Phase 1

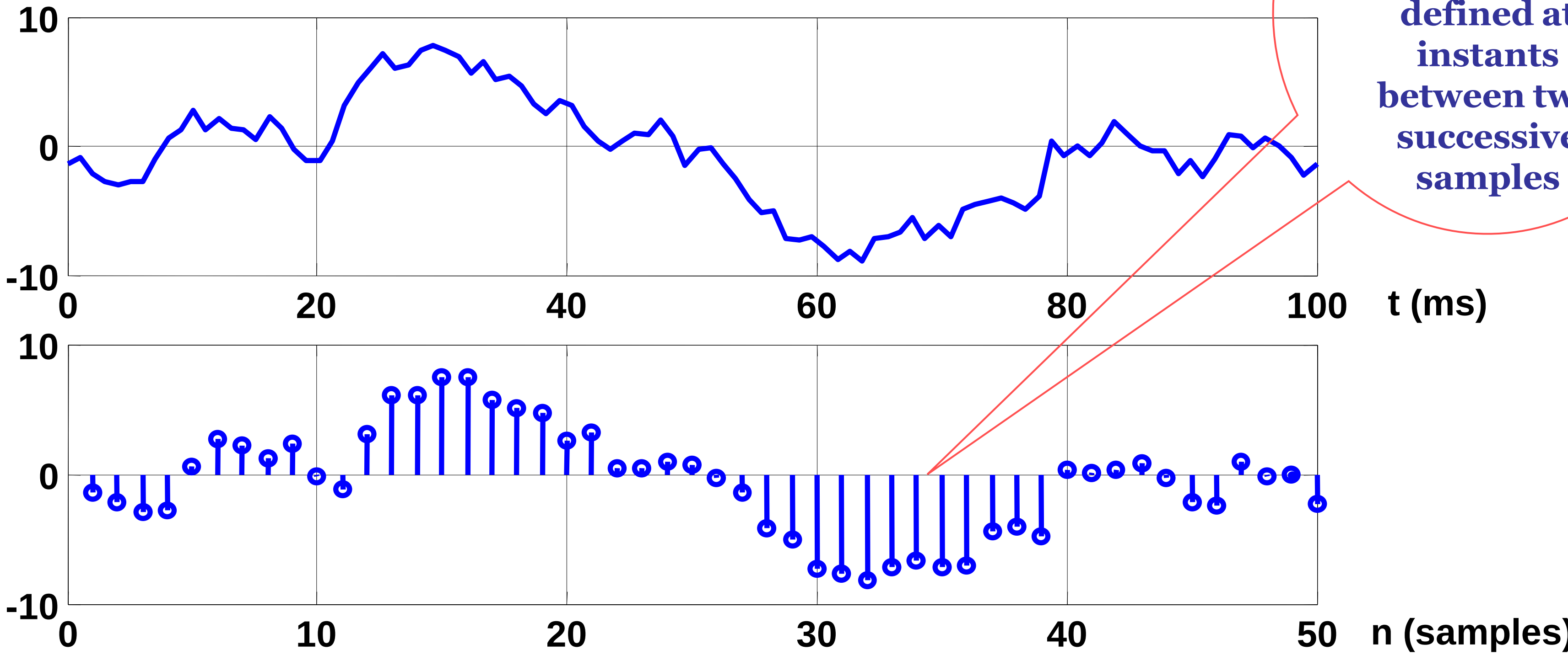
Topics

- Overview of discrete time signals and systems
- Generation and synthesis of basic sequences
- Convolution
- Response from system difference equation
- Correlation

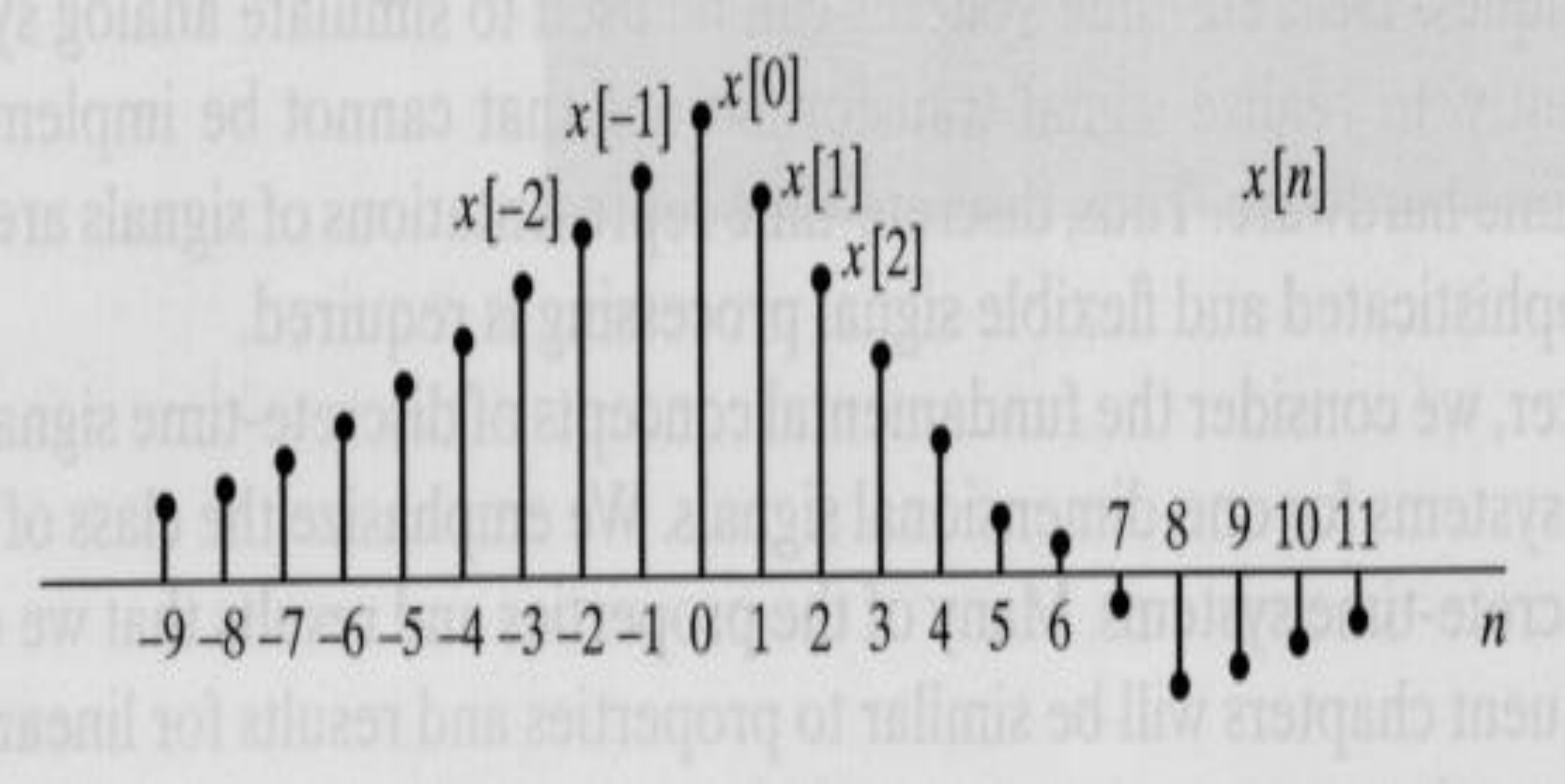
Discrete-Time(DT) Signals: Sequences

- Discrete-time signals are represented by sequence of numbers
  - The nth number in the sequence is represented with x[n]
- Often DT sequences are obtained by sampling of continuous-time signals
  - In this case x[n] is value of the analog signal at $x_c(nT)$
  - Where T is the sampling period

It is important to note that a discrete-time signal is not defined at instants between two successive samples



- Graphical Representation of a Discrete-Time-Signal:



Basic Sequences and Operations

- Delaying (Shifting) a sequence

$$y[n] = x[n - n_o]$$

- Unit sample (impulse) sequence

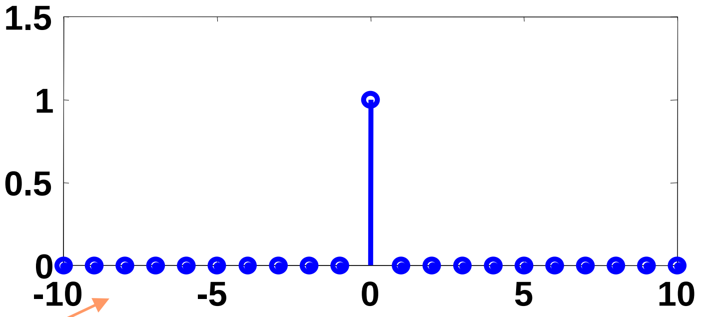
$$\delta[n] = \begin{cases} 0 & n \neq 0 \\ 1 & n = 0 \end{cases}$$

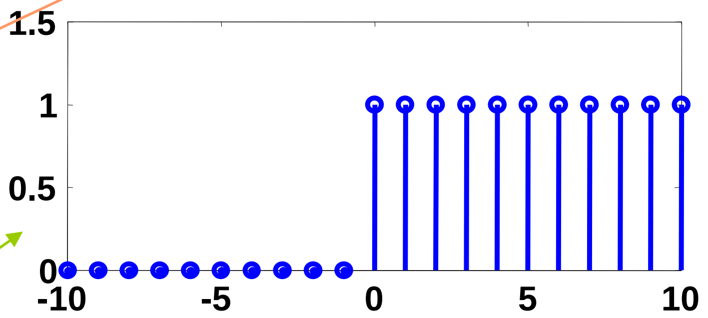
- Unit step sequence

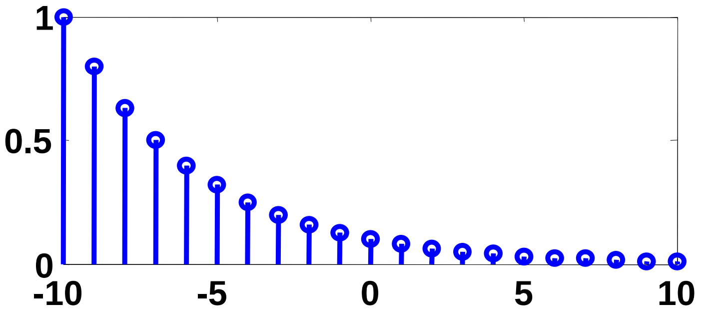
$$u[n] = \begin{cases} 0 & n < 0 \\ 1 & n \geq 0 \end{cases}$$

- Exponential sequences

$$x[n] = A\alpha^n$$







Sinusoidal Sequences

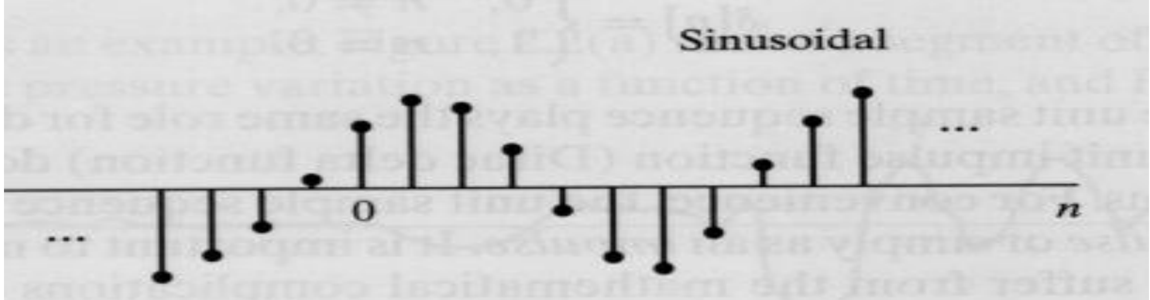
- Important class of sequences

$$x[n] = \cos(\omega_o n + \phi)$$

- An exponential sequence with complex $\alpha = |\alpha| e^{j\omega_o}$ and $A = |A| e^{j\phi}$

$$x[n] = A\alpha^n = |A| e^{j\phi} |\alpha|^n e^{j\omega_o n} = |A||\alpha|^n e^{j(\omega_o n + \phi)}$$

$$x[n] = |A||\alpha|^n \cos(\omega_o n + \phi) + j|A||\alpha|^n \sin(\omega_o n + \phi)$$



(d)

Discrete-Time Systems

- Discrete-Time Sequence is a mathematical operation that maps a given input sequence x[n] into an output sequence y[n]

$$y[n] = T\{x[n]\} \qquad x[n] \longrightarrow \boxed{T\{.\}} \longrightarrow y[n]$$

- Example Discrete-Time Systems
  - Moving (Running) Average

$$y[n] = x[n] + x[n-1] + x[n-2] + x[n-3]$$

  - Maximum

$$y[n] = \max\{x[n], x[n-1], x[n-2]\}$$

  - Ideal Delay System

$$y[n] = x[n - n_o]$$

Memoryless System

- Memoryless System
  - A system is memoryless if the output y[n] at every value of n depends only on the input x[n] at the same value of n

- Example Memoryless Systems
  - Square

$$y[n] = (x[n])^2$$

  - Sign $\quad y[n] = \text{sign}\{x[n]\}$

- Counter Example
  - Ideal Delay System $\quad y[n] = x[n - n_o]$

Linear Systems

- Linear System: A system is linear if and only if

$$T\{x_1[n]+x_2[n]\} = T\{x_1[n]\}+T\{x_2[n]\} \quad \text{(additivity)}$$

and

$$T\{ax[n]\} = aT\{x[n]\} \quad \text{(scaling)}$$

- Examples
  - Ideal Delay System $\quad y[n] = x[n-n_o]$

$$\begin{aligned}
T\{x_1[n]+x_2[n]\} &= x_1[n-n_o]+x_2[n-n_o] \\
T\{x_2[n]\}+T\{x_1[n]\} &= x_1[n-n_o]+x_2[n-n_o] \\
T\{ax[n]\} &= ax_1[n-n_o] \\
aT\{x[n]\} &= ax_1[n-n_o]
\end{aligned}$$

