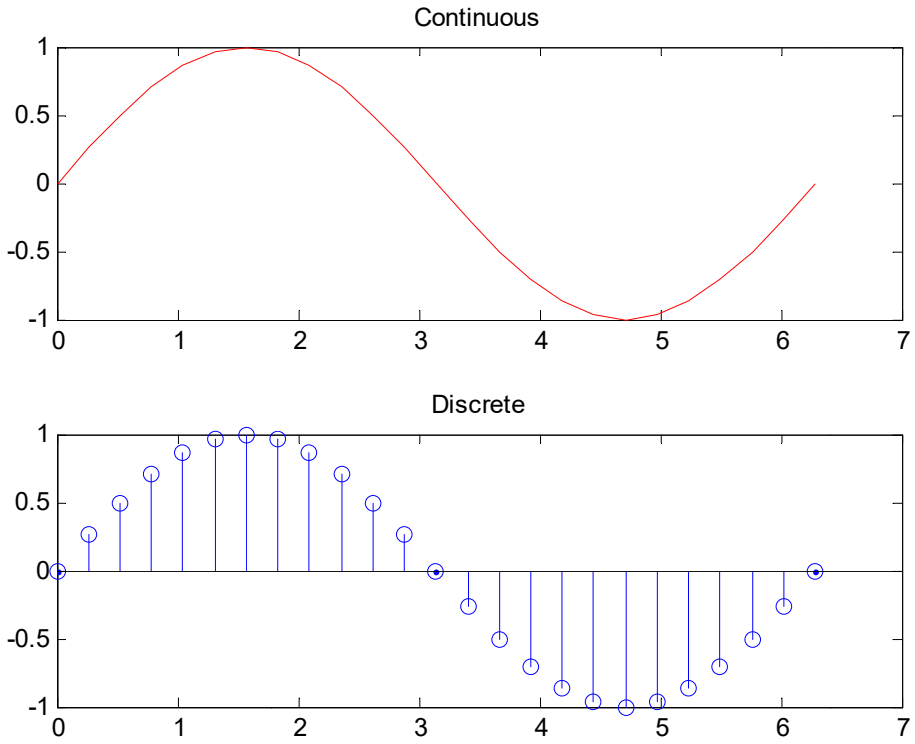
A signal is called a discrete-time signal if the independent variable (time) is defined in a discrete interval (e.g., the set of integer number), while the dependent variable is defined in a continuous set of values. In the following example, the discrete-time signal $y[n] = cos[n]$ is plotted. Note that when referring to discrete time the variable *n* is typically used to represent the time.

A discrete time signal x[n] is usually obtained by sampling a continuous-time signal $x(t)$ at a constant rate. Suppose Ts is the sampling period, that is every T_s we sample the value of $x(t)$. Suppose also that $n \in \mathbb{Z}$, *i.e.* $n = 0, \pm 1, \pm 2, \pm 3, \ldots \ldots$..The sequence of the sample is derived from the continuous time signal $x(t)$.

Digital Signals

Digital signals are the signals that both independent and dependent variables take values from a discrete set. In the following example, the signal $y[n] = cos[n]$ is again plotted, but we use the command round to limit the set of values that $y[n]$ can take, That is, $y[n]$ can be $-1, 0$ *or* 1.

```
% Program to understand formatting a plot: Continuous time and
  Discrete time signals.
    clc;
    clear all;
    close all
        x=linspace(0,2*pi,25); % 0 to 2pi is divided in 25 equidistant
    points
        y=sin(x);
        subplot(2,1,1);
        plot(x,y,'r');
        title('Continuous');
        subplot(2,1,2);
        stem(x,y);
        title('Discrete');
```



