**Advanced Signal Processing**

EEE411

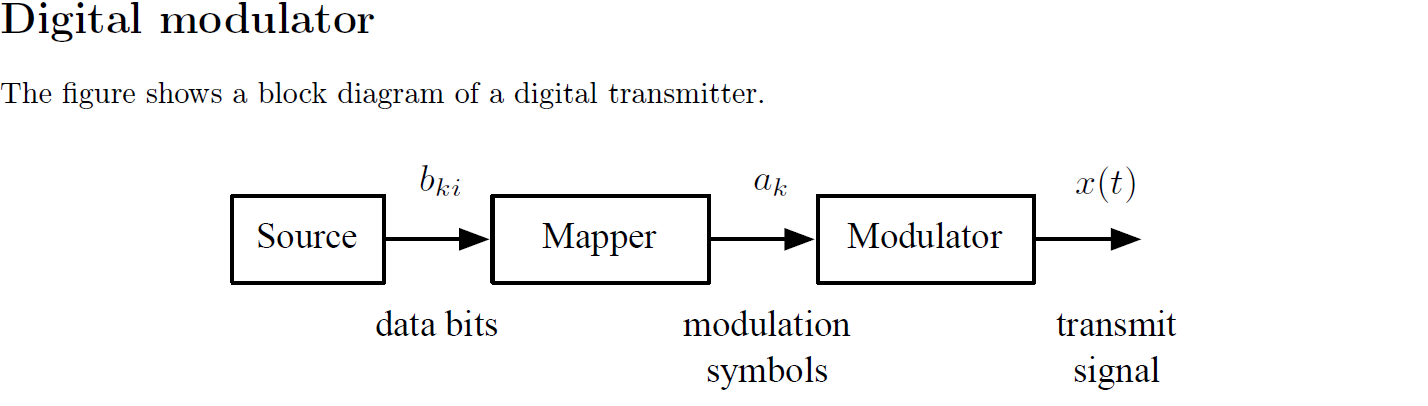
Assessment 2 - Report

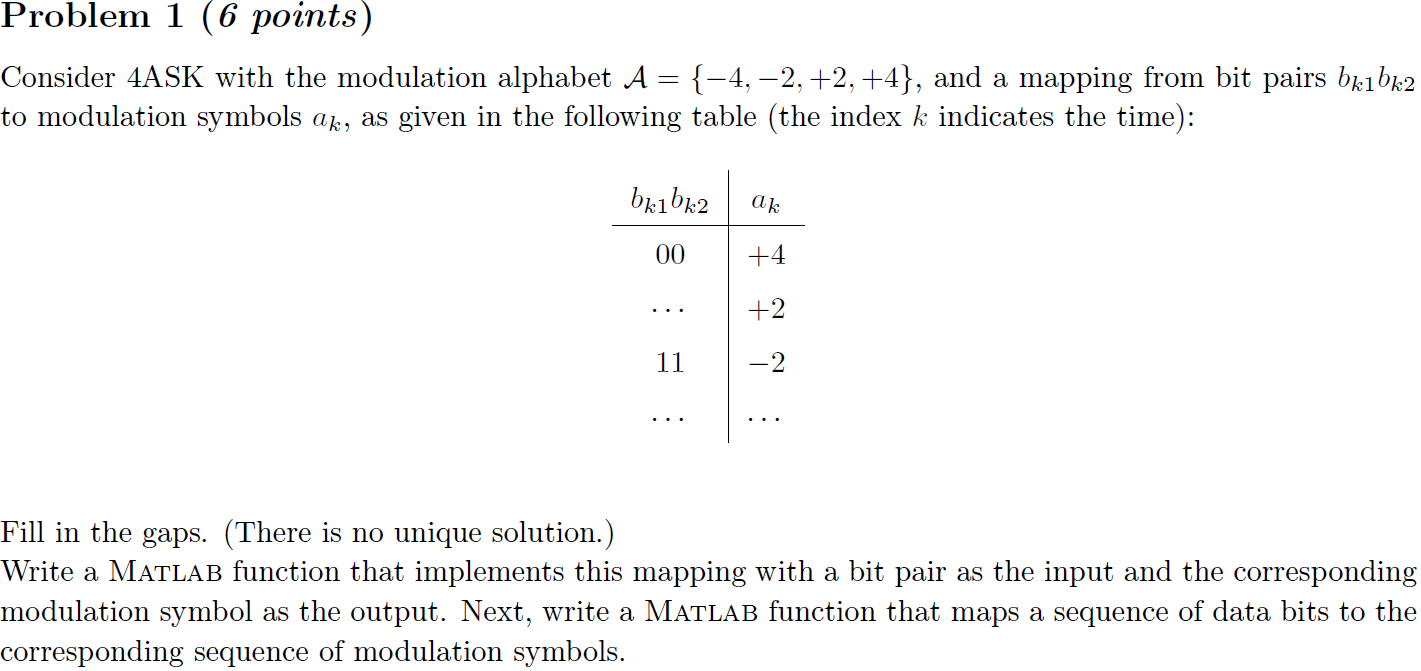
Name: Cong Yin

ID Number: 1715471

Introduction

These practical deals with the simulation of a digital communication system with Matlab. We will implement a transmitter and a receiver of a baseband system, and investigate the effect of noise to the system. For this assessment that will divided into three parts, the first part is digital modulator, the second part is digital demodulator and the last part is digital transmission system.





**Solution**

4ASK(Amplitude Shift Keying for quaternary) is a kind of method to modulate. Compare with 4PSK, 4ASK use the 00,01,10,11 represent the different range, and 4PSK use the 00,01,10,11 represent the different phase position.

The listing below shows a Matlab function that was used to solve the question. For Matlab script, the ‘y\_mapping’ are symbols. Data bits is a random value, so each process of run the script will get the different value.

function signal\_mapping = symbols\_mapping(imput)

new=zeros(4,1);

for i=1:4

if imput(1,i)==0

if imput(2,i)==0

new(i)=4;

elseif imput(2,i)==1

new(i)=-4;

end

elseif imput(1,i)==1

if imput(2,i)==1

new(i)=2;

elseif imput(2,i)==0

new(i)=-2;

end

end

end

signal\_mapping=new;

end

Then run the following Matlab script and get the result as shown in figure 1-1 and figure 1-2

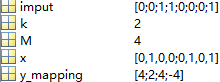
M=4;

k=log2(M);

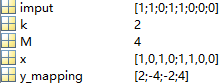
imput=randi([0,1],8,1);

x=reshape(imput,k,[]);

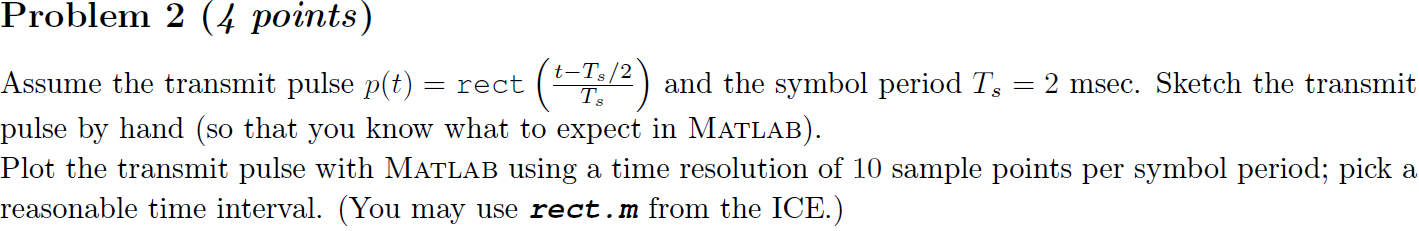
y\_mapping = symbols\_mapping(x);



**Figure 1-1**



**Figure 1-2**



**Solution**

The listing below shows a Matlab function that was used to solve the question, which from the ICE.

function x = rect(t)

% Rectangular pulse between -1/2 and +1/2.

%

% Usage: x = rect(t)

% (c) Siyi Wang, EEE/XJTLU, 29 August 2014

x = zeros(size(t));

x( abs(t) < 1/2 ) = 1;

x( abs(t) == 1/2 ) = 1/2;

Then run the following Matlab script and get the result as shown in figure 2-1, sketch the transmit pulse by hand as shown in figure 2-2.

Ts=0.002; %define the numerical value for Ts

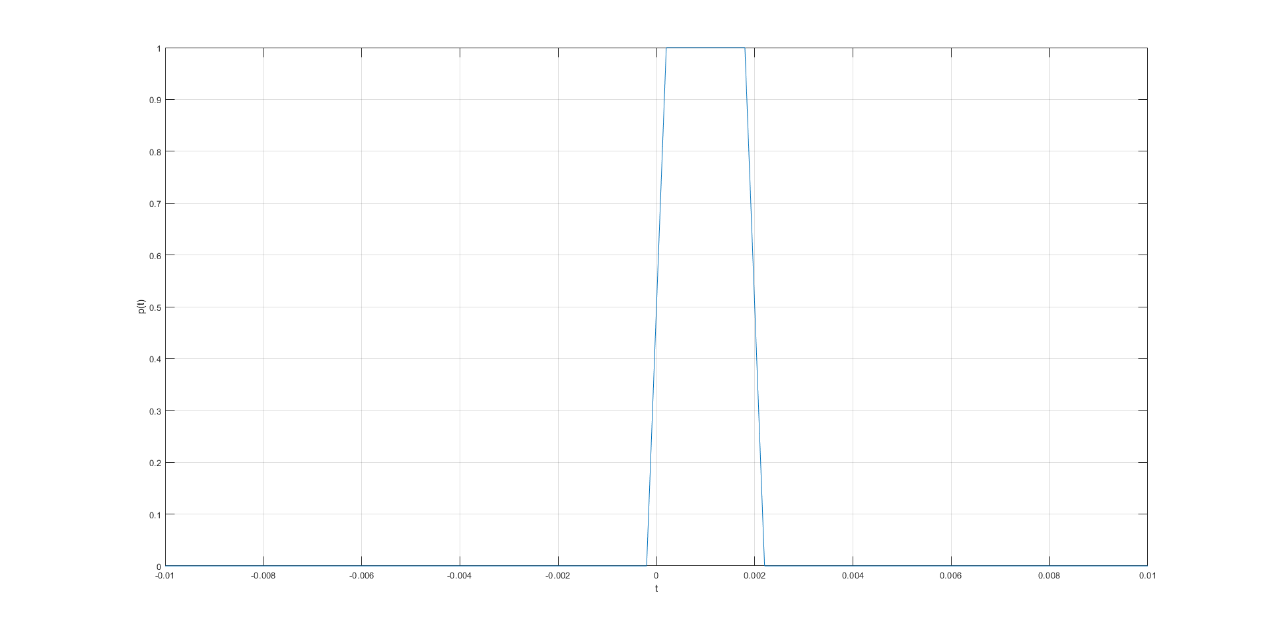
t=-0.01:Ts/10:0.01; %Ts/10=T0

p=rect((t-Ts/2)/Ts);