Time-Invariant Systems

- Time-Invariant (shift-invariant) Systems
  - A time shift at the input causes corresponding time-shift at output

$$y[n] = T\{x[n]\} \Rightarrow y[n-n_o] = T\{x[n-n_o]\}$$

- Example
  - Square

$$y[n] = (x[n])^2$$

Delay the input the output is $y_1[n] = (x[n-n_o])^2$

Delay the output gives $y[n - n_o] = (x[n-n_o])^2$

- Counter Example
  - Compressor System

$$y[n] = x[Mn]$$

Delay the input the output is $y_1[n] = x[Mn-n_o]$

Delay the output gives $y[n - n_o] = x[M(n-n_o)]$

Causal System

- Causality
  - A system is causal if it's output is a function of only the current and previous samples

- Example
  - Backward Difference

$$y[n] = x[n] - x[n-1]$$

- Counter Example
  - Forward Difference

$$y[n] = x[n+1] + x[n]$$

Stable System

- Stability (in the sense of bounded-input bounded-output BIBO)
  - A system is stable if and only if every bounded input produces a bounded output

$$|x[n]| \leq B_x < \infty \Rightarrow |y[n]| \leq B_y < \infty$$

- Example
  - Square

$$y[n] = (x[n])^2$$

if input is bounded by $|x[n]| \leq B_x < \infty$

output is bounded by $|y[n]| \leq B_x^2 < \infty$

- Counter Example
  - Log

$$y[n] = \log_{10}(|x[n]|)$$

even if input is bounded by $|x[n]| \leq B_x < \infty$

output not bounded for $x[n] = 0 \Rightarrow y[0] = \log_{10}(|x[n]|) = -\infty$

Topics

- Overview of discrete time signals and systems
- Generation and synthesis of basic sequences
- Convolution
- Response from system difference equation
- Correlation

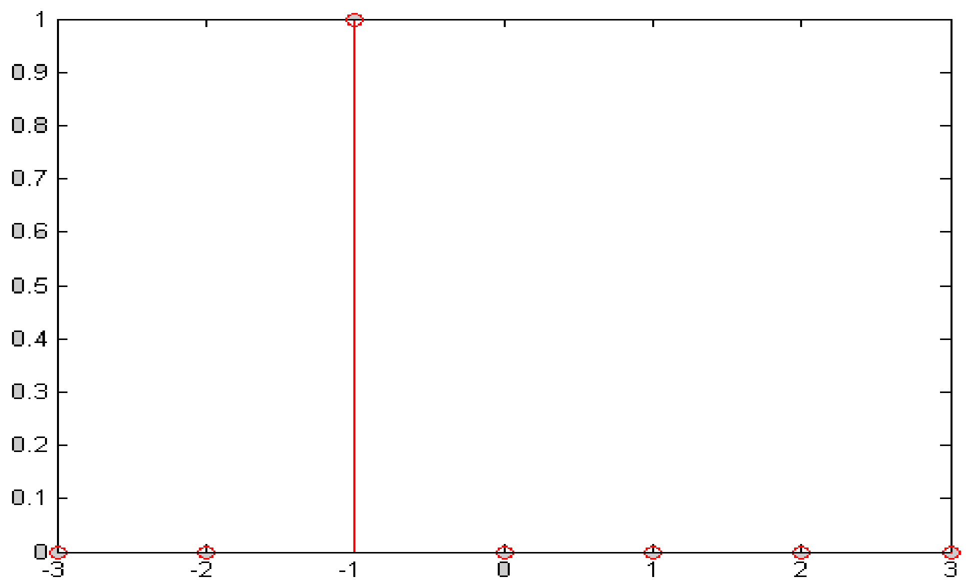
Generation and synthesis of basic sequences

PART-A: Implementation of unit impulse sequence:

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

consider, n_1= -3 and n_2=3 (7 point sequence).

lag, n_o = -1



Generation and synthesis of basic sequences

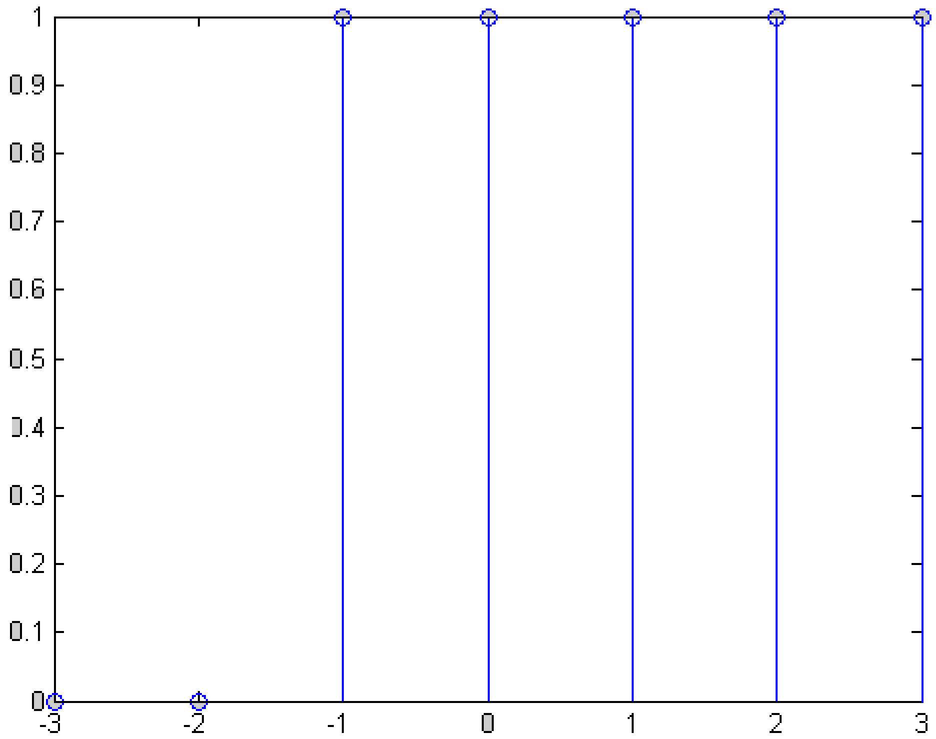
(a) Implementation of unit impulse sequence:

MATLAB Implementation:

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

Generation and synthesis of basic sequences

(b) Implementation of unit step sequence :



Only change : x2 = [(n-no)>=0] in the above program

Generation and synthesis of basic sequences

- **Lab Exercise : A.3**
- up-sampler

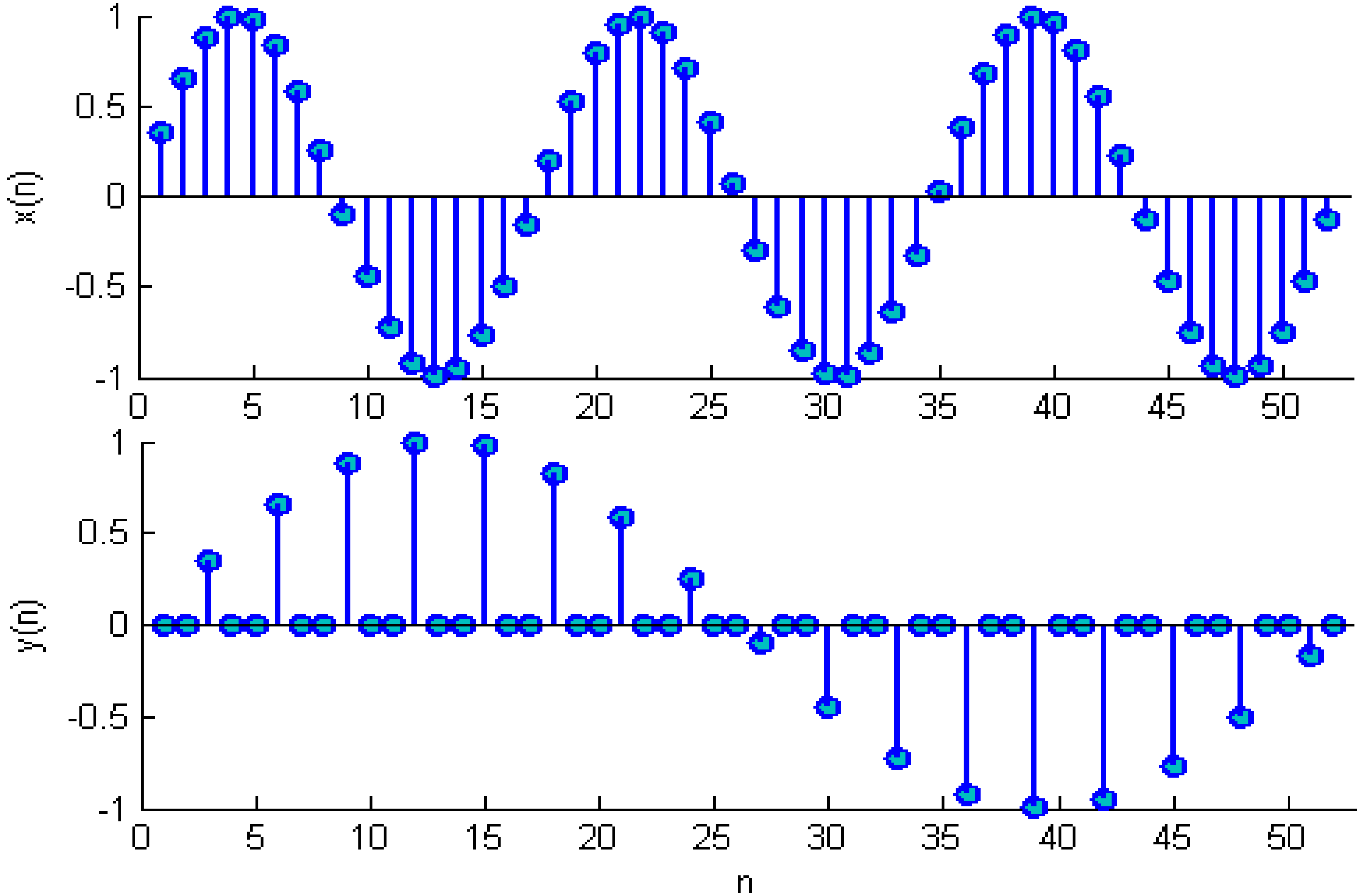
$$y(n) = \begin{cases} x(\frac{n}{L}), n = 0, \pm L, \pm 2L, \ldots \\ 0, otherwise \end{cases}$$

L is called the up-sampling factor which is a positive integer greater than 1.

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.

Generation and synthesis of basic sequences

- Result: up-sampler

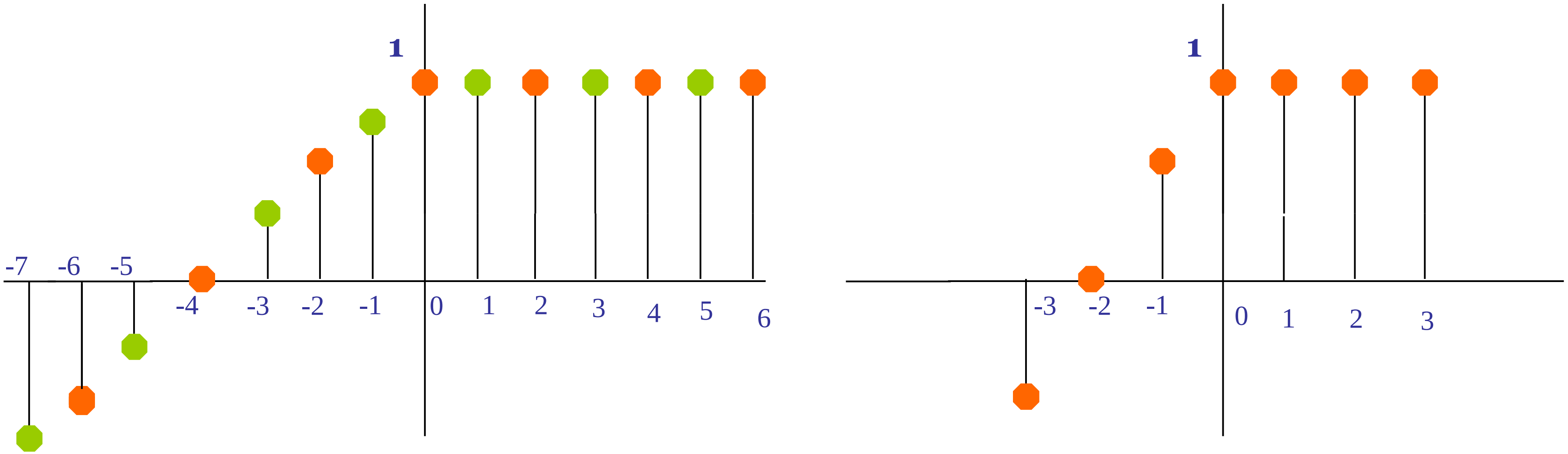


Generation and synthesis of basic sequences

- Down-sampler

$$y(n) = x(nL), n = 0, \pm 1, \pm 2, \ldots$$

L=2



x(n)

