Xi'an Jiaotong-Liverpool University

西交利物浦大学

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE412 Advanced Signal Processing

Assignment I: Signals and Spectra with Matlab

Student Name Zhiyu.Xu

Student ID 1926905

Email Address Zhiyu,Xu19@student.xjtlu.edu.cn

November 2019

Contents

Problem 1 Mapping by 4ASK....... 1
1.1 Result and code....... 1
1.2 Explanation....... 1
Problem 2....... 2
2.1 Result and code....... 2
2.1 Explanation....... 2
Problem 3....... 4
3.1 Explanation....... 4
Problem 4....... 5
4.1 Result....... 5
4.2 Code....... 6
4.3 Explanation....... 6
Problem 5....... 7
5.1 Result....... 7
5.2 Code....... 7
5.3 Contradistinction....... 8
Problem 6....... 9
6.1 Impulse response of the receive filter....... 9
Problem 7 Formula Derivation....... 10
Problem 8....... 11
8.1 Expected result....... 11
8.2 Matlab result....... 11
Problem 9....... 13
9.1 z1(t)....... 13
9.1.1 Result and code....... 13
9.1.2 Explanation....... 14
9.2 z2(t)....... 14
9.2.1 Result and code....... 14
9.2.2 Explanation....... 15
9.3 z(t) by x1(t), x2(t), x3(t) and x4(t)....... 15
9.3.1 Result and code....... 15
9.3.2 Explanation....... 17
Problem 10 The optimal sampling times....... 18
Problem 11 Add GWN....... 20
11.1 Result and code....... 20
11.2 Discussion....... 21
Problem 12 Decision block....... 22
Problem 13 Demapper....... 23
Problem 14 Digital transmission system....... 24
Problem 15 Test....... 26

Problem 1 Mapping by 4ASK

1.1 Result and code

Input	[0 0]	[0 1]	[1 0]	[1 1]	[0 1 1 1]	[1 0 0 0 1 1 0 0 1 0]
Output	4	2	-4	-2	[2,-2]	[-4,4-2,4,-4]

Table 1 4ASK mapping test result

```
% Problem 1 4ASK mapping
clc;clear;close all;
y=fourASK(input('Please enter a string of binary 
numbers:'));
```

Code 1 Main code

```
function [Output] = fourASK(inputArg)
%ak = (1-2bk1)*(4-2bk2);
b=reshape(inputArg,2,[])';
Output= (1-2*b(:,1)).*(4-2*b(:,2));
end
```

Code 2 4ASK function

1.2 Explanation

According to the assignment's require, as shown in the table below:

bk1bk2	ak
00	+4
01	+2
10	-2
11	-4

Assuming that 4ASK mapping equation is:

$$a_k = (1-2b_{k1})\times(4-2b_{k2}) \cdots\cdots\cdots (1)$$

In order to avoid inputting too much data at one time, an error is caused. Use the reshape function to convert the input signal into an array of N*2 for operation.

