EXPERIMENT NO 2

NAME: Vishal Shashikant Salvi **UID: 2019230069**

CLASS: TE COMPS **BATCH: C**

Aim: To Perform various operations on a given signal like Shift, folding, time scaling, addition and multiplication.

Theory:

1. Time Shifting

Suppose that we have a signal $x(t)$ and we define a new signal by adding/subtracting a finite time value to/from it. We now have a new signal, $y(t)$. The mathematical expression for this would be $x(t \pm t_0)$.

Graphically, this kind of signal operation results in a positive or negative "shift" of the signal along its time axis. However, note that while doing so, none of its characteristics are altered. This means that the time-shifting operation results in the change of just the positioning of the signal without affecting its amplitude or span.

Let's consider the examples of the signals in the following figures in order to gain better insight into the above information.

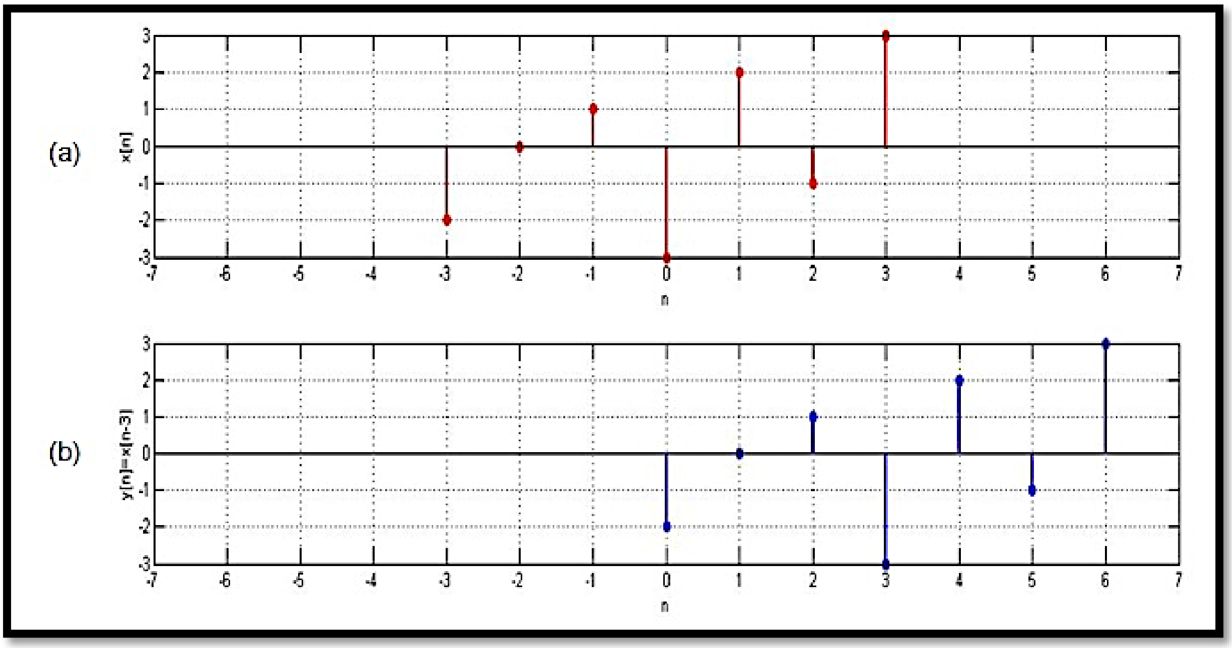


Figure 1. Original signal and its time-delayed version

Here the original signal, $x[n]$, spans from n = -3 to n = 3 and has the values -2, 0, 1, -3, 2, -1, and 3, as shown in Figure 1(a).

Time-Delayed Signals

Suppose that we want to move this signal right by three units (i.e., we want a new signal whose amplitudes are the same but are shifted right three times).
This means that we desire our output signal $y[n]$ to span from $n = 0$to $n = 6$. Such a signal is shown as Figure 1(b) and can be mathematically written as $y[n] = x[n-3]$.
This kind of signal is referred to as time-delayed because we have made the signal arrive three units late.

Time-Advanced Signals

On the other hand, let's say that we want the same signal to arrive early. Consider a case where we want our output signal to be advanced by, say, two units. This objective can be accomplished by shifting the signal to the left by two time units, i.e., $y[n] = x[n+2]$.
The corresponding input and output signals are shown in Figure 2(a) and 2(b), respectively. Our output signal has the same values as the original signal but spans from $n = -5$ to $n = 1$ instead of $n = -3$ to $n = 3$. The signal shown in Figure 2(b) is aptly referred to as a time-advanced signal.

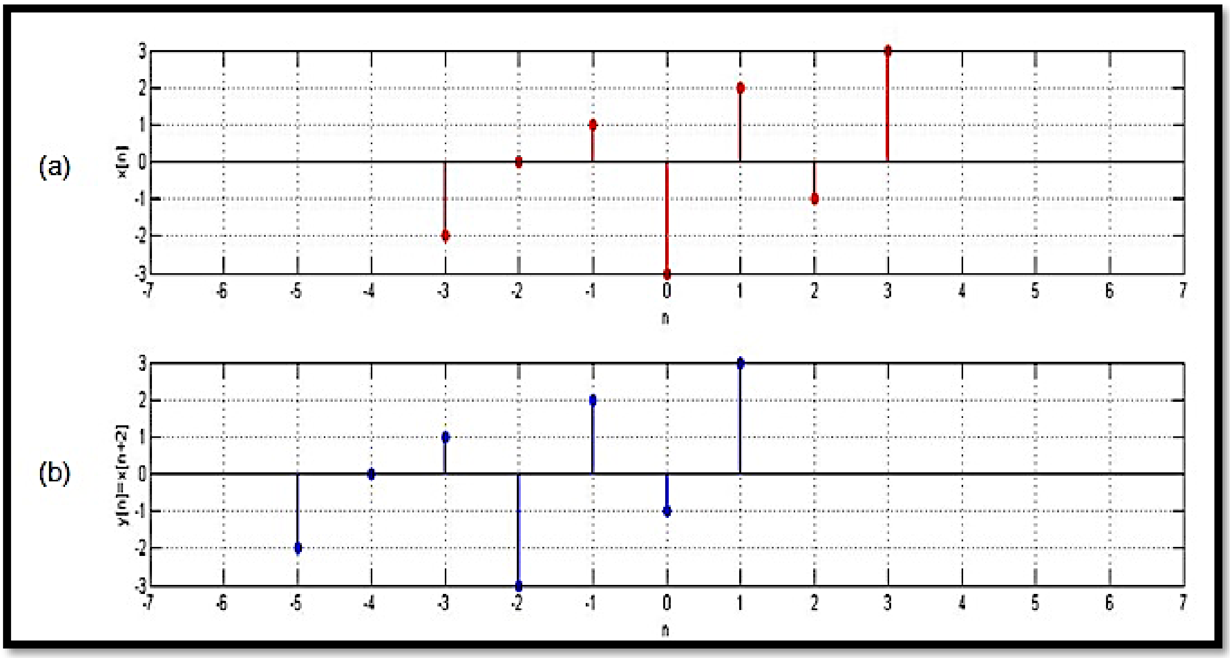


Figure 2. Original signal and its time-advanced version

For both of the above examples, note that the time-shifting operation performed over the signals affects not the amplitudes themselves but rather the amplitudes with respect to the time axis. We have used discrete-time signals in these examples, but the same applies to continuous-time signals.

Practical Applications

Time-shifting is an important operation that is used in many signal-processing applications. For example, a time-delayed version of the signal is used when performing autocorrelation. (You can learn more about autocorrelation in my previous article, Understanding Correlation.) Another field that involves the concept of time delay is artificial intelligence, such as in systems that use Time Delay Neural Networks.

2. Time Scaling

Now that we understand more about performing addition and subtraction on the independent variable representing the signal, we'll move on to multiplication.