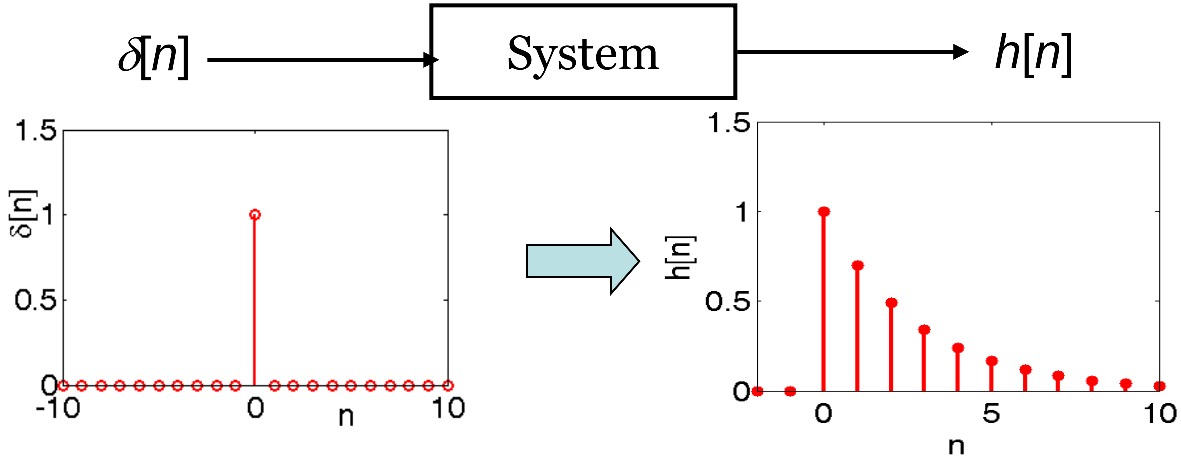
y(n)=x(2n)

Topics

- Overview of discrete time signals and systems
- Generation and synthesis of basic sequences
- Convolution
- Response from system difference equation
- Correlation

Convolution

- A very important way to analyze a system is to study the response signal, $h[n]$, when the input is an impulse $\delta[n]$



- Loosely speaking, this corresponds to giving the system a unit kick at $n=0$, and then seeing what happens.

- The specific notation, $h[n]$, is used to denote the **impulse response signal**, rather than the more general $y[n]$.

Convolution

- x(n) = input signal
- h(n) = impulse response of the system
- y(n) = output response of the system for the input x(n)

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

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

which is known as the convolution sum.

Convolution

- *h(n)* contains all the information displayed in a difference equation.
- Consider the following difference equation

$$y[n] = x[n] - x[n-1] + 2x[n-3]$$

- From convolution equation

$$y[n] = \sum_{k=0}^{\infty} h(k)x(n-k)$$

$$= h(0)x(n-0) + h(1)x(n-1)$$

$$+ h(2)x(n-2) + h(3)x(n-3)$$

$$= 1.x(n) + (-1)x(n-1)$$

$$+ 0.x(n-2) + 2.x(n-3)$$

***h(n)* ={1,-1,0,2}**

Convolution

The digital convolution steps are summarized as:

1. Flip (reverse) one of the digital functions.
2. Shift it along the time axis by one sample.
3. Multiply the corresponding values of the two digital functions.
4. Summate the products from step 3 to get one point of the digital convolution.
5. Repeat steps 1-4 to obtain the digital convolution at all times that the functions overlap.

Go to internet (JHU website)

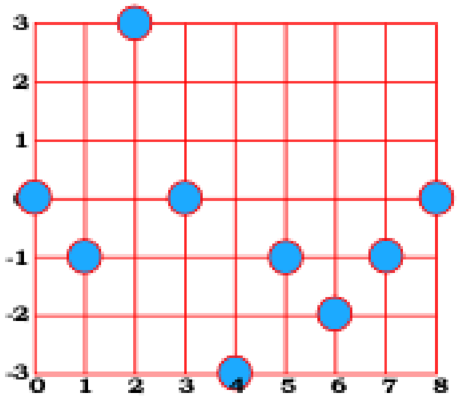
Convolution

For example, let x= {-1,3,0,-3,-1,-2,-1}
and h = {1,-2,1,}

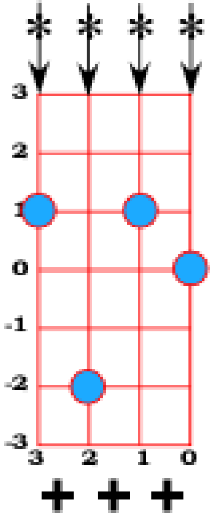
Convolution of these two discrete signals, y={-1,5,-7,0,5,-3,2,0,-1}

Diagrammatically

Convolution



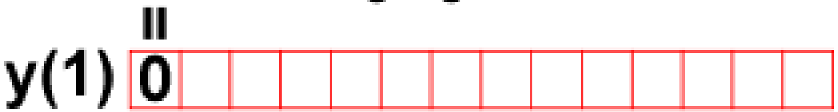
x=[0 -1 3 0 -3 -1 -2 -1 0]



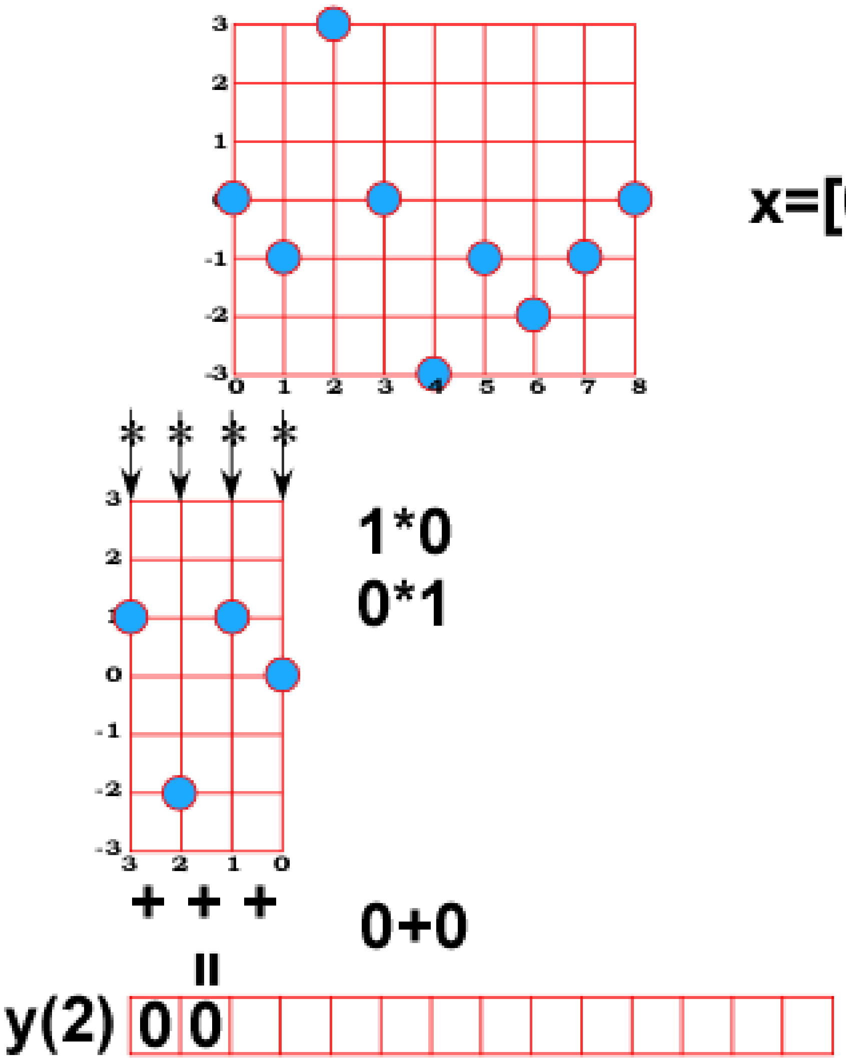
0*0

h=[0 1 -2 1 0]

0+0



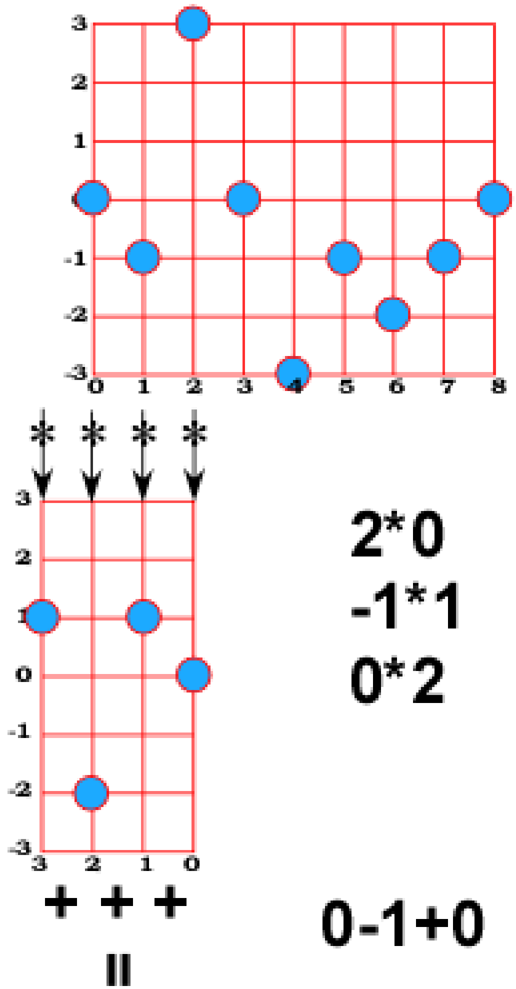
Convolution



x=[0 -1 3 0 -3 -1 -2 -1 0]

h=[0 1 -2 1 0]

Convolution



x=[0 -1 3 0 -3 -1 -2 -1 0]

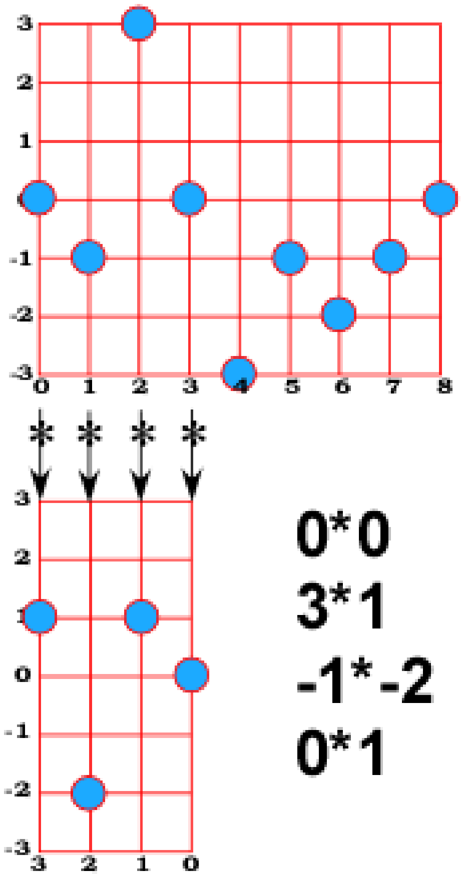
2*0
-1*1
0*2

h=[0 1 -2 1 0]

0-1+0

y(3) 0 0 -1

Convolution



x=[0 -1 3 0 -3 -1 -2 -1 0]

0*0
3*1
-1*-2
0*1

h=[0 1 -2 1 0]

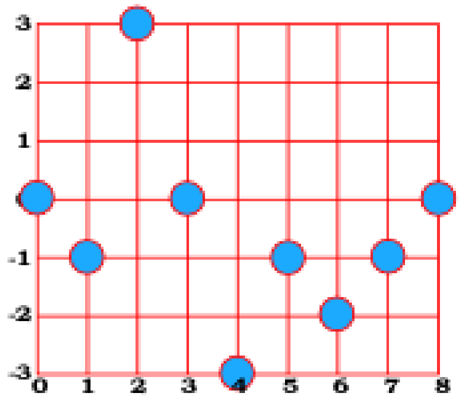
\+ + +

0+3+2+0

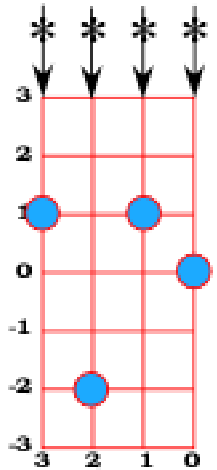
=

y(4) 0 0 -1 5

Convolution



x=[0 -1 3 0 -3 -1 -2 -1 0]



-3*0
0*1
3*-2
-1*1

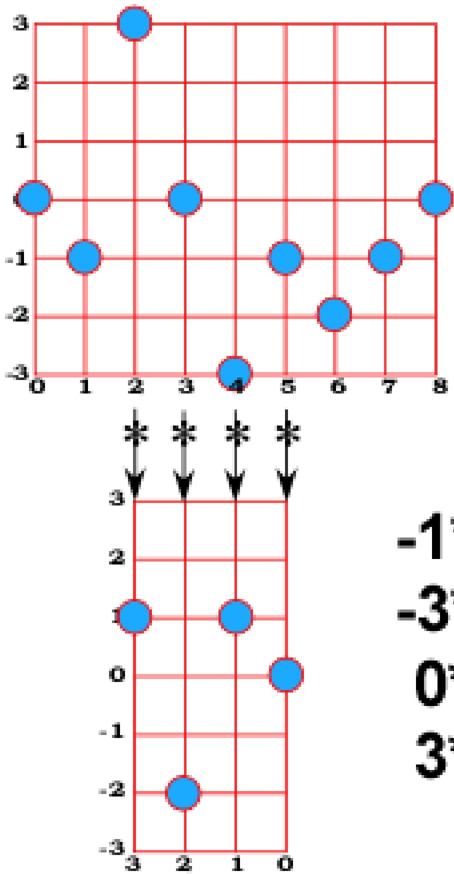
h=[0 1 -2 1 0]

