Heaven's Light is Our Guide

RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY

Department of Electrical & Computer Engineering

DIGITAL SIGNAL PROCESSING SESSIONAL

LAB REPORT

Number of Experiment: 03

Course Title: ***Digital Signal Processing Sessional***

Course No: ECE 4124

Date of Submission: May 15, 2023

Submitted by:	**Submitted to:**
Marjia Afroj	***Hafsa Binte Kibriya***
Series: 18	*Lecturer*
Roll: 1810029	*Dept. of ECE, RUET*
4th year Odd semester	
Dept. of ECE, RUET	

3.1 Experiment No: 03

3.2 Experiment Date: May 08, 2023

3.3 Experiment name: Study of correlation and implementation in Matlab

3.4 Theory:

In digital signal processing (DSP), correlation is a fundamental operation used to measure the similarity between two signals or sequences. It is commonly used for various applications, including signal analysis, pattern recognition, and communication systems. Correlation helps identify patterns, detect signal delays, estimate channel properties, and more.

There are two main types of correlation in DSP: cross-correlation and autocorrelation.

1. Cross-correlation: Cross-correlation measures the similarity between two different signals. It determines how much one signal resembles another when one is shifted over time. It is commonly used in applications such as audio processing, image registration, and communication systems.

Mathematically, the cross-correlation of two discrete-time signals x(n) and y(n) is given by:

$$\mathrm{R\_xy(n)} = \sum_{k=-\infty}^{\infty} \mathrm{x(k)} \, * \, \mathrm{y(k-n)}$$

2. Autocorrelation: Autocorrelation measures the similarity of a signal with itself at different time instants. It provides information about the periodicity or repeating patterns present in a signal. Autocorrelation is useful in applications like speech processing, audio analysis, and radar systems.

Mathematically, the autocorrelation of a discrete-time signal x(n) is given by:

$$\mathrm{R\_xy(n)} = \sum_{k=-\infty}^{\infty} \mathrm{x(k)} \, * \, \mathrm{x(k-n)}$$

3.5 Used Platform: MATLAB

3.6 Codes & Outputs:

1. Auto correlation:

```
clc;
clear all;
close all;

x = input('Enter your sequence: ');
h=fliplr(x);
x
h
C=x.'*h;
z = xcorr(x);

n1 = length(x);

n = (2*n1)-1;

y = zeros(1,n);

for i= 1:n
    for j = 1:n1
        k=i-j;
        m=i-j+1;
        if(k<n1 && m>0)
            y(i)=y(i)+C(m,j);
        end
    end
end
z
y
subplot(3,1,1);
stem(x);
title('The signal');
subplot(3,1,2);
stem(y);
title('Auto correlation without bilt in function');
subplot(3,1,3);
stem(z);
title('Auto correlation with built in function');
```

Output:

```
Enter your sequence: [-1 2 1]

x =

    -1     2     1


h =

     1     2    -1


z =

   -1.0000         0    6.0000         0   -1.0000


y =

    -1     0     6     0    -1
```

