BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING,
DHAKA-1000, BANGLADESH.

COURSE NO.: EEE 312

Experiment No. 2

Time domain analysis of discrete time signals and systems

% "The purpose of computing is insight, not numbers." – R.W. Hamming

In this experiment, mainly time domain properties of discrete time signals and systems will be analyzed. First of all different discrete time sequence generation and synthesis will be performed. Then LTI system response in terms of convolution and difference equation will be observed. Last but not the least, correlation of two signals, its uses in different fields will be analyzed.

Besides some MATLAB based practical pertinent examples will be presented too. Several related exercises and real-time applications are listed for proper comprehension of the theory.

Pre-lab Work:

1. Familiarize yourself with the experiment manual before attending the lab.
2. Practice the examples and exercises listed in this experiment at home for better class performance.
3. DO NOT bring any relevant MATLAB codes, neither in any portable device nor in written form.

Important MATLAB functions used in this experiment:

stem(), **fliplr**(), **min**(), **max**(), **conv**(), **filter**(), **xcorr**(), **randn**(), **rand**()

Part A

Generating basic sequences:

a) To implement unit impulse sequence

$$\delta(n-n_0)=\begin{cases}1, & n=n_0\\ 0, & n\neq n_0\end{cases} \quad \text{over} \quad n_1 \le n_0 \le n_2 \tag{1}$$

% Generate impulse sequence
% Say, n_1= -3 and n_2=3 (7 point sequence). Consider lag, n_o = -1

```
>>n1= -3;
>>n2= 3;
>>n=n1:n2;
>>no= -1;
>>x1=[(n-no) = = 0]; % using logical argument
>>stem(n,x1);
```

b) To implement unit step sequence

$$u(n-n_0)=\begin{cases}1, & n\ge n_0\\ 0, & n< n_0\end{cases} \quad \text{over} \quad n_1 \le n_0 \le n_2 \tag{2}$$

just set x_2 = [(n-n_o)>=0] in the above program.
Sample results are shown in Figure 1.

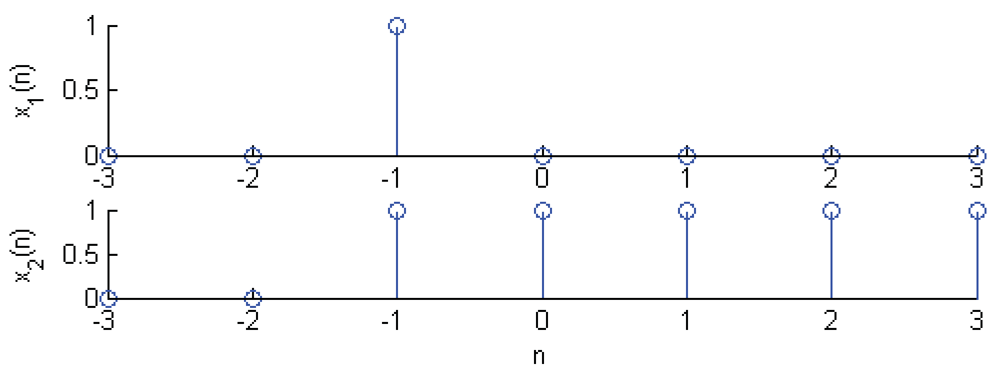


Figure 1: $x_1(n) = \delta(n+1)$ and $x_2(n) = u(n+1)$

Lab Exercise : A.1

Generate a ramp sequence (shown in Figure 2) following the above program.(length and lag must be variable).

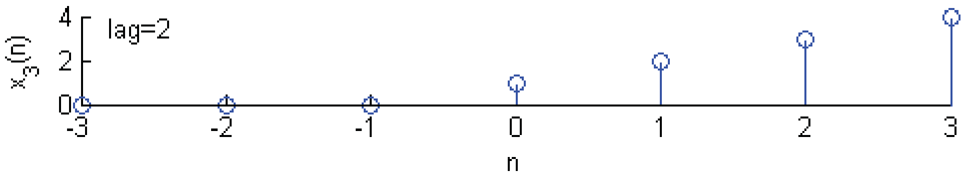


Figure 2: Result of Lab Exercise: A.1

Lab Exercise: A.2

Let n_1 denotes the time index for $x_1(n)$ and n_2 represent the time index for $x_2(n)$.Write a general program to find $x(n)=x_1(n)+x_2(n)$.The index of $x(n)$ will start with the minimum of n_1 and n_2 and end with the maximum of n_1 and n_2.

Given, $x_1(n)=\{\underset{\uparrow}{0}, 1, 2, 3\}$and $x_2(n)=\{0, \underset{\uparrow}{1}, 2, 3\}$.

Lab Exercise : A.3

The up-sampler is a discrete-time system defined by the input-output relation

$$y(n) = \begin{cases} x(\frac{n}{L}), & n = 0, \pm L, \pm 2L, ... \\ 0, otherwise \end{cases} \tag{3}$$

L is called the up-sampling factor which is a positive integer greater than 1. L-1 equidistant zero samples are inserted by the up-sampler between two input samples. It finds applications in sampling rate alteration process. Say input is a sine wave with frequency 0.36 rad/sec and L = 3 . Take time index, n = 1 to 52. Observe the up-sampled signal up to 52 samples. (Result shown in Figure 3)

Lab Exercise : A.4

Find the even and odd parts of the ramp signal obtained from Exercise 1.1
(Result shown in Figure 4)

Some helpful m-file routines:

```
function [y,n]=sigshift(x,m,n0) % For signal
shifting
n=m+n0;
y=x;

function [y,n]=sigfold(x,n) % For signal folding
y=fliplr(x);
n=-fliplr(n);
```

```
function [y,n]=sigadd(x1,n1,x2,n2) % For adding
%two signals
n=min(n1(1),n2(1)):max(n1(end),n2(end));
y1=zeros(1,length(n));
y2=y1;
y1(find((n>=n1(1))&(n<=n1(end))))=x1;
y2(find((n>=n2(1))&(n<=n2(end))))=x2;
y=y1+y2;
```

