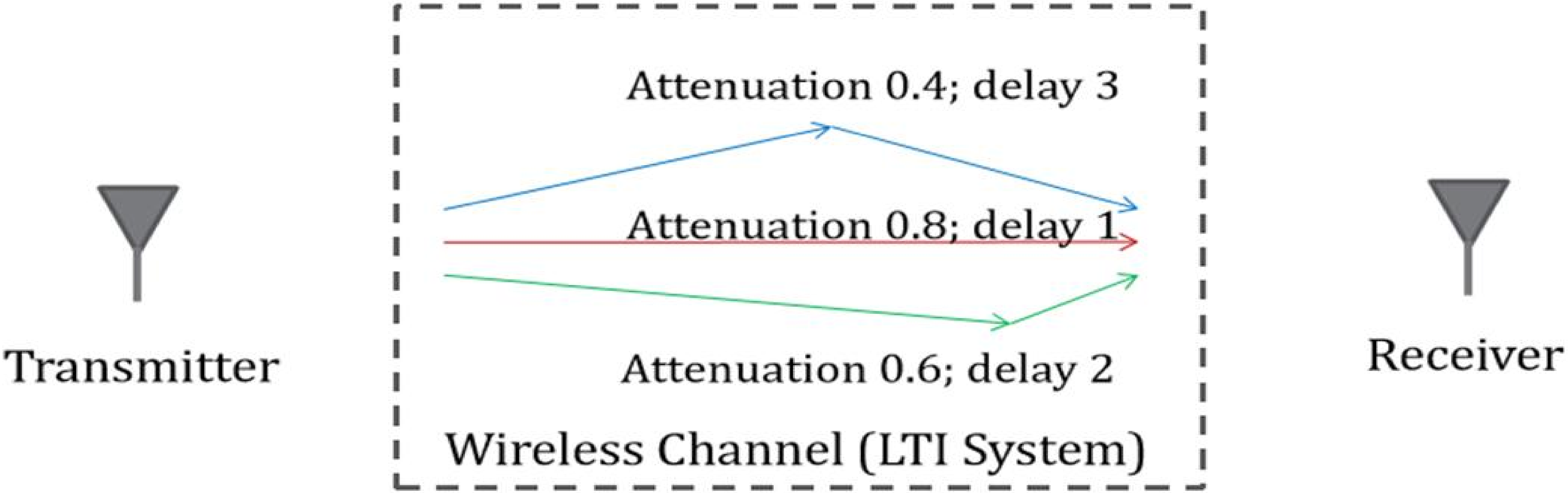
Part III: Application--Simplified Wireless Channel

Application: Simplified Wireless Channel



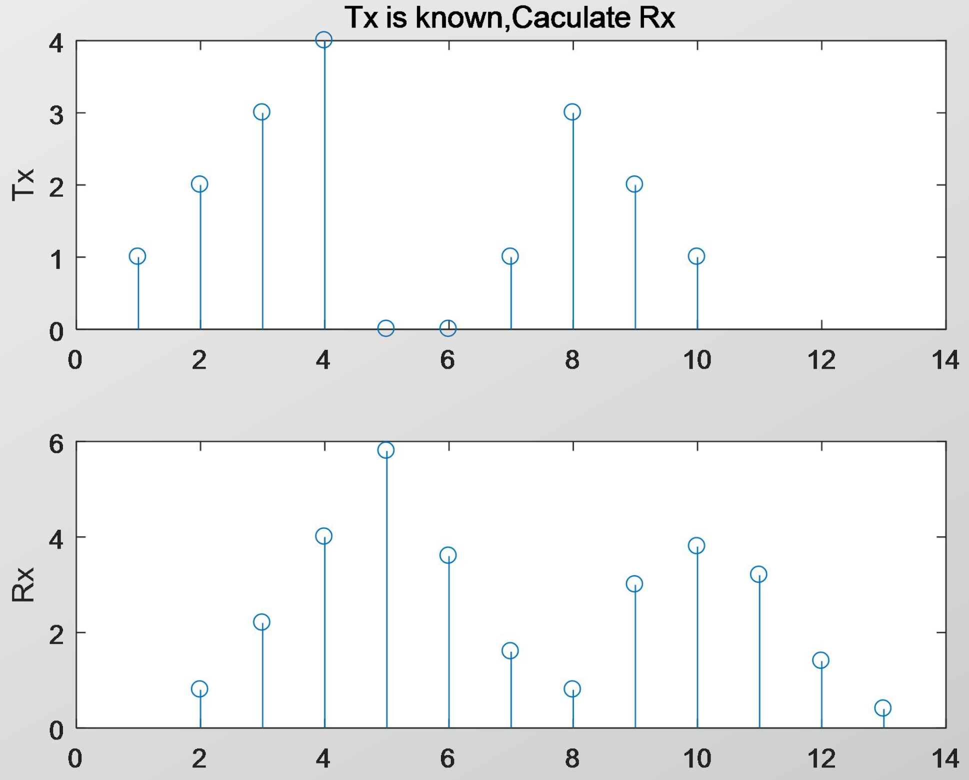
Impulse response: $h1[n] = 0.8\delta[n-1] + 0.6\delta[n-2] + 0.4\delta[n-3]$

Difference equation: $y[n] = 0.8x[n-1] + 0.6x[n-2] + 0.4x[n-3]$

CONVOLUTION

$$h1[n] = 0.8\delta[n-1] + 0.6\delta[n-2] + 0.4\delta[n-3]$$

```
%impulse response：
h=[0.8 0.6 0.4];
nh=[1:3]
% Generate Tx Signal:
x=[1 2 3 4 0 0 1 3 2 1];
nx=[1:10];
% Generate Rx Signal:
w=conv(x,h)
nw=2:13;
subplot(2,1,1), stem(nx,x),ylabel('Tx'),xlim([0 14]);
title('Tx is known,Caculate Rx') ;
subplot(2,1,2), stem(nw,w),ylabel('Rx'),xlim([0 14]);
```

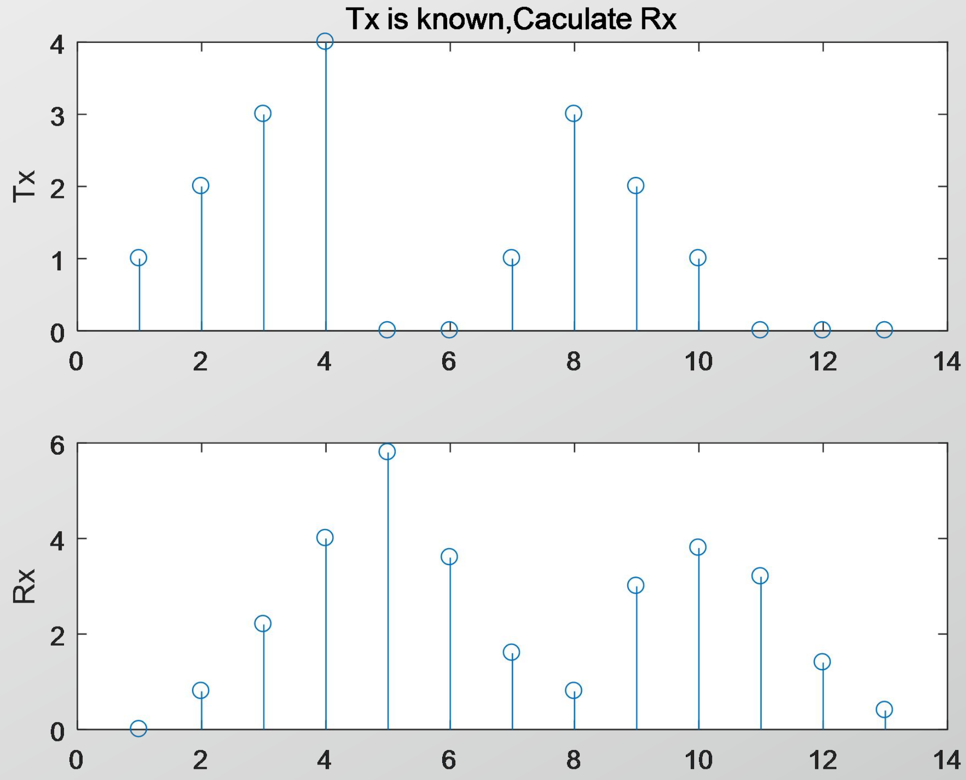


Rx= 0.8000 2.2000 4.0000 5.8000 3.6000 1.6000 0.8000 3.0000 3.8000 3.2000 1.4000 0.4000