Lab Task

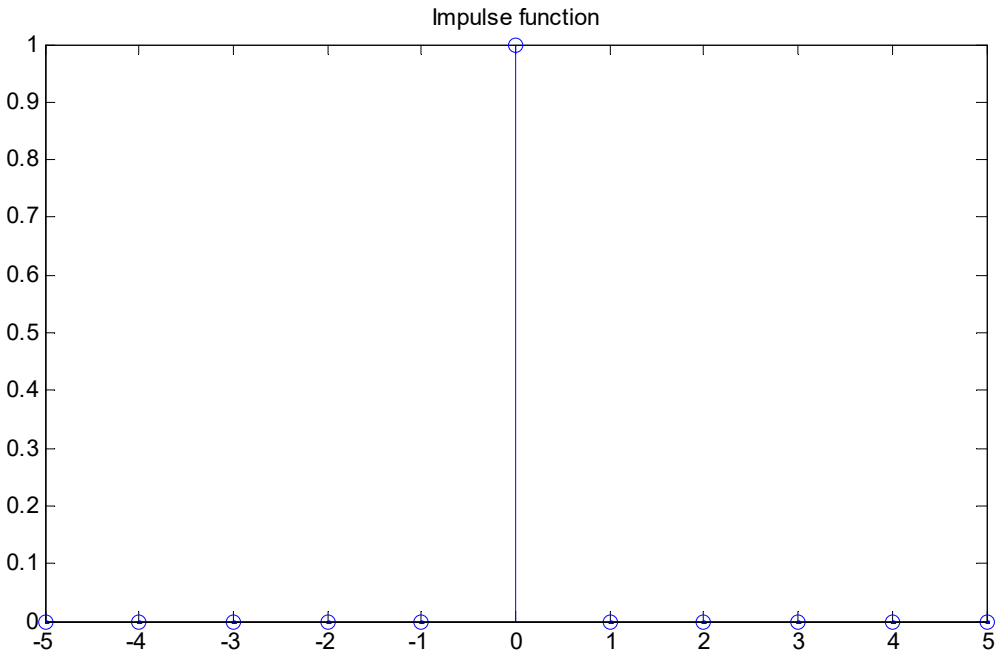
Plot the continuous time signal $x(t) = \cos(t)$ *over the interval* $-$ *Roll Number* $\leq t \leq$ *Roll Number* .

Impulse function

Impulse function given by

∂ (t) = { 1 for t=0 ; 0 otherwise }

```
n1=input('Lower limit')
n2=input('Upper limit')
x=n1:n2;
y=[x==0];
stem(x,y);
```