Problem 2

2.1 Result and code

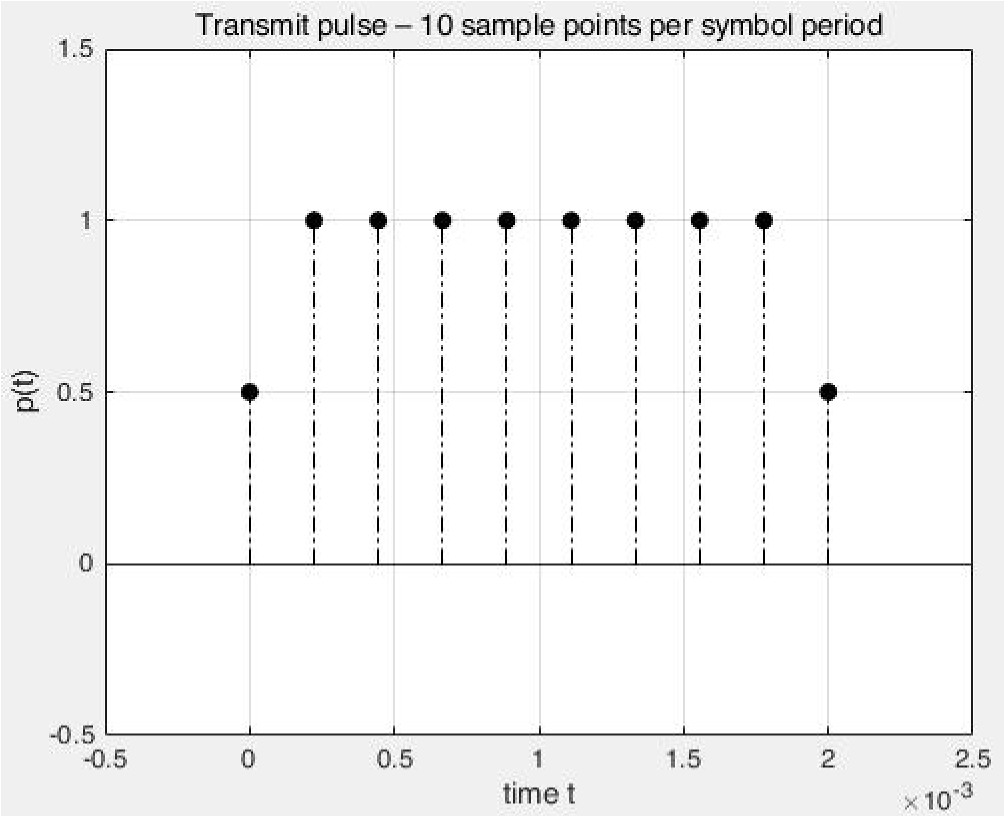

Figure 1 Transmit pulse result

```
% Problem 2 transmit pulse
clc;clear;close all;
Ts=0.002;
t=0:Ts/9:Ts;
p=rect((t-Ts/2)/Ts);
stem(t,p,'filled','k-.');
title('Transmit pulse – 10 sample points per symbol
period');
axis([-0.0005 0.0025 -0.5 1.5])
xlabel('time t');
ylabel('p(t)');grid on;
```

Code 3 Main code

2.1 Explanation

Known pulse signal is:

$$p(t) = rect(\frac{t - Ts/2}{Ts}) \quad \cdots\cdots\cdots\cdots (2)$$

The above signal is derived from rect(), as shown in the following figure:

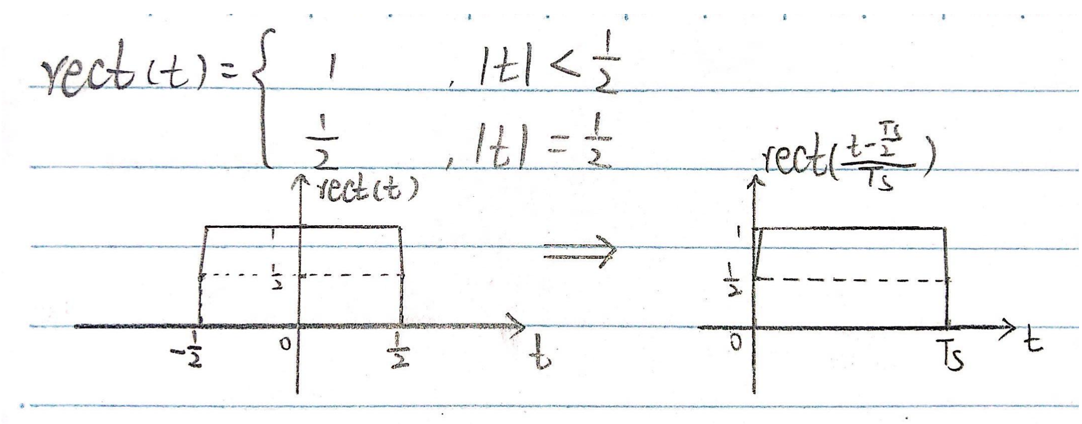

Figure 2 Transmit pulse by hand

According to requirement of the Problem2, Ts = 2msec, and we need to select 10 sample points. So set the point spacing equal to Ts/9, than we get the result as Figure1.