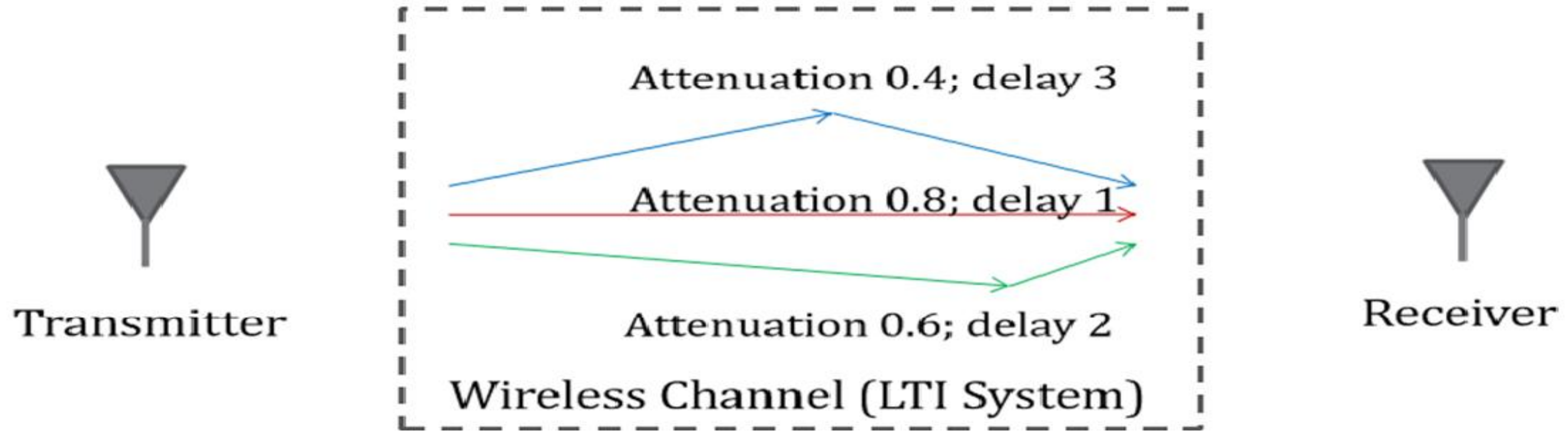
FILTER

$$y[n] = 0.8x[n-1] + 0.6x[n-2] + 0.4x[n-3]$$

```
% two coefficient vectors
A1 = 1;
B1 = [0 0.8:-0.2:0.4];
% Generate Tx Signal:
x=[1 2 3 4 0 0 1 3 2 1 0 0 0];
% Generate Rx Signal:
y = filter(B1, A1, x);
subplot(2,1,1), stem(x),ylabel('Tx');
title('Tx is known,Caculate Rx') ;
subplot(2,1,2), stem(y),ylabel('Rx');
```



y= 0 0.8000 2.2000 4.0000 5.8000 3.6000 1.6000 0.8000 3.0000 3.8000 3.2000 1.4000 0.4000



Impulse response: $h1[n] = 0.8\delta[n-1] + 0.6\delta[n-2] + 0.4\delta[n-3]$
Difference equation: $y[n] = 0.8x[n-1] + 0.6x[n-2] + 0.4x[n-3]$

Invertibility

Detected Signal z[n]
$z[n] = x[n]$ 或
$z[n]=x[n-k]$

Signal Detector
（LTI） —— h2[n]

Received signal
y[n]

Impulse response: $h2[n] * h1[n] = \delta[n]$ or $\delta[n-k]$
Difference Equation: $0.8z[n-1] + 0.6z[n-2] + 0.4z[n-3] = y[n]$

IIR

$$0.8z[n-1] + 0.6z[n-2] + 0.4z[n-3] = y[n]$$

```
a=[0 0.8 0.6 0.4];
b=[1];
y=[0.8000 2.2000 4.0000 5.8000 3.6000 1.6000 0.8000
3.0000 3.8000 3.2000 1.4000 0.4000];
z=filter(b,a,y)
subplot(2,1,1), stem(y),ylabel('Rx');
title(Rx is known,Caculate Tx') ;
subplot(2,1,2), stem(z),ylabel('Tx')；
```

```
Error using filter
First denominator filter coefficient must be non-zero.
```

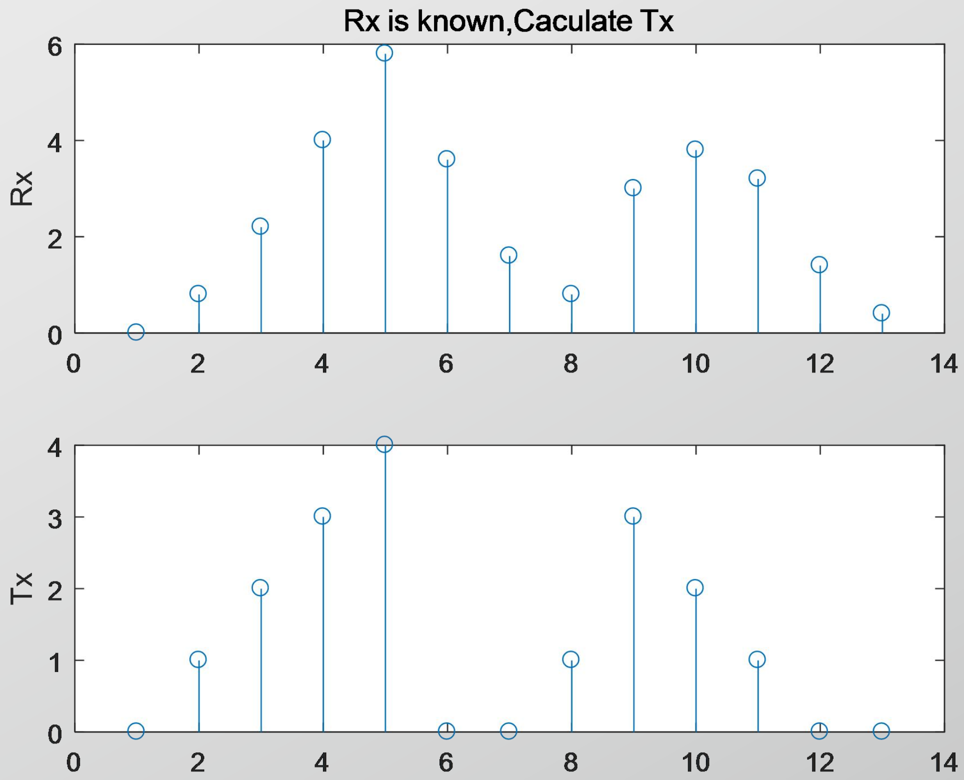
$$0.8z[n-1] + 0.6z[n-2] + 0.4z[n-3] = y[n]$$

Let $z'[n] = z[n-1]$

$$0.8z'[n] + 0.6z'[n-1] + 0.4z'[n-2] = y[n]$$

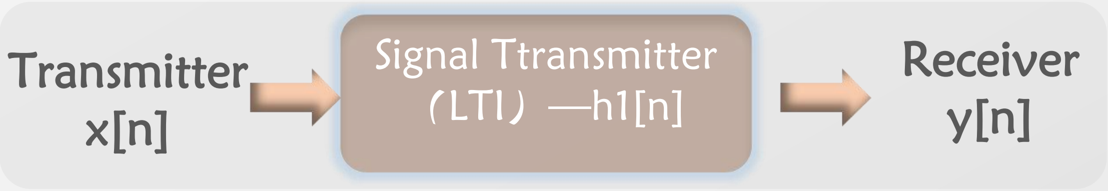
$$0.8z'[n] + 0.6z'[n-1] + 0.4z'[n-2] = y[n]$$