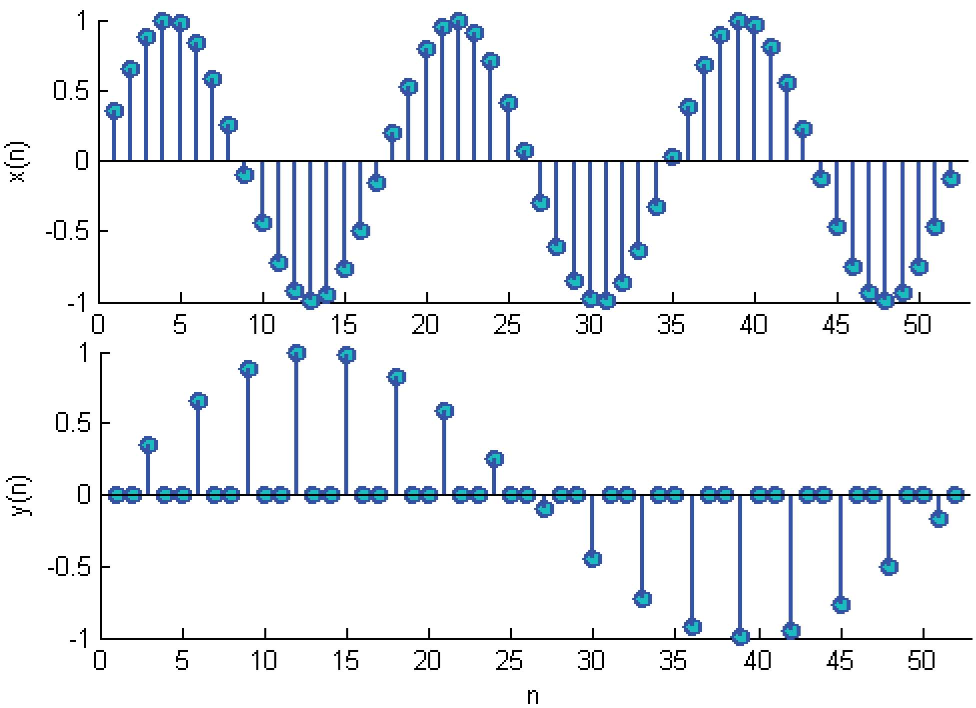


Figure 3 : Result of Lab Exercise : A.3

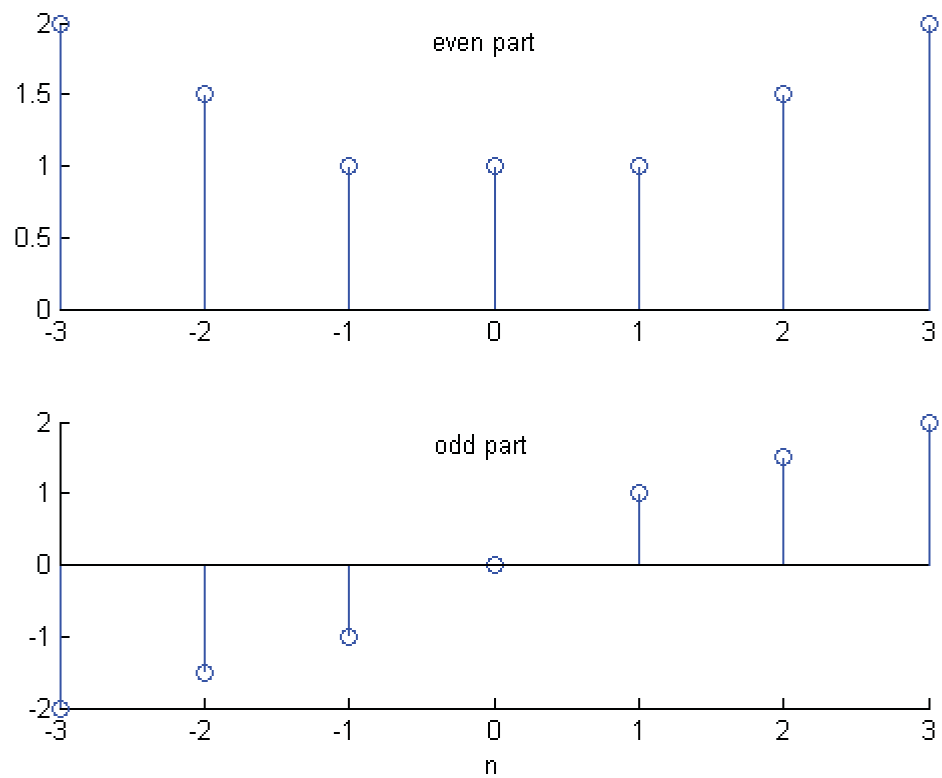


Figure 4 : Result of Lab Exercise : A.4

Part B

Response of LTI Systems to Arbitrary inputs: Convolution

For a relaxed LTI system, the response *y(n)* to a given input signal *x(n)* can be obtained if we know the system impulse response *h(n)* . Then response *y(n)* is given by the following relation

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) = x(n) * h(n) \tag{4}$$

which is known as the convolution sum.

Example:

Using **conv**() function , we can compute the convolution sum of two discrete time signals. But the outcome of conv() function does not reveal the timing information. So time index of the sum should be derived from the signals to be convolved. Recall that the lowest time index of the convolution sum is the sum of the lowest time indices of the two signals to be convolved. Same thing applies for highest time index of the sum. Let

$\mathbf{x}_1=[4\ 2\ \underset{\uparrow}{6}\ 3\ 8\ 1\ 5]$; $\mathbf{x}_2=[3\ 8\ 6\ 9\ \underset{\uparrow}{6}\ 7]$;

```
>>x1=[4 2 6 3 8 1 5];
>>n1=[-2:4];% generating index
>>x2=[3 8 6 9 6 7];
>>n2=[-4:1];
>>kmin=n1(1)+n2(1);% left edge of convolved result
>>kmax=n1(end)+n2(end); % right edge of convolved result
>>y=conv(x1,x2);
>>k=kmin:kmax; % generating index of the result
>>subplot(311),stem(n1,x1)
>>subplot(312),stem(n2,x2)
>>subplot(313),stem(k,y)
```

Lab Exercise: B.1

In the previous example, $x_1(n)$, $x_2(n)$ and y(n) are all of different lengths and scale, and this is far from ideal (Figure 5). Write a general program to bring three plots into same horizontal scale.(Figure 6)