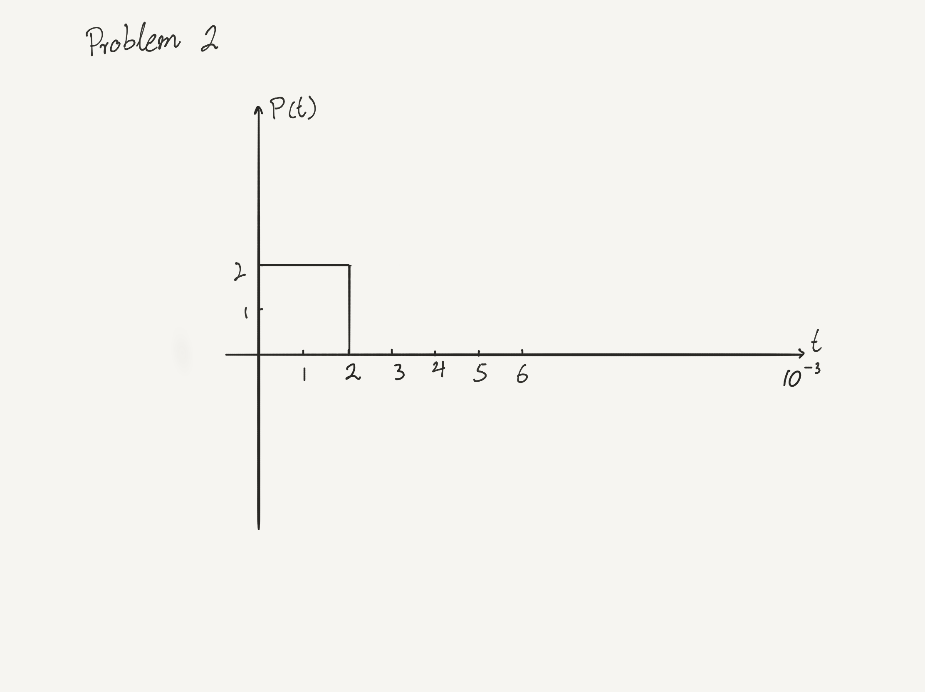
plot(t,p);

grid on

xlabel ('t');ylabel ('p(t)');

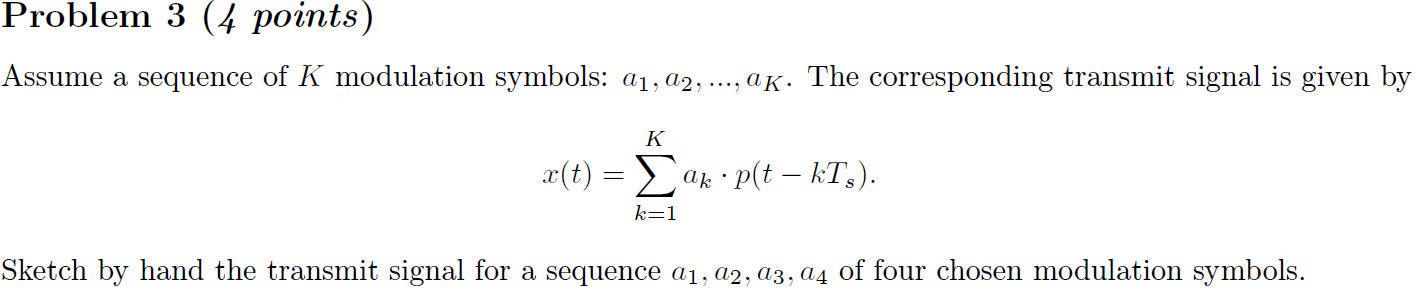


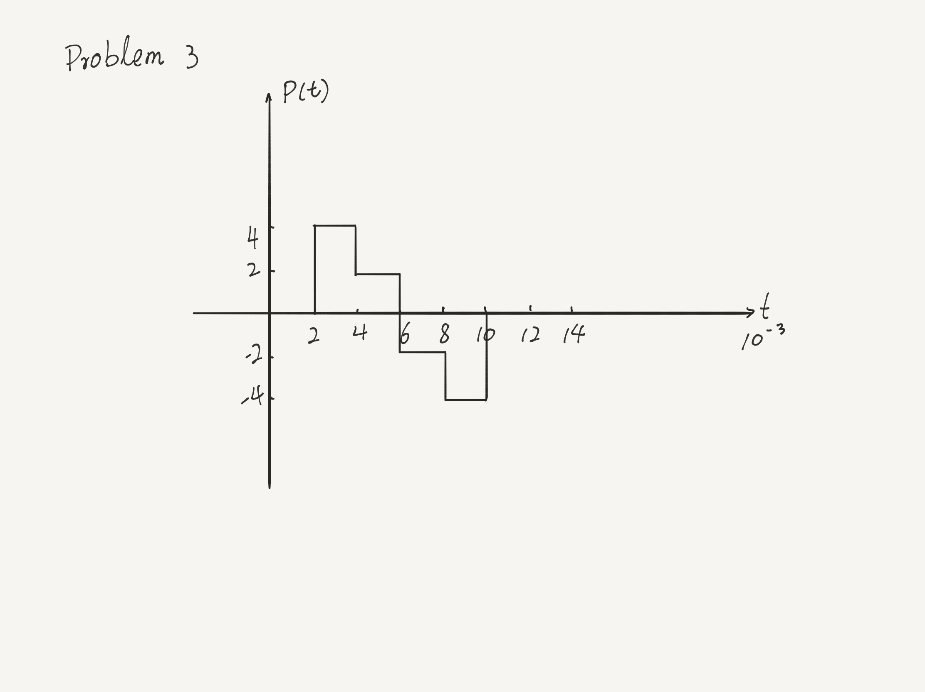
**Figure 2-1 Transmit pulse**

****

**Figure 2-2 Sketch the transmit pulse by hand**

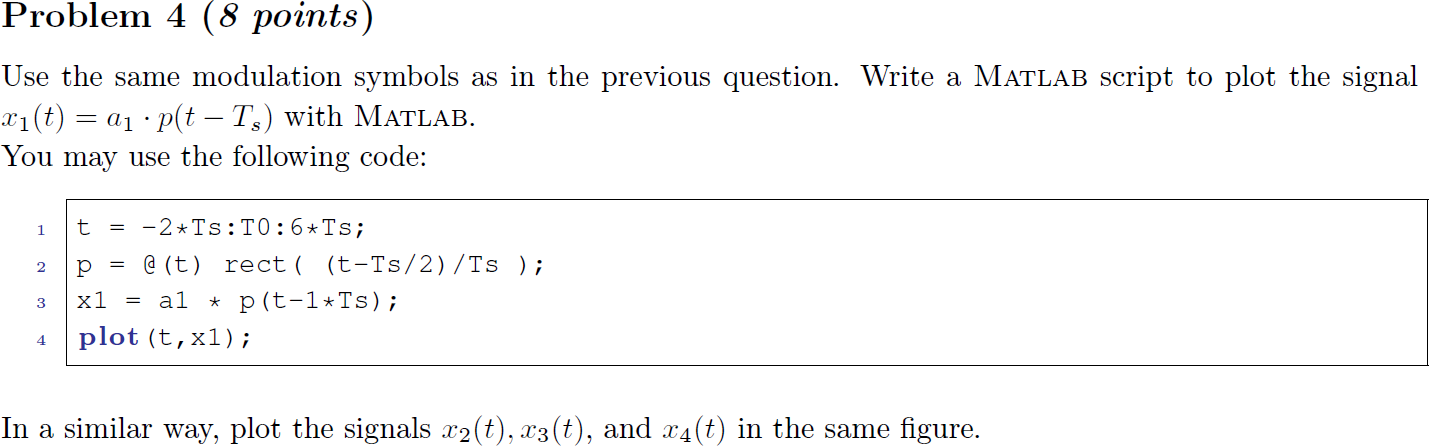
As shown in figure 2-2, the square wave is a ideal rectangular pulse, however the Figure 2-1 has a little difference between second figure, the MATLAB cannot print the ideal rectangular pulse. When the p(t) equals to , however in ideal conditions, there are not



****

**Figure 3-1 Sketch by hand the transmit signal**

The function was given by the question, the four symbols should be chosen that divide to a1, a2,a3 and a4, the number of K will become to 4. After shift, when t=0, there are no value, the symbol duration is very important for this question.