\+ + + 0+0+-6+-1

=

y(5)	0	0	-1	5	-7								

Convolution



x=[0 -1 3 0 -3 -1 -2 -1 0]

-1*0
-3*1
0*-2
3*1

h=[0 1 -2 1 0]

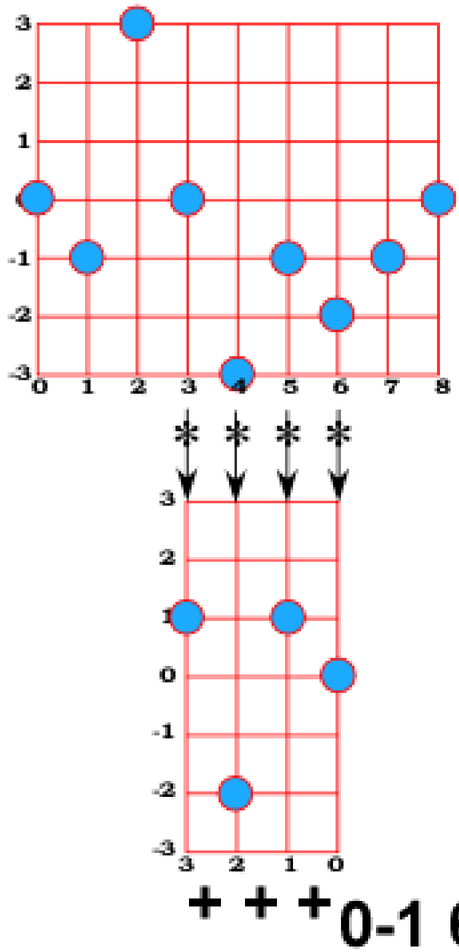
\+ + +

0-3 0+3

=

y(6)	0	0	-1	5	-7	0							

Convolution



x=[0 -1 3 0 -3 -1 -2 -1 0]

-2*0
-1*1
-3*-2
 0*1

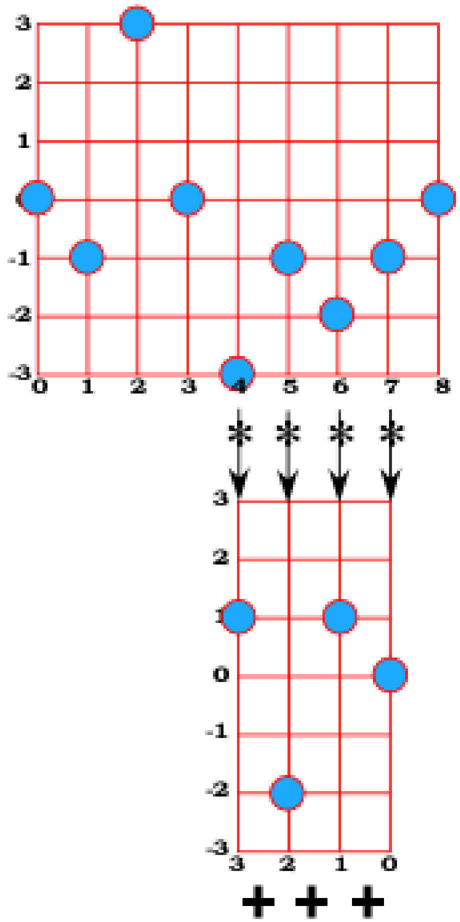
h=[0 1 -2 1 0]

+ + + 0-1 6+0

=

y(7)	0	0	-1	5	-7	0	5							

Convolution



x=[0 -1 3 0 -3 -1 -2 -1 0]

-1*0
-2*1
-1*-2
-3*1

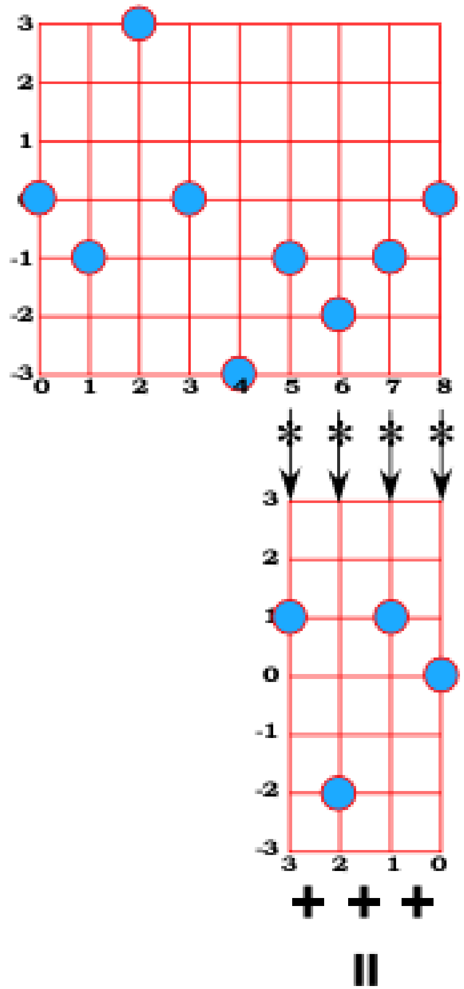
h=[0 1 -2 1 0]

0-2 2-3

=

y(8) | 0 | 0 | -1 | 5 | -7 | 0 | 5 | -3 | | | | | | |

Convolution



x=[0 -1 3 0 -3 -1 -2 -1 0]

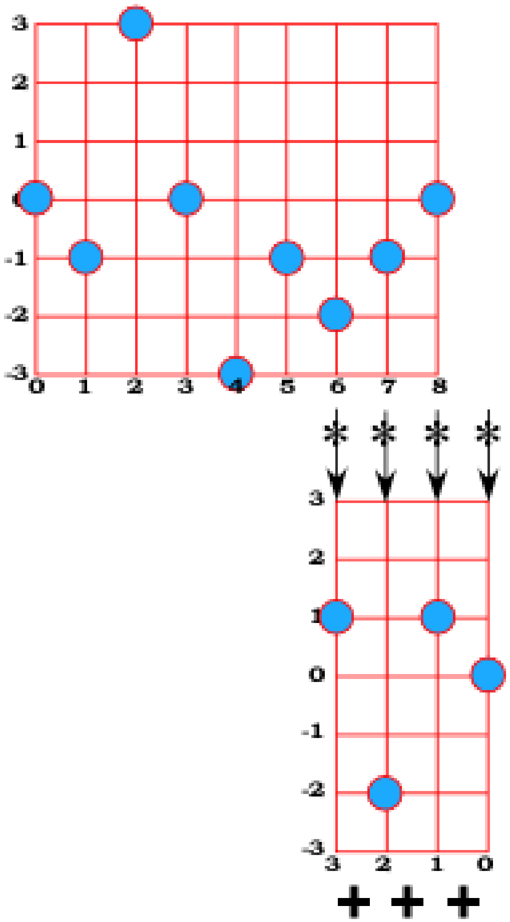
0*0
-1*1
-2*-2
-1*1

h=[0 1 -2 1 0]

0-1 4-1

y(9)	0	0	-1	5	-7	0	5	-3	2					

Convolution



x=[0 -1 3 0 -3 -1 -2 -1 0]

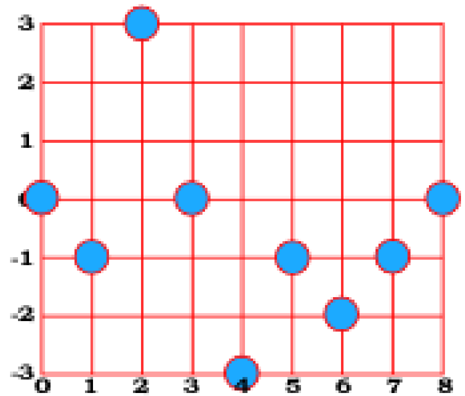
0*0
0*1
-1*-2
-2*1

h=[0 1 -2 1 0]

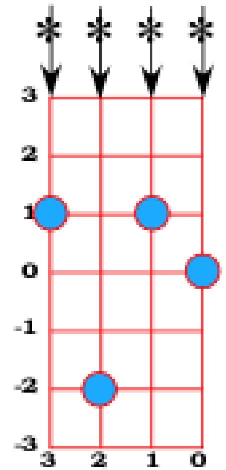
0 0 -2 -2

y(10) 0 0 -1 5 -7 0 5 -3 2 0

Convolution



x=[0 -1 3 0 -3 -1 -2 -1 0]



0*0

0*1

0*-2 h=[0 1 -2 1 0]

-1*1

+ + + 0 0 0 -1

=

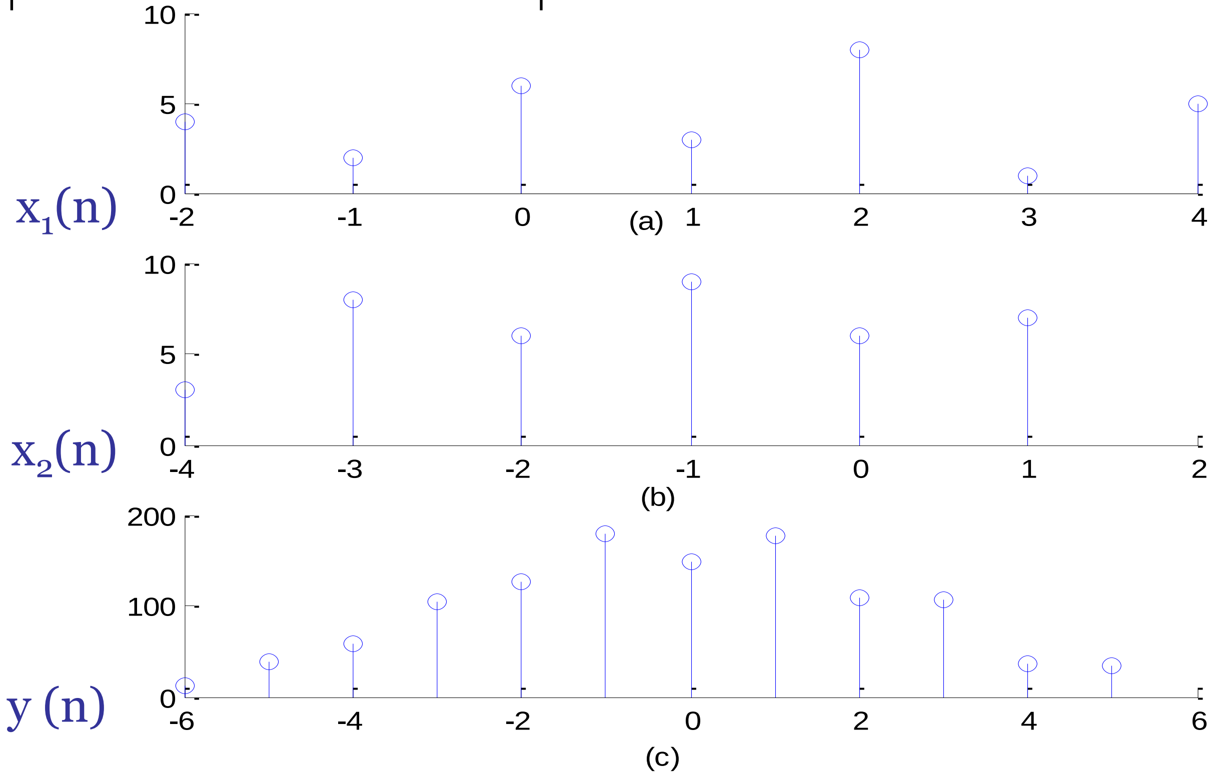
y(11)	0	0	-1	5	-7	0	5	-3	2	0	-1			

Convolution

PART-B: Example:

Find the convolution of the following two signals

$\mathbf{x_1}$=[4 2 6 3 8 1 5]; $\mathbf{x_2}$=[3 8 6 9 6 7];



Topics

- Overview of discrete time signals and systems
- Generation and synthesis of basic sequences
- Convolution
- Response from system difference equation
- Correlation

System Response from difference equations

- LTI discrete time (DT) systems is characterized by a linear constant co-efficient difference equation:

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

x(n) = input signal , y(n) = output response for the input x(n)
$\{a_k\}$ and $\{b_k\}$ are the coefficients

$$y(n) = -\sum_{k=1}^{N} \frac{a_k}{a_0} y(n-k) + \sum_{k=0}^{M} \frac{b_k}{a_0} x(n-k)$$