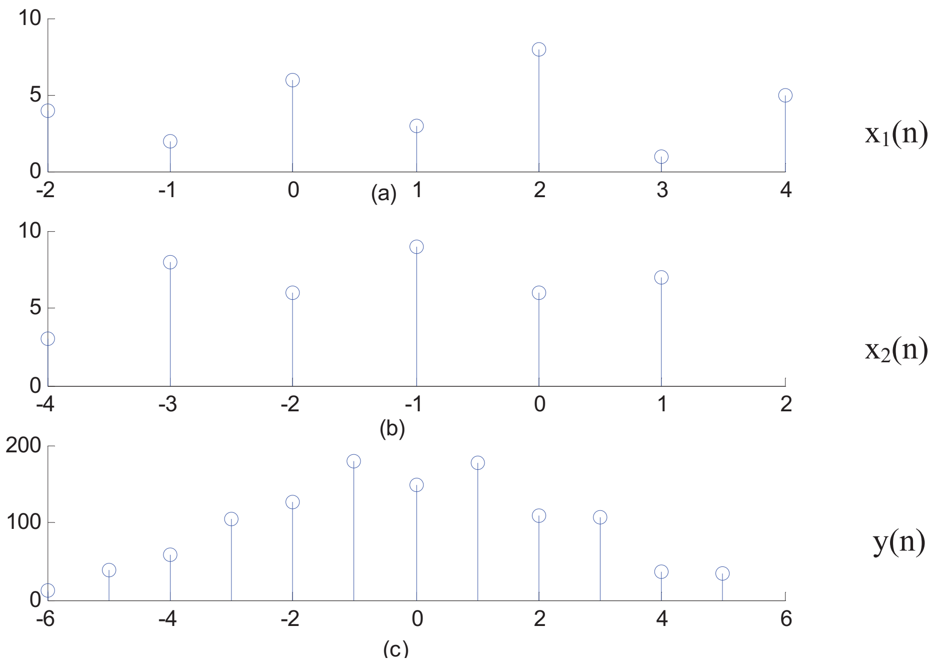


Figure 5 : With different horizontal scale

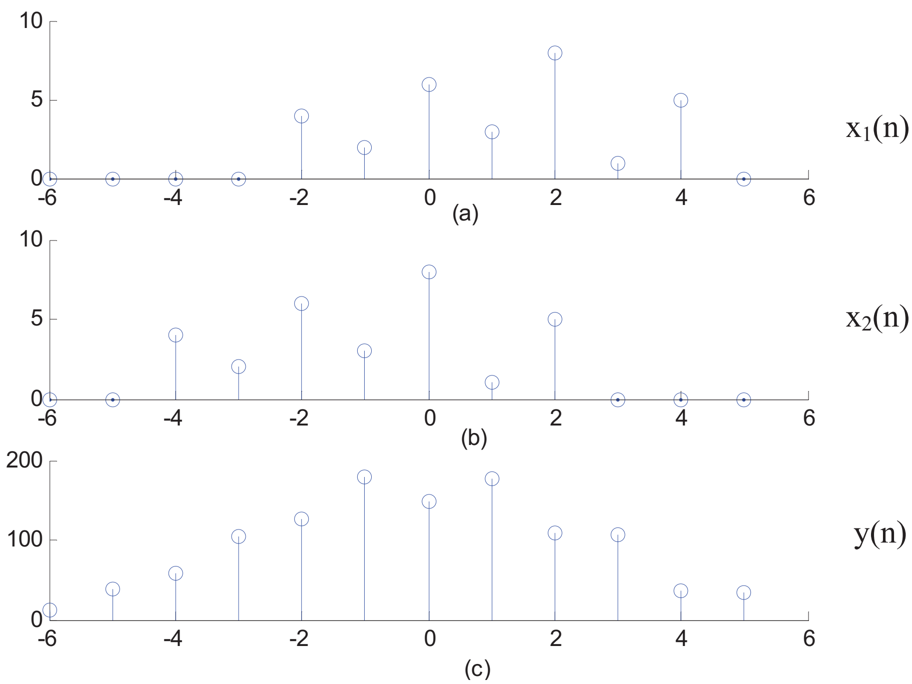


Figure 6 : With same horizontal scale

m-file which returns both convolved result and index :

```
function [y ny]=conv_m(x,nx,h,nh)
nyb=nx(1)+nh(1);
nye=nx(length(x))+nh(length(h));
ny=[nyb:nye];
y=conv(x,h)
```

Part C

System Response from difference equations:

An important subclass of LTI discrete time (DT) systems is characterized by a linear constant co-efficient difference equation of the form

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k) \quad (5)$$

where x(n) and y(n) are, respectively, the input and output of the system. (N≥M assumed). $\{a_k\}$ and $\{b_k\}$ are the coefficients. We can modify the equation as

$$y(n) = -\sum_{k=1}^{N} \frac{a_k}{a_0} y(n-k) + \sum_{k=0}^{M} \frac{b_k}{a_0} x(n-k)$$. Taking a_0 inside a_k and b_k,we get

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \quad (6)$$

Based on the above equation, a causal LTI system can be simulated in MATLAB using **filter**() function. Its general form is

y = filter(b,a,x); % remember co-efficient a_0 must be non-zero.
If a_0 is not equal to 1, filter() normalizes the system coefficients by a_0

Lab Exercise: C.1

Consider a system described by the following equation

y(n) + 0.6y(n-1)= x(n)

Use MATLAB to find impulse response, step response and sinusoidal response for x(n)=0.5sin(n)u(n) in the range -10≤n≤20. Verify the result using conv() function for step and sinusoidal responses.

Lab Exercise C.2:

Now consider

y(n) + 0.6y(n-1)= x(n)+x(n-2)

Obtain step response for this system applying superposition theorem. DO NOT use filter() function in that case. Verify your result using filter() function later on.