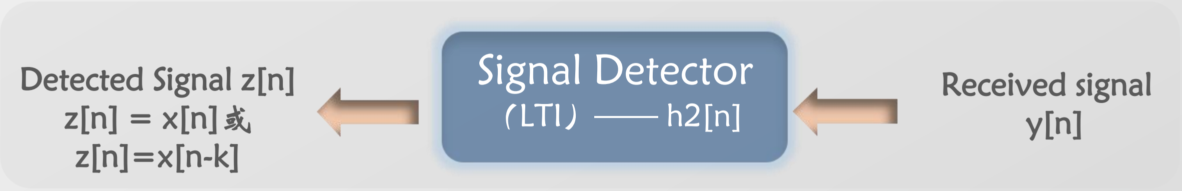
```
a=[0.8 0.6 0.4];
b=[1];
y=[0 0.8000 2.2000 4.0000 5.8000 3.6000 1.6000 0.8000
3.0000 3.8000 3.2000 1.4000 0.4000];
z=filter(b,a,y)
subplot(2,1,1), stem(y),ylabel('Rx');
title('Rx is known,Caculate Tx') ;
subplot(2,1,2), stem(z),ylabel('Tx');
```



z = 1.0000 2.0000 3.0000 4.0000 0 0 1.0000 3.0000 2.0000 1.0000 0 0

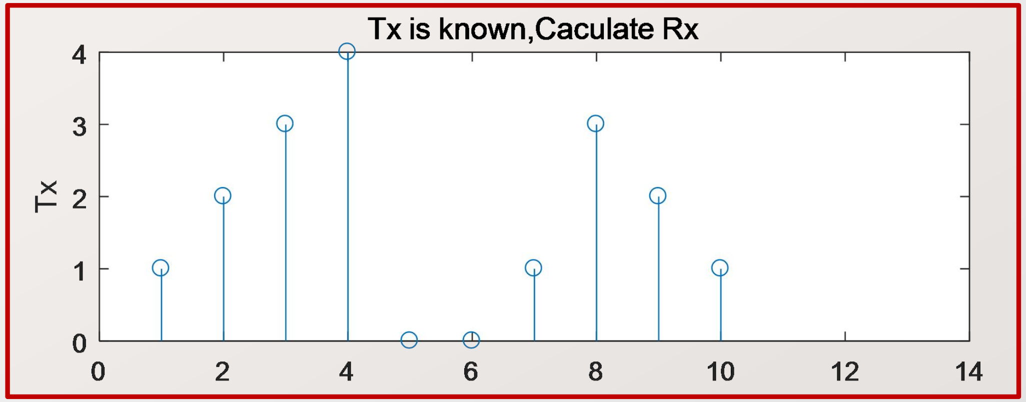
x=[1 2 3 4 0 0 1 3 2 1 0 0 0];

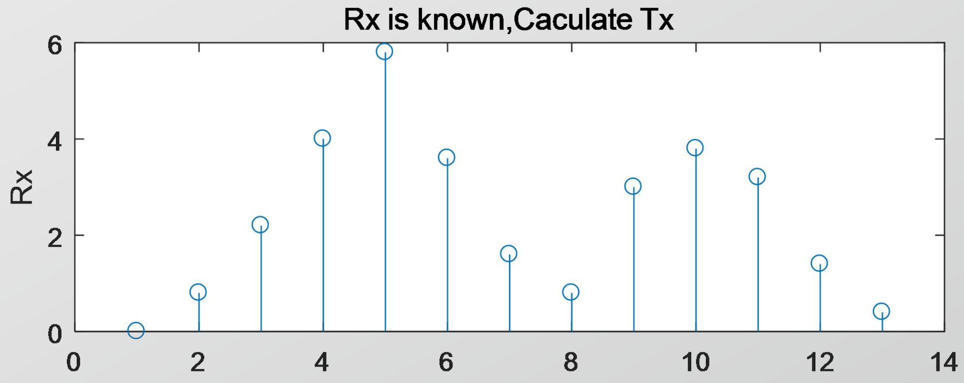

Transmitter
x[n]
Signal Ttransmitter
(LTI) —h1[n]
Receiver
y[n]

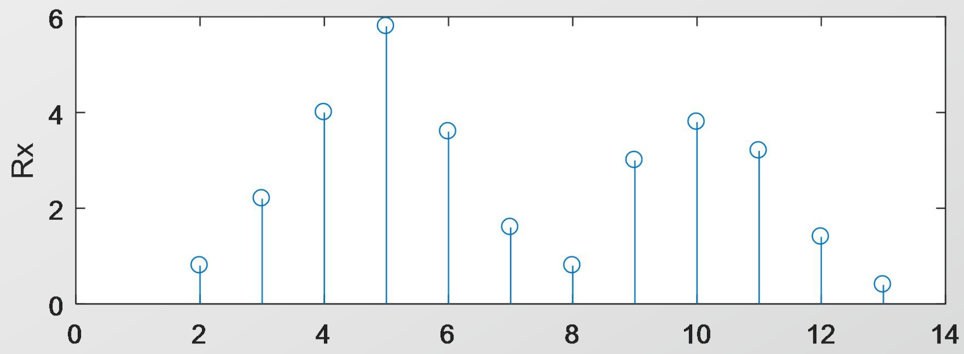

Detected Signal z[n]
z[n] = x[n] 或
z[n]=x[n-k]
Signal Detector
(LTI) —— h2[n]
Received signal
y[n]

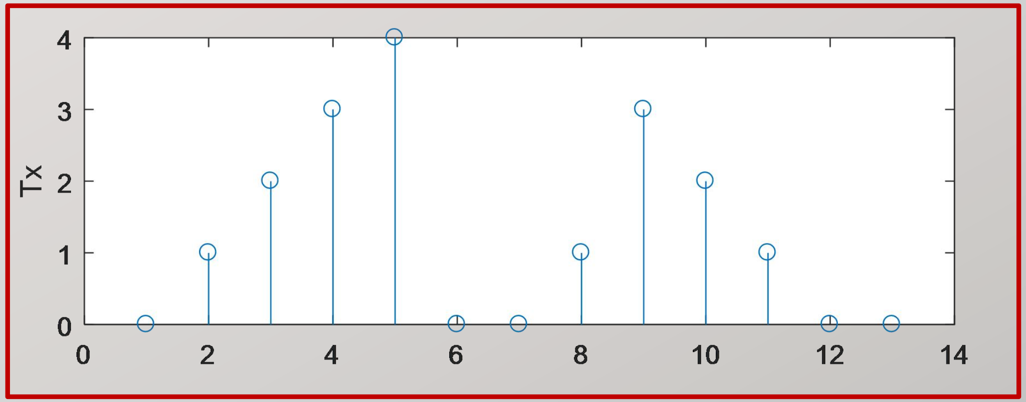
x[n]

z[n]


Tx is known,Caculate Rx
Tx
0
1
2
3
4
0
2
4
6
8
10
12
14


Rx is known,Caculate Tx
Rx
0
2
4
6
0
2
4
6
8
10
12
14


Rx
0
2
4
6
0
2
4
6
8
10
12
14


Tx
0
1
2
3
4
0
2
4
6
8
10
12
14

令 $z'[n] = z[n-1]$

■ 2.10 Echo Cancellation via Inverse Filtering

In this exercise, you will consider the problem of removing an echo from a recording of a speech signal. This project will use the audio capabilities of MATLAB to play recordings of both the original speech and the result of your processing. To begin this exercise you will need to load the speech file `lineup.mat`, which is contained in the Computer Explorations Toolbox. The Computer Explorations Toolbox can be obtained from The MathWorks at the address provided in the Preface. If this speech file is already somewhere in your MATLABPATH, then you can load the data into MATLAB by typing

```
>> load lineup.mat
```

You can check your MATLABPATH, which is a list of all the directories which are currently accessible by MATLAB, by typing `path`. The file `lineup.mat` must be in one of these directories.

Once you have loaded the data into MATLAB, the speech waveform will be stored in the variable `y`. Since the speech was recorded with a sampling rate of 8192 Hz, you can hear the speech by typing

```
>> sound(y,8192)
```

You should hear the phrase "line up" with an echo. The signal $y[n]$, represented by the vector `y`, is of the form

$$y[n] = x[n] + \alpha x[n - N], \tag{2.21}$$

where $x[n]$ is the uncorrupted speech signal, which has been delayed by N samples and added back in with its amplitude decreased by $\alpha < 1$. This is a reasonable model for an echo resulting from the signal reflecting off of an absorbing surface like a wall. If a

Autocorrelation Function

Definition:

Autocorrelation, also known as serial correlation or cross-autocorrelation, is the cross-correlation of a signal with itself at different points in time (that is what the cross stands for). Informally, it is the similarity between observations as a function of the time lag between them.