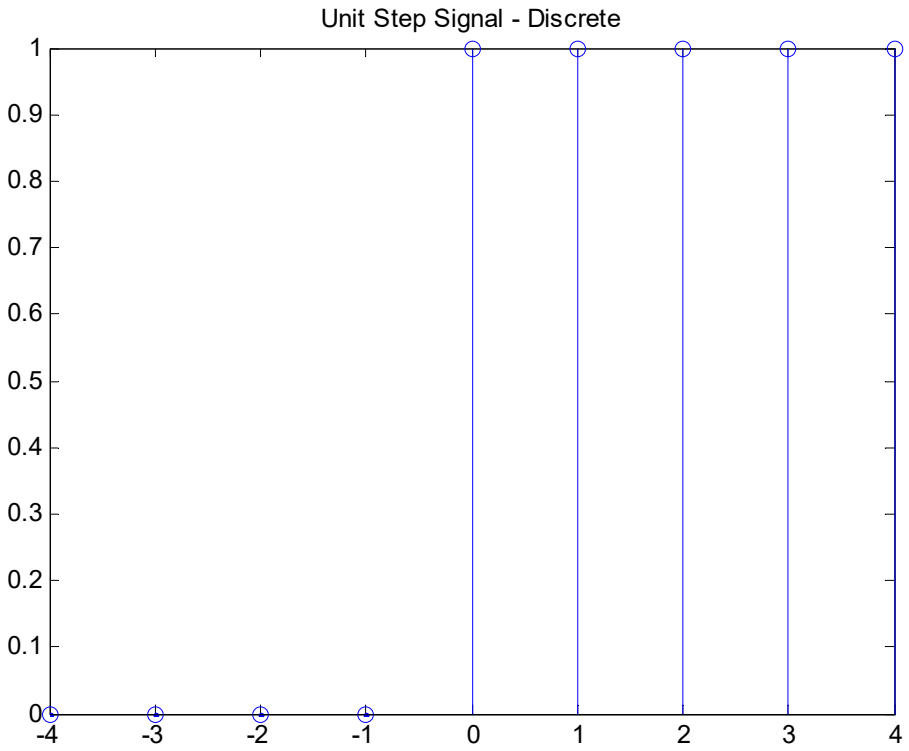


Unit Step Signal

u(t)={ 1 for t >=0 ; 0 for t< 0 }

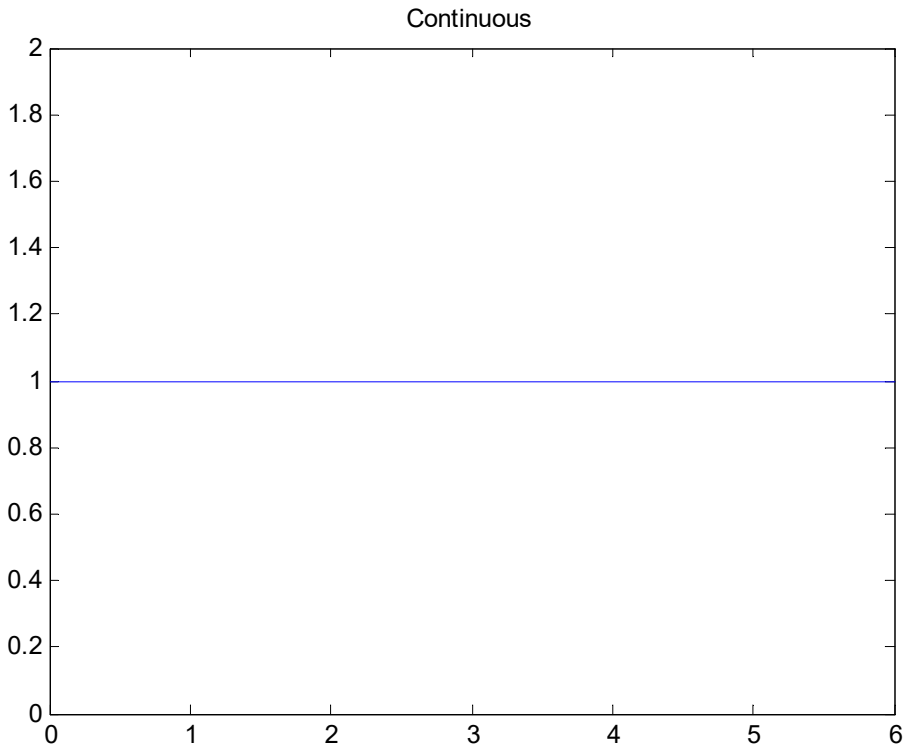
Discrete

```
n1=input('Enter the lower limit');
n2=input('Enter the upper limit');
n=n1:n2;
x=[n>=0];
stem(n,x);
title('Unit Step Signal - Discrete');
```



Continuous

```
n=input('Enter the upper limit');
t=0:n;
x=[t>=0];
plot(t,x);
title('Continuous');
```