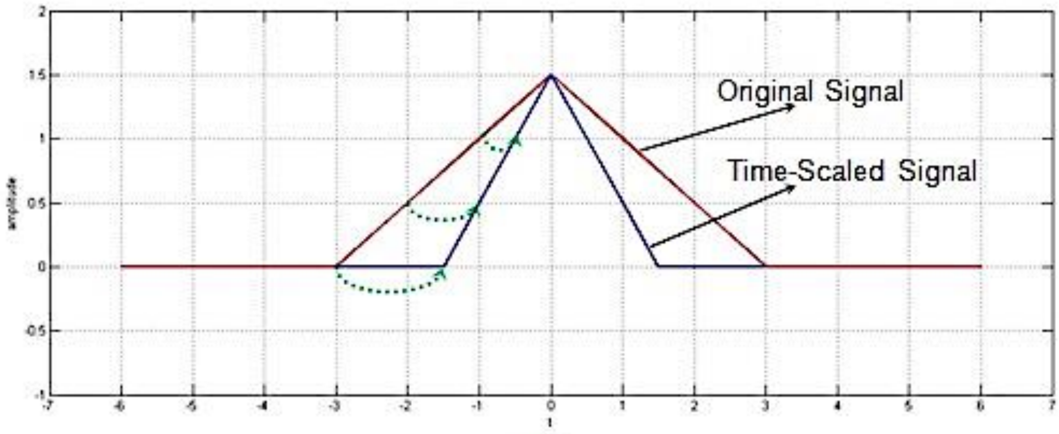
For this, let's consider our input signal to be a continuous-time signal $x(t)$ as shown by the red curve in Figure 3.

Now suppose that we multiply the independent variable (t) by a number greater than one. That is, let's make t in the signal into, say, $2t$. The resultant signal will be the one shown by the blue curve in Figure 3.

From the figure, it's clear that the time-scaled signal is contracted with respect to the original one. For example, we can see that the value of the original signal present at t = -3 is present at t = -1.5 and those at t = -2 and at t = -1 are found at t = -1 and at t = -0.5 (shown by green dotted-line curved arrows in the figure).

This means that, if we multiply the time variable by a factor of 2, then we will get our output signal contracted by a factor of 2 along the time axis. Thus, it can be concluded that the multiplication of the signal by a factor of n leads to the compression of the signal by an equivalent factor.

Now, does this mean that dividing the variable t by a number greater than 1 will cause the signal to become expanded? That is, if we divide the variable t by a factor of n, will we get a signal which is stretched by an equivalent factor?



(a)

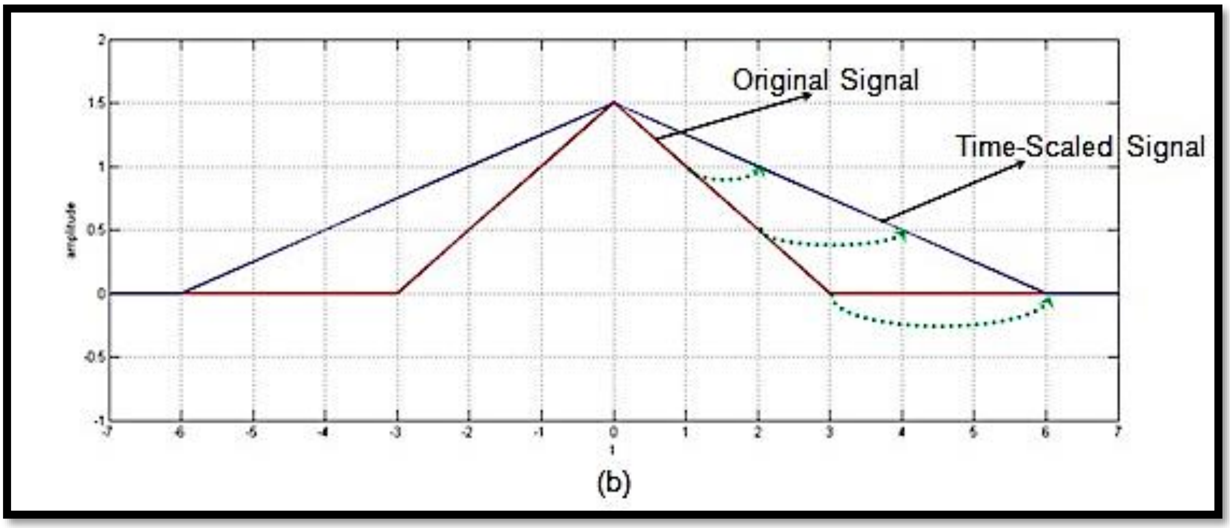


Figure 3. Original signal with its time-scaled versions

For this, let's consider our signal to be the same as the one in Figure 3 (the red curve in the figure). Now let's multiply its time-variable *t* by ½ instead of 2. The resultant signal is shown by the blue curve in Figure 3(b). You can see that, in this time-scaled signal indicated by the green dotted-line arrows in Figure 3(b), we have the values of the original signal present at the time instants t = 1, 2, and 3 to be found at t = 2, 4, and 6.

This means that our time-scaled signal is a stretched-by-a-factor-of-*n* version of the original signal. So the answer to the question posed above is "yes."

Although we have analysed the time-scaling operation with respect to a continuous-time signal, this information applies to discrete-time signals as well. However, in the case of discrete-time signals, time-scaling operations are manifested in the form of decimation and interpolation.

Practical Applications

Basically, when we perform time scaling, we change the rate at which the signal is sampled. Changing the sampling rate of a signal is employed in the field of speech processing. A particular example of this would be a time-scaling-algorithm-based system developed to read text to the visually impaired.

Next, the technique of interpolation is used in Geodesic applications (PDF). This is because, in most of these applications, one will be required to find out or predict an unknown parameter from a limited amount of available data.

3. Folding

Until now, we have assumed our independent variable representing the signal to be positive. Why should this be the case? Can't it be negative?

It can be negative. In fact, one can make it negative just by multiplying it by -1. This causes the original signal to flip along its *y*-axis. That is, it results in the reflection of the signal along its vertical axis of reference. As a result, the operation is aptly known as the time reversal or time reflection of the signal.

For example, let's consider our input signal to be $x[n]$, shown in Figure 4(a). The effect of substituting $-n$ in the place of n results in the signal $y[n]$ as shown in Figure 4(b).