**Solution**

Run the following Matlab script and get the result as shown in figure 4-1

Ts=0.002;

T0=Ts/10;

t = -2\*Ts:T0:6\*Ts;

p = @(t)rect((t-Ts/2)/Ts);

x1 = 4\*p(t-1\*Ts);

x2 = 2\*p(t-2\*Ts);

x3 = -2\*p(t-3\*Ts);

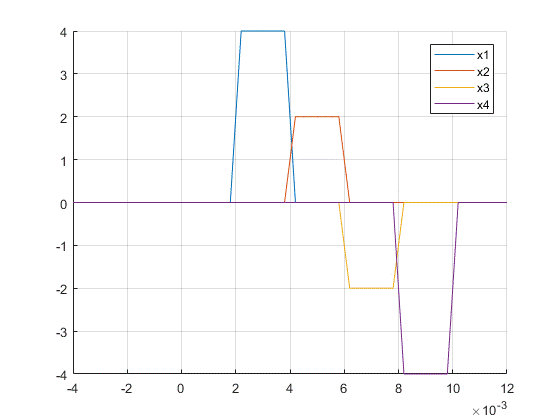
x4 = -4\*p(t-4\*Ts);

hold on;

grid on;

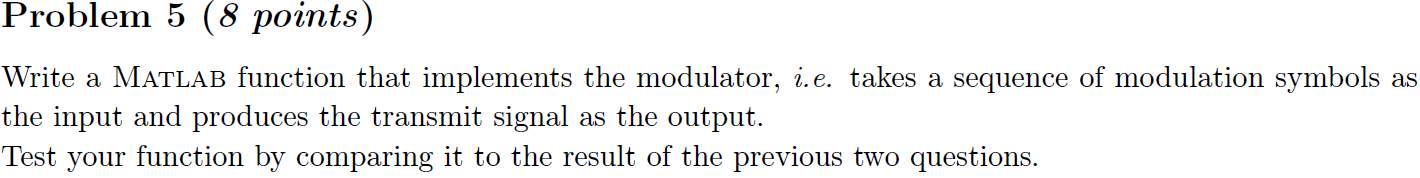
plot(t,x1,t,x2,t,x3,t,x4);

legend('x1','x2','x3','x4');

****

**Figure 4-1 Plot the four signal in the same figure**

The figure 4-1 shows the different signal with different color.



**Solution**

The fourth question shows the code, which was for plot the different signal x1 to x4. For this question, the process is same as fourth question, it is just need add a for loop to show this part . The result is vary similar to the figure 4-1, it just has a little different, which generated by the transmitting process.

The listing below shows a Matlab function that was used to solve the question

function [t,ak]=modulator(symbols)

Ts=0.002;

T0=Ts/10;

t=-2\*Ts:T0:6\*Ts;

n=size(symbols,2);

p=@(t)rect((t-Ts/2)/Ts);

ak=0;

for k=1:n%the step is calculated by the function, which from the question 3

ak=ak+symbols(1,k)\*p(t-k\*Ts);

end

end

Then run the following Matlab script and get the result as shown in figure 5-1

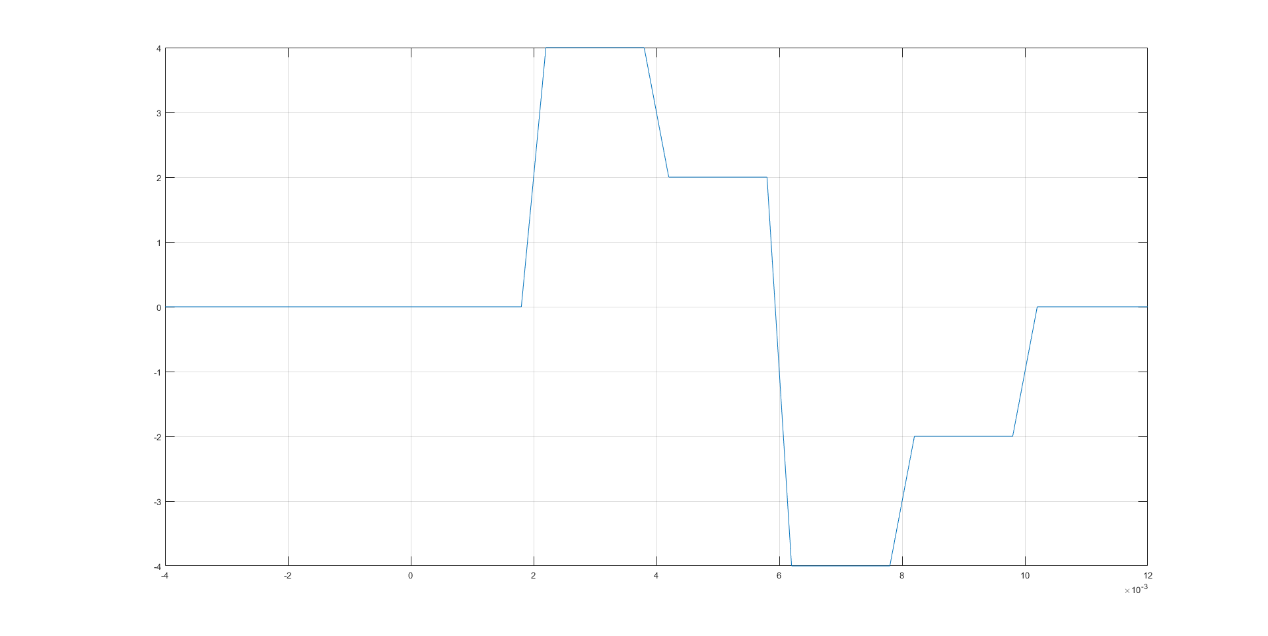
clc,clear;

symbols=[4,2,-2,-4];

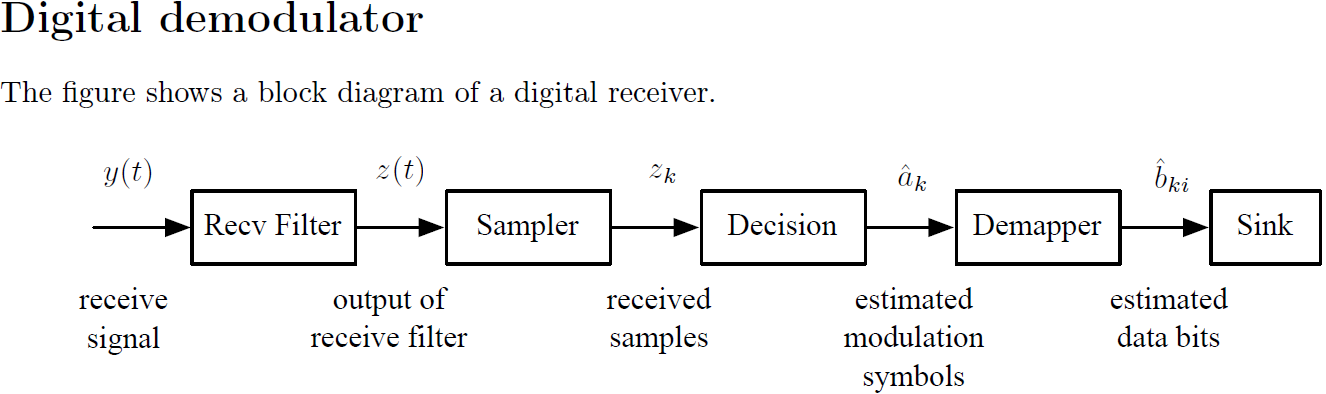
[t,modulation]=modulator(symbols);

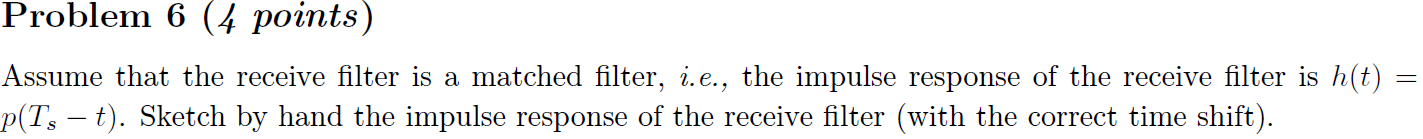
plot(t,modulation)