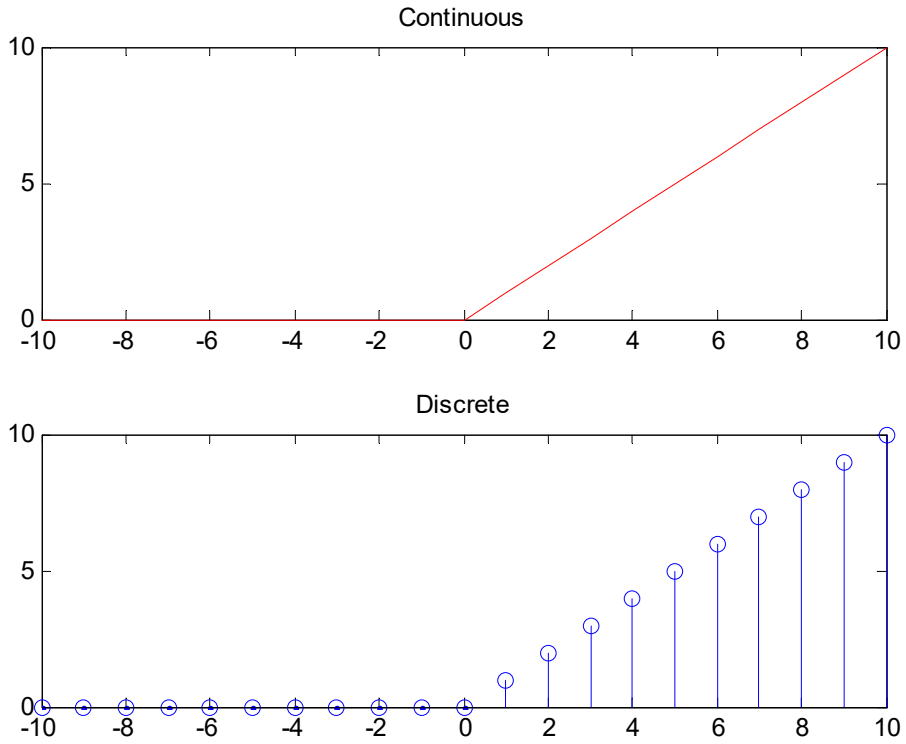
Unit step signal – Continuous
n = 6

III. Unit Ramp Signal

r(t)={ t for t >= 0 ; 0 for t <0 }

Recall that ramp signal r(t)=t*u(t) where u(t) is unit step signal

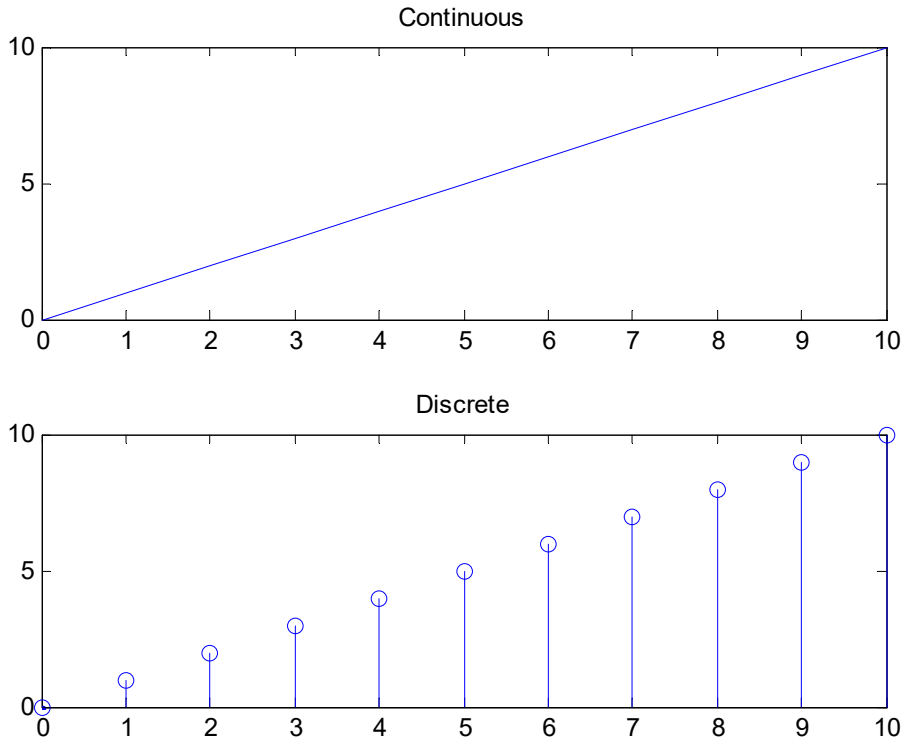
```
n1=input('Enter lower limit');
n2=input('Enter upper limit');
n=n1:n2;
x=n.*[n>=0];
subplot(2,1,1);
plot(n,x,'r');
title('Continuous');
subplot(2,1,2);
stem(n,x,'b');
title('Discrete');
```



Unit Ramp Function
n1=-10 ; n2=10

Shortcut method for Generating Ramp Signal

```
n=input('Enter the upper limit')
t=0:n;
subplot(2,1,1);
plot(t,t);
title('Continuous');
subplot(2,1,2);
stem(t,t);
title('Discrete');
```