Figure 1.1: Output of Auto correlation

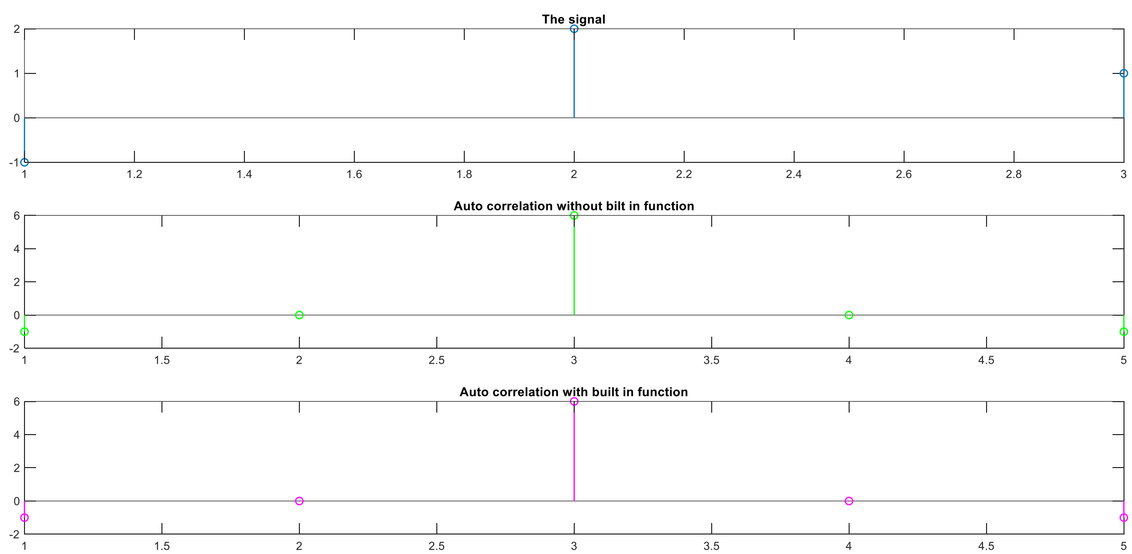

Figure 1.2: Graphical representation of auto correlation

2. Cross correlation:

```
clc;
clear all;
close all;

x = input('Enter your sequenceX: ');
h = input('Enter your sequenceY: ');
h1=fliplr(h);
C = x.'*h1;
z = xcorr(x,h);

n1 = length(x);
```

```
n2 = length(h1);


n = n1+n2-1;
y = zeros(1,n);


for i=1:n

    for j=1:n1
        s=i-j;
        m=i-j+1;
        if(s<n1 && m>0)
            y(i)=y(i)+C(m,j);
        end
    end
end
display(y);
display(z);

subplot(4,1,1);
stem(x);
title('Signal 1');
subplot(4,1,2);
stem(h);
title('Signal 2');
subplot(4,1,3);
stem(y);
title('Cross correlation without built in function');
subplot(4,1,4);
stem(z);
title('Cross correlation with built in function');
```

Output:

```
Enter your sequenceX: [1 2 3 4]
Enter your sequenceY: [4 2 3 1]

y =

     1     5    11    21    26    20    16


z =

    1.0000    5.0000   11.0000   21.0000   26.0000   20.0000   16.0000
```