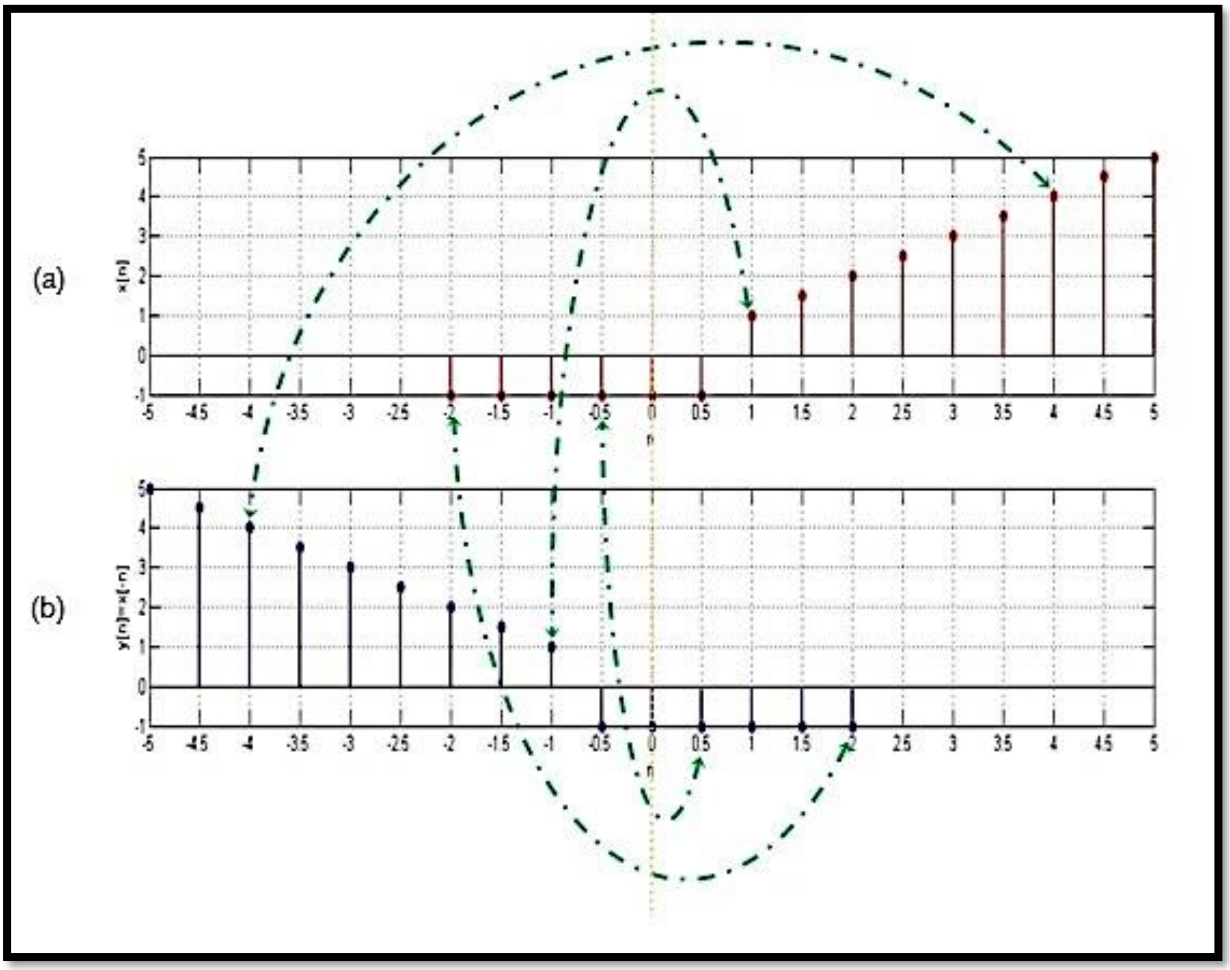


Figure 4. A signal with its reflection

Here you can observe that the value of $x[n]$ at the time instant n = -2 is -1. This is equal to the value of $y[n]$ at n = 2. Likewise, $x[-0.5] = y[0.5] = -1$, $x[1] = y[-1] = 1$, and $x[4] = y[-4] = 4$ (as indicated by the green dotted-dashed-line arrows).

This indicates that the graph of $y[n]$ is nothing but the original signal $x[n]$ reflected along the vertical axis (shown as a dotted orange line in the figure).

This applies to both continuous- and discrete-time signals.

Practical Applications

Time reversal is an important preliminary step when computing the convolution of signals: one signal is kept in its original state while the other is mirror-image and slid along the former signal to obtain the result. Time-reversal operations, therefore, are useful in various image-processing procedures, such as edge detection.

A time-reversal technique in the form of the time reverse numerical simulation (TRNS) method can be effectively used to determine defects. For example, the TRNS method aids in finding out the exact position of a notch which is a part of the structure along which a guided wave propagates.

4. Addition

The first and foremost operation which we will consider will be addition. The addition of signals is very similar to traditional mathematics. That is, if $x_1(t)$ and $x_2(t)$ are the two continuous time signals, then the addition of these two signals is expressed as $x_1(t) + x_2(t)$.

The resultant signal can be represented as $y(t)$ from which we can write

$$y(t) = x_1(t) + x_2(t)$$

Similarly for discrete time signals, $x_1[n]$ and $x_2[n]$, we can write

$$y[n] = x_1[n] + x_2[n]$$

Figure 1 shows an example of addition operation performed over the continuous time signals $x_1(t)$ and $x_2(t)$.

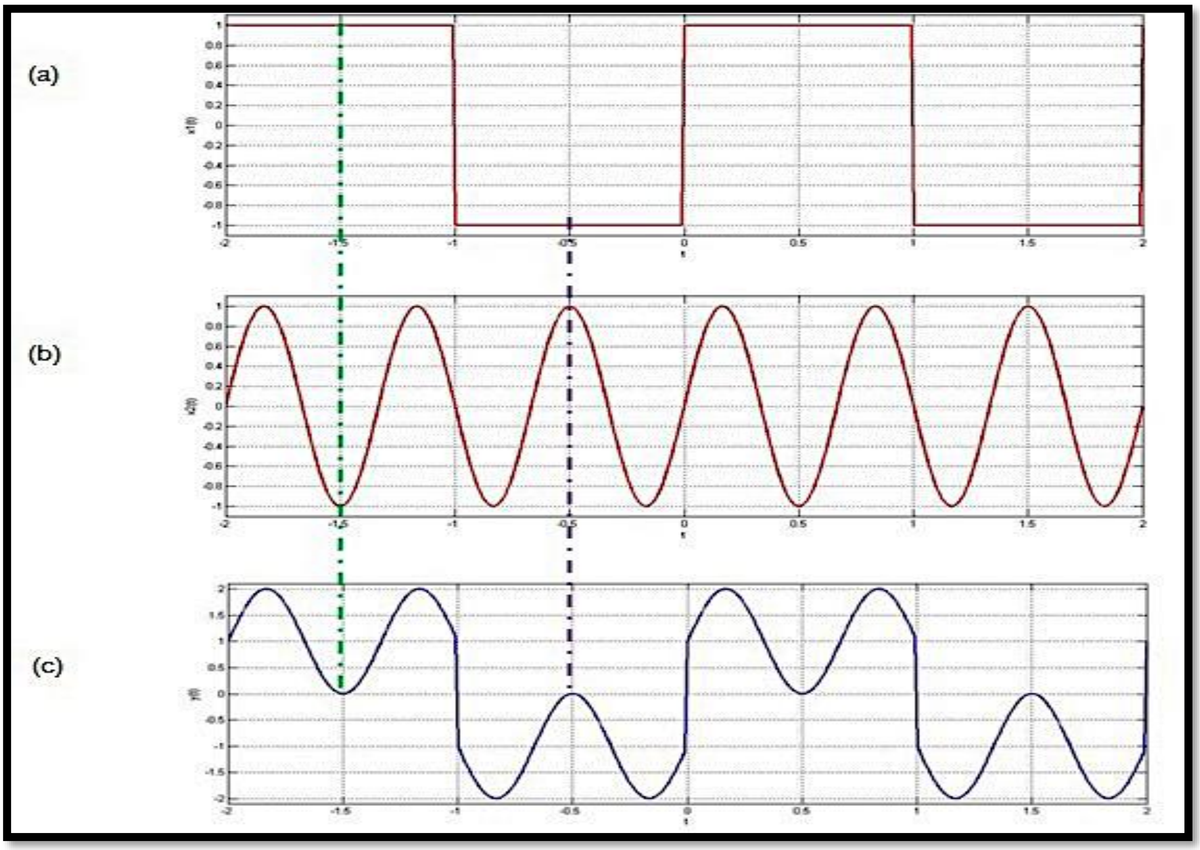


Figure 5: Addition operation performed on two continuous time signals.

By following the green-coloured dotted line in Figure 1, you can note the value of $y(t)$ at t = -1.5 to be 0 which is nothing but the summation of $x_1(t)$ at t = -1.5 which is 1 and that of $x_2(t)$ at t = -1.5 which is -1. Similarly, by moving along the purple-coloured dotted line, the value of $y(-0.5)$ is seen to be 0 which is equal to $x_1(-0.5) + x_2(-0.5) = -1 + 1$.

Hence it can be concluded that all the values of the resultant signal $y(t)$ can be obtained by adding the corresponding values of the signals $x_1(t)$ and $x_2(t)$. Although we have depicted the example of continuous time signals, the conclusion stated holds good even for discrete time signals.

Practical Scenario:

A practical aspect in which signal addition plays its role is in the case of transmission of a signal through a communication channel. This is because, here, we see that the undesired noise gets added up with the desired signal.

Another example which can be quoted is of dithering where the noise is added to the signal intentionally. This is because, when done so, one can effectively reduce undesired artifacts created as an aftermath of quantization errors.