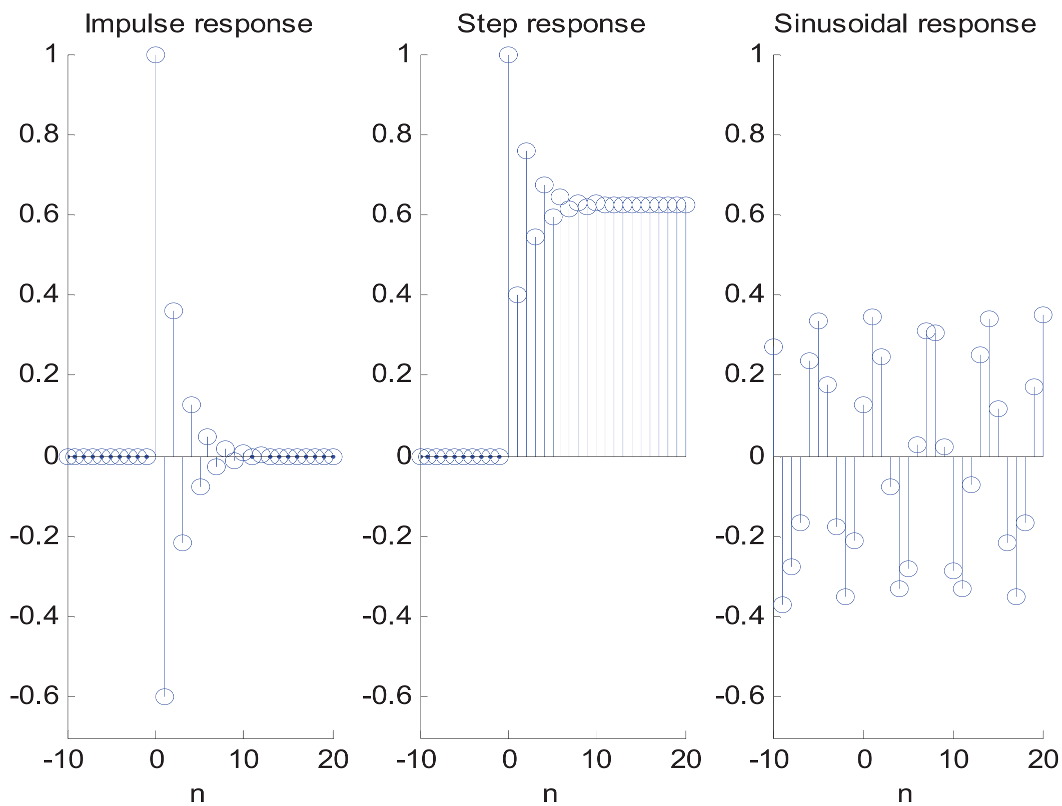


Figure 7 : Result of Exercise C.1

Part D

Correlation:

The cross-correlation of two deterministic real discrete-time finite energy sequences x(n) and y(n) is a third sequence $r_{xy}(l)$ defined as

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n - l)\,. \qquad (7)$$

In dealing finite duration sequences where x(n)=y(n)=0 for n<0 and n>N-1

$$r_{xy}(l) = \sum_{n=i}^{N-|k|-1} x(n)y(n - l) \qquad (8)$$

where $i = l$, $k = 0$ for $l \geq 0$ and $i = 0$,$k = l$ for $l < 0$(See Appendix for clarification)

A distinct peak in the CCF indicates that the two signals are matched for that particular time shift. This has important applications in signal detection and system identification.

Example:

Consider,

y(n)=x(n-2) where x(n)=[3 11 7 0 -1 4 2]. Then cross-correlating x(n) with y(n) we get
↑

In the following figure, it is apparent by inspection that the CCF is very small if k=0 but substantial if k=2. This is because y(n) lags behind x(n) by 2 samples and a match therefore occurs at 2 samples delay.

```
>>x = [3,11,7,0,-1,4,2];
>>n = -3:3;
>>[y,ny] = sigshift(x,n,2)
>>[x,nx] = sigfold(x,n);
>>[rxy,nxy] = conv_m(x,nx,y,ny);
>>stem(nxy,rxy/max(rxy))
```

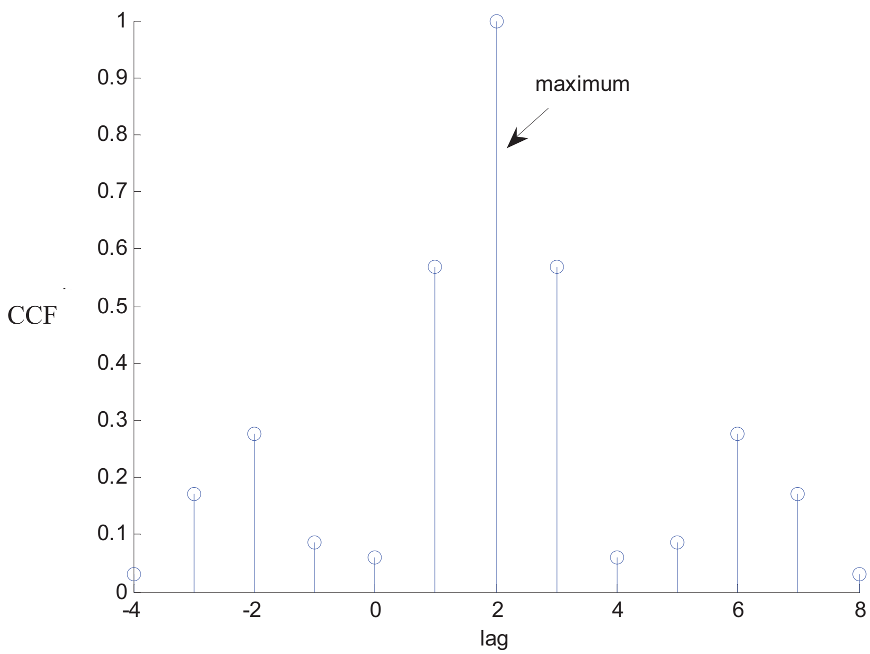


Figure 8 : Result of cross-correlation between x(n) and y(n)

There is a connection between convolution and correlation.

For convolution, we set $x(n) * y(n) = \sum_{n=-\infty}^{\infty} x(n)y(k-n)$

Then if we set, $x(n) * y(-n) = \sum_{n=-\infty}^{\infty} x(n)y(n-k)$ we get the desired result for CCF.

$$r_{xy}(k) = \sum_{n=-\infty}^{\infty} x(n)y(n-k) = x(n) * y(-n) \qquad (9)$$

So we can compute CCF by **conv**() function too.

r_{xy} = conv(x,fliplr(y));% Here **fliplr**() function folds y.

If the signals are ergodic then CCF is defined as

$$\hat{r}_{xy}(k) = \lim_{M \to \infty} \frac{1}{2M + 1} \sum_{n=-M}^{M} x(n)y(n - k) \tag{10}$$

For random signals we have to calculate $r_{xy}(k)$ as E[x(n)y(n-k)] to be exact. But to calculate an expected value accurately we need infinite data points. In practice data is finite so we take the estimated CCF as averaged result.