Figure 1.3: Output of cross correlation

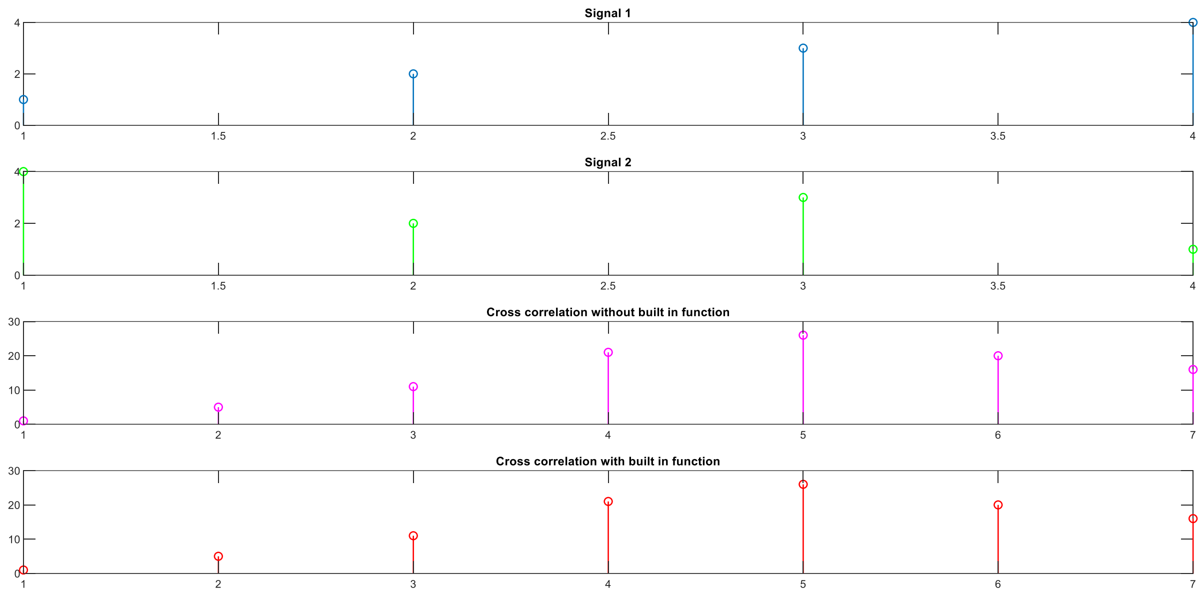

Figure 1.4: Graphical representation of cross correlation

3. Determining the delay of two signals using cross correlation:

```
clc;
close all;
clear all;

t = 0: 0.1: 100;
y1 = 10*sin((0.1)*pi*t);
y2 = 10*sin((0.1)*pi*(t-5));
z= xcorr(y1,y2);

[max_val, max_idx] = max(abs(z));
% Compute the time delay
time_delay = (max_idx - 1) / 10;
% Plot the cross-correlation
lag = -length(y1)+1:length(y1)-1;
figure;
subplot(2,1,1);
plot(t,y1);
hold on;
plot(t,y2);
xlabel('Time (s)');
ylabel('Amplitude');
title('Signal1 and Signal2');

subplot(2,1,2);
plot(lag/10, z);
xlabel('Time Delay (s)');
ylabel('Cross-correlation');
title('Cross-correlation of Signal1 and Signal2');

fprintf('The time delay between y1 and y2 is %.3f seconds.\n', time_delay);
```

Output:

```
The time delay between y1 and y2 is 95.100 seconds.
>>
```

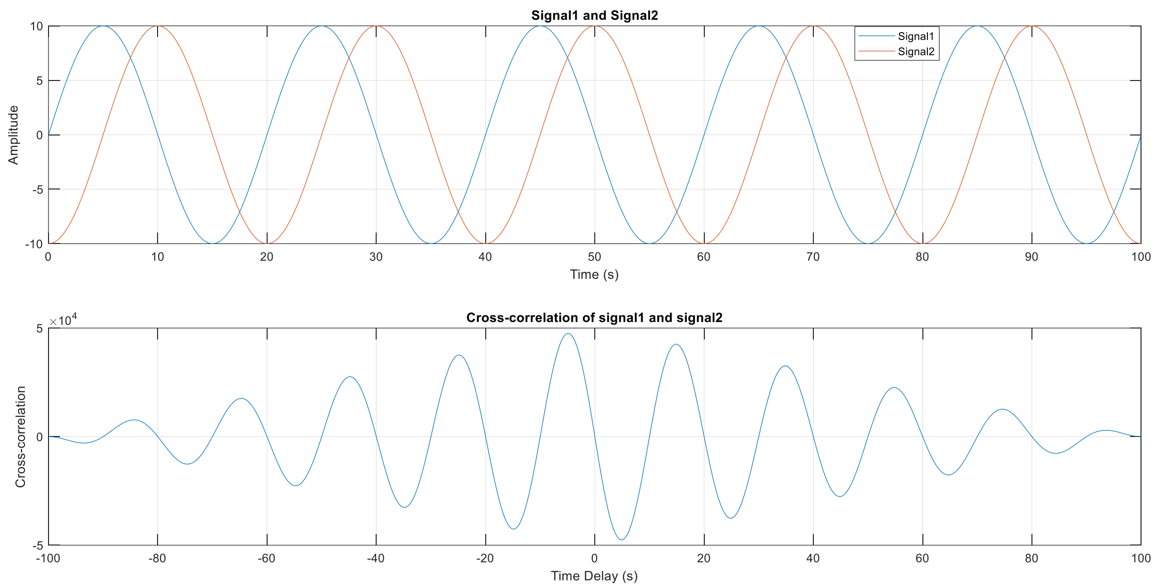

Figure 1.5: Graphical representation of time delay using cross correlation

3.7 Discussion:

The experiment was about to implement correlation in MATLB without using built in xcorr() function. It was implemented successfully using some for loop and mathematical logic.

Time delay between two signals was determined using cross correlation.

3.8 Conclusion:

The experiment was performed successfully. The output of the codes was similar to the theory.