5. Subtraction

Similar to the case of addition, subtraction deals with the subtraction of two or more signals in order to obtain a new signal. Mathematically it can be represented as

$y(t) = x_1(t) - x_2(t)$ … for continuous time signals, $x_1(t)$ and $x_2(t)$

and

$y[n] = x_1[n] - x_2[n]$ … for discrete time signals, $x_1[n]$ and $x_2[n]$

Subtraction operation performed over two discrete time signals $x_1[n]$ and $x_2[n]$ is shown in Figure 6.

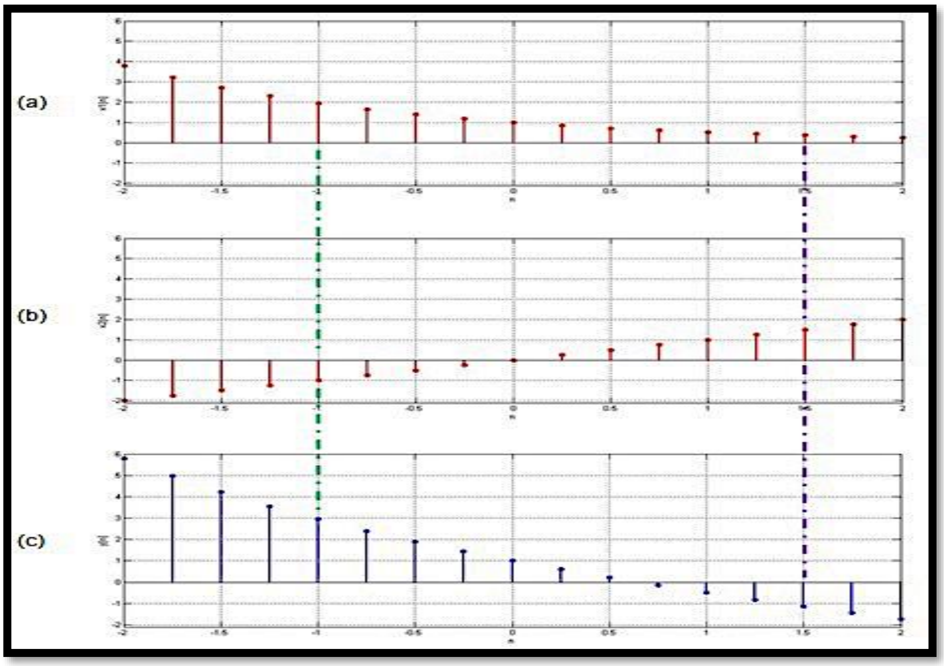


Figure6: Subtraction operation performed on two discrete time signals.

Even in the case of subtraction operation, all the values of the resultant signal $y[n]$ can be obtained by subtracting the corresponding values of the signals $x_1[n]$ and $x_2[n]$.
This is evident from the figure as the discontinuous green-colored dotted line shows $y[-1] = 3$ which is equal to $x_1[-1] - x_2[-1] = 2 – (-1)$. Another example of a similar kind is shown by the discontinuous purple-colored dotted line, wherein $y[1.5] = x_1[1.5] - x_2[1.5] = 0.4 – 1.5 = -1.1$.
From the discussion presented, it can be stated that the conclusion we arrive at in the case of subtraction operation is very similar to that of the addition operation and applies to both continuous and discrete time signals.

6. Multiplication

The next basic signal operation performed over the dependent variable is multiplication. In this case, as you might have already guessed, two or more signals will be multiplied so as to obtain the new signal.
Mathematically, this can be given as:
$y(t) = x_1(t) \times x_2(t)$ … for continuous-time signals $x_1(t)$ and $x_2(t)$
and
$y[n] = x_1[n] \times x_2[n]$ … for discrete-time signals $x_1[n]$ and $x_2[n]$
Figure 7(c) shows the resultant discrete-time signal $y[n]$ obtained by multiplying the two discrete-time signals $x_1[n]$ and $x_2[n]$ shown in Figures 7(a) and 7(b), respectively.

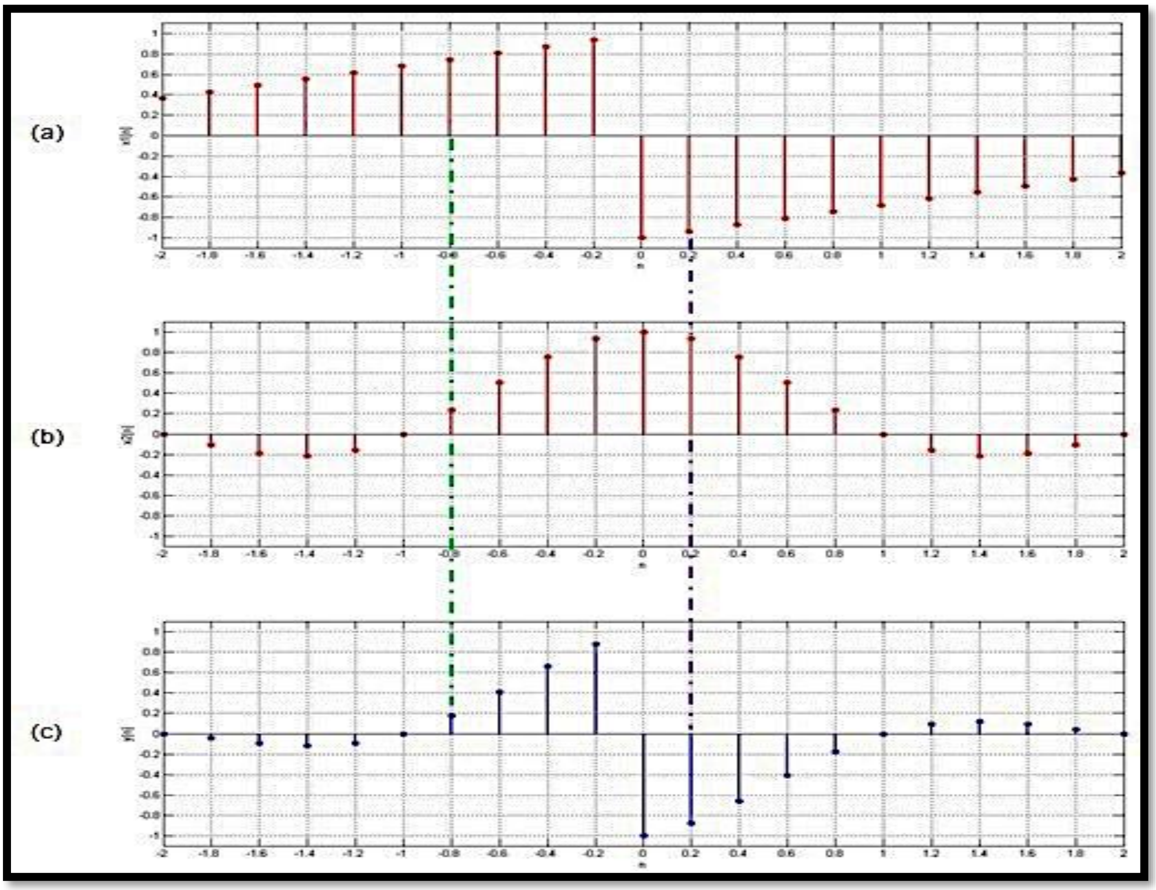


Figure 7. Multiplication operation performed over two discrete-time signals

Here the value of $y[n]$ at n = -0.8 is seen to be 0.17, which is found to be equal to the product of the values of $x_1[n]$ and $x_2[n]$ at n = -0.8, which are 0.75 and 0.23, respectively. In other words, by tracing along the green dotted-dashed line, one gets 0.75 × 0.23 = 0.17.
Similarly, if we move along the purple dotted-dashed line (at n = 0.2) to collect the values of $x_1[n]$, $x_2[n]$, and $y[n]$, we find that they are -0.94, 0.94, and -0.88, respectively. Here also we find that -0.94 × 0.94 = -0.88, which in turn implies $x_1[0.2] \times x_2[0.2] = y[0.2]$.
Thus, we can conclude that the multiplication operation results in the generation of a signal whose values can be obtained by multiplying the corresponding values of the original signals. This is true irrespective of whether we are dealing with a continuous-time or discrete-time signal.