For N point periodic signals a simpler alternative is

$$\hat{r}_{xy}(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)y(n - k) \tag{11}$$

Similarly auto-correlation function can be defined. Just replace y(n) with x(n).

Note: ACF of a periodic signal is periodic.

<u>**Exercise: D.1**</u>

a) Determine the ACF of a sine wave sin(w_ot) analytically.(Homework)

b) Take period, T=2ms, tstep as T/100.

1. Take t=tstep:tstep:T ;
   Find ACF for x=2*sin(2*pi*t/T).
2. Repeat 1 for t=tstep:tstep:3*T

Explain for (b.2) why ACF magnitude decreases for successive periods? What happens if we set t(end)=5T , 10T etc.?

c) Observe the ACF of a random white noise sequence.

Applications of ACF/CCF:

1. Detecting a periodic input corrupted by AWGN

This is an example (See Figure 9-10) that illustrates the use of ACF to identify a hidden periodicity in an observed signal. This example will demonstrate how white noise gets suppressed in auto-correlation domain. For instance, you can observe the ACF of a white noise sequence which is characterized by an impulse (i.e noise power) at zero lag and nearly zero at other locations. The more the data points you take, the more the result matches an impulse.

```
%observe the result for different SNR(0,-5,-10 dB)
>>T=2e-3; % period=2ms
>>tstep=T/100;
>>t=tstep:tstep:3*T; % taking time index upto 3 periods
>>x=2*sin(2*pi*t/T); % Input
>>Px=sum(x.^2)/length(x); % Input power
>>SNR= -10; % in dB
>>Py=Px/10^(SNR/10);
>>n=sqrt(Py)*randn(1,length(t));% generate white noise
>>y=x+n; % Corrupted input
>>ACF_x=normalize(xcorr(x)); % Normalizing the peak to 1
>>ACF_n=normalize(xcorr(n));
>>ACF_y=normalize(xcorr(y));
>>ACF_y(length(x))=.4*max(ACF_y); %You can enable this line for better
%understanding. Scaling value should be decreased for lower SNR
>>figure(1)
>>subplot(211),plot(t,x)
>>subplot(212),plot(t,n)
>>figure(2)
>>subplot(221),plot(tstep*(1:length(ACF_x)),ACF_x)% showing ACF w.r.t. time
>>subplot(222),plot(tstep*(1:length(ACF_y)),ACF_y)% showing ACF w.r.t. time
% hold on

function R=normalize(x)
R=x/max(x);
```