grid on

****

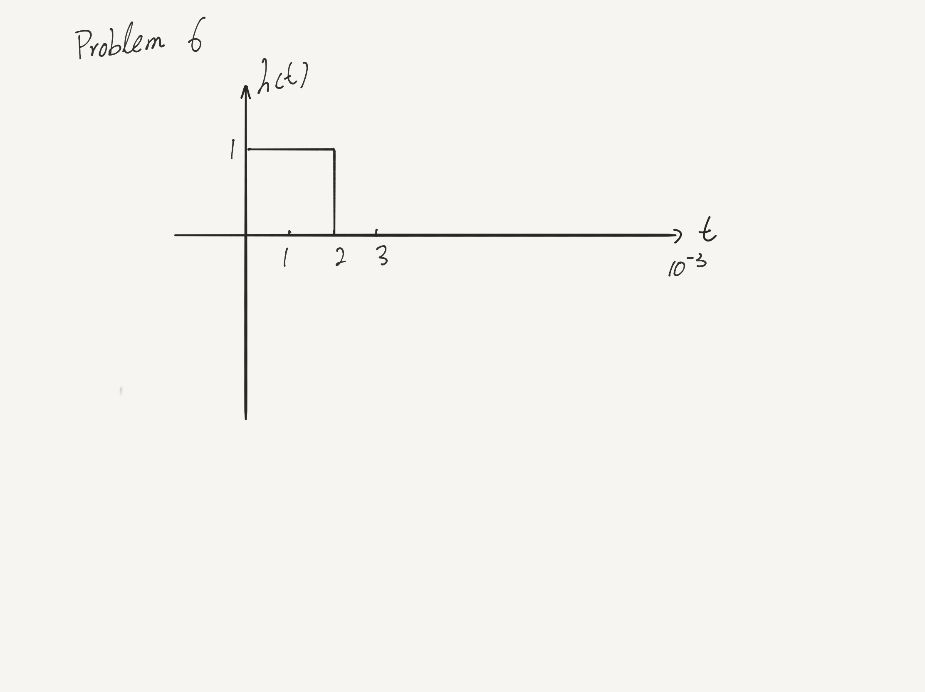
**Figure 5-1 Transmitted signal**



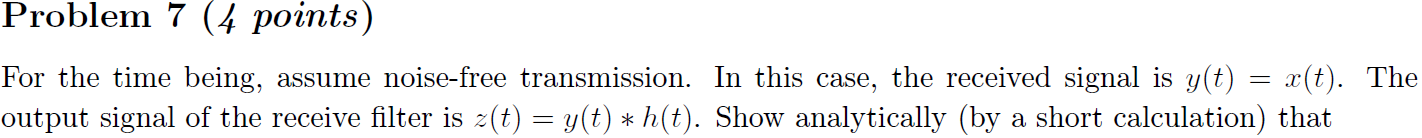
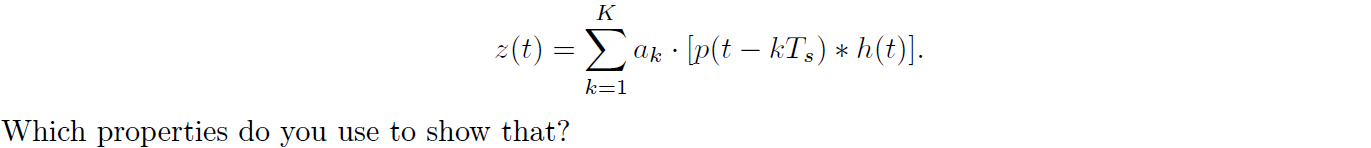


**Solution**

For the function , the first step is make the p(t) become to p(-t), and then shift p(-t) will get the following figure 6-1

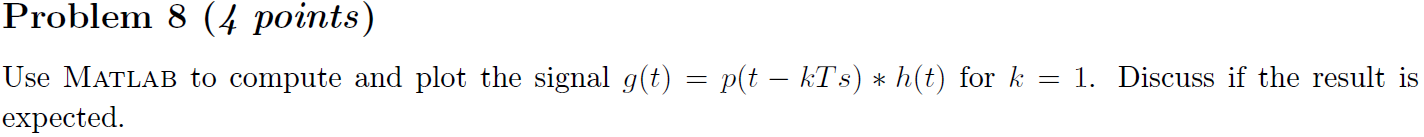
****

**Figure 6-1 Sketch by hand the impulse response of the receive filter (with the correct time shift)**

**Solution**

The above function is based on the third question the different between and is h(t)(match filter) that used the distribution law . was calculated by ak1 to ak4(4,2,-2,-4).



**Solution**

Run the following Matlab script and get the result as shown in figure 8-1

if the result is expected, it is shown as figure 8-1, when k=1, y-p(t-Ts) and h=p(Ts-t), the result of convolution is depend on the value of Ts.

clc,clear;

Ts=0.002;

t=0:Ts/10:6\*Ts;

p=@(t)rect((t-Ts/2)/Ts);

k=1;

y=p(t-k\*Ts);

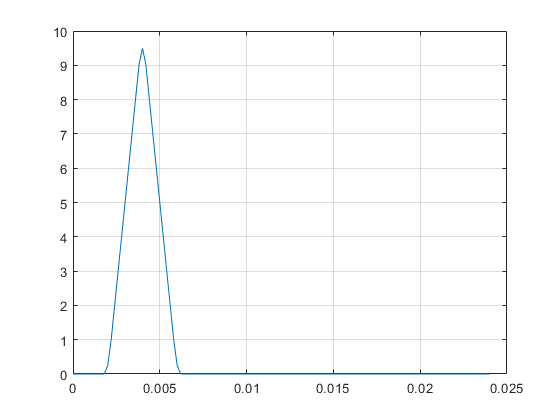
h=p(Ts-t);

t1=0:Ts/10:12\*Ts;

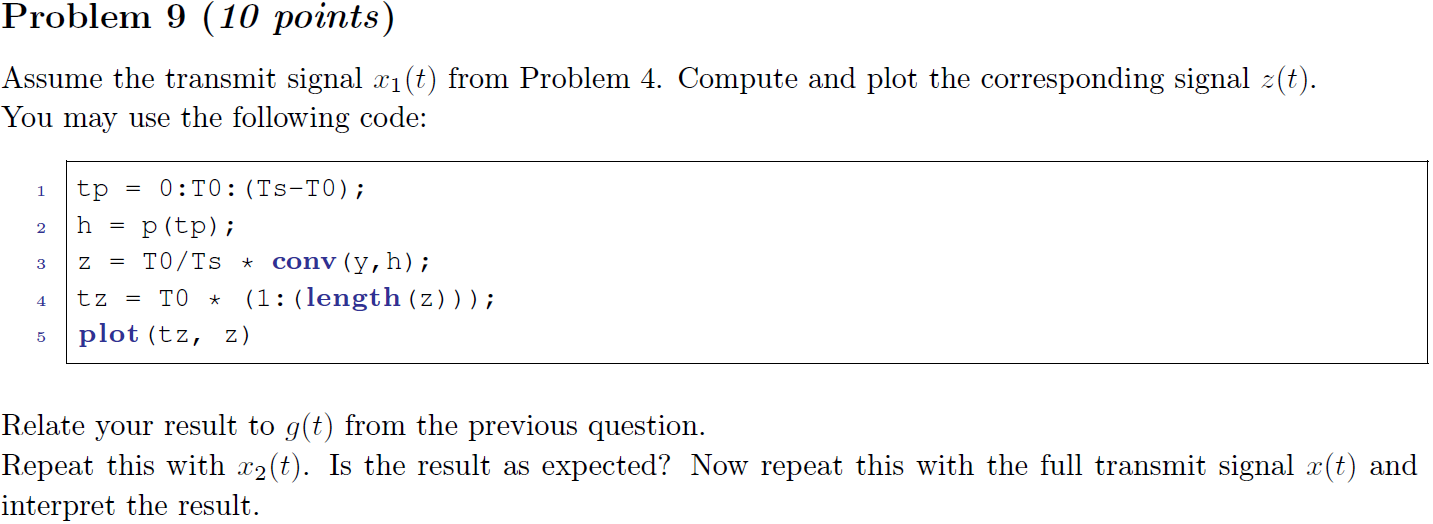
g=conv(y,h);

plot(t1,g);

grid on

****

**Figure 8-1** **signal** **g(t)= p(t - kTs)\*h(t) for k = 1**



**Solution**

Run the following Matlab script and get the result as shown in figure 9-1, 9-2 and 9-3

Ts=0.002;

t=0:Ts/10:6\*Ts;

p=@(t)rect((t-Ts/2)/Ts);

T0=Ts/10;

h=p(t);

y1=4\*p(t-1\*Ts);

z1=T0/Ts\* conv(y1,h);

tz1=T0\*(1:(length(z1)));

figure;plot(tz1,z1);

grid on;

y2=2\*p(t-1\*Ts);

z2=T0/Ts\*conv(y2,h);

tz2=T0\*(1:(length(z2)));

figure;plot(tz2,z2);

grid on;

symbols=[4,2,-2,-4];

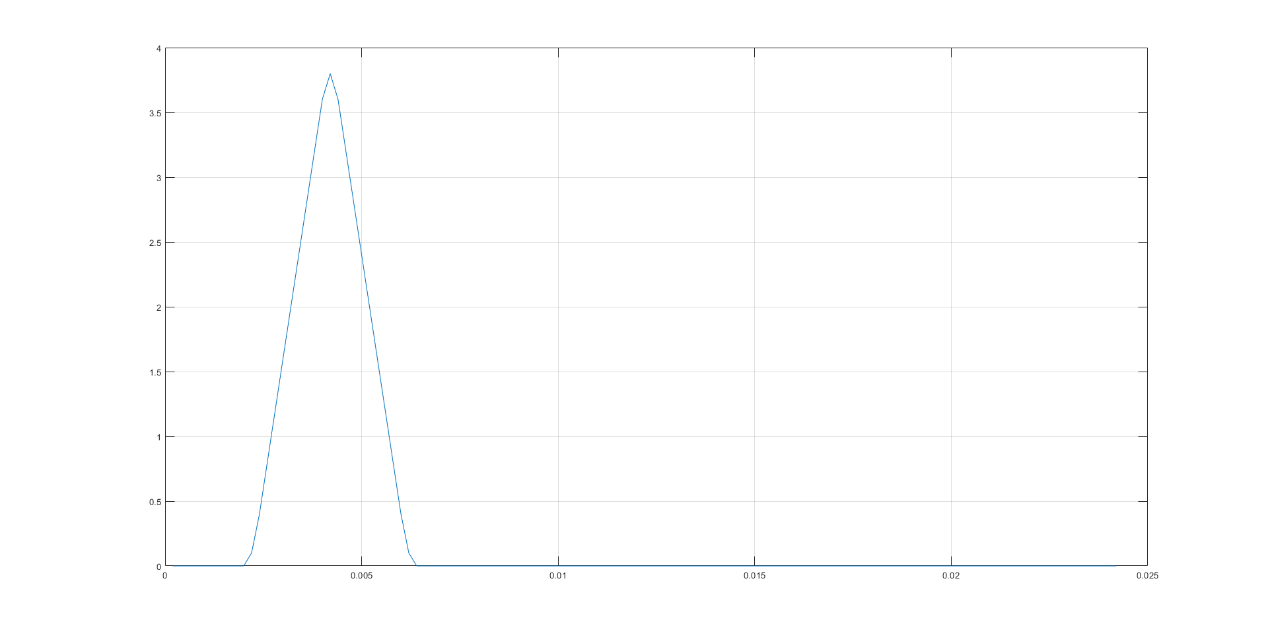
[t,y]=modulator(symbols);

z=T0/Ts\*conv(y,h);

