

DEPARTMENT OF
ROBOTICS AND MECHATRONICS ENGINEERING

LAB REPORT

DIGITAL SIGNAL PROCESSING
(CSE-401)

Submitted By:

Md. Tahmeed Abdullah

Roll: SH-092-002

4th year 1st semester

Submitted To:

Mr. Sujan Sarker

Lecturer

Dept. of RME

Experiment no. 4

Name of the experiment

Using cross-correlation to find delay between input and output signals

Objectives

- To learn how to use linear filters.
- To understand the basics of cross-correlation.
- To learn how radar and other distance sensors use correlation to find distance of objects

Theory

In general, correlation describes the mutual relationship which exists between two or more things. The same definition holds good even in the case of signals. That is, correlation between signals indicates the measure up to which the given signal resembles another signal. In other words, if we want to know how much similarity exists between the signals 1 and 2, then we need to find out the correlation of Signal 1 with respect to Signal 2 or vice versa.

There are two types of Correlation in signal processing. These are:

- This is a kind of correlation, in which the signal in-hand is correlated with another signal so as to know how much resemblance exists between them. The cross-correlation of the discrete time signals $x[n]$ and $y[n]$ is expressed as

$$R_{xx}[m] = \sum_{n=-\infty}^{\infty} x[n]x[n-m]$$

- Auto Correlation is a type of correlation in which the given signal is correlated with itself, usually the time-shifted version of itself. The auto correlation of a discrete time signal $x[n]$ is

$$R_{xx}[m] = \sum_{n=-\infty}^{\infty} x[n]x[n-m]$$

A signal is transmitted and received by a digital device. If there is no delay between the signals, the auto-correlation of the transmitted and received signal will have a peak at $n = 0$, because the two signal are exactly equal to each other(if we neglect transmission losses), at $n = 0$.

If there is a delay between the two signals, the peak in the correlation signal will be shifted from the origin. The amount of shift determines the time delay between the two signals.

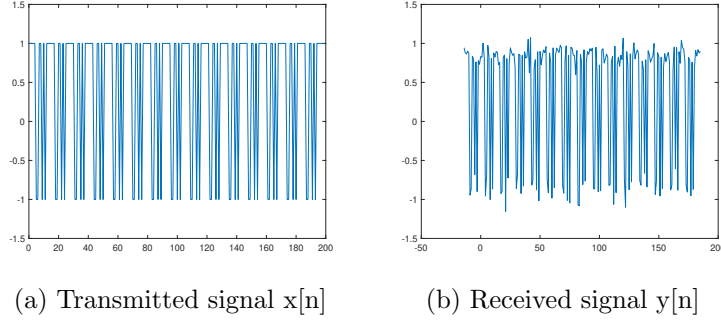


Figure 1: Transmitted and received signal

For a digital measuring device let $x[n]$ be the transmitted signal and $y[n]$ be the received signal where

$$y[n] = a \times x[n - D] + v$$

In figure 1a we have a signal $x[n]$ and in figure 1b we have the received signal $y[n]$. The input signal is attenuated by factor a , delayed by D samples and the noise induction in the signal is v .

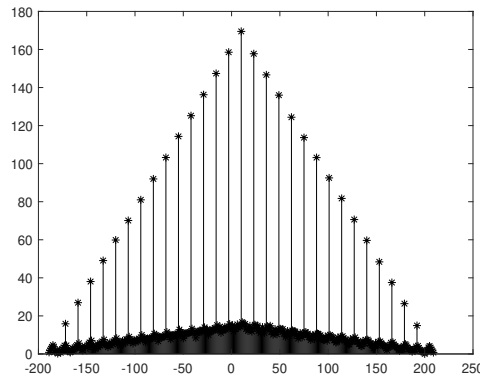


Figure 2: R_{xy} ; Cross correlation of x & y

If we calculate the cross correlation of the two signals we get the graph in figure 2. If observed, the peak is shifted by some sample N which is the

time delay between the two signals.

Implementation Code

main.m

```
1  clc; clear; close all;
2
3  %generating input signal
4  origin=1;
5  x = [1 1 1 1 1 -1 -1 1 1 -1 1 -1 1];
6  x = [x x x x x x x x x x x x x x x 1 1 1 1 1];
7  Xn = 0:199;
8
9  % Definitions
10 a = 0.85;
11 D=round(random('Uniform',10,20));
12 v = random('Normal' , 0, 0.1, [1 200]);
13
14 % generating output signal
15 [x,xn] =shift(x,Xn,D);
16 y = a*x + v;
17 yn = xn;
18
19 % cross correlate x,y
20 [rxy,lag] = ccor(x,Xn,y,yn);
21
22 % Find index of peak
23 [M, midx] = max(rxy);
24 lag(midx)
25
26 % plot the correlated signal
27 stem(lag,rxy,'k*');
```

Functions Used:

ccor.m

```
1  function [rxy,lag] = ccor(x,Xn,y,yn)
2      [y,yn] = fold(y,yn);
3      [rxy,lag] = convolute(x,Xn,y,yn);
4  end
```

convolute.m

```

1 function [outsignal, outIndex] = convolute(x,xidx,h,
    hidx)
2     [k, kidx] = fold(h,hidx);
3     l = size(x,2);
4     m = size(h,2);
5     y = zeros(1, l+m-1);
6     for i=1:l+m-1
7         for j=1:l
8             if i-j+1<=0 || i-j+1>m
9                 multi = 0;
10            else multi = h(i-j+1);
11                y(i) = y(i) + x(j)*multi;
12            end
13        end
14        outsignal = y ;
15        start = xidx(1,1)+hidx(1,1);
16        ending = xidx(1,l)+hidx(1,m);
17        outIndex = start:ending;
18 end

```

shift.m

```

1 function [outSignal,outindex] = shift(inSignal,
    index, shift)
2
3     outindex = index-shift;
4
5     outSignal = inSignal;
6
7 end

```

fold.m

```

1 function [out, index]= fold(in, n)
2     index = -fliplr(n);
3     out = fliplr(in);
4 end

```