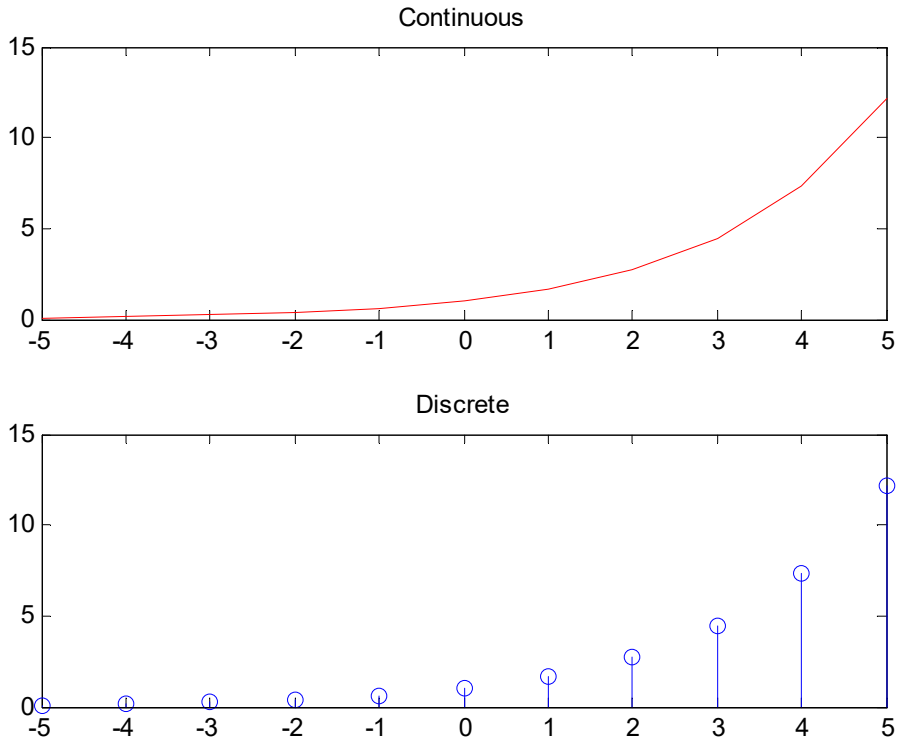


Unit Ramp Function Shortcut Method with n=10

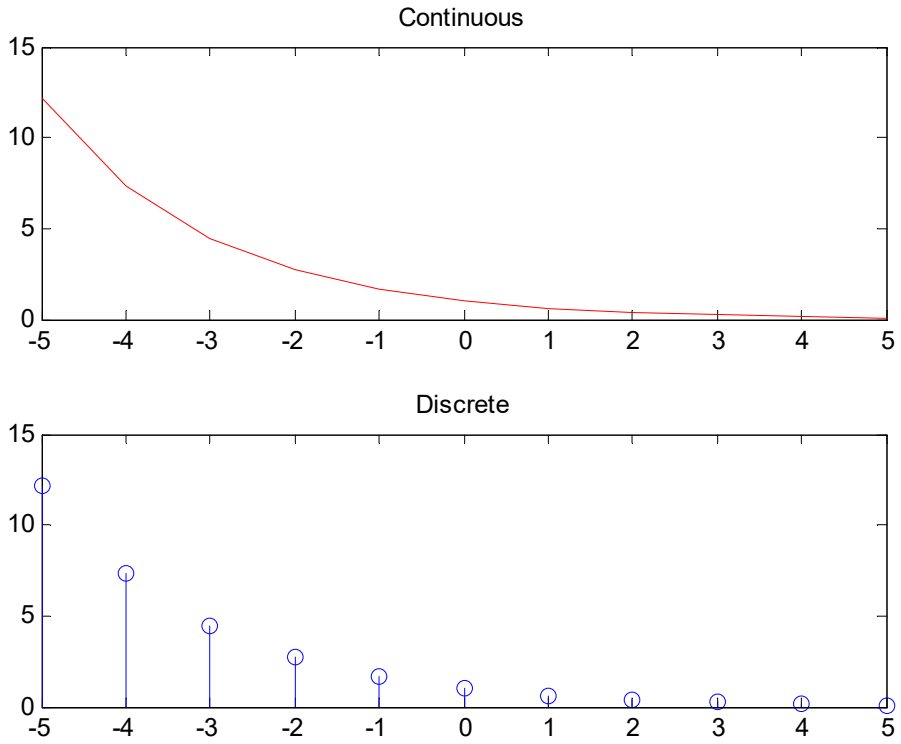
IV. Unit Exponential signal

$x(t)= e^{at}$

```
n1=input('Enter lower limit');
n2=input('Enter upper limit');
t=n1 : n2;
a=input('Enter the value for a');
y=exp(a*t);  # exp() function in matlab SYNTAX:exp(a)n-
returnstheexponentialfactorofa
subplot(2,1,1);
plot(t,y,'r');
title('Continuous');
subplot(2,1,2);
stem(t,y,'b');
title('Discrete');
```



Exponentially Growing Signal
n1 = -5 ; n2= 5 ; a = 0.5



Exponentially Decaying Signal
n1 = -5 ; n2 = 5 ; a = -0.5

Name of Experiment: Properties of signals and their transformations

Objectives

This experiment is designed to familiarize the students with the basic properties of continuous and discrete time signals. The main features of the experiment include:

- Operations on signals like time reversal or reflection time scaling and time shifting.

Transformations of the time Variable for Continuous-Time Signals

In many cases, there are signals related to each other with operations performed on the independent variable, namely, the time. In this section, we examine the basic operations that are performed on the independent variable.

Time Reversal or Reflection

The first operation discussed is the signal's reflection. A signal $y(t)$ is a reflection or a reflected version of $x(t)$ about the vertical axis if $y(t) = x(-t)$.