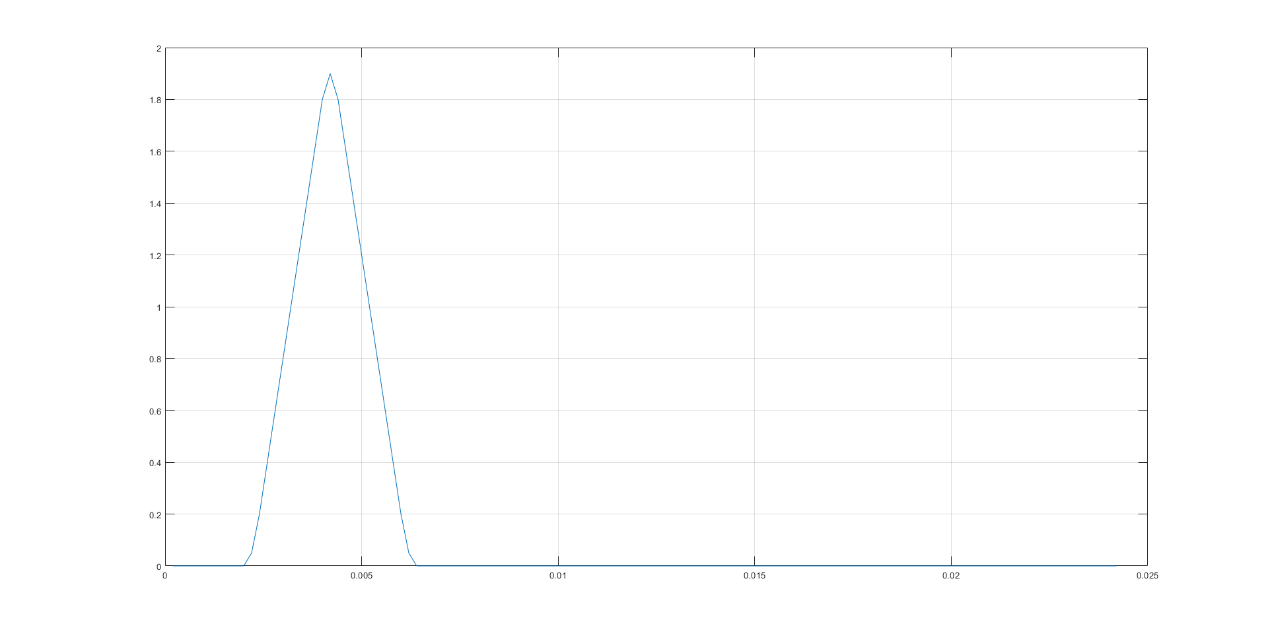
tz=T0\*(1:(length(z)));

figure;plot(tz,z);

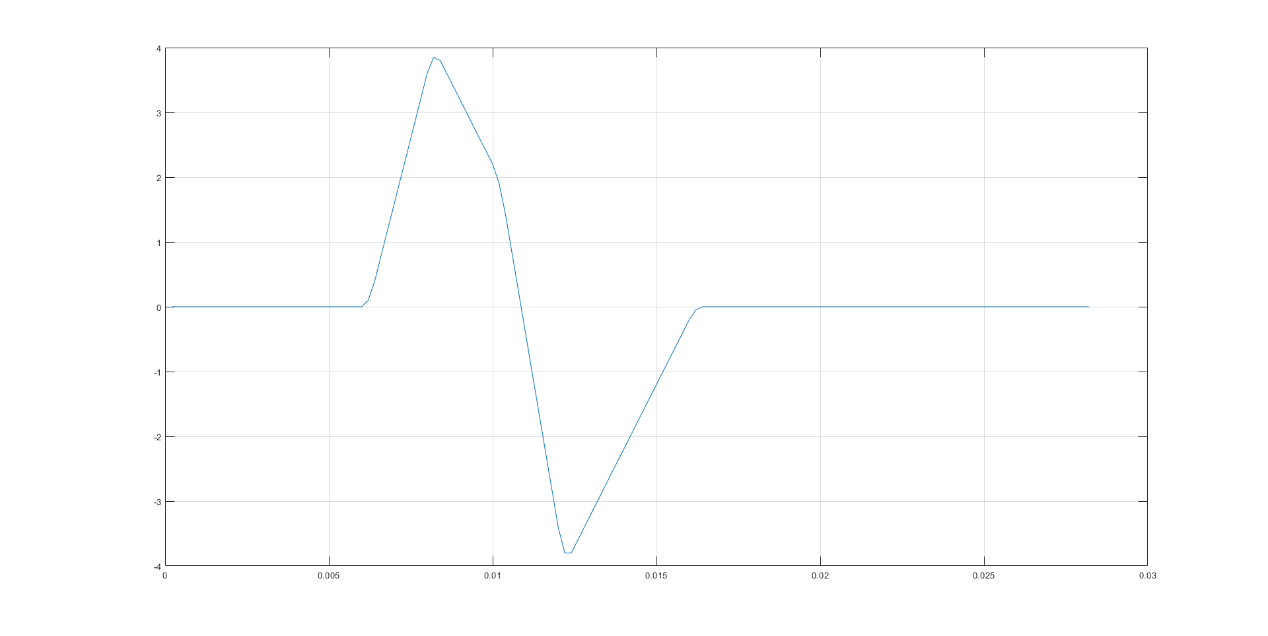
grid on;

****

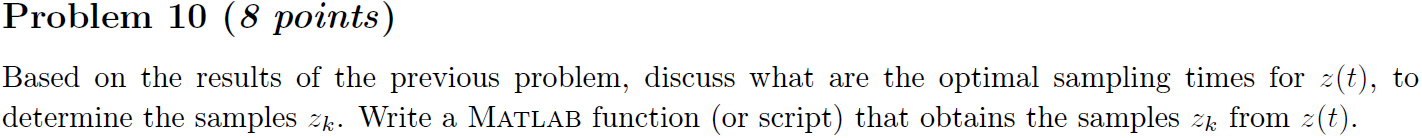
**Figure 9-1 corresponding signal z1(t)**

****

**Figure 9-2 corresponding signal z2(t)**

****

**Figure 9-3 corresponding signal z(t)**



Run the following Matlab script and get the result as shown in figure 10-1

clc,clear

symbols=[4,2,-2,-4];

[t,y]=modulator(symbols);

Ts=0.002;

T0=Ts/10;

t=0:Ts/10:6\*Ts;

p=@(t)rect((t-Ts/2)/Ts);

h=p(t);

z=T0/Ts\*conv(y,h);

tz=T0 \*(1:(length(z)));

plot(tz,z);

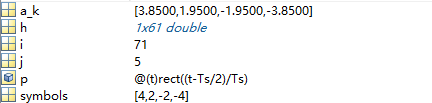
j=1;

for i=41:10:71

a\_k(j)=z(i);

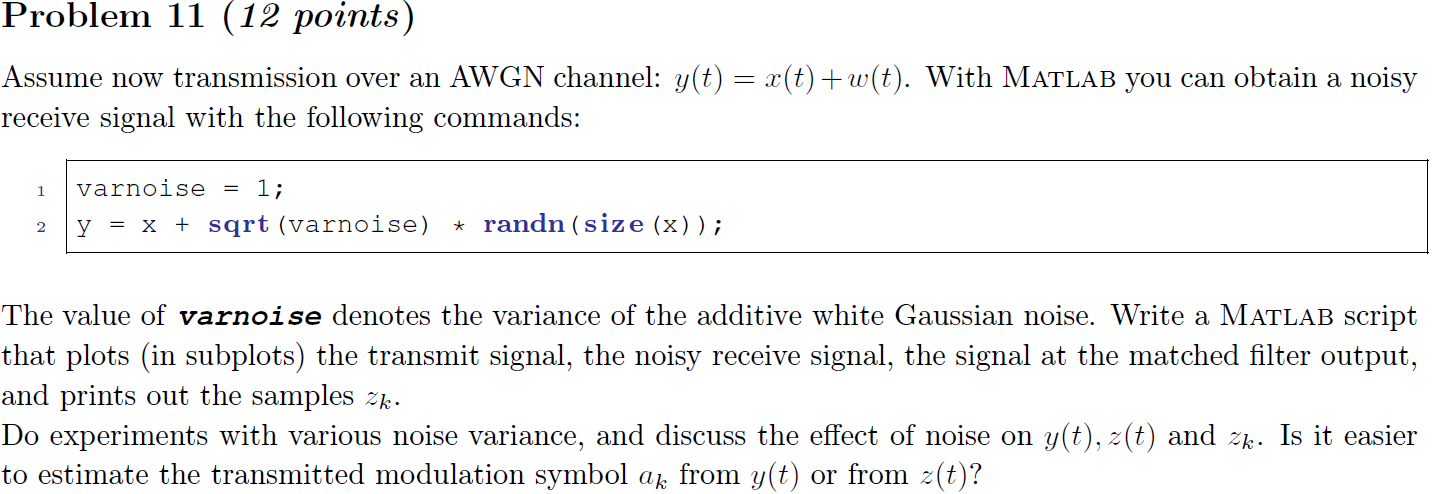
j=j+1;

end



**Figure 10-1 The result of the above script**

The value(3.85) is in the 41st, so the start point of signal sampling is 41st



Run the following Matlab script and get the result as shown in figure 11-1 to figure 11-6

clc,clear

varnoise=1;

symbols=[4,2,-2,-4];