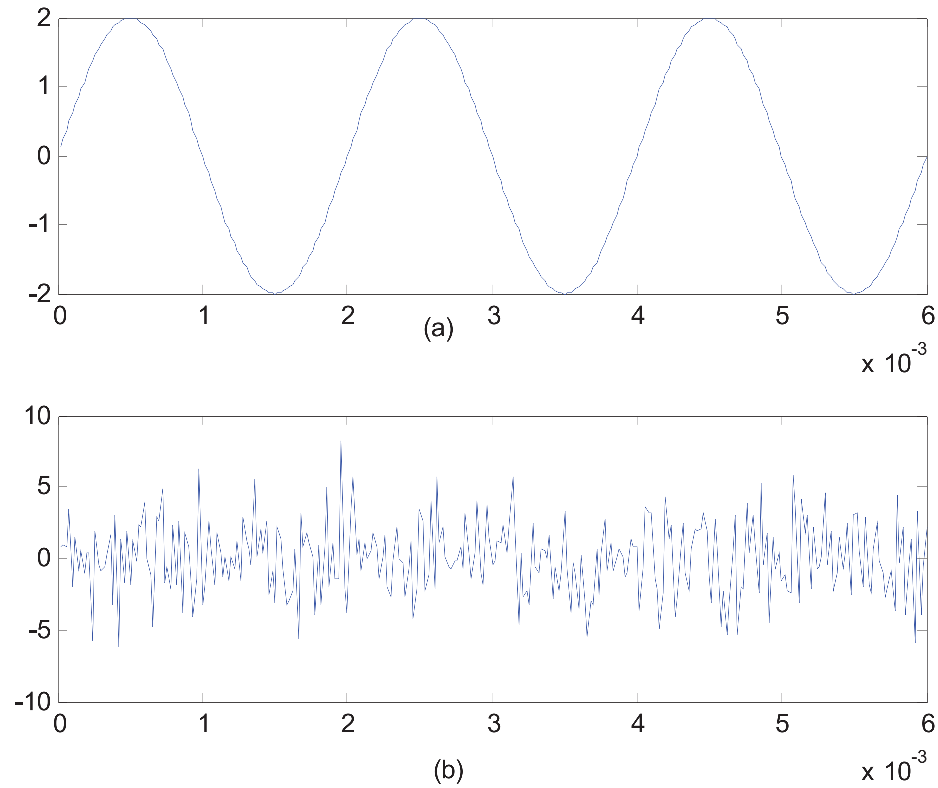


Figure 9: (a) Input periodic wave (b) Input corrupted by AWGN for -5dB SNR

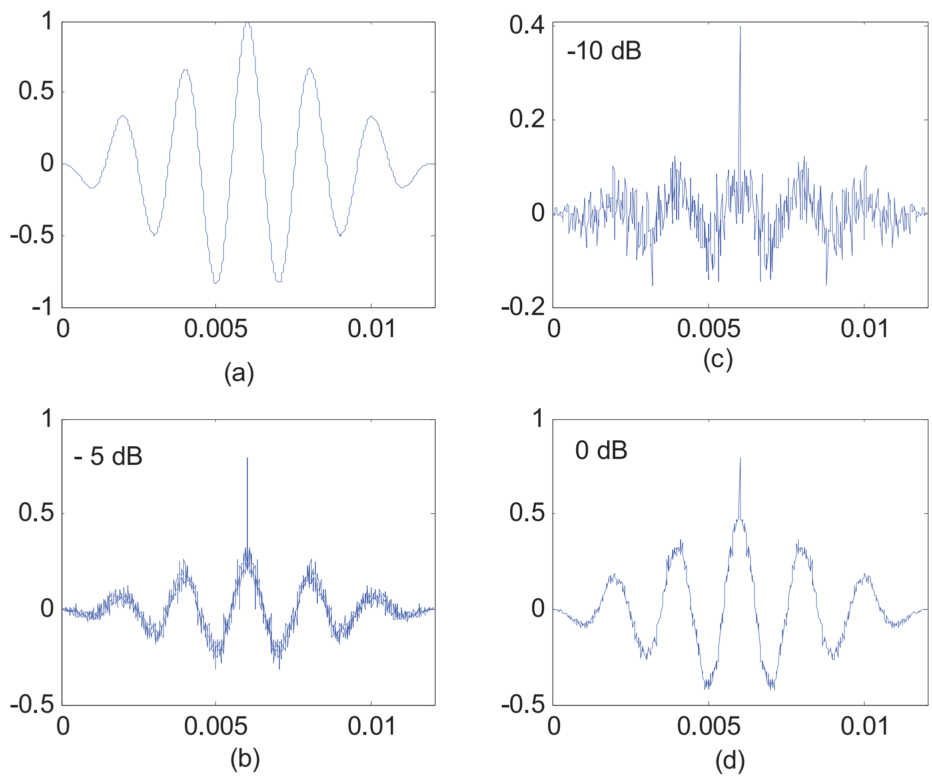


Figure 10: (a) ACF of input (b,c,d) ACF of noisy sequence.

Lab Exercise D.2

Derive a technique to obtain the original periodic signal if you have ACF of the noisy sequence. Test your technique for the cases displayed above. If the signal has a phase shift, could you recover it from ACF? Explain.

2. Estimation of impulse response:

Output of a relaxed system is given by y(n)=h(n) * x(n). If we cross-correlate the output of the system with a noisy input with normal pdf , then the cross-correlation

$$R_{yr}(k) = y(n) * r(-n) = [r(n) * h(n)] * r(-n) = r(n) * r(-n) * h(n) = R_{rr}(k) * h(n) \qquad (12)$$

where r(n) is a white noise input, $R_{rr}(k)$ is the auto-correlation of the noise input. As the auto-correlation of white noise sequence is like impulse sequence (In other words, the input noise has a much greater bandwidth than that of the system).we can write

$$R_{yr}(k) \approx h(n) \qquad (13)$$

Therefore, we say that if we correlate the white noise input with the output of the system, we will have an approximation of the impulse response of the system (See Figure 11).

Consider a system:

$$y(n) + 0.6y(n-1) = x(n)$$

```
%estimating system impulse response using cross-correlation
>>N=500;
>>nr=0:499;
>>ny=nr;
>>r=randn(1,N);
>>y=zeros(size(r));
>>for n=2:500
        y(n)=r(n)-0.6*y(n-1);
   end
>>rr=fliplr(r);
>>nrr=-fliplr(nr);
>>Ryr=conv(y,rr); % Calculating correlation
>>kmin=ny(1)+nrr(1);
>>kmax=ny(length(ny))+nrr(length(nrr));
>>k=kmin:kmax; % generating index
>>subplot(211),stem(k,Ryr/Ryr(N))
>>title('Approximate Impulse Response');
>>num=[1 0];
>>den=[1 0.5];
```

```
>>n=0:499;
>>x=zeros(size(n));
>>x(1)=1;
>>yy=filter(num,den,x);
>>subplot(212),stem(n,yy)
>>title('Actual Impulse Response')
```

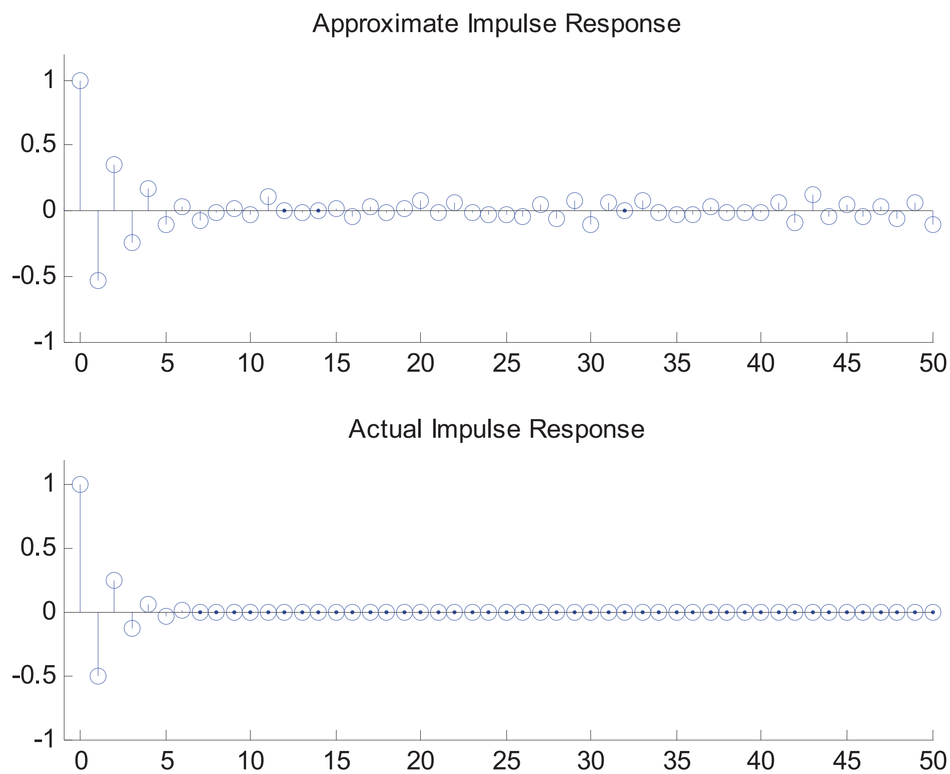


Figure 11: Actual and Estimated Impulse Response.

End of Experiment Exercises:

1. Write a general program to compute CCF of two DT signals. Signals can have different time scale. DO NOT use conv() or xcorr() function. Say two inputs are

$x(n)=\{ \ldots.0,1,\underset{\uparrow}{1},3,5,7,2,0\ldots.\}$ and $y(n)=\{ \ldots.0,5,5,5,3,3,\underset{\uparrow}{1},1,0\ldots.\}$

You can take insight from (8).This problem is just for proper understanding about what is happening inside conv() and xcorr() functions.