Problem 3

3.1 Explanation

According to the requirement, the formula for the image is as follows:

$$x(t) = \sum_{k=0}^{K} a_k \cdot p(t - kT_s) \quad \cdots\cdots\cdots\cdots \quad (3)$$

It can be deduced from formula (3) that a_k determines the amplitude of the image, and $p(t - kT_s)$ determines the unit in which the image is shifted to the right by kT_s on the basis of the original image. And we choose the a_k is [4, 2, -2, -4].

The hand drawing is as follows：

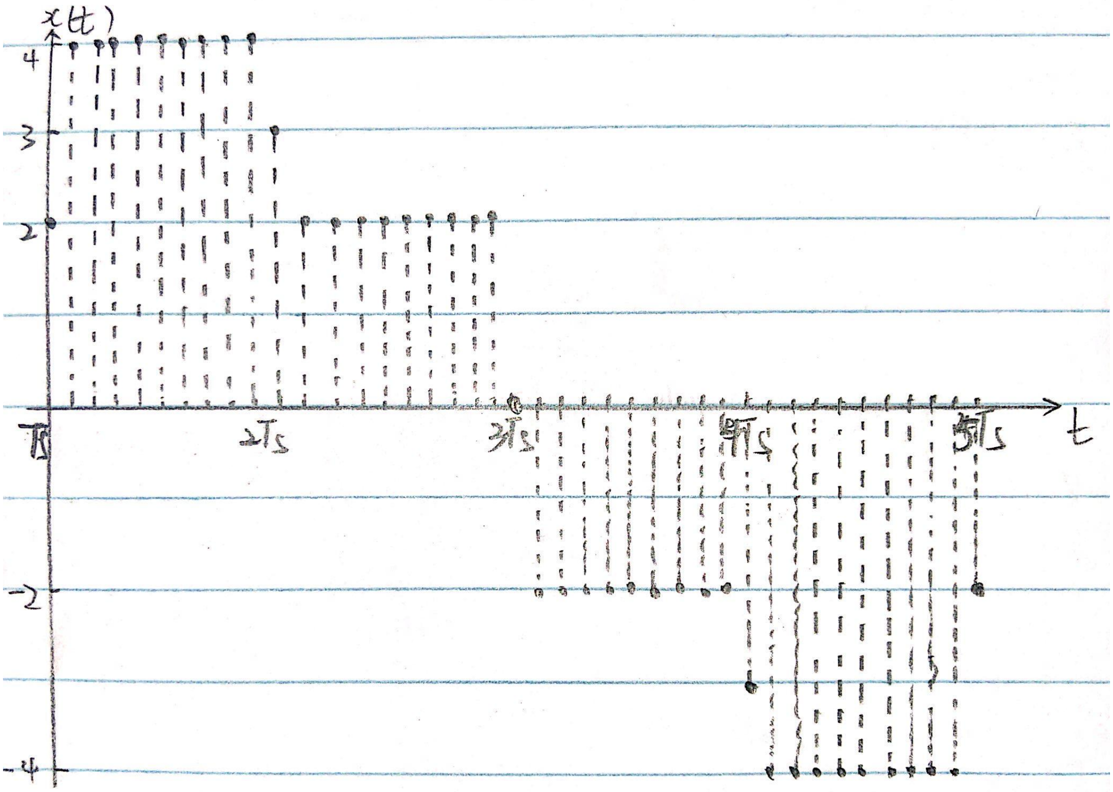

Figure 3 x(t) by hand drawing

Problem 4

4.1 Result

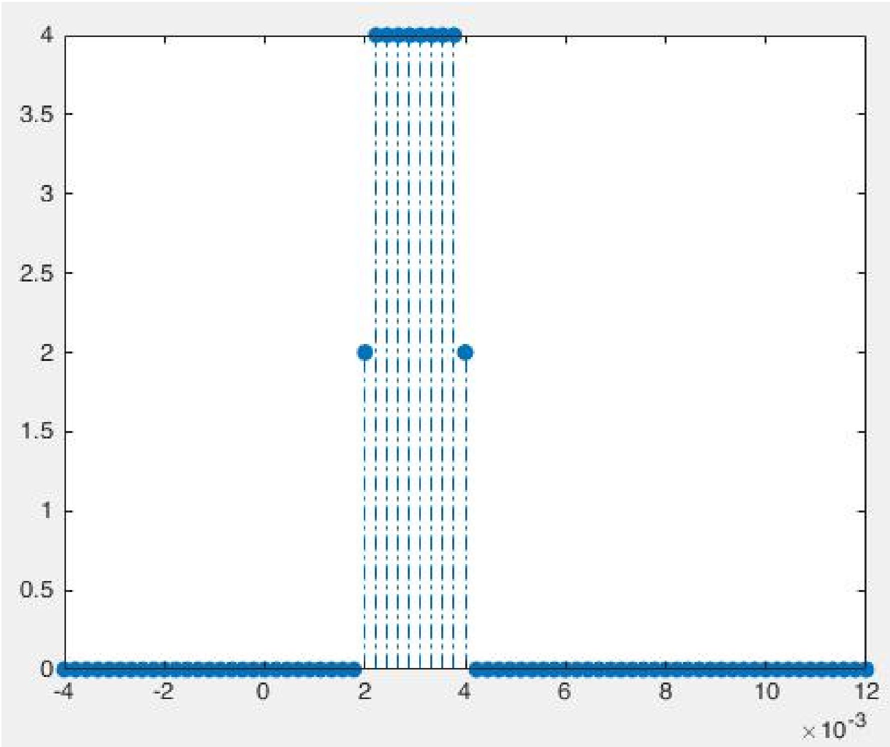

Figure 4 Result of x1(t)

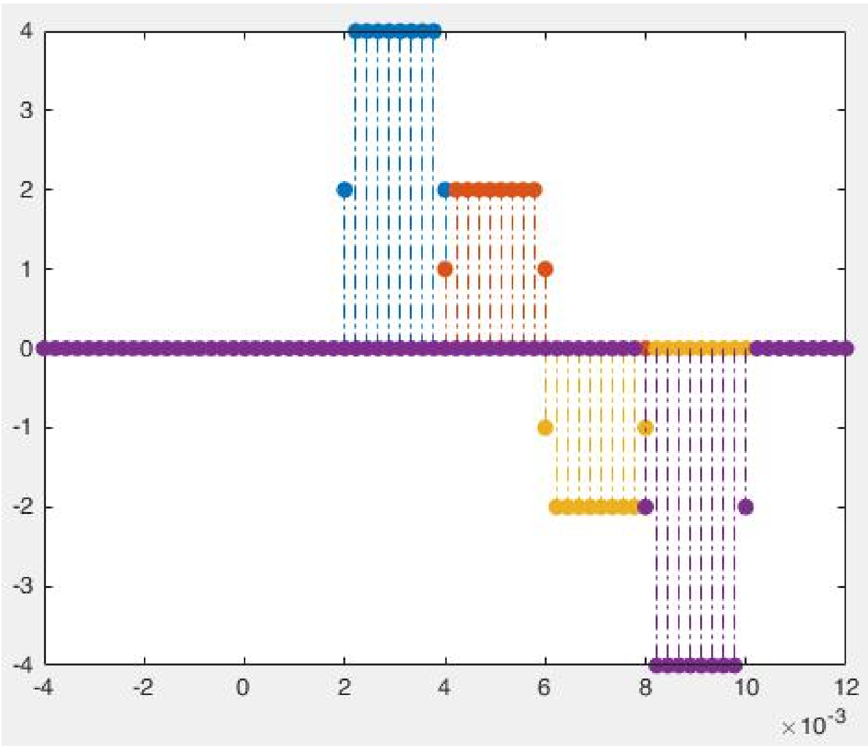

Figure 5 Result of x1(t), x2(t), x3(t) and x4(t)

4.2 Code

```
%Problem4
Ts = 0.002;
T0 = Ts/9;
t = -2*Ts:T0:6*Ts;
a = [1 2 3 4];
p = @(t) rect((t-Ts/2)/Ts);
x1 = a(1)*p(t-1*Ts);
x2 = a(2)*p(t-2*Ts);
x3 = a(3)*p(t-3*Ts);
x4 = a(4)*p(t-4*Ts);
stem(t,x1,'filled','-.');hold on;
stem(t,x2,'filled','-.');hold on;
stem(t,x3,'filled','-.');hold on;
stem(t,x4,'filled','-.');hold on;
```

Code 4 Code of plotting signal x1(t), x2(t), x3(t) and x4(t)

4.3 Explanation

Because the a_k is [4, 2, -2, -4], so in Figure 3 we can see four pulse signals with amplitudes of 4, 2, -2 and -4 each of which has a width of Ts, and the beginning and end of each two signals are connected.