Autocorrelation of u[n]: $w[n] = u[n] * u[-n]$ **Convolution**

Major Properties:

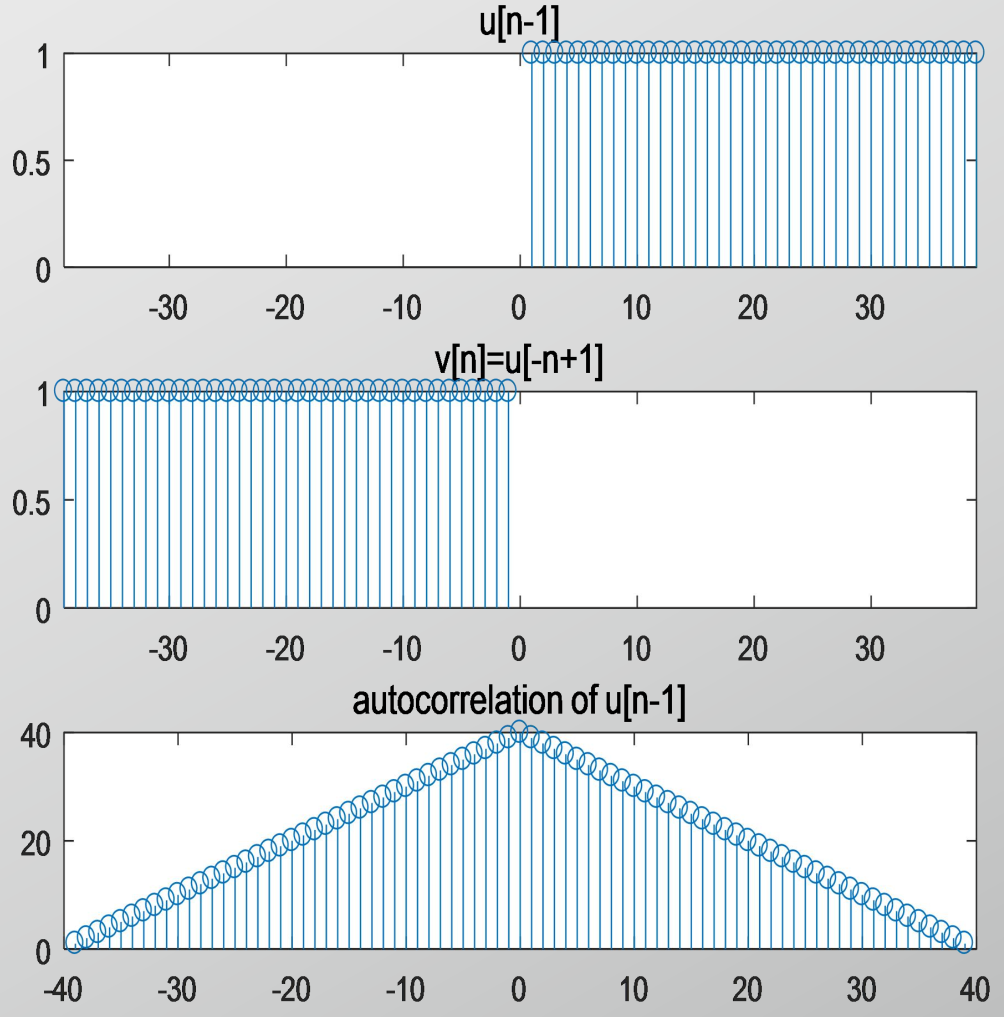
1. The autocorrelation function is a dual function, and its graph is symmetrical to vertical axis.
2. When t=0, the autocorrelation function has a maximum value;
3. The autocorrelation function of the periodic signal is still a periodic signal of the same frequency.

An Simple Example

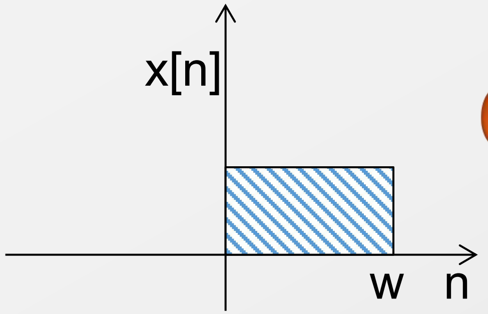
$\rightarrow x[n] = u[n-1]$, $Calculate\ its\ autocorrelation\ function\ \ Rxx$

```
u=ones(1,40);
nu = 1:40;
v=u(end:-1:1);
nv=-40:-1;
w=conv(u,v);
nw=nu(1)+nv(1):nu(end)+nv(end);
figure
subplot(3,1,1),stem(nu,u),title('u[n-1]'),xlim([nw(1) nw(end)]);
subplot(3,1,2),stem(nv,v),title('v[n]=u[-n+1]'),xlim([nw(1)
nw(end)]);
subplot(3,1,3),stem(nw,w),title('autocorrelation of u[n-1]');
```

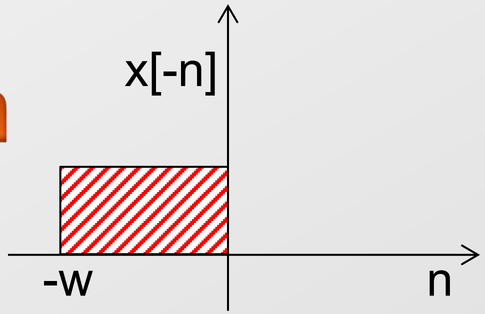
fliplr



Convolution

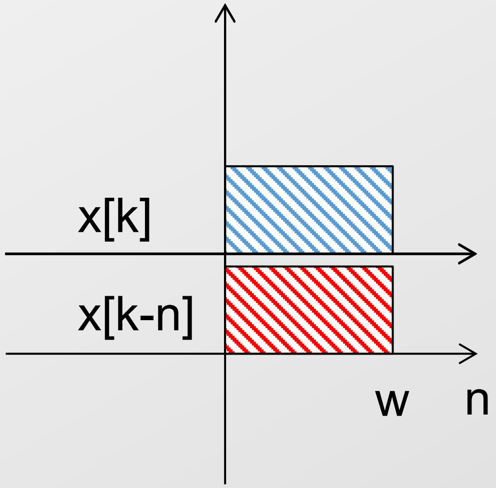


$*$

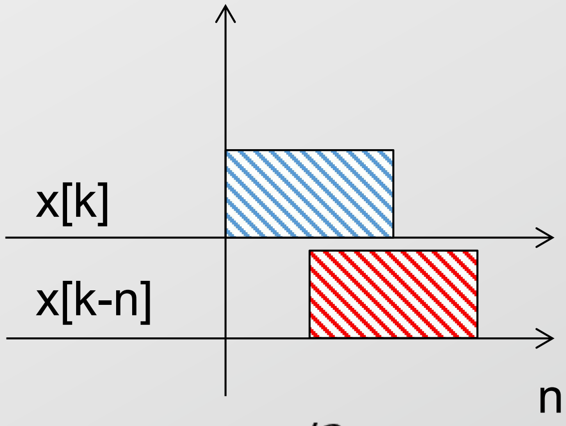


Flip Slide Multiply Sum

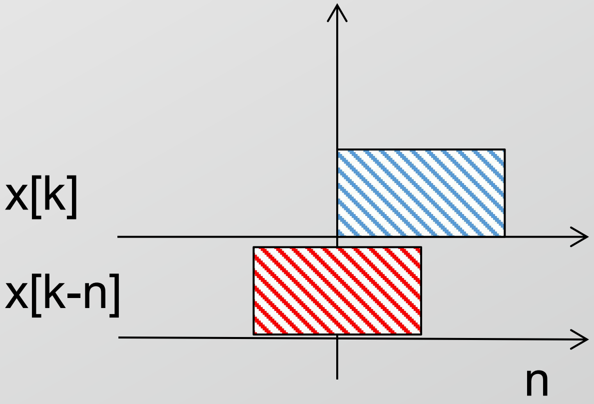
$$Rxx[n] = x[\mathrm{n}] * x[-\mathrm{n}] = \sum_{k=-\infty}^{+\infty} x[k]x[k-n]$$