Practical Scenario

Multiplication of signals is exploited in the field of analog communication when performing amplitude modulation (AM). In AM, the message signal is multiplied with the carrier signal so as to obtain a modulated signal.
Another example in which signal multiplication plays an important role is frequency shifting in RF (radio frequency) systems. Frequency shifting is a fundamental aspect of RF communication, and it is accomplished using a mixer, which is similar to an analog multiplier.

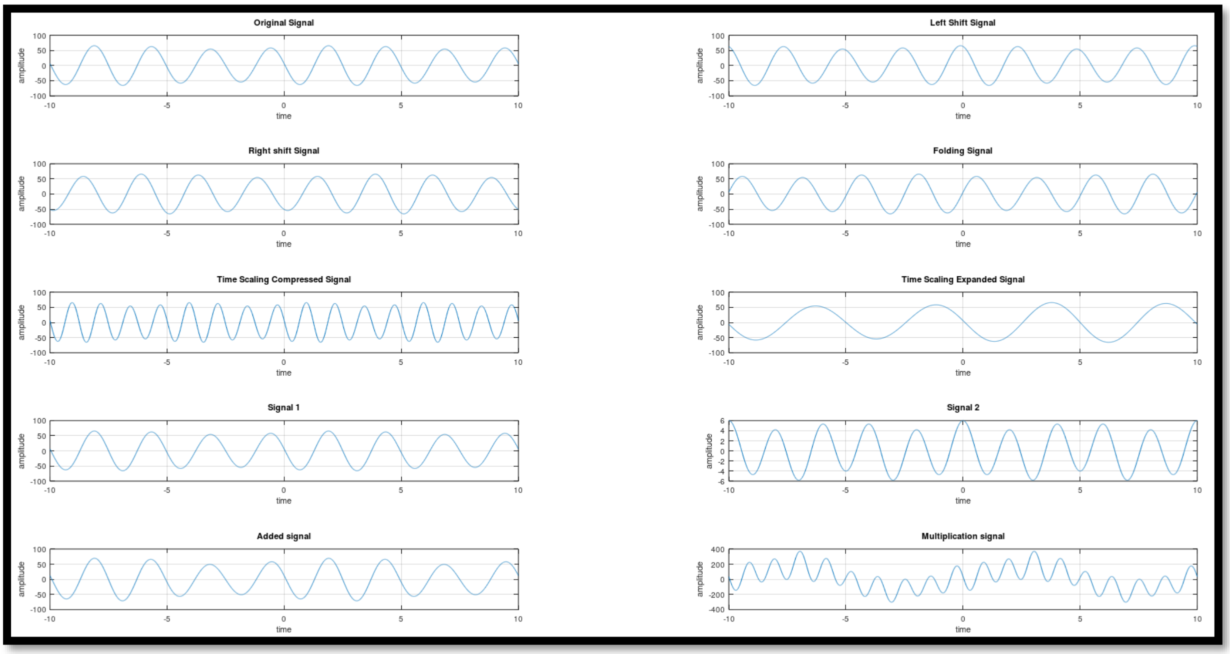
Code:

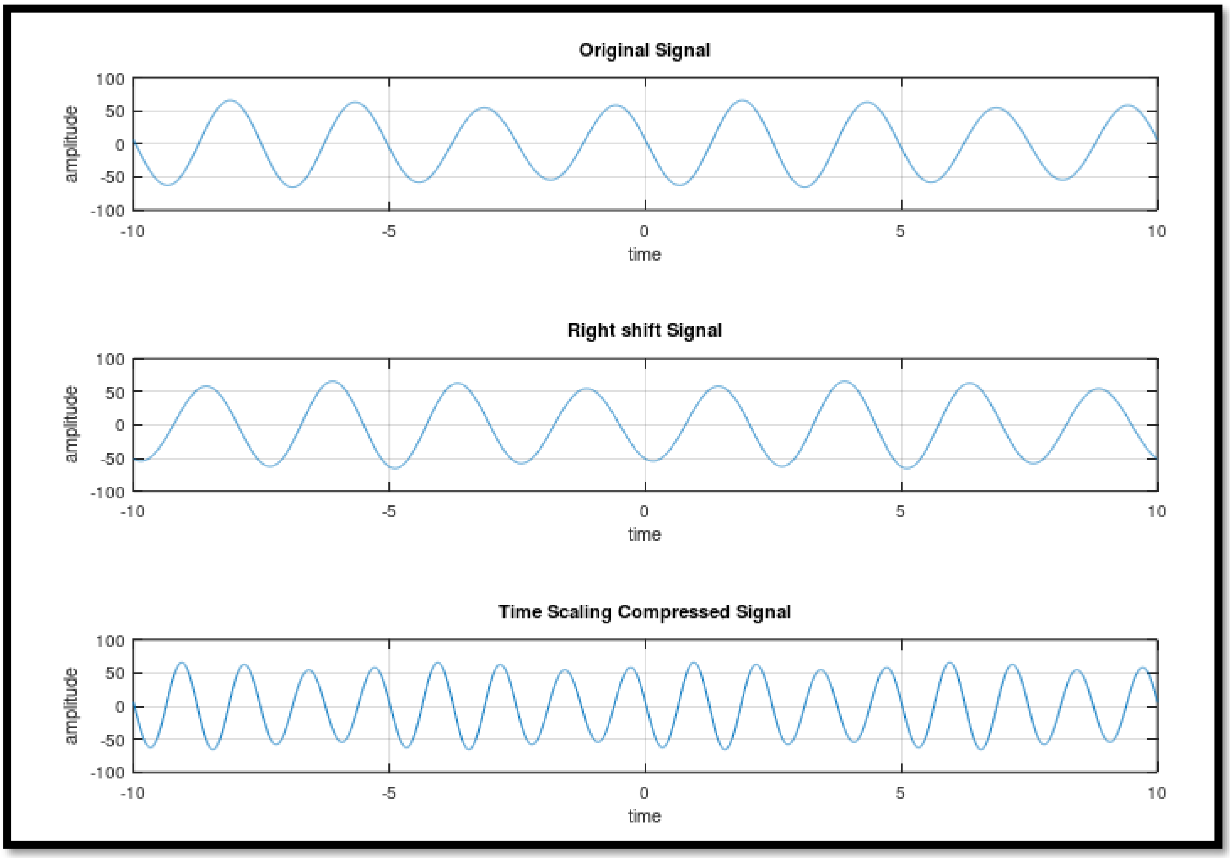
```
clc;clear;
t = -10:0.01:10;
#Original signals
x = 5*cos(pi*t) - 60*sin(0.8*pi*t) + cos(pi*t);
subplot(5,2,1);
plot(t,x);
title("Original Signal");
grid on;xlabel('time');ylabel('amplitude');

#Left Shift Signal
x = 5*cos(pi*(t+2)) - 60*sin(0.8*pi*(t+2)) + cos(pi*(t+2));
subplot(5,2,2);
plot(t,x);
title("Left Shift Signal");
grid on;xlabel('time');ylabel('amplitude');

#Right shift Signal
x = 5*cos(pi*(t-2)) - 60*sin(0.8*pi*(t-2)) + cos(pi*(t-2));
subplot(5,2,3);
plot(t,x);
title("Right shift Signal");
grid on;xlabel('time');ylabel('amplitude');
```