2. Detection of signals in noise by auto-correlation

Consider a radar transmitting a short tone burst of EM energy(Figure 12-a) and a weak echo from a distant target(Figure 12-b).In the absence of noise weak echo can be amplified and there is no problem detecting it. If there is background of noise whose amplitude exceeds that of the echo, the echo will be masked and not detectable((Figure 12-c),echo+noise).As we know noise is suppressed in ACF domain, then write a general program to detect the echo using correlation (Figure 12-d). If the tone burst has a gradual increase in frequency, what will happen? Is it advantageous for detection? Explain with necessary simulated figures.

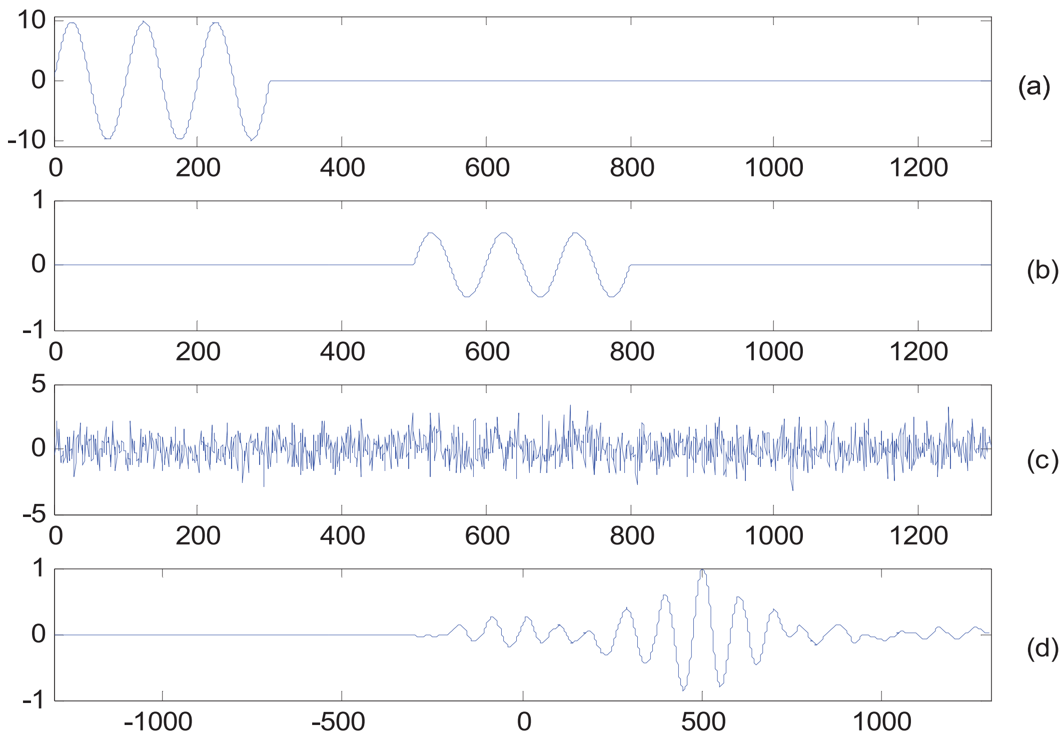


Figure 12:(a) Transmitted tone burst (b)Received weak echo(c)Received echo buried into background noise(d)CCF between (a) and (c) to locate a weak echo. It shows that after 500 units delay an echo arrives (location of the peak).

3. Detection of a transmitted sequence

Let in the transmitter, to transmit zero (**0**) we send $x_0(n)$ for $0 \le n \le$ L-1 and to transmit one (**1**) we send $x_1(n)$ for $0 \le n \le$ L-1 where $x_1(n) = - x_0(n)$. The signal received by the receiver

$$y(n) = x_i(n) + w(n) \quad i = 0,1 \text{ and } 0 \le n \le L-1 \tag{14}$$

$w(n)$ is additive white noise.
Present a technique to detect the sequence transmitted from $y(n)$.Assume that particular receiver knows $x_0(n)$ and $x_1(n)$.Write a general MATLAB program for this purpose.

4. Signal smoothing by a moving average (MA) system:

From (6) if we set $\{a_k\}=0$ for $k = 1,\ldots,N$ then (6) turns into

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) \tag{15}$$

which is a non-recursive linear time-invariant system. This system takes most recent M+1 points and add them after weighting. This type of system is called moving average (MA) system.

Simple working MA system can be expressed as

$$y(n) = \frac{1}{M} \sum_{k=0}^{M-1} x(n-k)\,. \tag{16}$$

Such a system is often used in smoothing random variations in data.

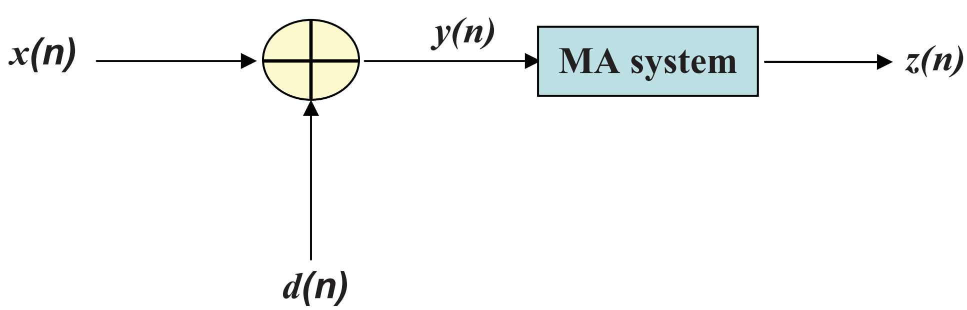


Figure 13: Signal smoothing by MA system

Consider, $x(n) = 2n(0.9)^n$. Take 50 point signal.
Noise, d = rand(1,50)-0.5; % to make noise bipolar
Take M=3.Write a sample program for the whole system shown (See Figure 14). Plot the M versus normalized error curve.