The operation of time reversal is actually an alternation of the signal values between negative and positive time. Assume that x is the vector that denotes the signal $x(t)$ in time t. The MATLAB statement that plots the reflected version of x(t) is plot (-t,x).

$$P_x = \frac{1}{T}\int_{t_o}^{t_o+T} |x(t)^2|\, dt$$

A signal $x(t)$ is called a power signal if $0 < P_x < \infty$. In order to calculate the energy or the power of a signal, recall that the command $limit(F, x, a)$ computes the limit of the function F when the symbolic variable x tends to a.

Time Scaling

The second operation discussed is timing scaling. A signal $x_1(t)$ is compressed version of $x(t)$ if $x_1(t) = x(at),\ a > 1$. The time compression practically means that the time duration of the signal is reduced by a factor a. On the other hand, a signal $x_2(t)$ is expanded version of $x(t)$ if $x_2(t) = x(at), 0 < a < 1$. In this case, the time duration of the signal is increased by a factor 1/a.

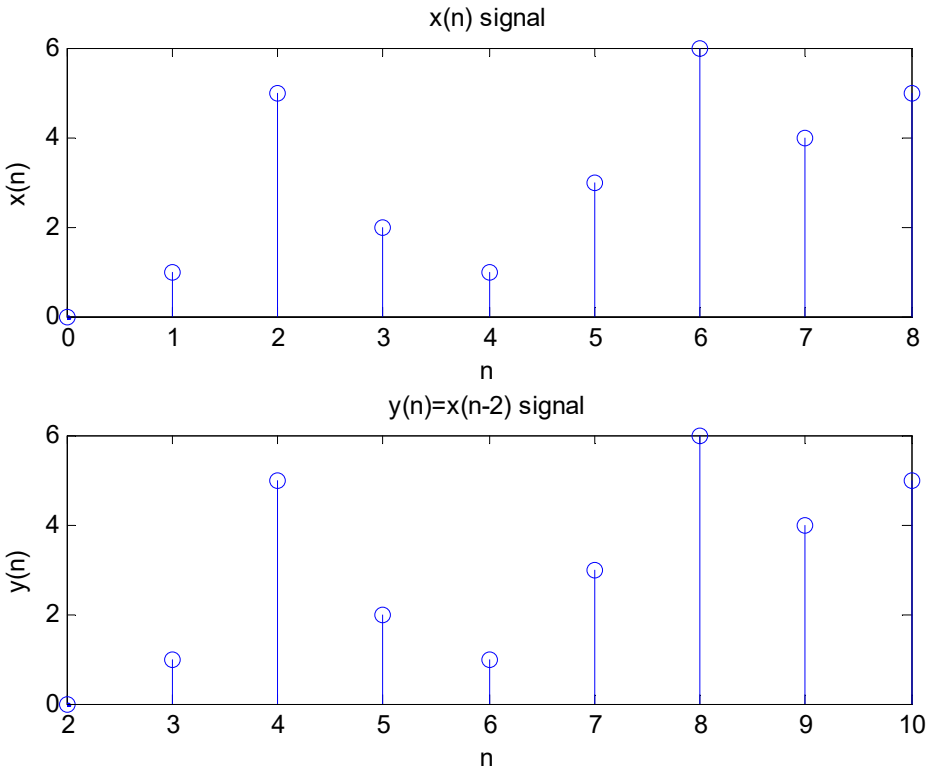
In order to plot in MATLAB a time-scaled version of $x(t)$, namely, a signal of the form $y(t) = x(at)$, the statement employed is $plot(\left(\frac{1}{a}\right) * t, x)$.In contrast to what someone would expect the vector of time t must be multiplied by 1/a and not a.