Problem 5

5.1 Result

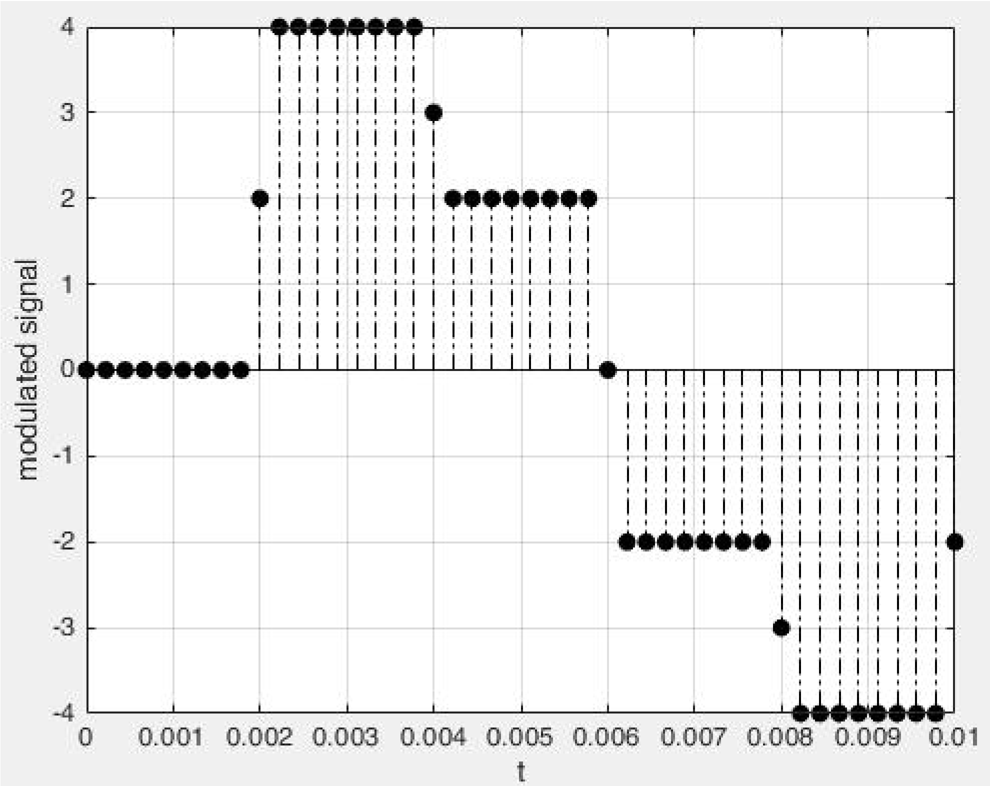

Figure 6 Result of modulator

5.2 Code

```
%Problem 5
clc;clear;close all;
a=[4; 2; -2; -4];
Ts=0.002;
To=Ts/9;
t=0:To:Ts*length(a);
xt=modu(a,Ts,t);
stem(t,xt,'fill','k-.');grid on;
xlabel('t');ylabel('modulated signal');
```

Code 5 Code of plotting signal x(t)

```
function [y] = modu(a,Ts,t)
%functiong of modulator
n = 1:length(a);
p = @(t,n)rect((t-n*Ts-Ts/2)/Ts);
y = a'*bsxfun(p,t,n');
end
```

Code 6 Function of modulator

5.3 Contradistinction

Comparing Figure 6 with Figure 5, we can see that the two images are basically the same(this is because that two cases choose the same a_k value), except that in Figure 5, four independent signals are superimposed, there are two values at the intersection, and Figure 6 adds the intersections. And the intersection points are t=0.002, t=0.004 and t=0.006.