Taking a_o inside a_k and b_k

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

System Response from difference equations

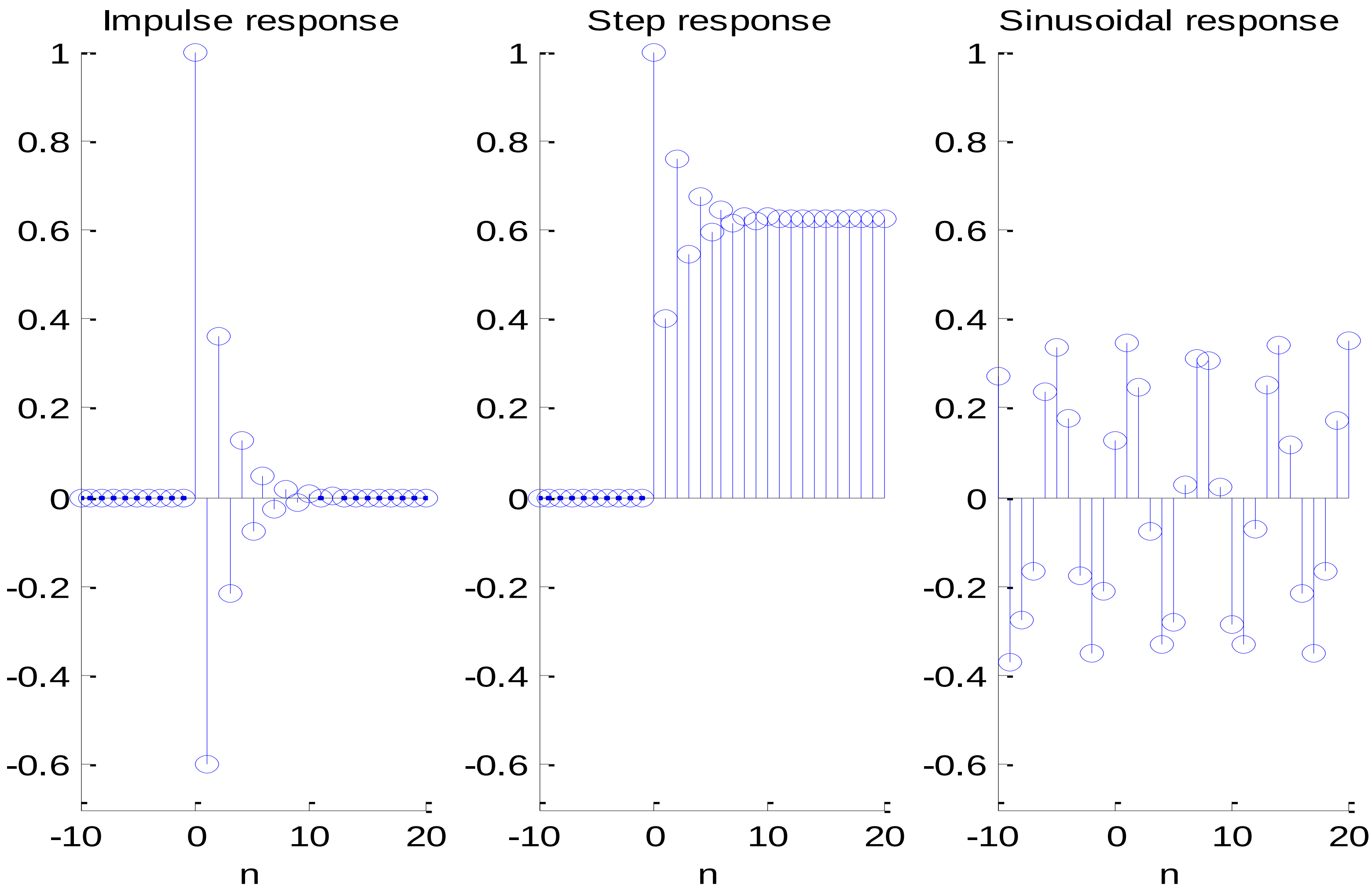
- **Lab Exercise: C.1**

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

Find
(1) impulse response
(2) step response and
(3) sinusoidal response
for $x(n)=0.5\sin(n)u(n)$ in the range $-10 \leq n \leq 20$.

System Response from difference equations

RESULT:



Lab Task

1. Let $x(n) = \begin{cases} r(n) & , n \geq 0 \\ -2u(-n) & , n < 0 \end{cases}$

Where N_l = left edge of window

N_r = right edge of window

m = scaling factor (+ve/-ve)

If n_o is lag variable then

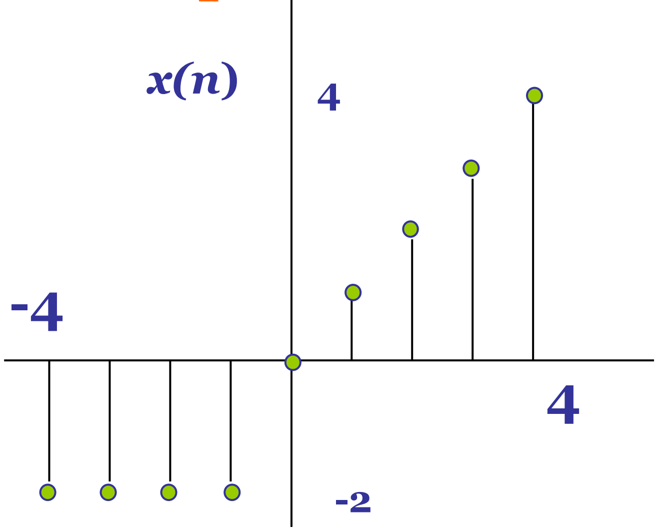
Input of the program

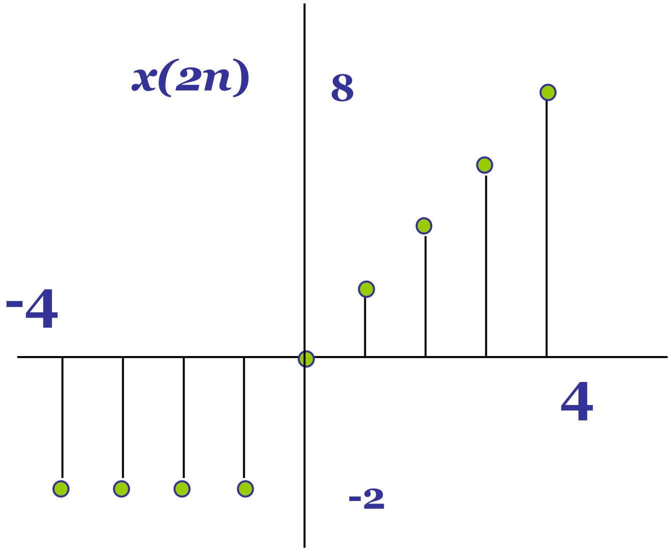
plot $x\{m(n-n_o)\}$

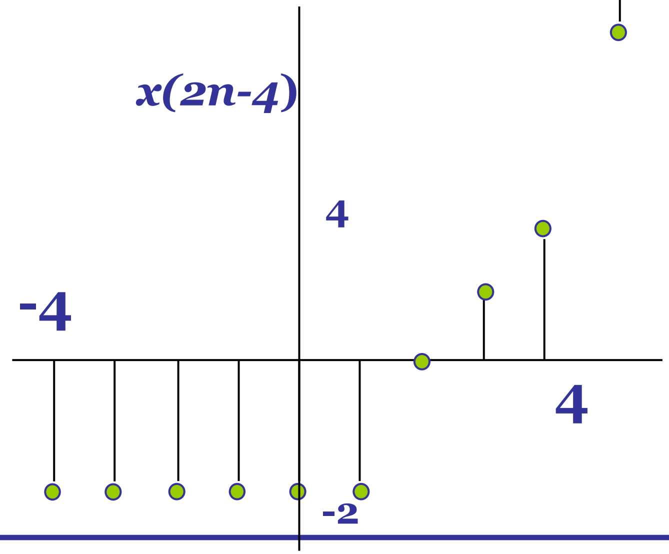
- All parameters must be variable.
- Output will be observed within the specified window

Lab Task

- Example $m = 2$, $n_o = 2$ $N_l = -4$, $N_r = 4$

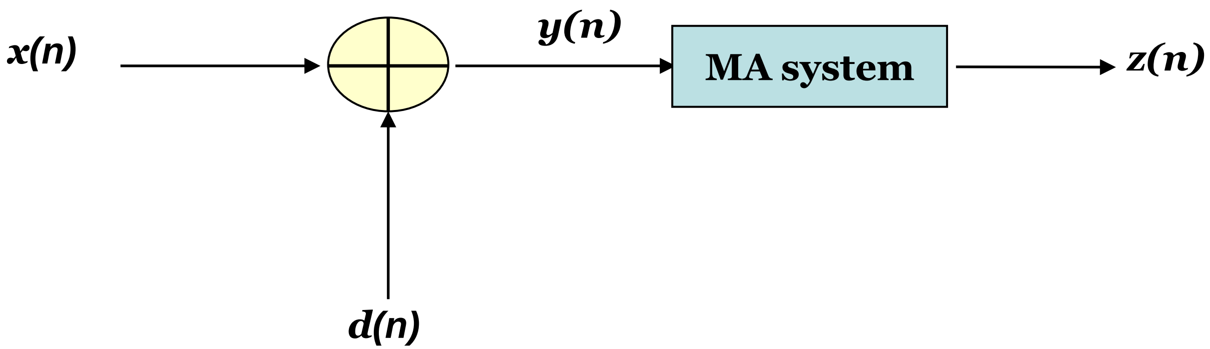






Lab Task

2. Do End of Experiment Exercise No. 4 (Page 16 of labsheet)



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

Consider, *x(n) = 2n(0.9)^n* .
Take 50 point signal.
Noise, d = rand(1,50)-0.5;
Take M = 3

Lab Task

3. Do Lab Exercise C.1 and C.2

(Page 7 of labsheet)

Verify superposition property for the LTI system given.

Lab Task

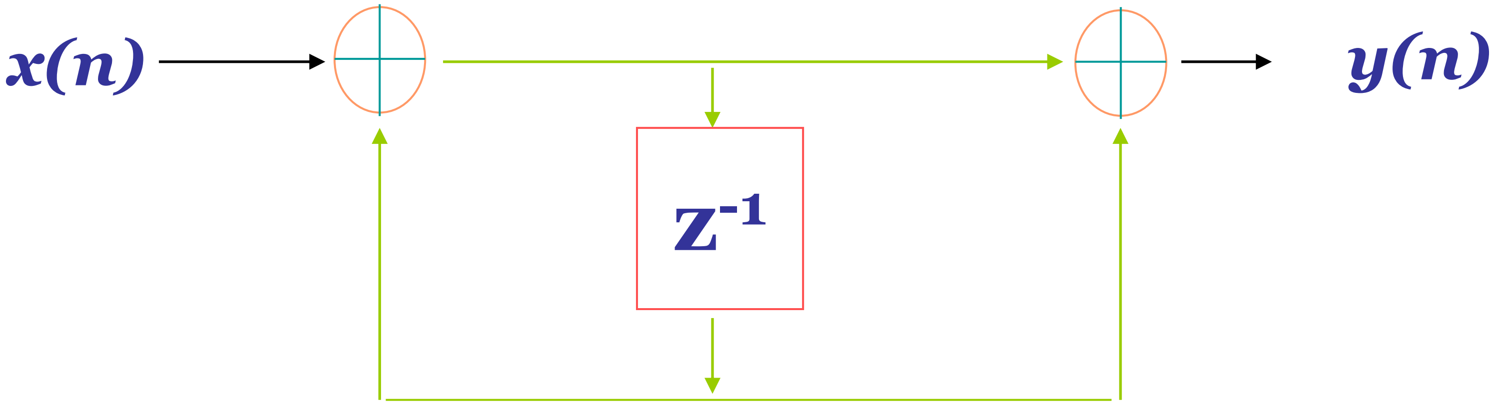
4. Write a general program to compute convolution of two DT signals. DO NOT use conv() function. Say two inputs are

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

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

Report

1. Develop a program to plot up-sampler and down-sampler results as function of time(sec). You can take any arbitrary real function as input.
2. Verify the output of Lab Exercise C.1 using conv() function.
3. Consider the discrete-time system



Plot 1st 10 samples of Impulse Response and then find step response

Topics

- Overview of discrete time signals and systems
- Generation and synthesis of basic sequences
- Convolution
- Response from system difference equation
- Correlation

Correlation:

- The Operation to determine some measure of similarity between two signals.
- Cross-Correlation between two signals x(n) and y(n) is defined as

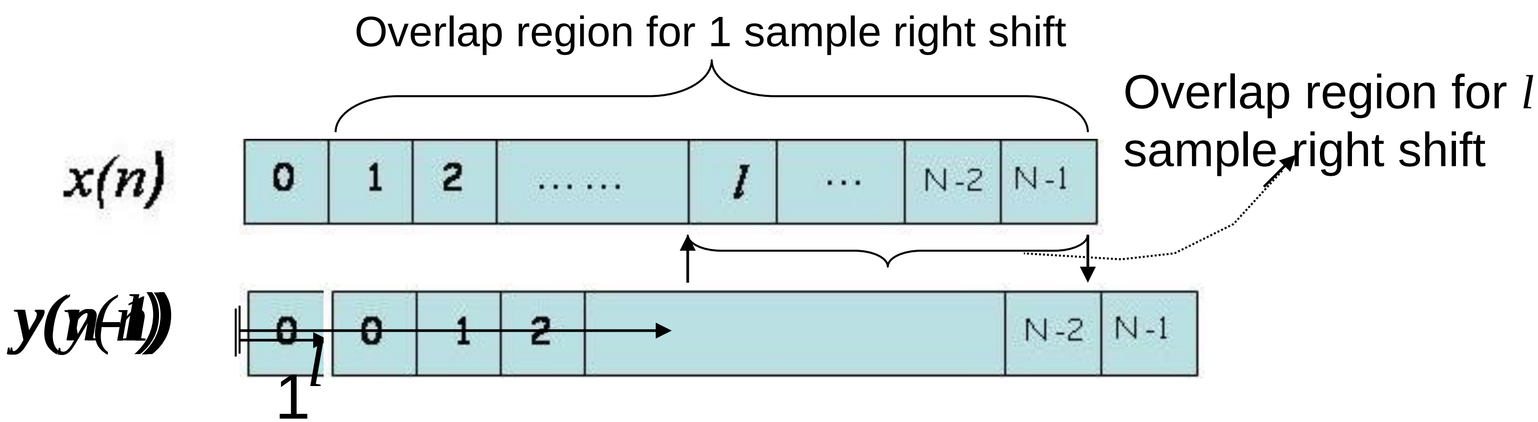
$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l)$$

- l is the index of correlation function $r_{xy}(l)$, and is referred to as, "lag"

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

- where $i = l, k = 0$ for $l \geq 0$ and $i = 0, k = l$ for $l < 0$
- Autocorrelation is a special case when x(n)=y(n).