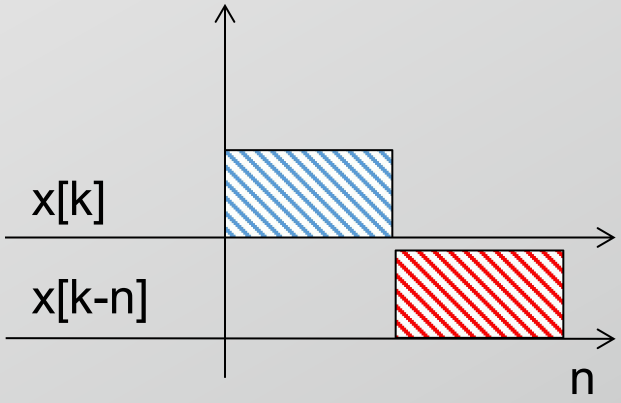
$n = 0$



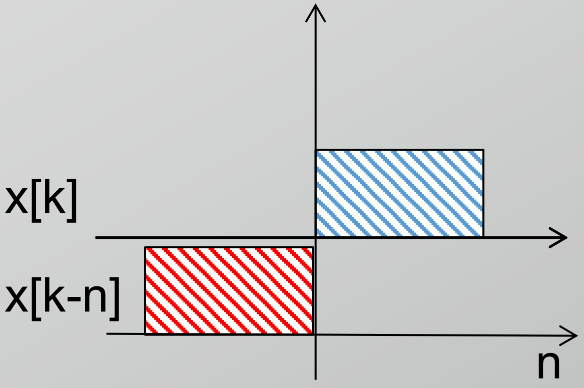
$n = w/2$



$n = -w/2$



$n = w$



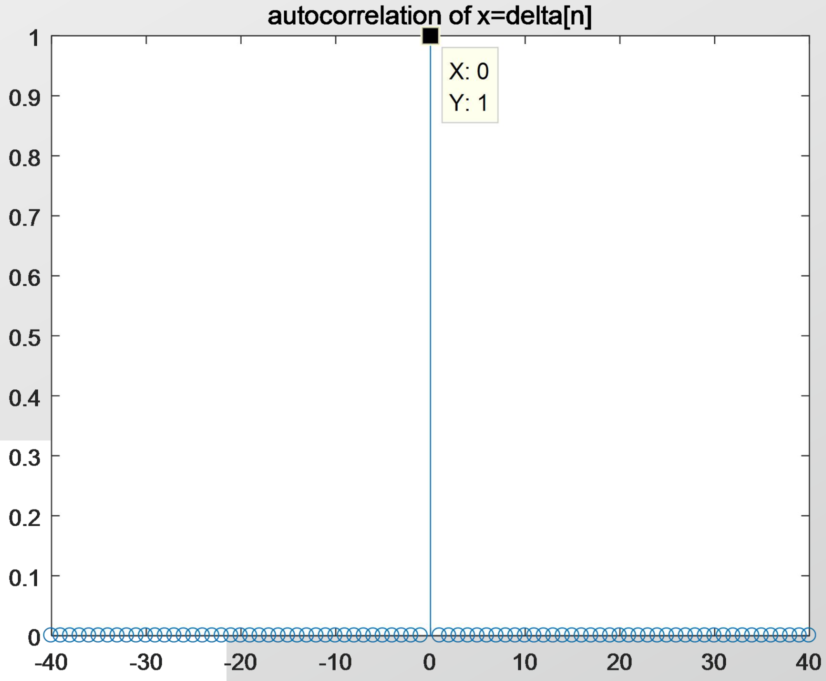
$n = -w$

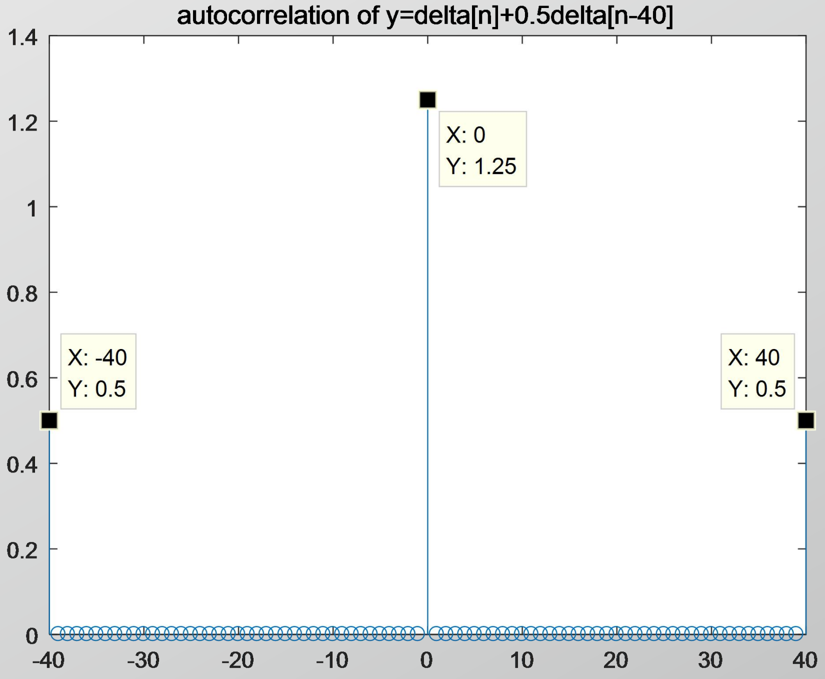
Another Simple Example of form 2.10

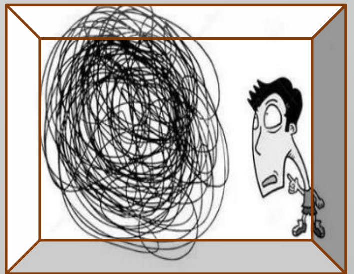
Suppose $y[n] = x[n] + 0.5x[n-40]$ $\quad$ $when\ x[n] = \delta[n]\ ,\ caculate\ Rxx$

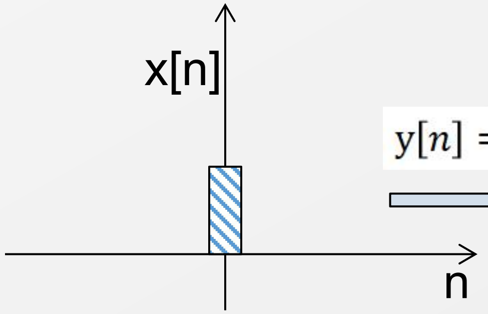
```
x=[zeros(1,20),1,zeros(1,20)];
nx=-20:1:20;
Rxx=conv(x,fliplr(x));
nRxx=-40:40;
stem(nRxx,Rxx);
title('autocorrelation of x=delta[n]');


y=filter([1,zeros(1,39),0.5],1,x);
ny=nx;
Ryy=conv(y,fliplr(y));
nRyy=nRxx;
figure
stem(nw,w);
title('autocorrelation of y=delta[n]+0.5delta[n-40]');
```