Normalized error, $e(n) = \left|\dfrac{x(n) - z(n)}{x(n)}\right|$

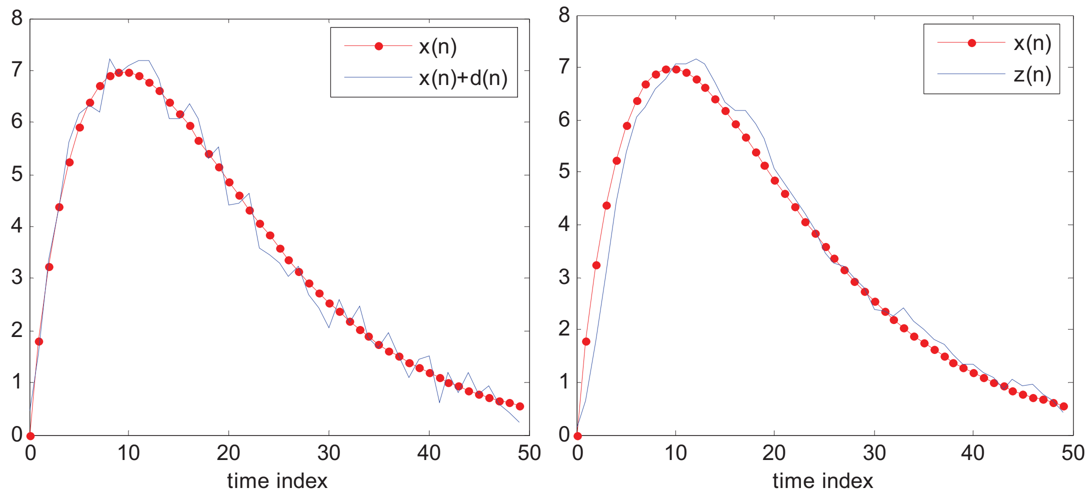
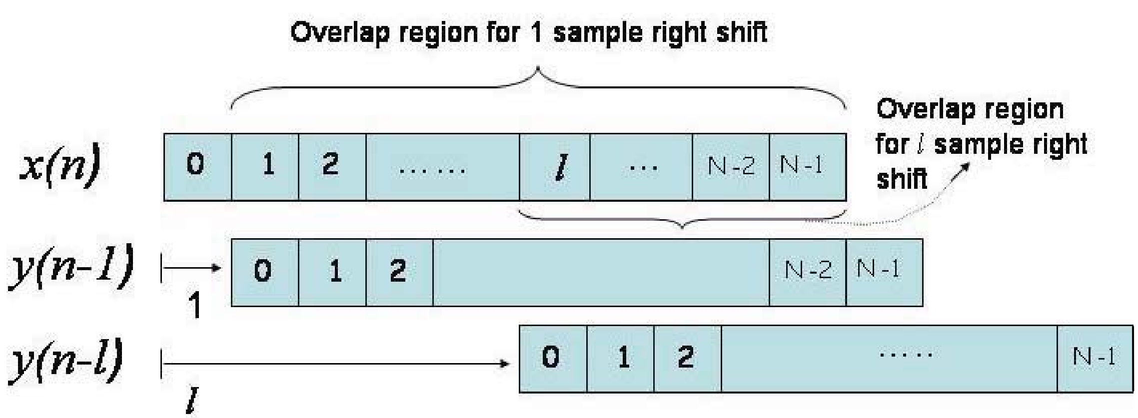


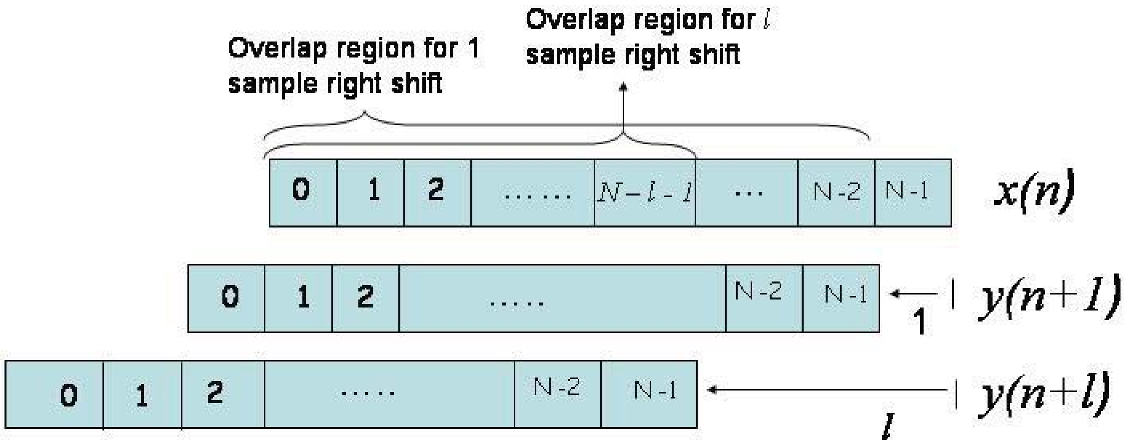
Figure 14: (left) *x(n)*=original sequence and *y(n)=x(n)+d(n)*,corrupted sequence. (right) *z(n)* ,output of the MA system and *x(n)*

Appendix :



$$r_{xy}(l) = \sum_{n=i}^{N-|k|-1} x(n)y(n-l) \quad \text{where } i=l,\ k=0 \text{ for } l \geq 0$$

(a)



$$r_{xy}(l) = \sum_{n=i}^{N-|k|-1} x(n)y(n-l) \quad \text{where } i=0,\ k=l \text{ for } l<0$$

(b)

Figure 15: (a , b)illustrates cross-correlation of two finite N point sequences (equation 8).

References:

1) Proakis & Manolakis,"Digital Signal Processing:Principles,Algorithms and Applications.",Chapter 2, 3rd Edition , Prentice Hall Ltd.
2) Mitra," Digital Signal Processing: A Computer Based Approach" Chapter 2, Edition 1998, Tata McGraw -Hill Co. Ltd.
3) Denbigh, "System Analysis and Signal Processing" Chapter (2,16), Edition 1998, Addison-Wesley.
4) Elali, "Discrete Systems and Digital Signal Processing with MATLAB® " Chapter 2,Edition 2004, CRC Press
5) Ingle & Proakis, "Digital Signal Processing using *MATLAB®* " Chapter 2,Edition 2000 Thomson-Brooks/Cole Co Ltd.

Laboratory manual for this experiment is prepared by

TOUFIQUL ISLAM
Lecturer, Dept of EEE, BUET

and supervised by

Dr. Md. KAMRUL HASAN
Professor, Dept of EEE, BUET

June 9, 2007