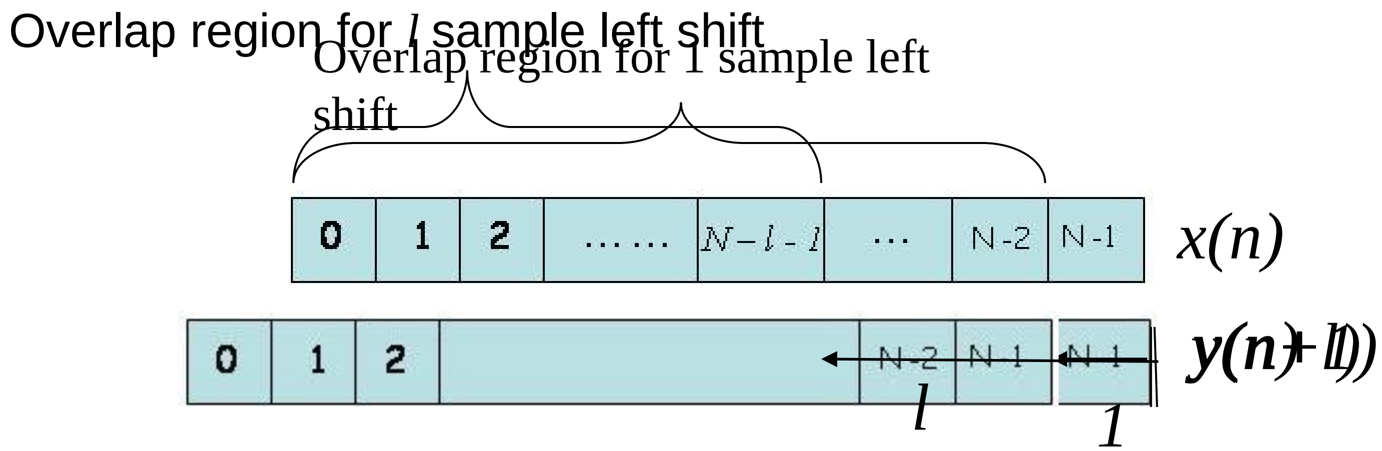
Correlation:



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

where $i = l, k = 0$ for $l \geq 0$

Correlation:



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

where $i = 0$, $k = l$ for $l < 0$

Correlation:

The steps of cross-correlation is summarized as:

1. Shift one of the signals along the time axis by one sample.
2. Multiply the corresponding values of the two signals.

3. Summate the products from step 2 to get one point of the correlation sequence.
4. Repeat steps 1-3 to obtain the total correlation sequence at all times that the signals overlap.

- For example, let x= {-1,3,0,-3,-1,-2,-1}and h ={1,-2,1,}
- Cross-correlation of these two discrete signals, R_{xh}={-1,5,-7,0,5,-3,2,0,-1}

LOOK! Same result was obtained for the convolution of 'x' and 'h', in the example shown diagrammatically.

Correlation:

- **Convolution and Correlation are same except for the flip**
- CONVOLUTION:

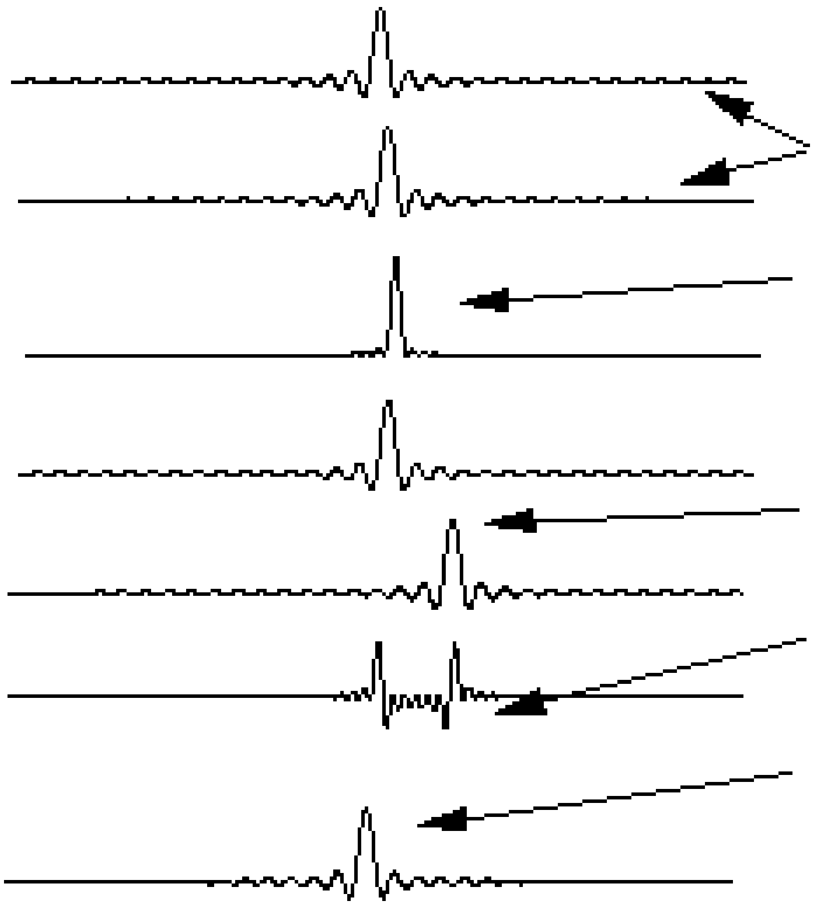
$$x(n) * y(n) = \sum_{n=-\infty}^{\infty} x(n)y(k-n)$$

- CORRELATION:

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

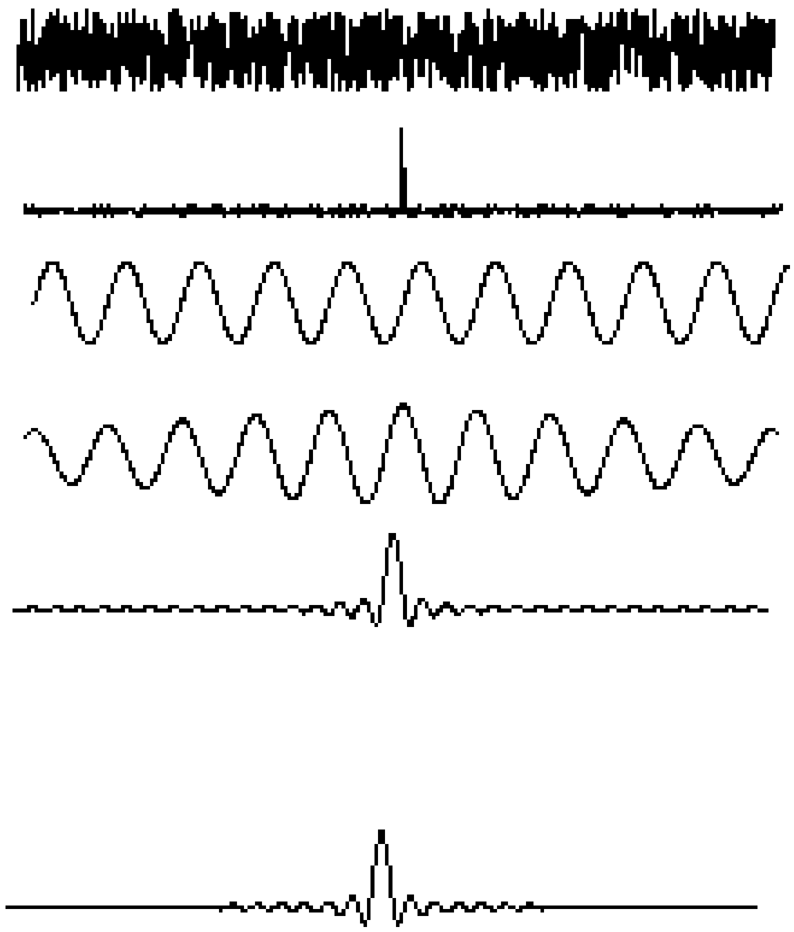
- we can compute Cross-correlation by **conv**() function too
- r_{xy}=conv(x,fliplr(y));
- % Here **fliplr**() function folds y.

Correlation:



- If two signals are similar and unshifted...
- their product is all positive.
- But as the shift increases...
- parts of it become negative...
- and the correlation function shows where the signals are similar and unshifted

Correlation:



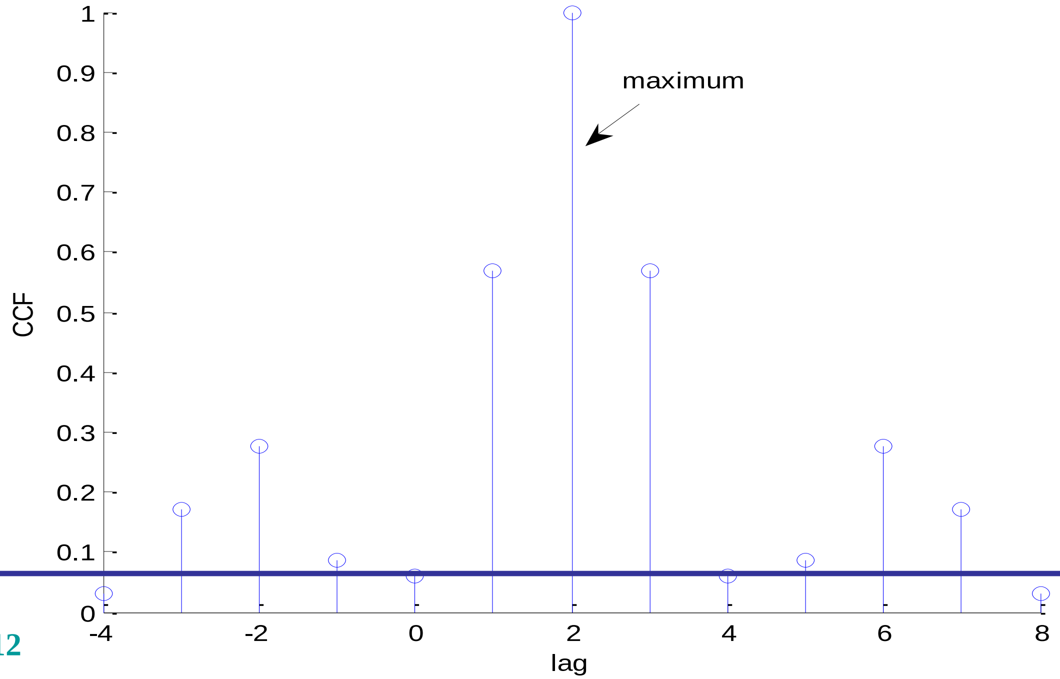
- Random noise is similar to itself, and in phase, only with no time shift at all
- so its correlation function is a spike
- periodic signals go in and out of phase as they are time shifted
- so their correlation functions are periodic
- signals that last only a short while are only similar while they last
- so their correlation functions are short

Correlation:

- **PART-D : EXAMPLE**

y(n)=x(n-2) , where
x(n)=[3 11 7 0 -1 4 2].

cross-correlation result:



Correlation:

- Applications of ACF/CCF:

(1) Detecting a periodic input corrupted by additive white Gaussian noise:

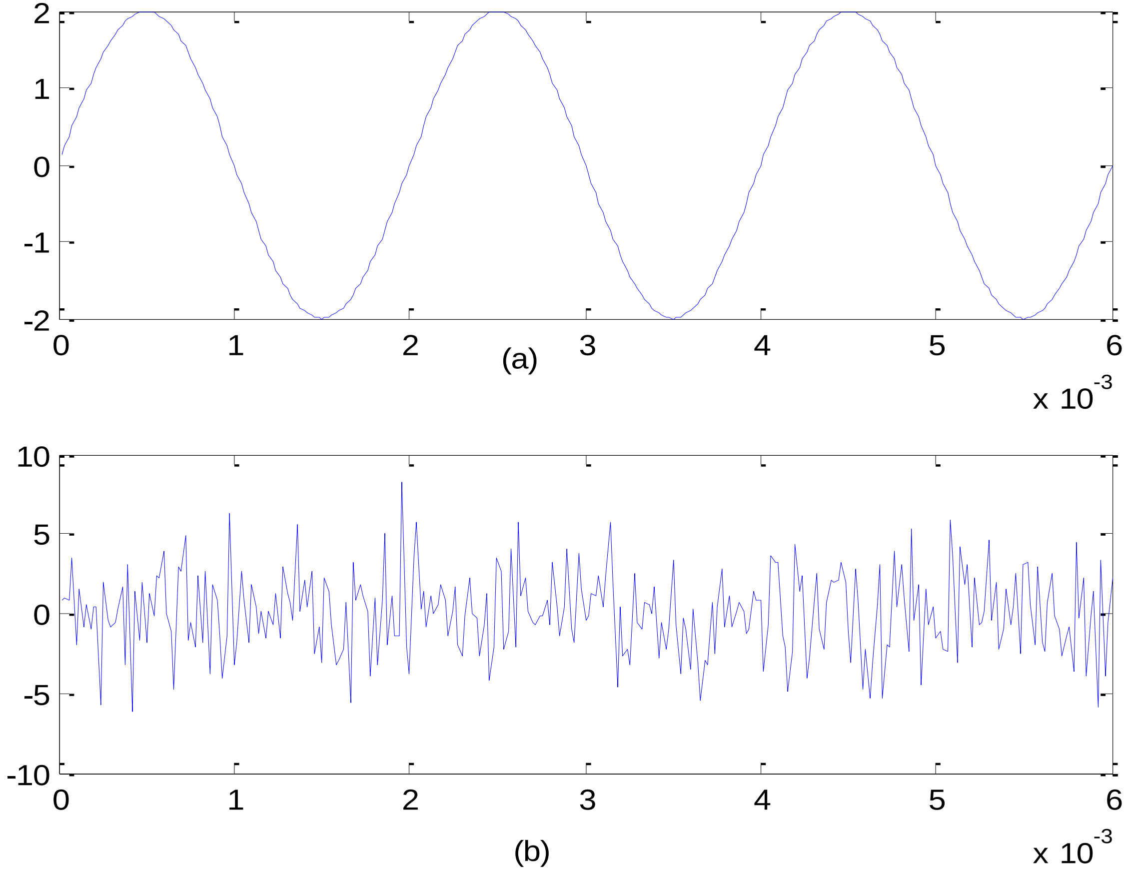


Figure: (a) Input periodic wave (b) Input corrupted by AWGN for -5dB SNR

Correlation:

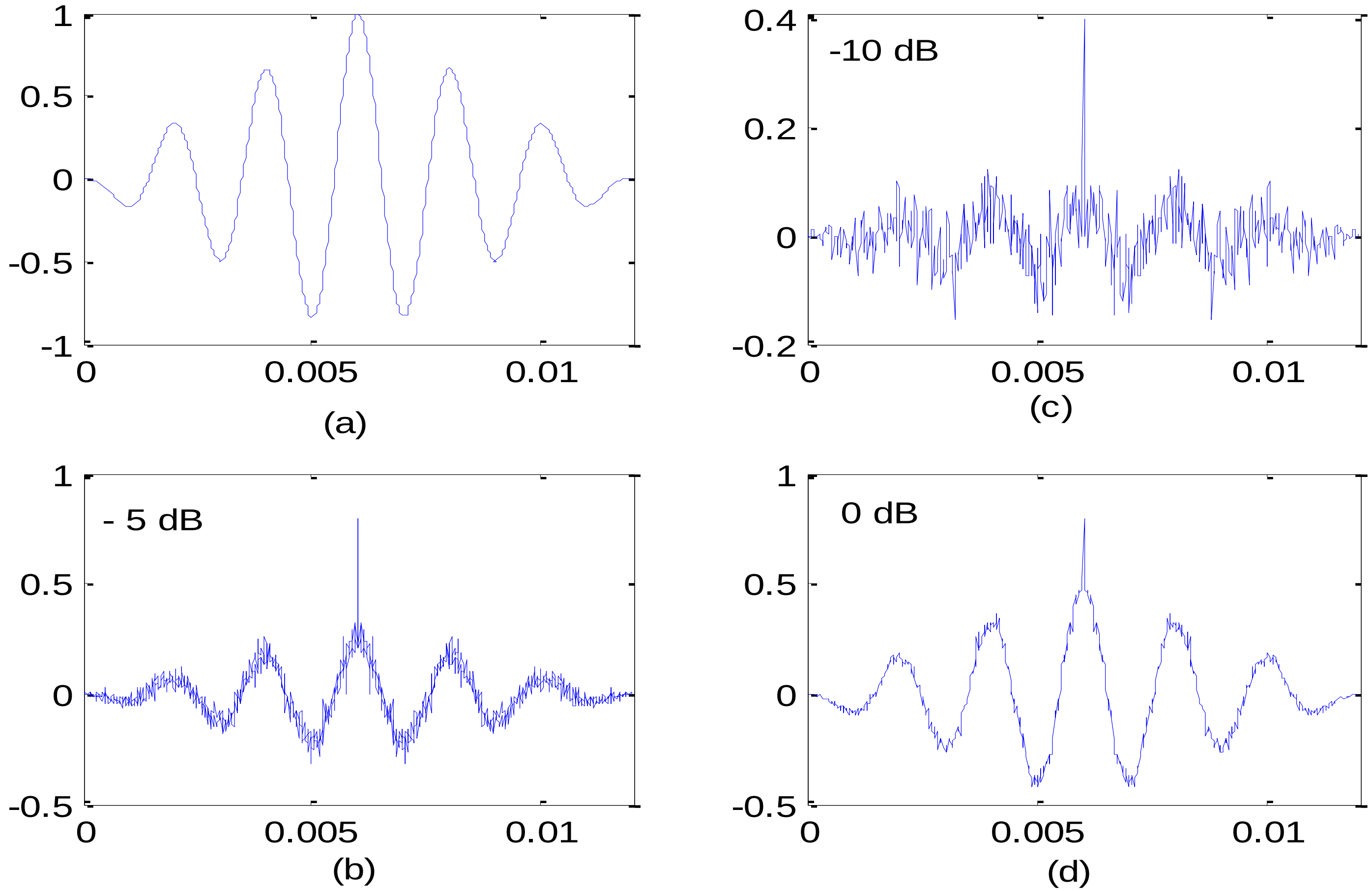


Figure: (a) ACF of input
(b,c,d) ACF of noisy sequence.

Correlation:

(2) Estimation of impulse response:

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

Where, r(n) is a white noise input,

$R_{rr}(k)$ is the auto-correlation of the noise input

auto-correlation of white noise sequence is like impulse sequence

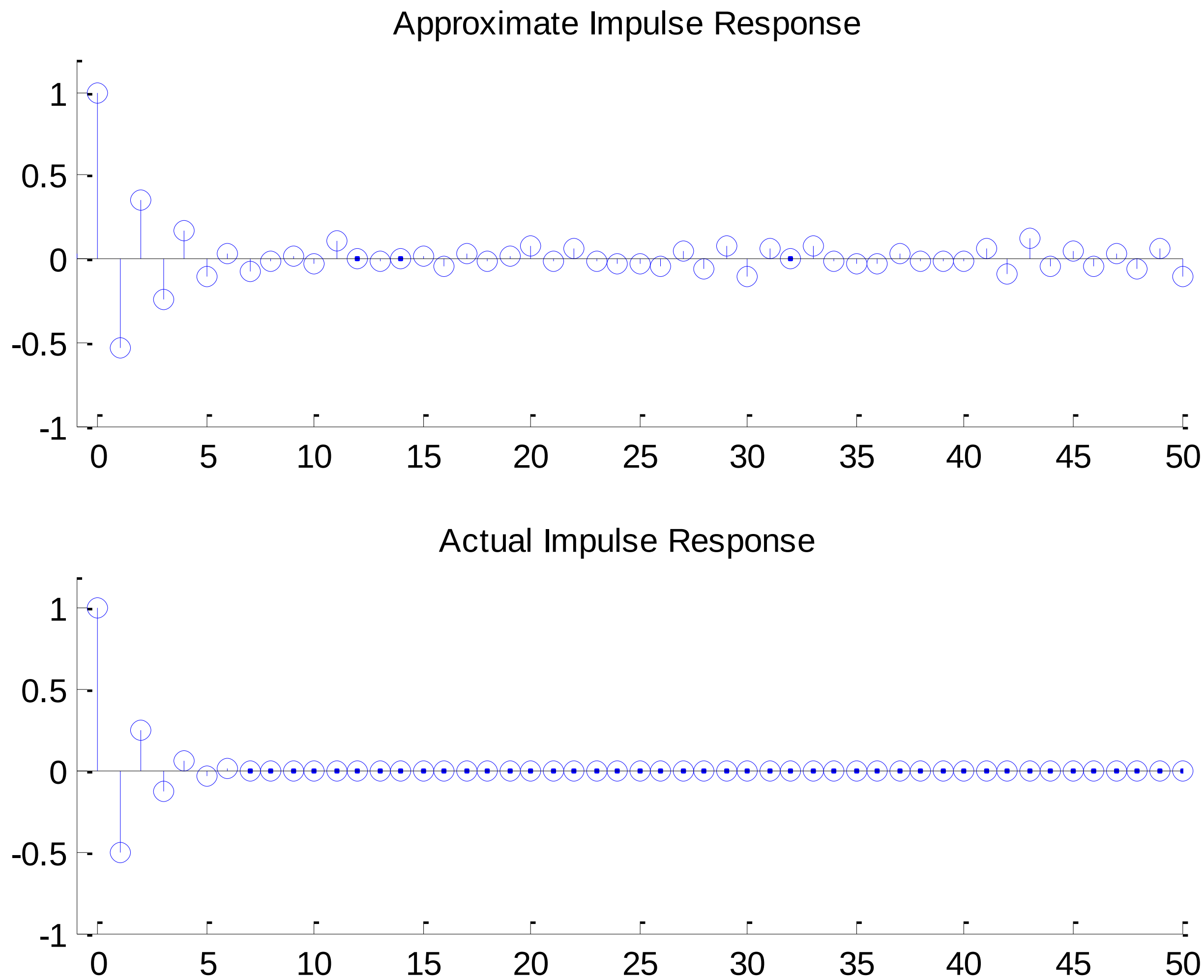
$$R_{yr(k)} \approx h(n)$$

Consider a system:

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

Correlation:

RESULT:



Correlation:

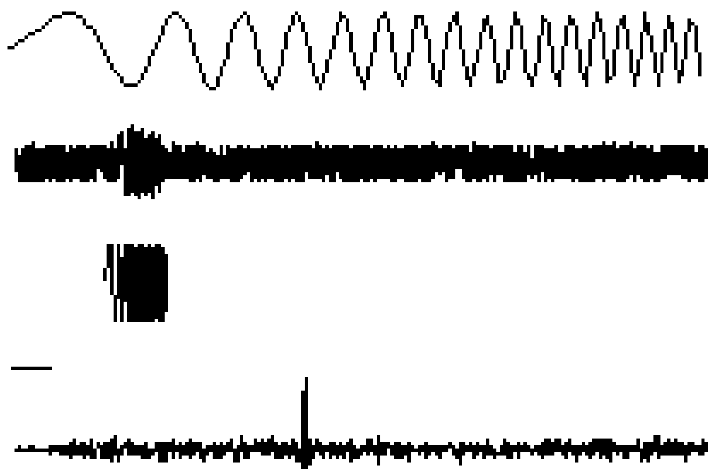
- **Detection of signals in noise by auto-correlation**

PROBLEM:

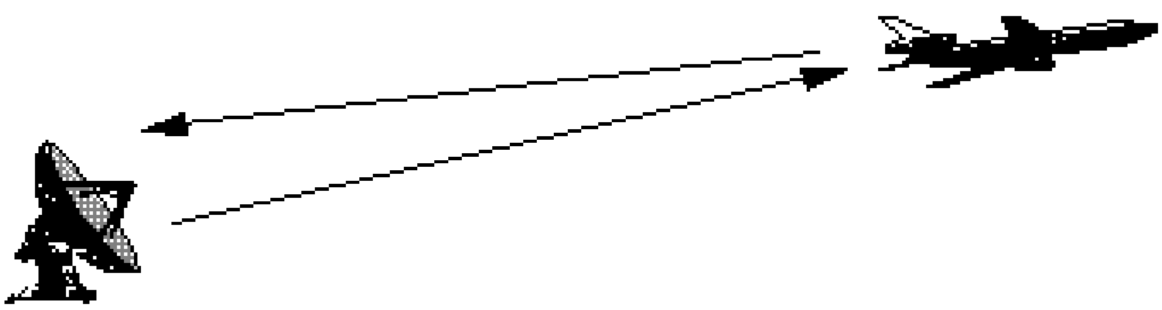
a radar transmitting a short tone burst of EM energy and receiving a weak echo from a distant target. In the absence of noise weak echo can be amplified and there is no problem detecting it. If there is background of noise whose amplitude exceeds that of the echo, the echo will be masked and not detectable (echo with noise). As we know noise is suppressed in ACF domain, then write a general program to detect the echo using correlation.