[t,ak]=modulator(symbols);

y = ak+sqrt(varnoise)\*randn(size(ak));

plot(t,y)

Ts=0.002;

T0=Ts/10;

t=0:Ts/10:12\*Ts;

p=@(t)rect((t-Ts/2)/Ts);

h=p(Ts-t);

figure;

z=T0/Ts\* conv(y,h);

tz=T0\*(1:(length(z)));

plot(tz,z)

j=1;

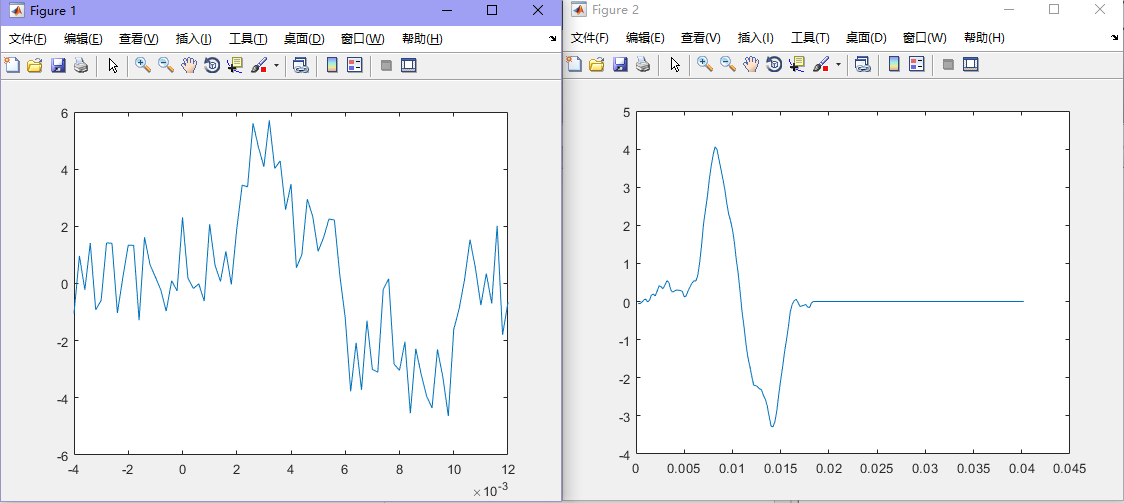
for i=41:10:71

a\_k(j)=z(i);

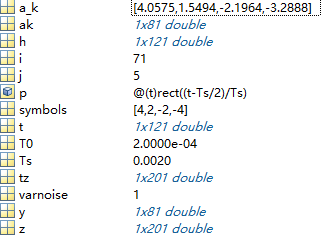
j=j+1;

end

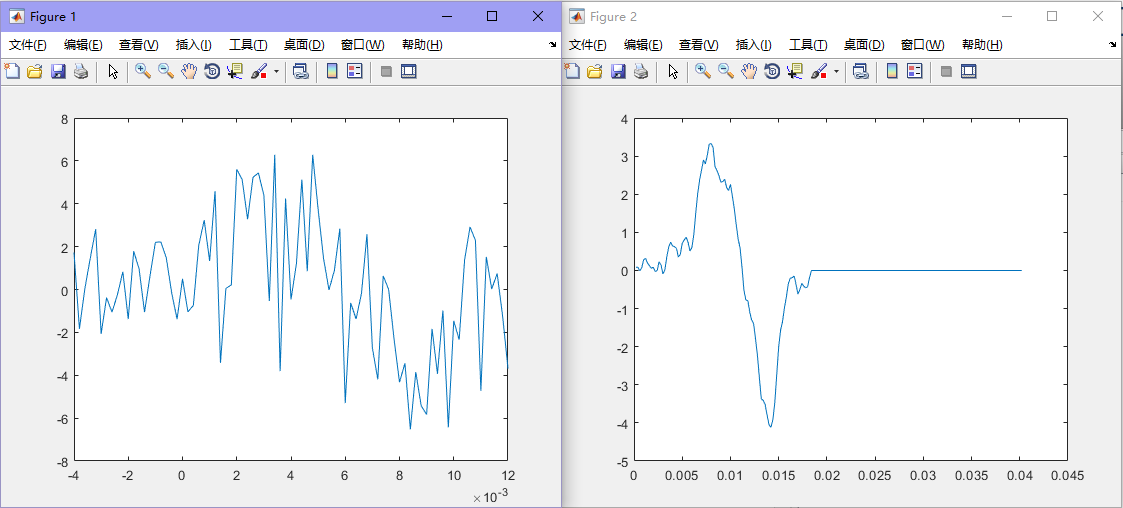
modulation\_symbol=determine\_modulation\_symbol(z\_k); %Question 12



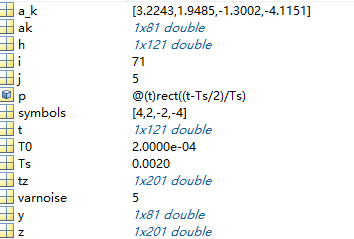
**Figure 11-1 varnoise=1**



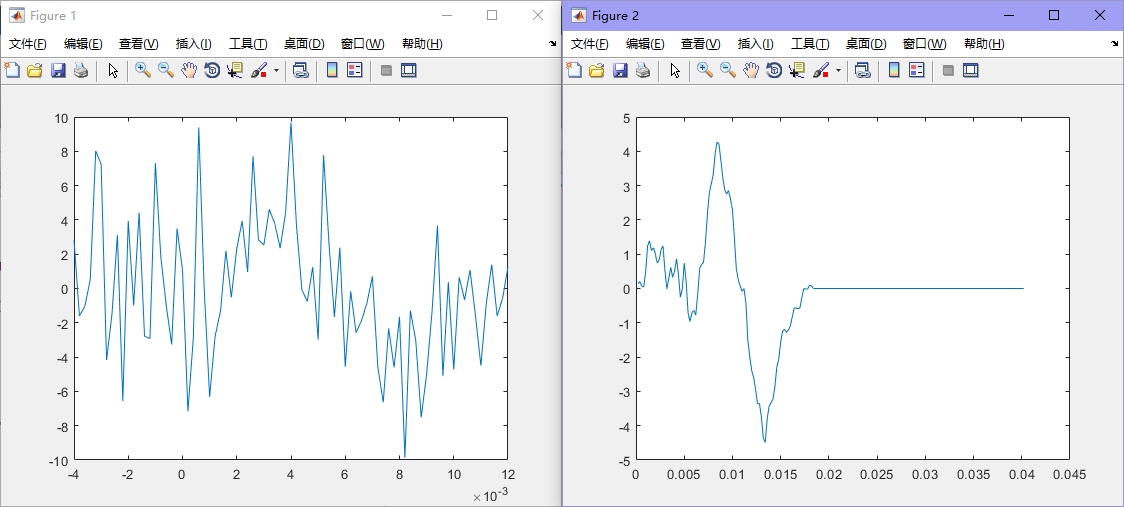
**Figure 11-2 varnoise=1**



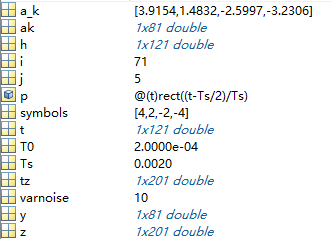
**Figure 11-3 varnoise=5**



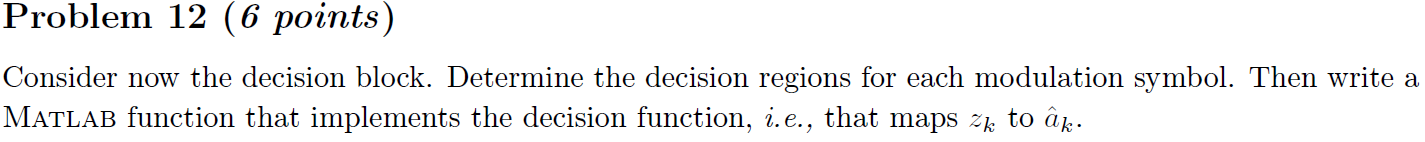
**Figure 11-4 varnoise=5**



**Figure 11-5 varnoise=10**



**Figure 11-6 varnoise=10**



**Solution**

The listing below shows a Matlab function that was used to solve the question

function modulation\_symbol=determine\_modulation\_symbol(received\_samples)

modulation\_symbol=[];

for i=1:length(received\_samples)

distance\_1=abs(4-received\_samples(i));

distance\_2=abs(2-received\_samples(i));

distance\_3=abs(-4-received\_samples(i));

distance\_4=abs(-2-received\_samples(i));

x1=[distance\_1,distance\_2,distance\_3,distance\_4];

x2=[4,2,-4,-2];

[m,n]=find(x1==min(x1));

f=x2(m,n);

modulation\_symbol=[modulation\_symbol f];

end

end



**Solution**

The listing below shows a Matlab function that was used to solve the question

function demapper = symbols\_demapping(imput)

L=length(imput);

for i=1:L

if imput(i)==4

demapper(:,i)=[0,0];

elseif imput(i)==2

demapper(:,i)=[0,1];

elseif imput(i)==-2

demapper(:,i)=[1,1];

elseif imput(i)==-4

demapper(:,i)=[1,0];

end

end

end

Run the following Matlab script and get the result as shown in figure 13-1

clc,clear

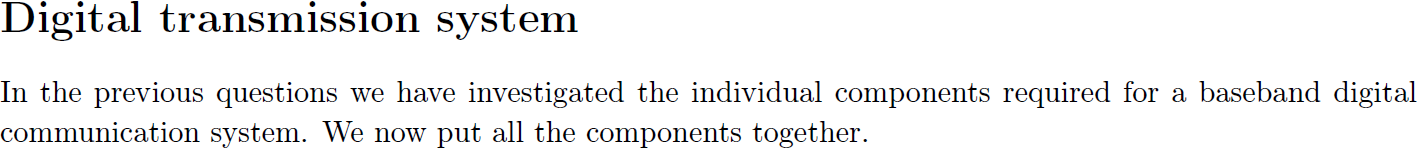
symbols=[4,2,-2,-4];