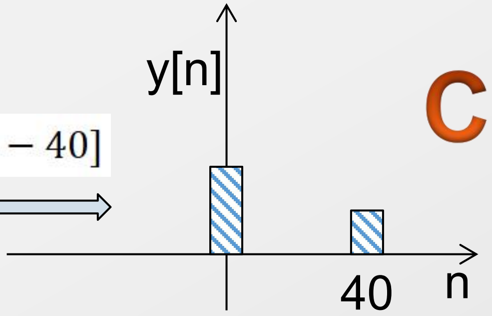






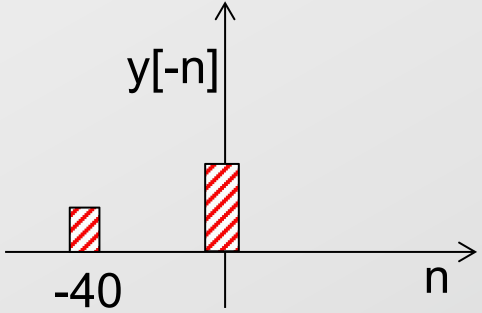


$y[n] = x[n] + 0.5x[n-40]$



Convution

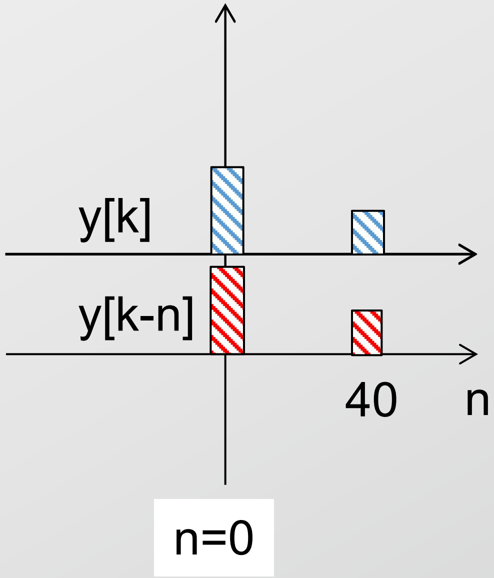
*

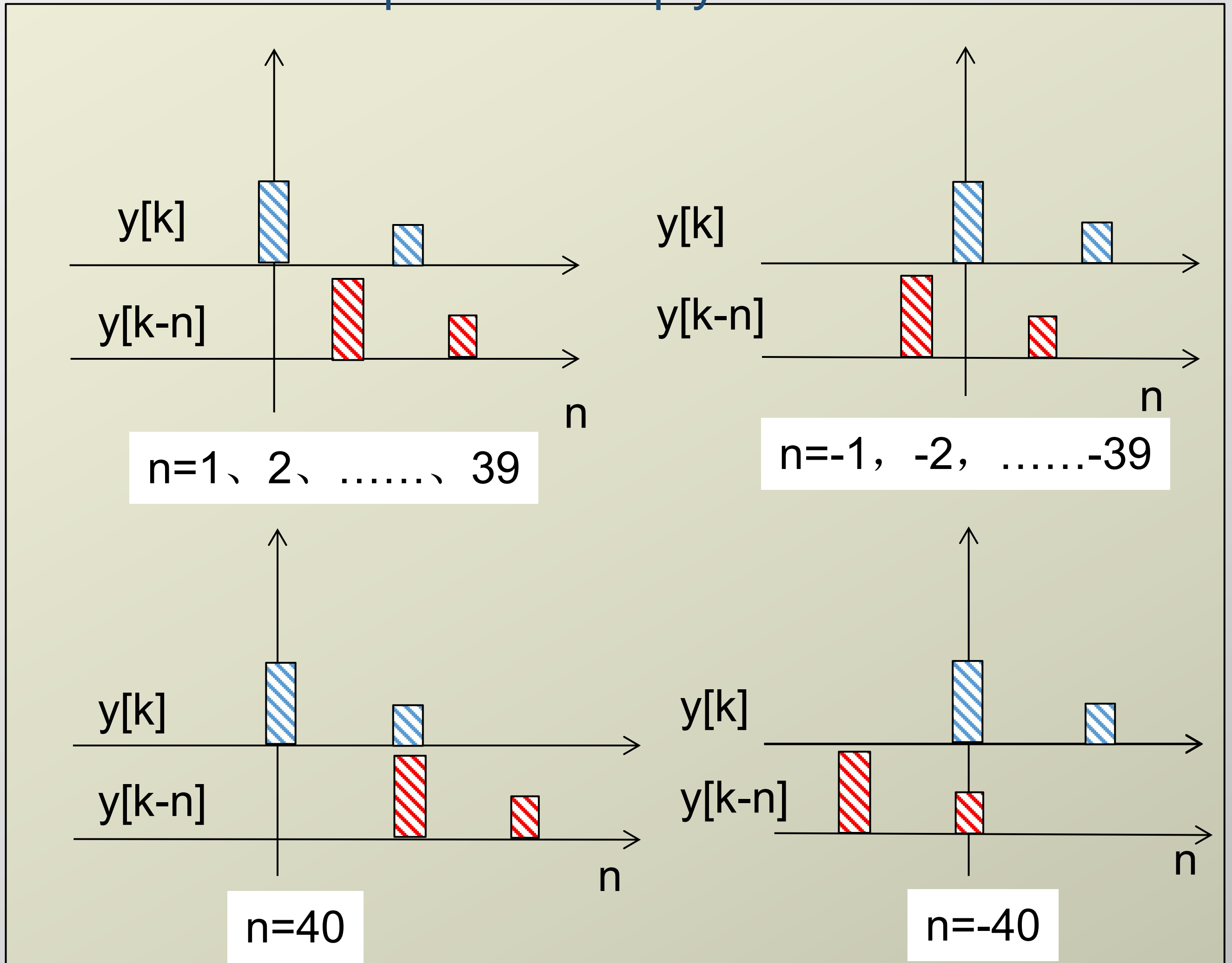


Flip Slide Multiply Sum

N √

$$\mathrm{Ryy}[n] = y[n] * y[-n] = \sum_{k=-\infty}^{k=\infty} y[k]y[k-n]$$





a ?

$$Ryy[n] = y[n] * y[-n] = x[n] * (\delta[n] + \alpha\delta[n-N]) * x[-n] * (\delta[-n] + \alpha\delta[-n-N])$$

$$= Rxx[n] * \{(\delta[n] + \alpha\delta[n-N]) * (\delta[-n] + \alpha\delta[-n-N])\}$$

$$= Rxx[n] * (\delta[n] + \alpha\delta[n-N] + \alpha\delta[n+N] + \alpha^2\delta[n-N] * \delta[n+N])$$

$$= Rxx[n] * \left((1+\alpha^2)\delta[n] + \alpha\delta[n+N] + \alpha\delta[n-N]\right)$$

$$= (1+\alpha^2)Rxx[n] + \alpha Rxx[n+N] + \alpha Rxx[n-N]$$

$$Ryy[0] = (1+\alpha^2)Rxx[0] + \alpha Rxx[N] + \alpha Rxx[-N] = (1+\alpha^2)Rxx[0] + 2\alpha Rxx[N]$$

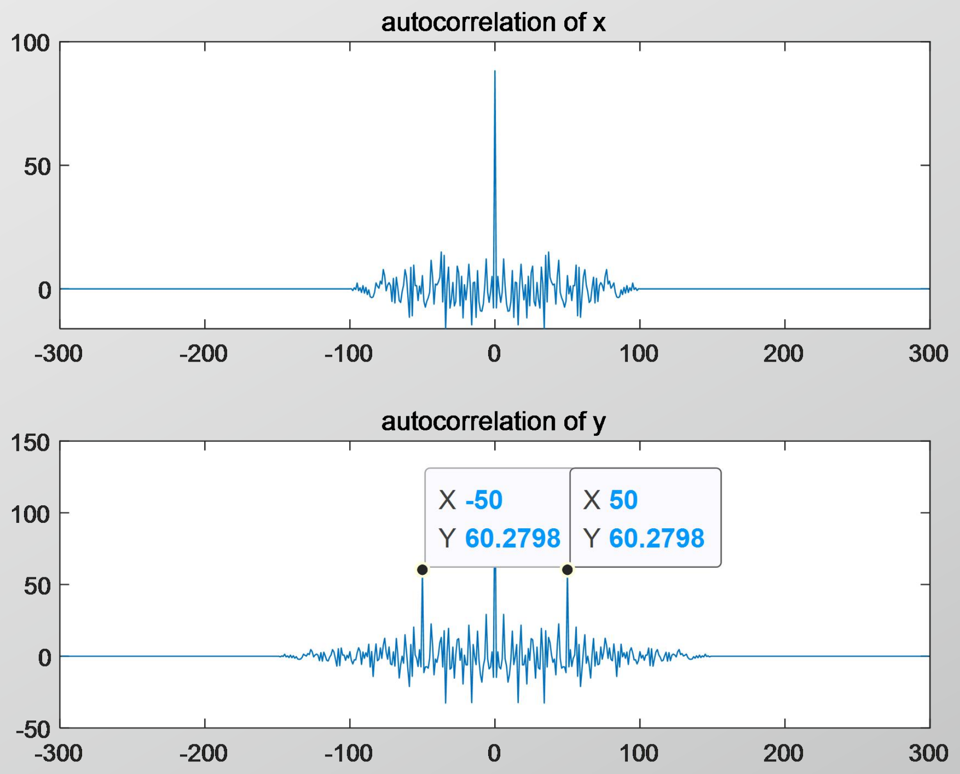
$$Ryy[N] = (1+\alpha^2)Rxx[N] + \alpha Rxx[2N] + \alpha Rxx[0]$$

$$Ryy[-N] = (1+\alpha^2)Rxx[-N] + \alpha Rxx[0] + \alpha Rxx[-2N]$$

$$y[n] = x[n] + \alpha x[n - N]$$

Suppose $N=50$, $\alpha = 0.9$, $x[n]$ is a random signal

```
%Construct a random signal x
NX = 100;     X = randn(1,NX);
%why should I do this??
 nX = 300;     x = [X,zeros(1,200)];
 %y[n]=x[n]+0.6x[n-50]，Calculate output y  using ‘filter’
 N =50;   alpha = 0.6;
 y = filter([1,zeros(1,N-1),alpha],1,x);
 %Calculate the autocorrelation of x and y respectively
 Rxx = conv(x,fliplr(x));
 Ryy = conv(y,fliplr(y));
 subplot(2,1,1), plot([-nX+1:nX-1],Rxx);
 title('autocorrelation of x');
 subplot(2,1,2), plot([-nX+1:nX-1],Ryy);
 title('autocorrelation of y');
```



We can know N=50 by observing the image

$$y[n] = x[n] + \alpha x[n-N]$$

$$Ryy[0] = (1+\alpha^2)Rxx[0] + \alpha Rxx[N] + \alpha Rxx[-N] = (1+\alpha^2)Rxx[0] + 2\alpha Rxx[N]$$

$$\alpha = \frac{-2Rxx[N] \pm \sqrt{(2Rxx[N])^2 - 4Rxx[0](Rxx[0]-Ryy[0])}}{2Rxx[0]}$$

```
%we have known Rxx, Ryy, N,  now let's calculate a
Rxx0=Rxx(nx);
Rxx_N1=Rxx(nx-N);
Rxx_N2=Rxx(nx+N);
Ryy0=Ryy(nx);
alpha1=(-2*Rxx_N1+sqrt((2*Rxx_N1)^2-4*Rxx0*(Rxx0-Ryy0)))/2/Rxx0
alpha2=(-2*Rxx_N1-sqrt((2*Rxx_N1)^2-4*Rxx0*(Rxx0-Ryy0)))/2/Rxx0
```

```
alpha1 = 0.6000
alpha2 = -0.7220
```

NOW 2.10

(f). Suppose that you were given $y[n]$ but did not know the value of the echo time, N, or the amplitude of the echo, α. Based on Eq. (2.21), can you determine a method of estimating these values? Hint: Consider the output **y** of the echo system to be of the form:

$$\boxed{y[n] = x[n] * (\delta[n] + \alpha\delta[n-N])}$$

and consider the signal,

$$R_{yy}[n] = y[n] * y[-n].$$

This is called the autocorrelation of the signal $y[n]$ and is often used in applications of echo-time estimation. Write $R_{yy}[n]$ in terms of $R_{xx}[n]$ and also plot $R_{yy}[n]$. You will

$$y[n] = x[n] + 0.5x[n-40] \qquad when\ x[n] = \delta[n]$$

$y[n] = x[n] + \alpha x[n-N]$ when N=50, α = 0.6, x is a random signal

x[n] is known

if x[n] is unknown, what can we do ?