Time Shifting

A third operation performed on time is one of time shifting. A signal $y(t)$ is a time-shifted version of $x(t)$ if $(t) = x(t - t_o)$, where t_o is the time shift. If $t_o > 0$, the signal $y(t) = x(t - t_o)$ is shifted by t_o units to the right (i.e., toward +∞); while if $t_o < 0$, the signal $y(t) = x(t - t_o)$ is shifted by t_o units to the left. The time shifting is implemented in MATLAB in an opposite way to what may be expected. More specifically, in order to plot the signal $x(t - t_o)$ the corresponding MATLAB statement is plot(t+t_o,x).

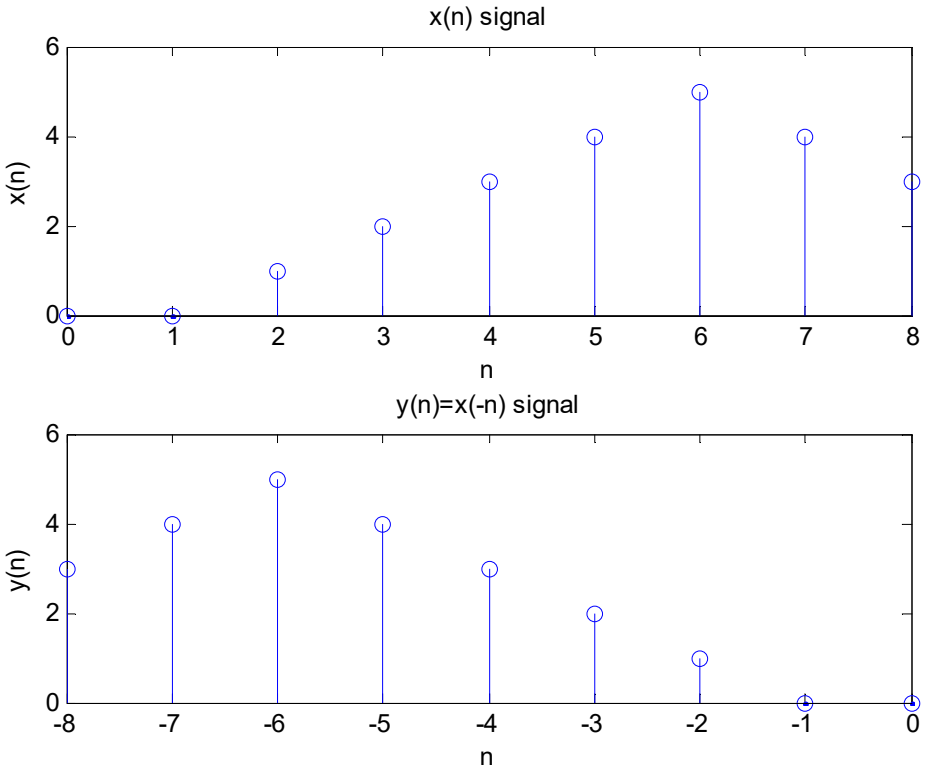
```
%Shifting a non-function Discrete-time signal
n = 0:8;
x = [0 1 5 2 1 3 6 4 5];
subplot(2,1,1);stem(n,x); title('x(n) signal');
xlabel('n'); ylabel('x(n)');

m=n+2; y=x;
subplot(2,1,2);stem(m,y); title('y(n)=x(n-2) signal');
xlabel('n'); ylabel('y(n)');
```