Correlation:

Detection of signals in noise by auto-correlation:



- A radar or sonar 'chirp' signal...
- bounced off a target may be buried in noise...
- but correlating with the 'chirp' reference
- clearly reveals when the echo comes



Correlation:

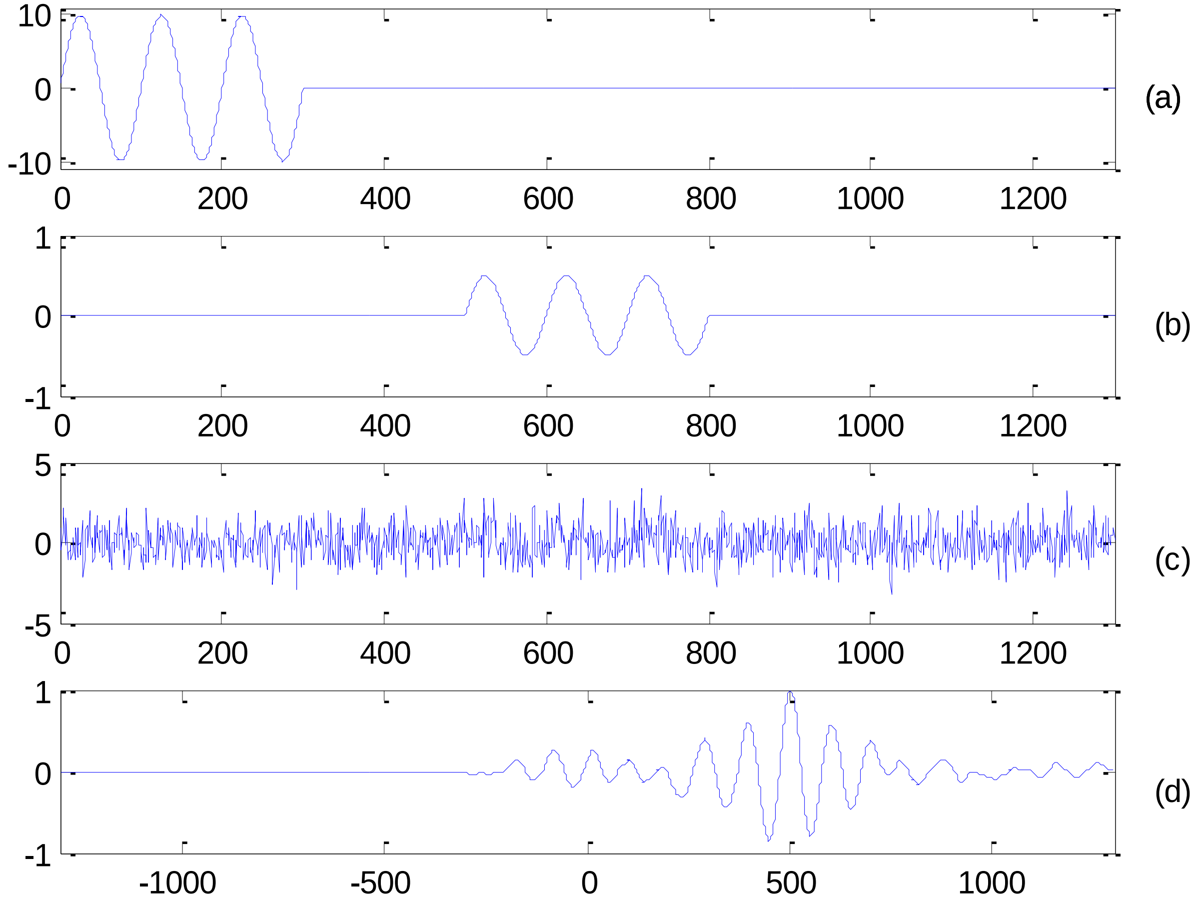


Figure:(a) Transmitted tone burst (b)Received weak echo(c)Received echo buried into background noise(d)CCF between (a) and (c) to locate a weak echo. It shows that after 500 units delay an echo arrives (location of the peak).

Correlation:

- **Signal smoothing by a moving average (MA) system:**

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

set $\{a_k\}=0$ for $k = 1,\ldots,N$

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

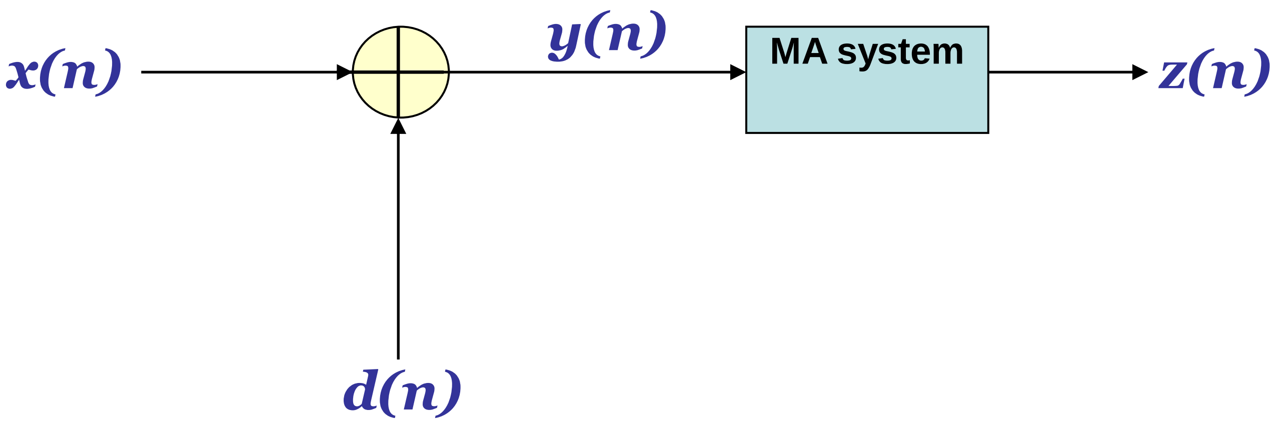
which is a non-recursive linear time-invariant system, called moving average (MA) system.

Simple working MA system can be expressed as :

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

Correlation:

Simple working moving average (MA) system:



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

Correlation:

- **Simple working moving average (MA) system:**

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

Consider,

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

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

Correlation:

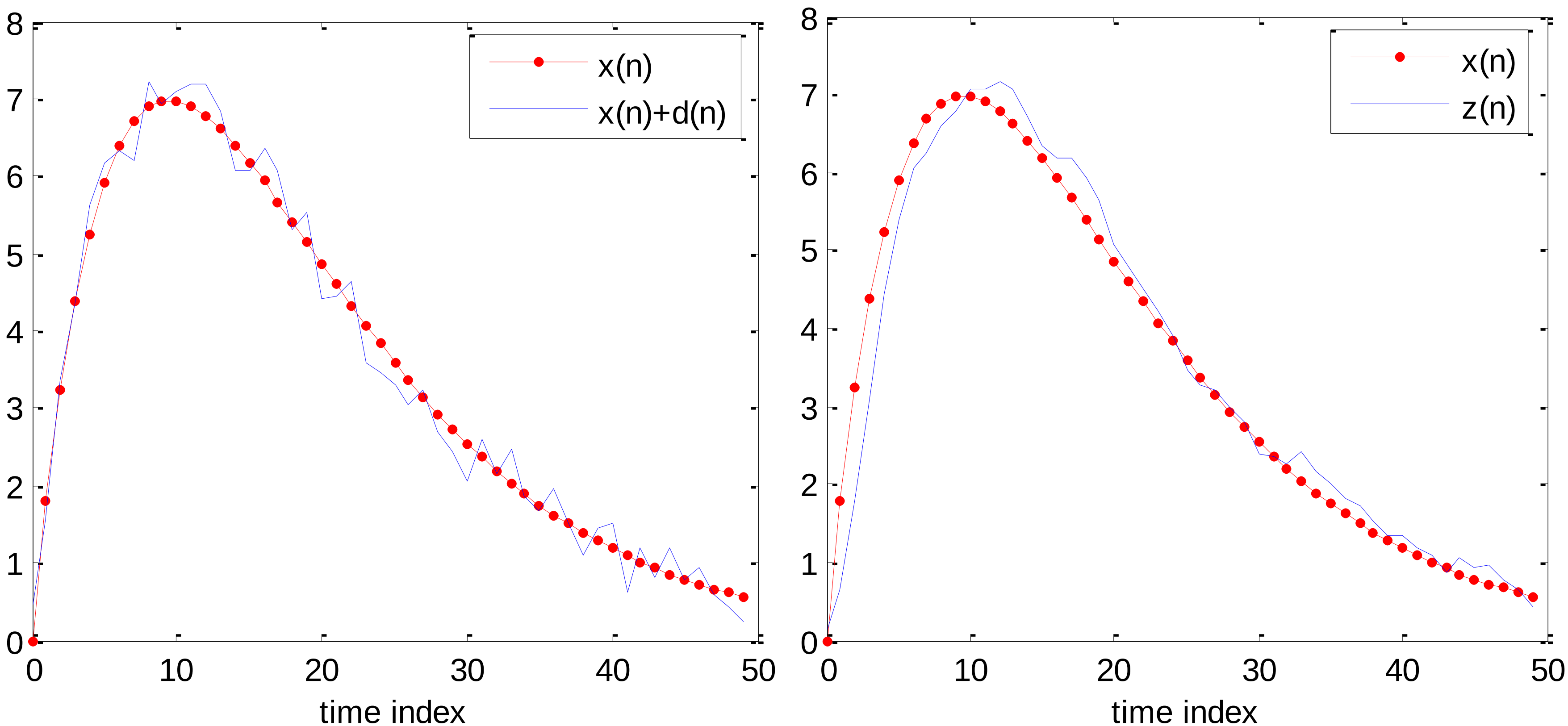


Figure:
(left) $x(n)$=original sequence and $y(n)=x(n)+d(n)$, corrupted sequence.
(right) $z(n)$=output of the MA system with original sequence, $x(n)$

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

MATLAB CODES:

MATLAB CODES

MATLAB CODES:

```
% generating sequences
n1=-3;
n2=3;
n=n1:n2;
no=-1;
x1=[(n-no)==0];%impulse
x2=[(n-no)>=0];%step
x3=[x1(1:find(x1)-1)
0:length(n)-find(x1)];%ramp
xe=0.5*(x3+fliplr(x3));
xo=0.5*(x3-fliplr(x3));
x4=xe+xo;
subplot(211),stem(n,xe)
subplot(212),stem(n,xo)
```

```
% upsampling
w=0.36;
n =1:52;
L = 3;
x=sin(w*n);
y=zeros(1,length(n));
for n=1:52
    if(rem(n,L)==0)
        y(n)=x(n/L);
    end
end
subplot(211),stem(1:n,x)
xlabel('n')
subplot(212),stem(1:n,y)
xlabel('n')
```

MATLAB CODES:

- PART-2:
- Example-1: CONVOLUTION

```
x1=[4 2 6 3 8 1 5];
n1=[-2:4];
x2=[3 8 6 9 6 7];
n2=[-4:1];
kmin=n1(1)+n2(1);
kmax=n1(end)+n2(end);
y=conv(x1,x2);
k=kmin:kmax;
subplot(311),stem(n1,x1)
subplot(312),stem(n2,x2)
subplot(313),stem(k,y)
```

MATLAB CODES:

- PART-2:
- Exercise-2.1: CONVOLUTION

```
%extension of the previous program

k1=min([n1(1) n2(1) kmin]);
k2=max([n1(end) n2(end) kmax]);
x11=[zeros(1,n1(1)-k1) x1];
x22=[zeros(1,n2(1)-k1) x2];
kc=k1:k2;
x11(length(kc))=0;
x22(length(kc))=0;
subplot(311),stem(kc,x11)
subplot(312),stem(kc,x22)
subplot(313),stem(kc,y)
```

MATLAB CODES:

- PART-2:
- Exercise-2.1: CONVOLUTION

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)
```

MATLAB CODES:

- Exercise-3.1: System Response from difference equations

```
n=[-10:20];
x1=[(n-0)==0];
x2=[(n-0)>=0];
x3=.5*sin(n);
b=[1];
a=[1 0.6];
impulse=filter(b,a,x1);
step=filter(b,a,x2);
sinusoidal=filter(b,a,x3);
subplot(131),stem(n,impulse)
axis([-10 20 -.7 1])
xlabel('n')
title('Impulse response')
subplot(132),stem(n,step)
axis([-10 20 -.7 1])
xlabel('n')
title('Step response')
subplot(133),stem(n,sinusoidal)
axis([-10 20 -.7 1])
xlabel('n')
title('Sinusoidal response')
```

MATLAB CODES:

- PART-4:
- Example :CORRELATION

```
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))
```

MATLAB CODES:

- Applications of ACF/CCF:

```
>> 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); >>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
```

MATLAB CODES:

- Estimation of impulse response:

```
>>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);
>>kmin=ny(1)+nrr(1);
```

```
>>kmax=ny(length(ny))+nrr(length(nrr));
>>k=kmin:kmax;
>>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')
```

MATLAB CODES:

- Detection of signals in noise by auto-correlation:

```
>>T=2e-3; % period=2ms
>>tstep=T/100;
>>t=tstep:tstep:3*T; % taking time index upto 3 periods
>>s=2*sin(2*pi*t/T); % Input
>>x=5*[s zeros(1,1000)];
>>e=[zeros(1,500) .25*s zeros(1,500)];
>>n=randn(1,length(x));
>>r=e+n;
>>n=-(length(n)-1):(length(n)-1);
>>Rxr=xcorr(r,x);
>>subplot(411),plot(x)
>>subplot(412),plot(e)
>>subplot(413),plot(r)
>>subplot(414),plot(n,Rxr/max(Rxr))
```

MATLAB CODES:

- Signal smoothing by a moving average (MA) system

```
>> R=50;
>> d=rand(1,R)-0.5;
>> m=0:R-1;
>> x=2*m.*(0.9).^m;
>> stem(x)
>> y=x+d;
>> subplot(121),plot(m,x,'r.-')
>> hold on
>> subplot(121),plot(m,y,'b-')
>> M=3;
>> b=ones(1,M)/M;
>> z=filter(b,1,y);
>> subplot(122),plot(m,x,'b.-')
>> hold on
>> subplot(122),plot(m,z,'r-')
>> legend('x(n)','z(n)')
```

Thank You

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

Presentation for this experiment is prepared by
SHANKHANAAD MALLICK
Lecturer, Dept of EEE, BUET
and
TOUFIQUL ISLAM
Lecturer, Dept of EEE, BUET