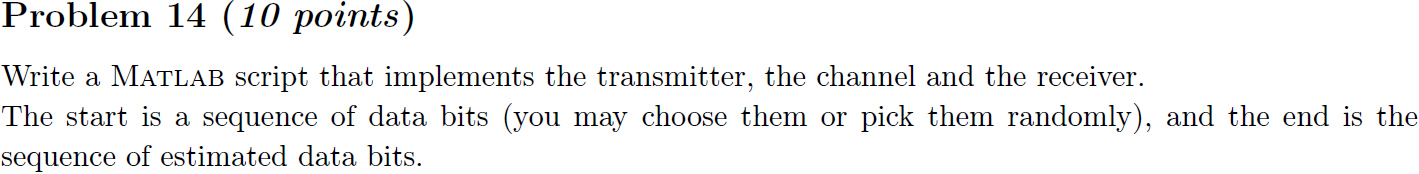
demapper = symbols\_demapping(symbols);

bits=reshape(demapper,1,[]);



**Figure 13-1 Demapper**





**Solution**

Run the following Matlab script and get the result as shown in figure 14-1 to 14-4

clc,clear

M=4;

k=log2(M);

imput=randi([0,1],8,1);

x=reshape(imput,k,[]);

y\_mapping = symbols\_mapping(x);

y\_mapping\_r=reshape(y\_mapping,[],4);

[t,ak]=modulator(y\_mapping\_r);

plot(t,ak);

figure;

Ts =0.002;

T0=Ts /10;

p=@(t)rect((t-Ts /2)/ Ts );

h=p(Ts-t);

plot(t,h);

figure;

z=T0/Ts\* conv(ak,h);

tz=T0\*(1:(length(z)));

plot(tz,z)

j=1;

for i=61:10:91

z\_k(j)=z(i);

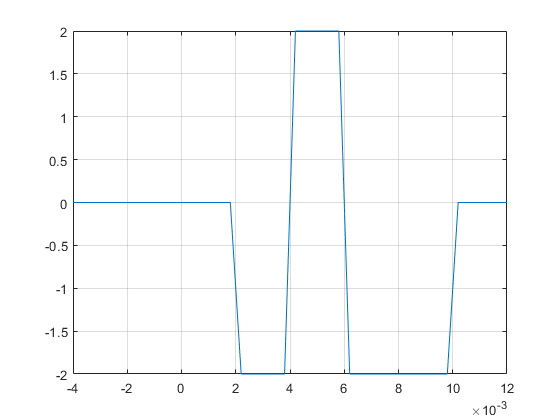
j=j+1;

end

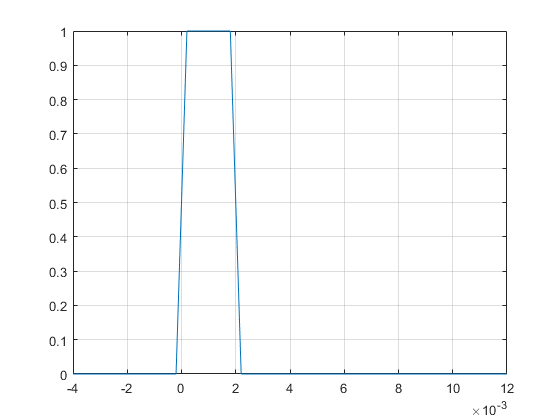
modulation\_symbol=determine\_modulation\_symbol(z\_k);

demapper = symbols\_demapping(modulation\_symbol);

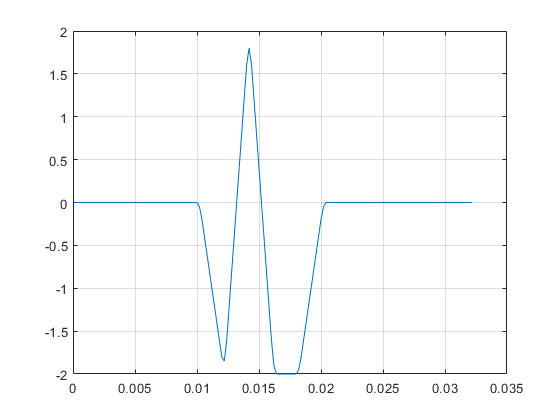
bits=reshape(demapper,1,[]);

****

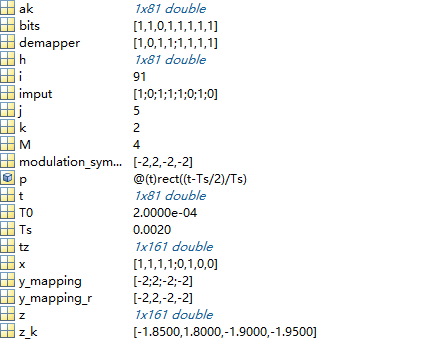
**Figure 14-1 the process of a digital transmitter**

****

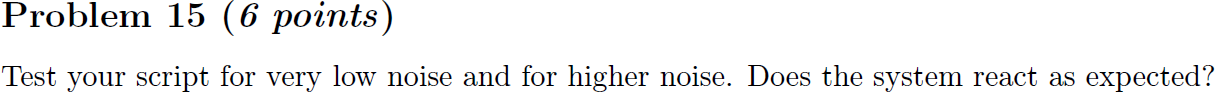
**Figure 14-2 the process of a digital transmitter**

****

**Figure 14-3 the process of a digital transmitter**



**Figure 14-4 the process of a digital transmitter**



**Solution**

clc,clear

M=4;

k=log2(M);

imput=randi([0,1],8,1);

x=reshape(imput,k,[]);

y\_mapping = symbols\_mapping(x);

y\_mapping\_r=reshape(y\_mapping,[],4);

[t,ak]=modulator(y\_mapping\_r);

plot(t,ak);

grid on

figure;

varnoise=1;

y=ak+sqrt(varnoise)\*randn(size(ak));

plot(t,y);

grid on

figure;

Ts=0.002;

T0=Ts/10;

p=@(t)rect((t-Ts/2)/Ts);

h=p(Ts-t);

plot(t,h);

grid on

figure;

z=T0/Ts\* conv(y,h);

tz=T0\*(1:(length(z)));

plot(tz,z)

grid on

j=1;

for i=61:10:91

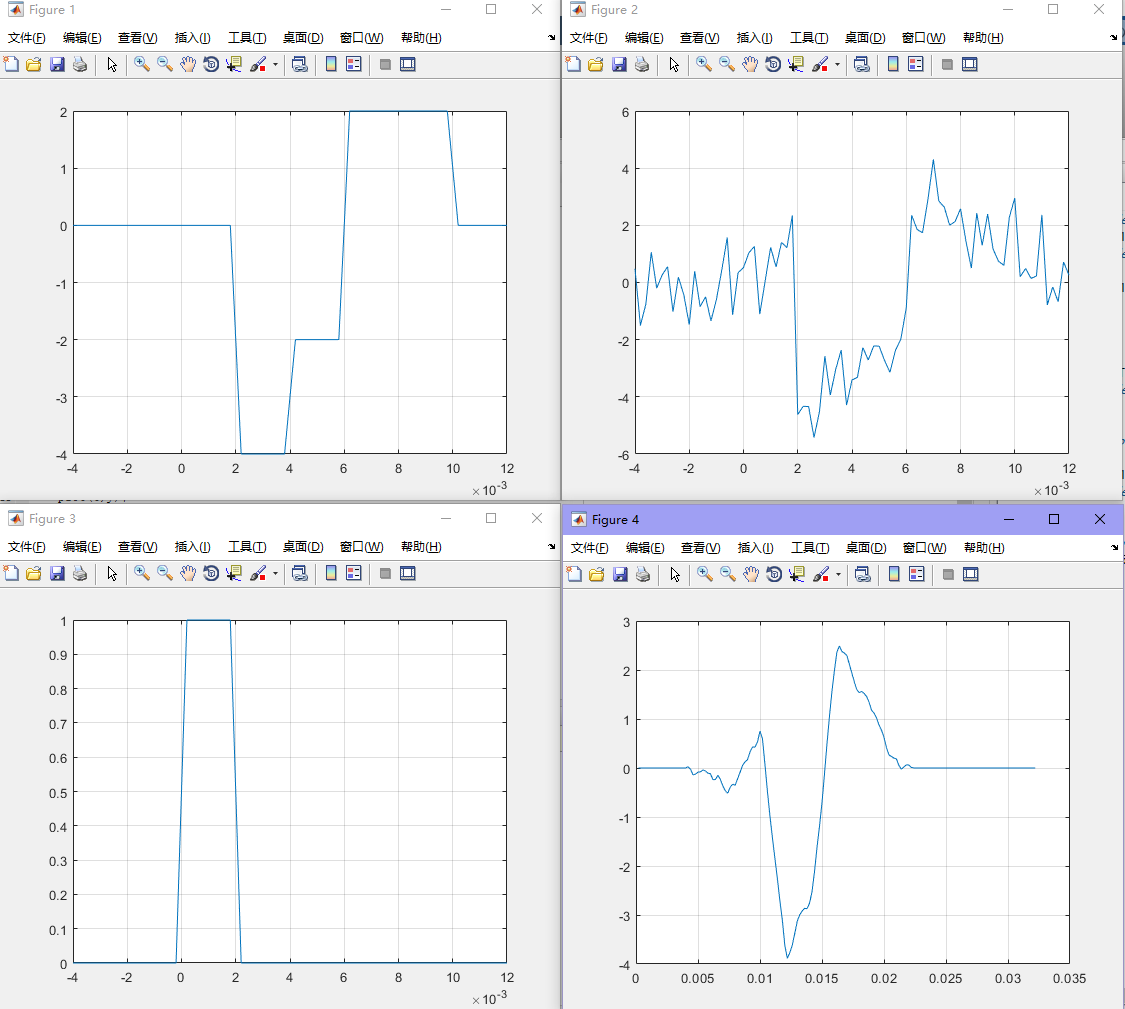
z\_k(j)=z(i);

j=j+1;