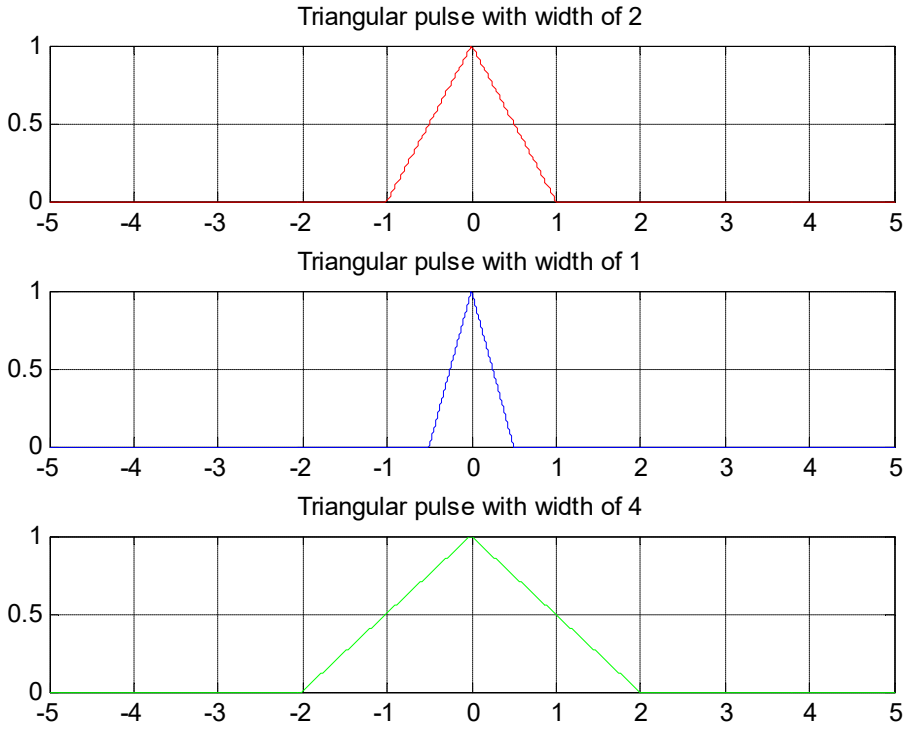



```
%Folding a Discrete-time signal
n = 0:8;
x = [0 0 1 2 3 4 5 4 3];
subplot(2,1,1);stem(n,x); title('x(n) signal');
xlabel('n'); ylabel('x(n)');

m=-fliplr(n); y=fliplr(x);
subplot(2,1,2);stem(m,y); title('y(n)=x(-n) signal');
xlabel('n'); ylabel('y(n)');
```




```
%Scaling a trianguler pulse signal
clc;
clear all;
close all
t = -5:0.0001:5; % time vector of 5 seconds
x = tripuls (t ,2) ; % triangular pulse of width 2
subplot (3 ,1 ,1)
plot(t ,x ,'r') , grid
title ('Triangular pulse with width of 2')
t1 = 2* t ; % new time vector scaled by 2
x1 = tripuls (t1 ,2) ; % scaled triangular pulse
subplot (3 ,1 ,2)
plot(t ,x1 ,'b') , grid
title ('Triangular pulse with width of 1')
t3 = 1/2* t; % new time vector scaled by 0.5
x3 = tripuls (t3 ,2) ; % scaled triangular pulse
subplot (3 ,1 ,3)
plot(t ,x3 ,'g') , grid
title ('Triangular pulse with width of 4')
```


Triangular pulse with width of 2
Triangular pulse with width of 1
Triangular pulse with width of 4
1
0.5
0
-5
-4
-3
-2
-1
0
1
2
3
4
5