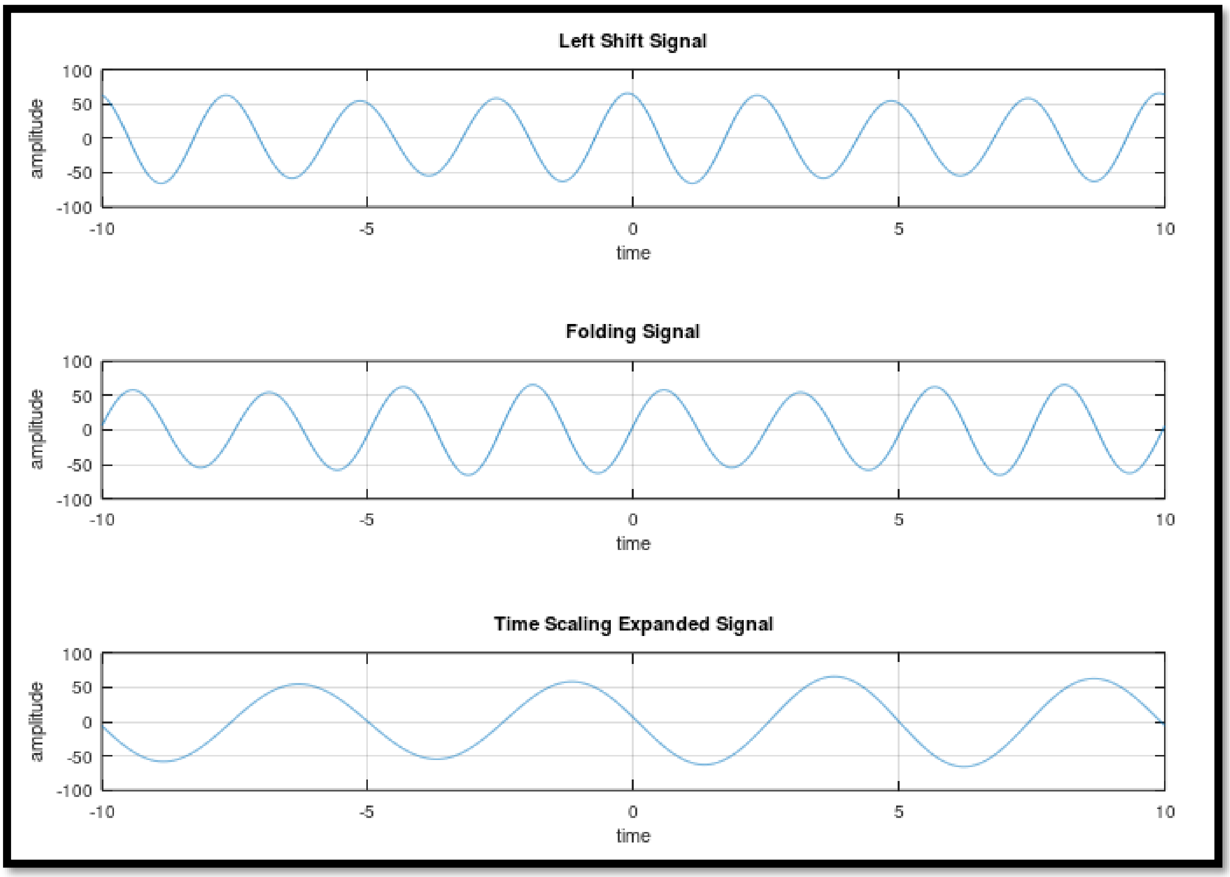
```
#Folding Signal
x = 5*cos(pi*(-t)) - 60*sin(0.8*pi*(-t)) + cos(pi*(-t));
subplot(5,2,4);
plot(t,x);
title("Folding Signal");
grid on;xlabel('time');ylabel('amplitude');

#Time Scaling Compressed Signal
x = 5*cos(pi*2*t) - 60*sin(0.8*pi*2*t) + cos(pi*2*t);
subplot(5,2,5);
plot(t,x);
title("Time Scaling Compressed Signal");
grid on;xlabel('time');ylabel('amplitude');

#Time Scaling Expanded Signal
x = 5*cos(pi*0.5*t) - 60*sin(0.8*pi*0.5*t) + cos(pi*0.5*t);
subplot(5,2,6);
plot(t,x);
title("Time Scaling Expanded Signal");
grid on;xlabel('time');ylabel('amplitude');

#Signal 1
x = 5*cos(pi*t) - 60*sin(0.8*pi*t) + cos(pi*t);
subplot(5,2,7);
plot(t,x);
title("Signal 1");
grid on;xlabel('time');ylabel('amplitude');

#Signal 2
y = 5*cos(pi*t) + cos(0.4*pi*t);
subplot(5,2,8);
plot(t,y);
title("Signal 2");
grid on;xlabel('time');ylabel('amplitude');

%addition
subplot(5,2,9);
plot(t,x+y);
title("Added signal")'
grid on;xlabel('time');ylabel('amplitude');

%Multiplication
subplot(5,2,10);
```

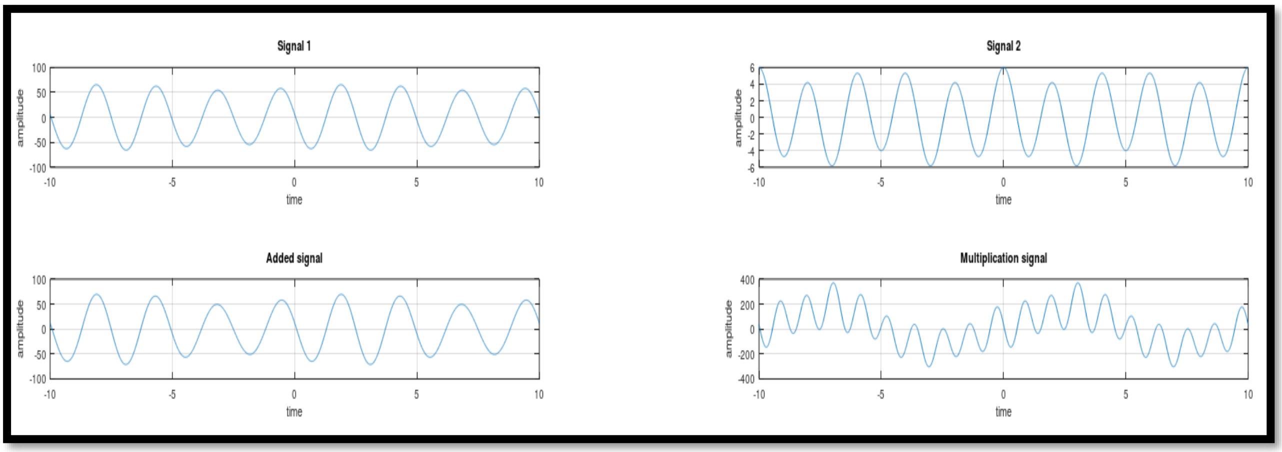
```
plot(t,x.*y);
title("Multiplication signal");
grid on;xlabel('time');ylabel('amplitude');
```

Output:





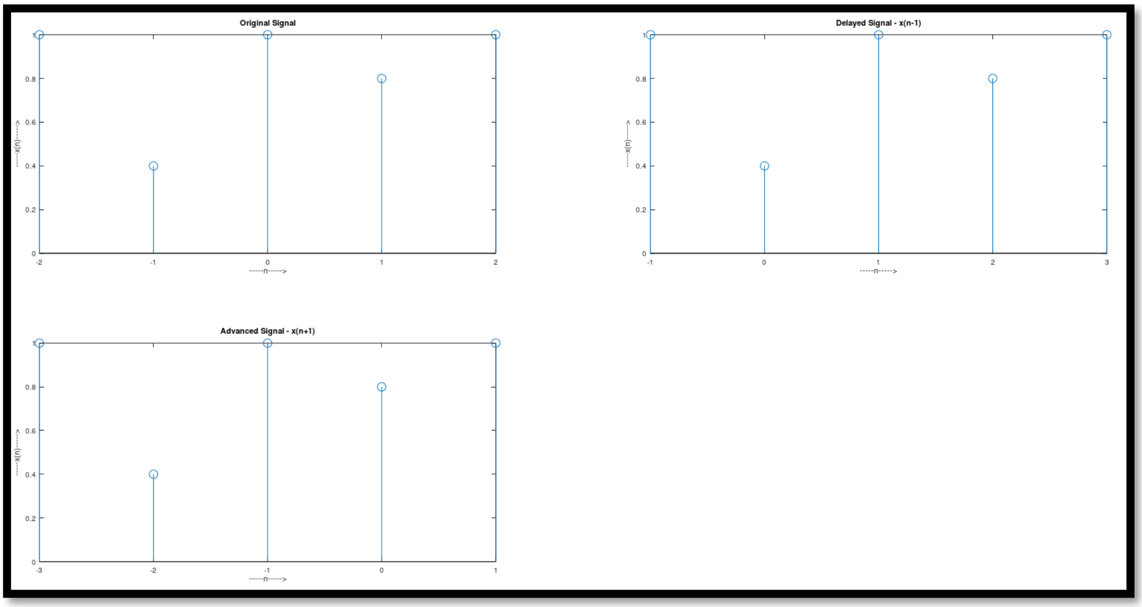

Left Shift Signal
amplitude
time
Folding Signal
amplitude
time
Time Scaling Expanded Signal
amplitude
time