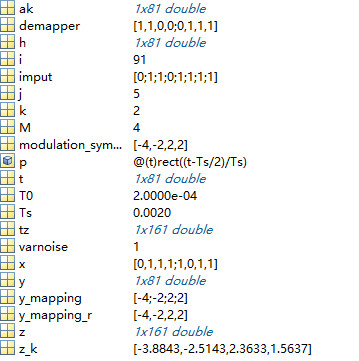
end

modulation\_symbol=determine\_modulation\_symbol(z\_k);

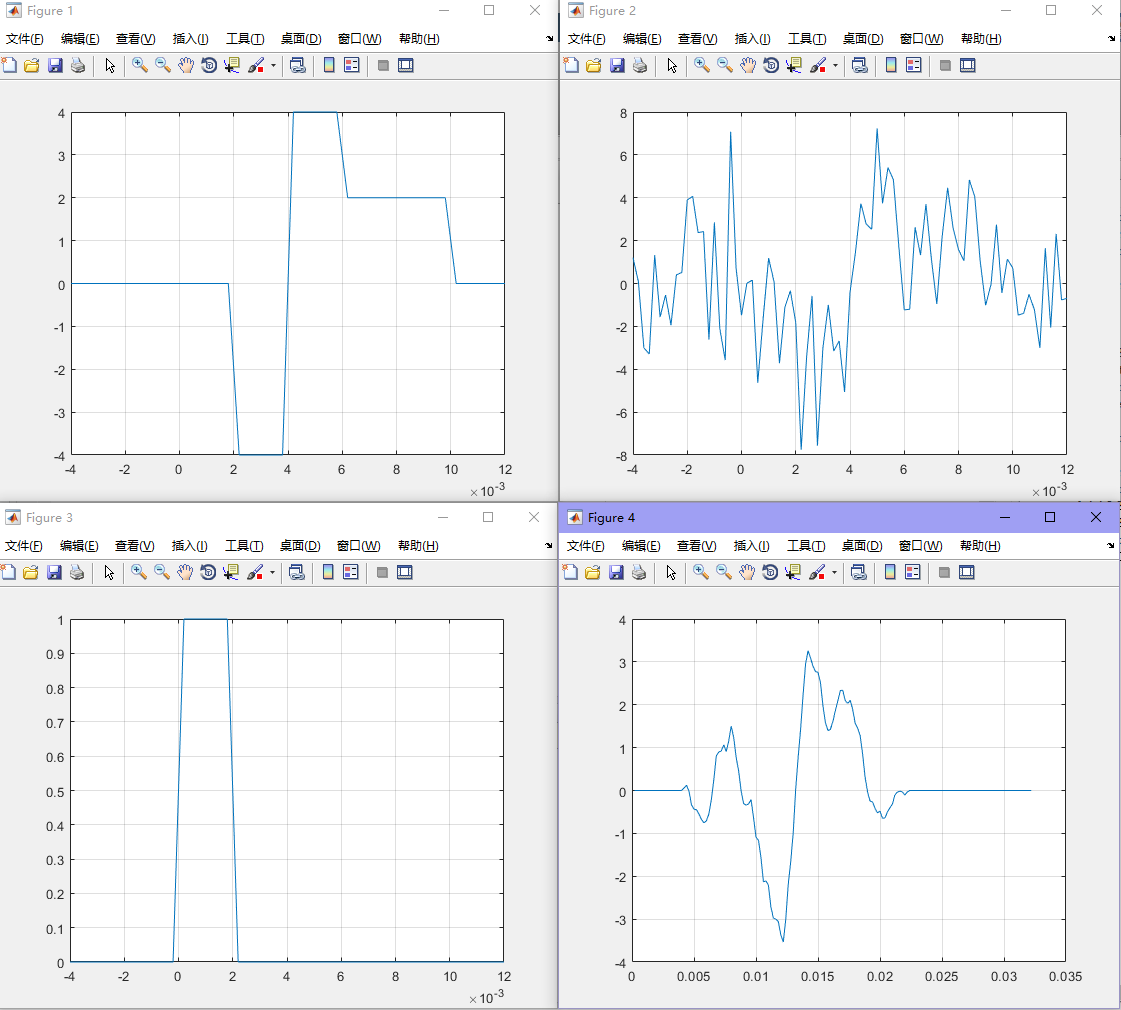
demapper = symbols\_demapping(modulation\_symbol);



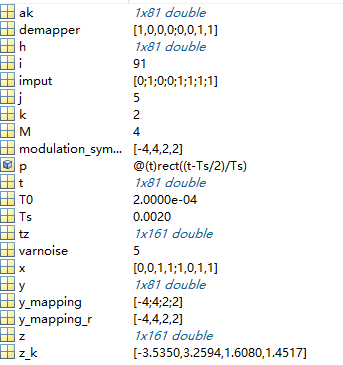
**Figure 15-1 varnoise=1**



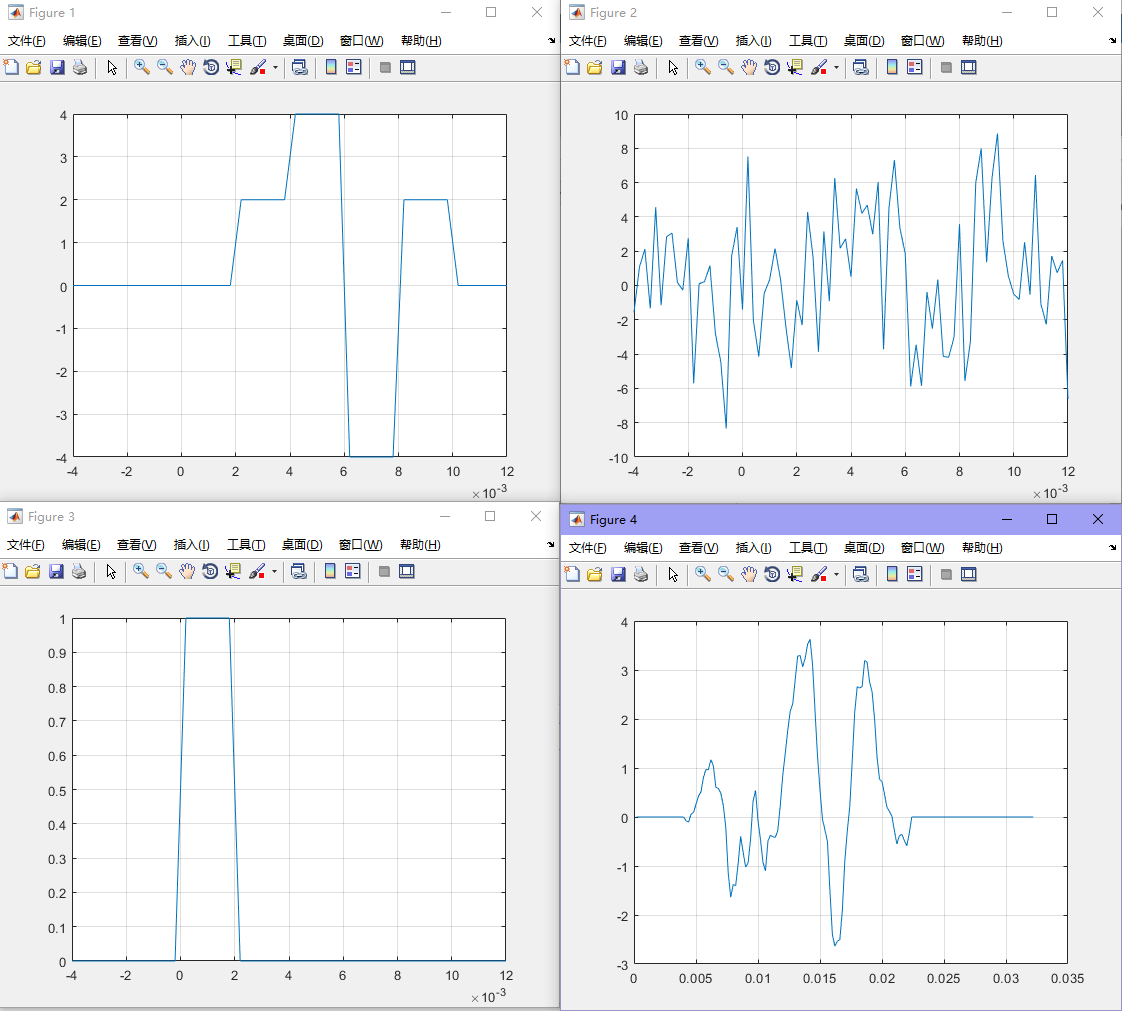
**Figure 15-2 varnoise=1**



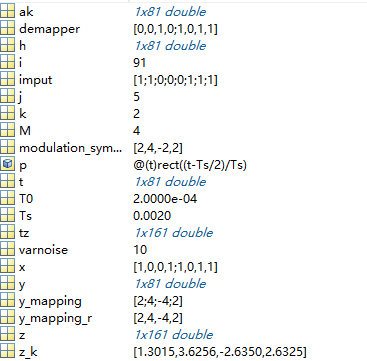
**Figure 15-3 varnoise=5**



**Figure 15-4 varnoise=5**



**Figure 15-5 varnoise=10**



**Figure 15-6 varnoise=10**