主观题 10分

设置

Determine the relationship between $x[n]$ and $y[n]$;

Determine the relationship between $x[n]$ and $y_2[n]$;

Determine the relationship between $x[n]$ and $y_3[n]$;

作答

x[n] is unknown

$$y[n] = x[n] + \alpha x[n-N]$$

y is known, x is unkbown, caculate N， α。

$$\begin{aligned}
Ryy[n] &= y[n] * y[-n] = x[n] * \left(\delta(n) + \alpha\delta(n-N)\right) * x[-n] * \left(\delta(-n) + \alpha\delta(-n-N)\right) \\
&= Rxx[n] * \left\{\left(\delta(n) + \alpha\delta(n-N)\right) * \left(\delta(-n) + \alpha\delta(-n-N)\right)\right\} \\
&= Rxx[n] * \left(\delta(n) + \alpha\delta(-n-N) + \alpha\delta(n-N) + \alpha^2\delta(n-N) * \delta(-n+N)\right) \\
&= Rxx[n] * \left((1+\alpha^2)\delta(n) + \alpha\delta(n+N) + \alpha\delta(n-N)\right) \\
&= (1+\alpha^2)Rxx[n] + \alpha Rxx[n+N] + \alpha Rxx(n-N)
\end{aligned}$$

Traversal

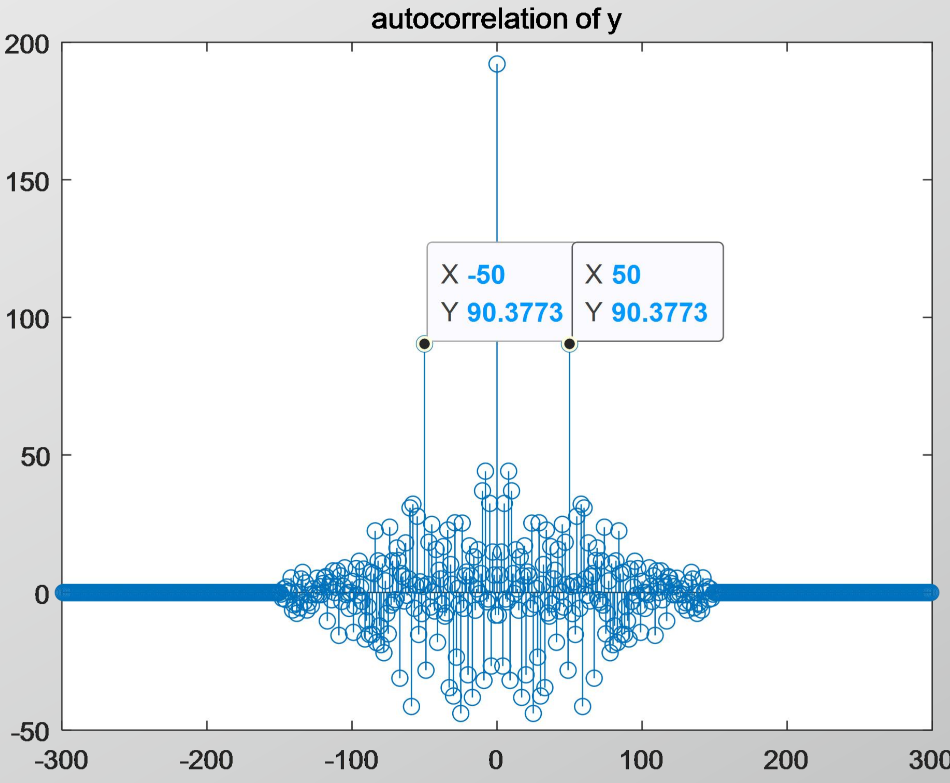
$y[n] = x[n] + \alpha x[n-1]$ $\xrightarrow{\text{draw Ryy}}$ get N

y[n] N is known, assume 'α'

$\hat{x}[n] + \alpha\hat{x}[n-1] = y[n]$ $\xrightarrow{\hat{x}[n]}$ $\widehat{Rxx}$

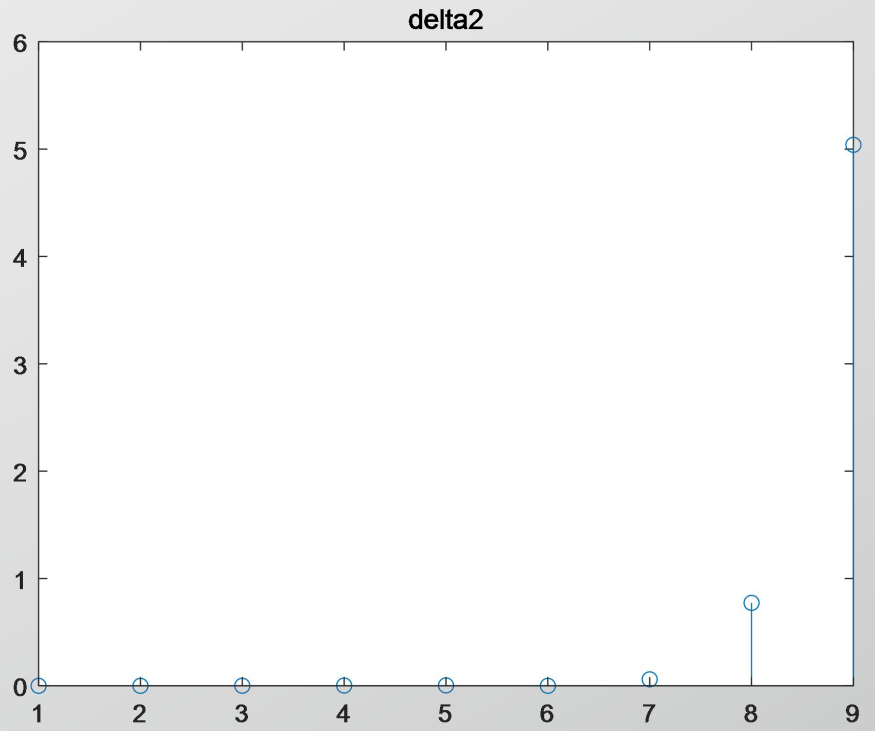
FEASIBILITY ANALYSIS

```
%x[n]
x = randn(1,100);
NX = 300;
x = [x,zeros(1,200)];
%y[n]=x[n]+0.9x[n-50]， caculate y
N =50;
alpha = 0.6;
y = filter([1,zeros(1,N-1),alpha],1,x);
%from now on, we only know y[n] and y[n]=x[n]+ax[n-N], let’t calculate α， N
Ryy = conv(y,fliplr(y));
stem([-NX+1:NX-1],Ryy)， title('autocorrelation of y');
```