Signal 1
amplitude
time
Signal 2
amplitude
time
Added signal
amplitude
time
Multiplication signal
amplitude
time

Time Shifting:

```
clc;clear;

x1=[1 0.4 1 0.8 1];
n=-2:2;

subplot(2,2,1);
stem(n,x1);
title('Original Signal');
xlabel('-----n----->');
ylabel('-----x(n)----->');

%time shifting delayed
n1=n+1;
subplot(2,2,2);
stem(n1,x1);
title('Delayed Signal - x(n-1)');
xlabel('-----n----->');
ylabel('-----x(n)----->');

%time shifting advanced
n2=n-1;
subplot(2,2,3);
stem(n2,x1);
title('Advanced Signal - x(n+1)');
xlabel('-----n----->');
ylabel('-----x(n)----->');
```

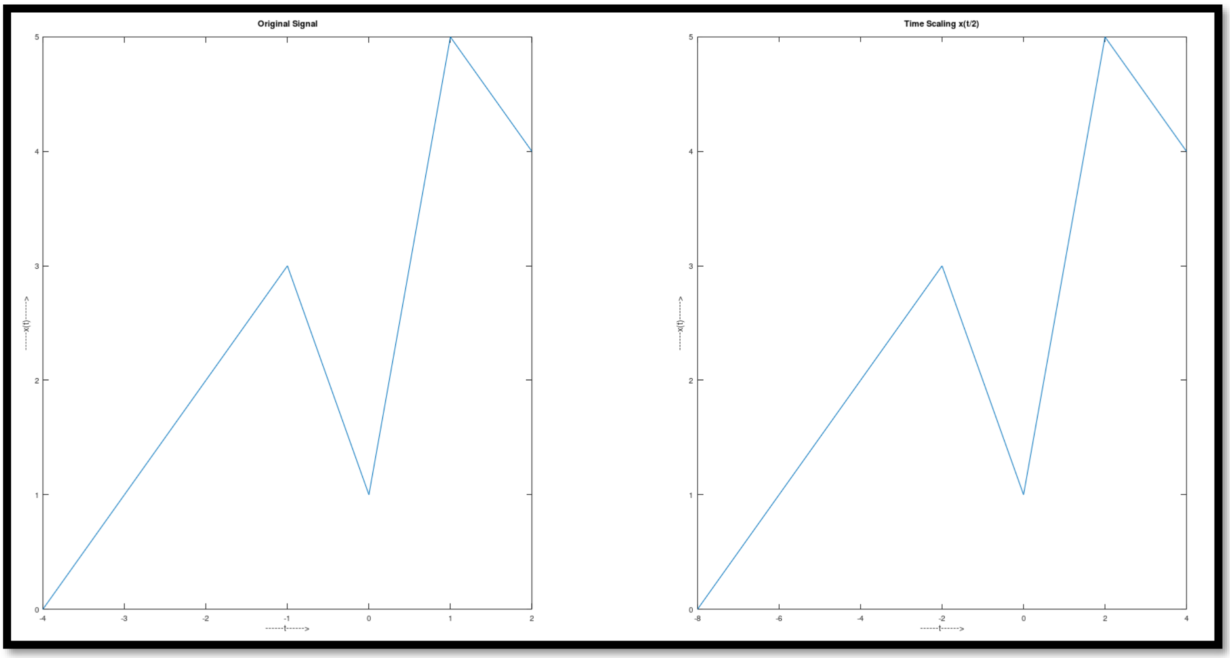
Output:



Time Scalling:

```
clc;clear;
x=[0 1 2 3 1 5 4];
t=-4:2;

subplot(1,2,1);
plot(t,x);
xlabel('------t------>');
ylabel('------x(t)------>');
title('Original Signal');

%time scaling = t/2
t1=t.*2;
subplot(1,2,2);
plot(t1,x);
xlabel('------t------>');
ylabel('------x(t)------>');
title('Time Scaling x(t/2)');
```

Output:



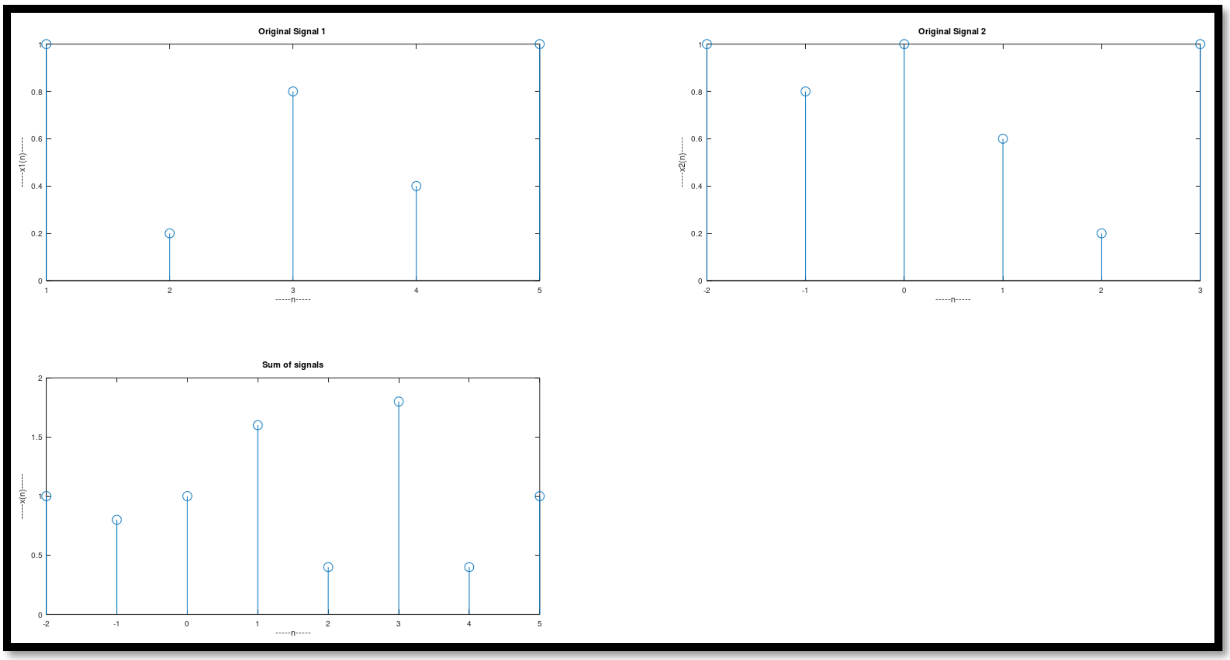
Addition:

```
clc;clear;
x1=[1 0.2 0.8 0.4 1];
n1=1:5

x2=[1 0.8 1 0.6 0.2 1];
n2=-2:3

subplot(2,2,1);
stem(n1,x1);
title('Original Signal 1 ');
xlabel('-----n-----');
ylabel('-----x1(n)-----');

subplot(2,2,2);
stem(n2,x2);
title('Original Signal 2 ');
xlabel('-----n-----');
ylabel('-----x2(n)-----');

[r,n]=signaladdition(x1,x2,n1,n2);

subplot(2,2,3);
stem(n,r);
title('Sum of signals');
xlabel('-----n-----');
ylabel('-----x(n)-----');
```