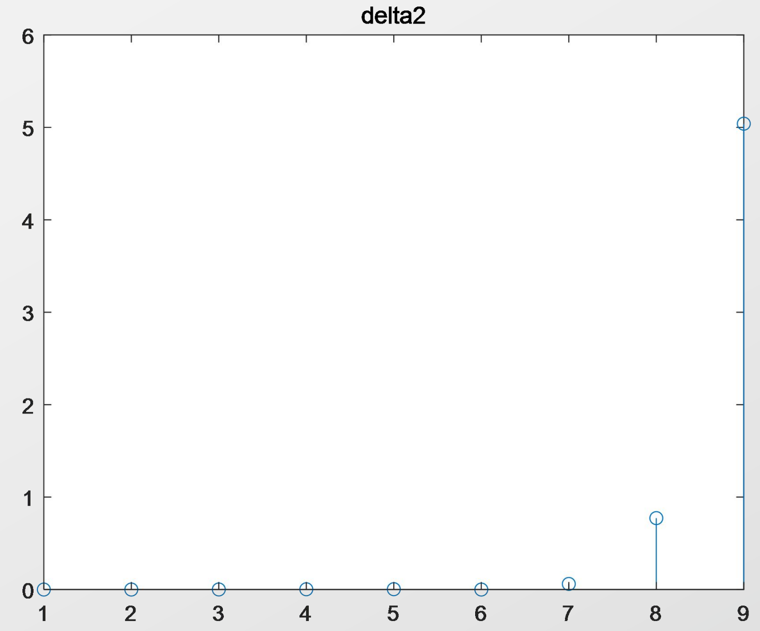


from the figure, we know N=50

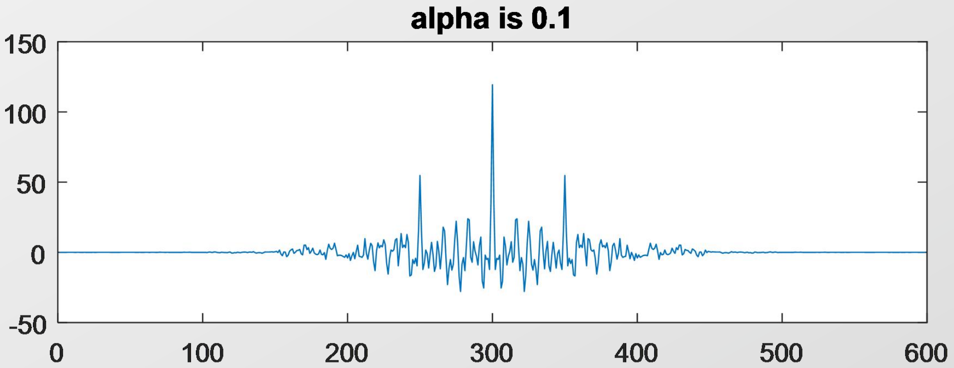
```
N=50;
for i=1:1:9
alpha=i*0.1;
%caculate x_re
x_re=filter(1,[1,zeros(1,N-1),alpha],y);
%Rxx_re
Rxx_re=conv(x_re,fliplr(x_re));
%Estimate the error between Rxx_re and Ryy
M_diff=Ryy-(1+alpha^2)*Rxx_re-alpha*[zeros(1,N),Rxx_re(1:2*NX-1-N)]-
alpha*[Rxx_re(N+1:2*NX-1),zeros(1,N)];
delta2(i)=std(M_diff.^2);
figure
subplot(2,1,1),plot(Rxx_re),title(['alpha is ',num2str(alpha)])
subplot(2,1,2),plot(M_diff),title('M_diff')
end
figure
stem(delta2),title('delta2')
```

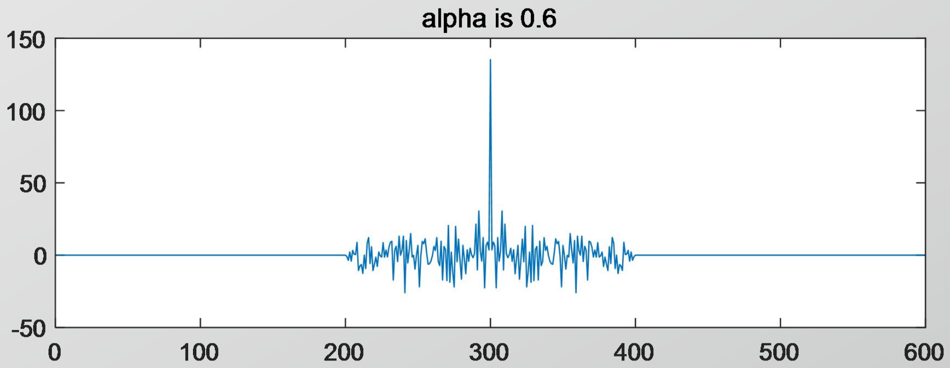


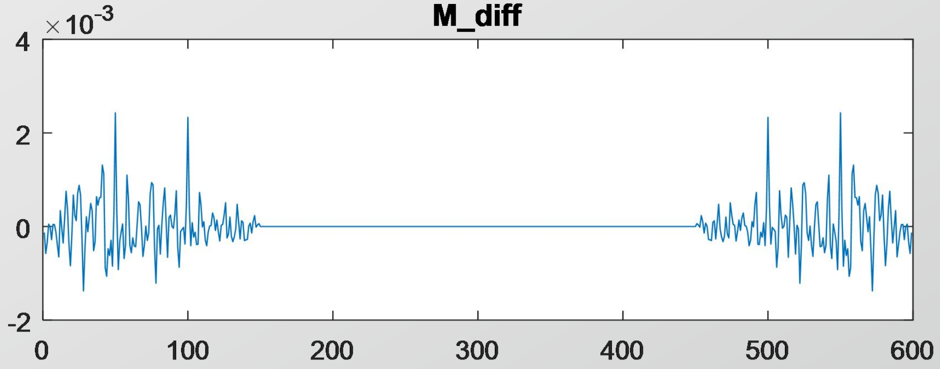
optimal solution of α

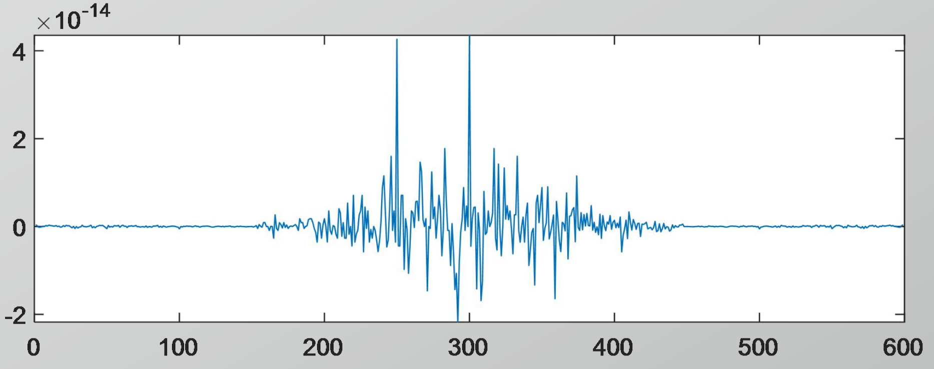


optimal solution of α









The received echo signal y,y2,y3 are known， and

$$y[n] = x[n] + 0.5x[n - 1000]$$

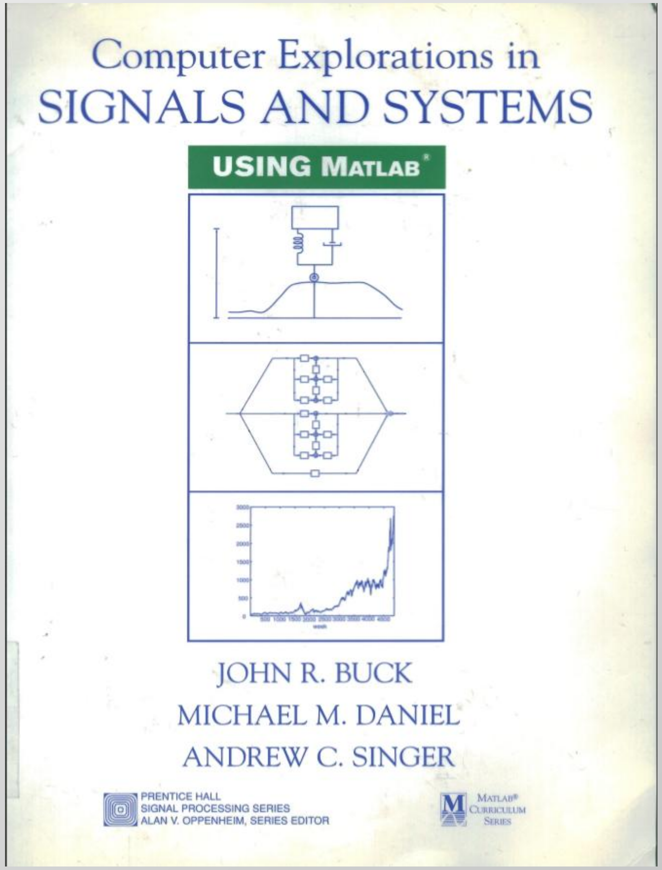
$$y2[n] = x[n] + \alpha x[n - N]$$

$$y3[n] = x[n] + \alpha 1x[n - N1] + \alpha 2x[n - N2]$$

Now, let´t caculate α and N！

Lab2 Assignments

- Read tutorial 2.1、2.2、2.3
- homeworks: 2.4，2.10.



Tips：

1 Download **lineup.mat** for 2.10-----BB

2 y、y2、y3 are column vectors----fliplr vs flipud

3 For 2.10 f , there are two methods to caculate N and α

a: The original signal x[n] is known---Solve equations.

b: The original signal x[n] is unknown---Traversal .

Tips：Solve equations by Matlab

```
%Solve One Variable Quadratic Equation
syms x
result=solve(x^2+2*x-19.25==0,x)
double(result)
```

```
%Solve Two Variables Quadratic Equation
syms x y;
S=solve([x^2+y*2-17.54==0,x+y==5.8],[x,y])
double(S.x)
double(S.y)
```