Function:

```
function[result,n]=signaladdition(x1,x2,n1,n2);
n=min(min(n1),min(n2)):max(max(n1),max(n2));
a=zeros(size(n));
b=a;
a(find(n==min(n1)):find(n==max(n1)))=x1;
b(find(n==min(n2)):find(n==max(n2)))=x2;
result = (a+b);
end
```

Output:



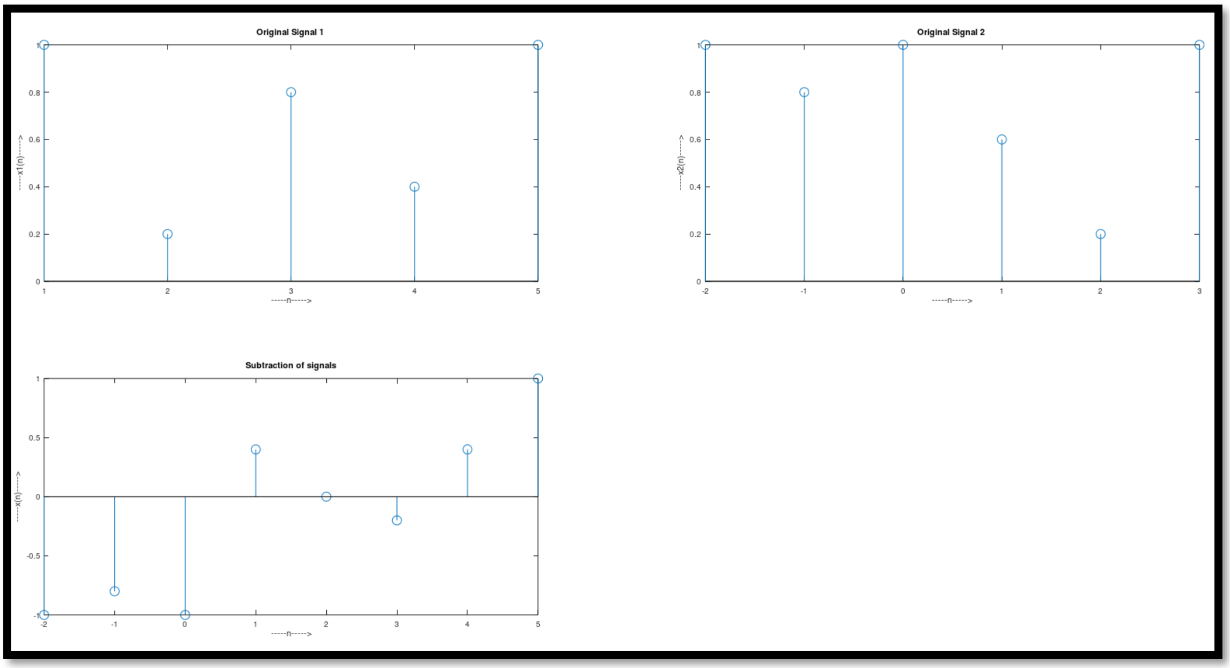
Subtraction:

```
clc;clear;
x1=[1 0.2 0.8 0.4 1];
n1=1:5

x2=[1 0.8 1 0.6 0.2 1];
n2=-2:3

subplot(2,2,1);
stem(n1,x1);
title('Original Signal 1 ');
xlabel('-----n----->');
ylabel('-----x1(n)----->');

subplot(2,2,2);
stem(n2,x2);
title('Original Signal 2 ');
xlabel('-----n----->');
ylabel('-----x2(n)----->');

[r,n]=signaladdition(x1,x2,n1,n2);

subplot(2,2,3);
stem(n,r);
title('Subtraction of signals');
xlabel('-----n----->');
ylabel('-----x(n)----->');
```

Function:

```
function[result,n]=signaladdition(x1,x2,n1,n2);
n=min(min(n1),min(n2)):max(max(n1),max(n2));
a=zeros(size(n));
b=a;
a(find(n==min(n1)):find(n==max(n1)))=x1;
b(find(n==min(n2)):find(n==max(n2)))=x2;
result = (a-b);
end
```

Output:



Multiplication:

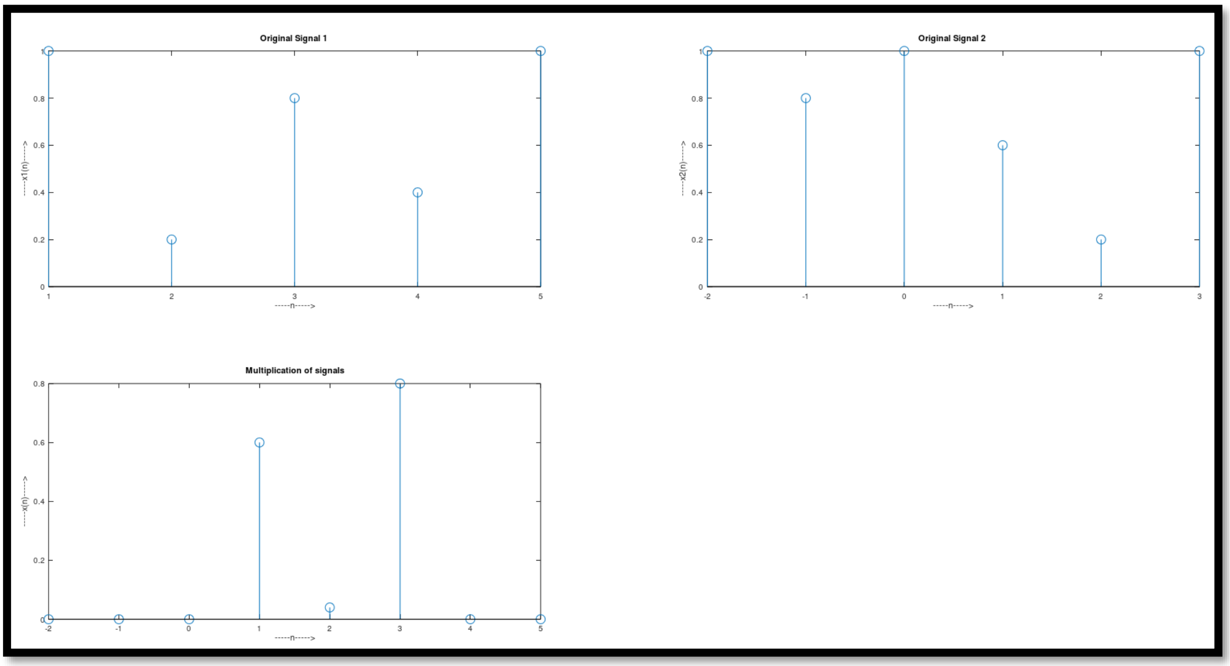
```
clc;clear;
x1=[1 0.2 0.8 0.4 1];
n1=1:5

x2=[1 0.8 1 0.6 0.2 1];
n2=-2:3

subplot(2,2,1);
stem(n1,x1);
title('Original Signal 1 ');
xlabel('-----n----->');
ylabel('-----x1(n)----->');

subplot(2,2,2);
```

```
stem(n2,x2);
title('Original Signal 2 ');
xlabel('-----n----->');
ylabel('-----x2(n)----->');

[r,n]=signaladdition(x1,x2,n1,n2);

subplot(2,2,3);
stem(n,r);
title('Multiplication of signals');
xlabel('-----n----->');
ylabel('-----x(n)----->');
```

Function:

```
function[result,n]=signaladdition(x1,x2,n1,n2);
n=min(min(n1),min(n2)):max(max(n1),max(n2));
a=zeros(size(n));
b=a;
a(find(n==min(n1)):find(n==max(n1)))=x1;
b(find(n==min(n2)):find(n==max(n2)))=x2;
result = (a.*b);
end
```

Output:



Conclusion:

Thus, we studied and performed different signal manipulation operations such as shifting, scaling, folding, addition & multiplication.

References:

1) https://www.allaboutcircuits.com/technical-articles/basic-signal-operations-in-dsp-time-shifting-time-scaling-and-time-reversal/

2) https://www.tutorialspoint.com/signals_and_systems/signals_basic_operations.htm

3) https://www.youtube.com/watch?v=04IocihsuC4