Comparing Figure 6 with Figure 3, they are basically the same graphics.

Problem 6

6.1 Impulse response of the receive filter

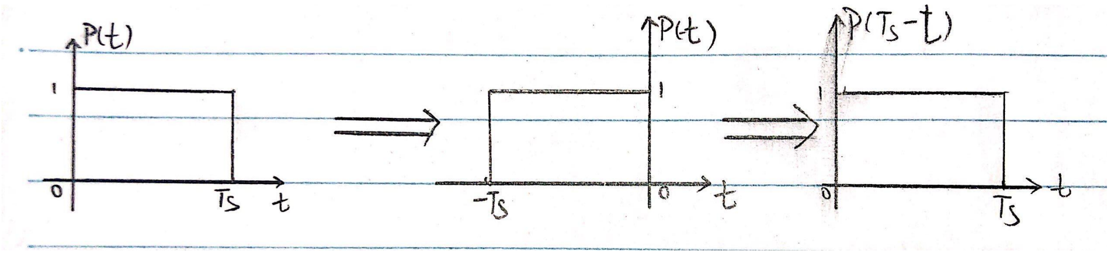

Figure 7 Impulse response by hand drawing

As known $h(t) = p(T_s - t)$, so first we should reverse p(t) to p(-t), than shift the image of p(-t) horizontally to the right along the x-axis by T_s units, finally we got the signal h(t) which we need.

Problem 7 Formula Derivation

In this case, assume noise-free transmission, the received signal is $y(t) = x(t)$, and the output signal of the receive filter is $\mathrm{z}(\mathrm{t}) = \mathrm{y}(\mathrm{t}) * \mathrm{h}(\mathrm{t})$. In Problem 3, we know the formula(3) which is the x(t) expression. Then we can get the following derivation process:

$$\begin{aligned} \mathrm{z}(t) &= y(t) * h(t) \\ &= x(t) * h(t) \\ &= \left[\sum_{k=1}^{K} a_k \cdot p(t - kTs)\right] * h(t) \end{aligned}$$

Assume K equal 4, then:

$$\begin{aligned} \mathrm{z}(\mathrm{t}) &= \left[\sum_{k=1}^{4} a_k \cdot p(t - kT_s)\right] * h(t) \\ &= \left[a_1 \cdot p(t - T_s) + a2 \cdot p(t - 2T_s) + a_3 \cdot p(t - 3T_s) + a_4 \cdot p(t - 4T_s)\right] * h(t) \\ &= a_1 \cdot p(t - T_s) * h(t) + a2 \cdot p(t - 2T_s) * h(t) + a_3 \cdot p(t - 3T_s) + a_4 \cdot p(t - 4T_s) * h(t) \\ &= \sum_{k=1}^{4} a_k \cdot p(t - kT_s) * h(t) \end{aligned}$$

So when K takes any real number, the following formula is true:

$$\mathrm{z}(\mathrm{t}) = \sum_{k=1}^{K} a_k \cdot \left[p(t - kT_s) * h(t)\right]$$

And the property what I used here is distributive law of convolution.

Problem 8

8.1 Expected result

In this Problem, we need to compute and plot the signal $\mathrm{g}(\mathrm{t}) = p(t-kT_s) * h(t)$ for k = 1.

So the output signal is $\mathrm{g}(\mathrm{t}) = p(t-T_s) * p(T_s - t)$, and assume $\mathrm{t} \in [0, \mathrm{T_s}]$.

$$
\begin{aligned}
\mathrm{g}(\mathrm{t}) &= \sum_{k=0}^{T_s=10T_0} p(k) * p(t-k) \\
&= p(0) * p(t) + p(T_0) * p(t-T_0) + \ldots + p(T_s) * p(t-1) \\
&= \frac{1}{2} p(t) + p(t-T_0) + \ldots + p(t-9T_0) + \frac{1}{2} p(t-10T_0) \\
&= \frac{1}{2}\left[\frac{1}{2}\delta(t) + \delta(t-T_0) + \ldots + \delta(t-9T_0) + \frac{1}{2}\delta(t\text{ - }10T_0)\right] + \\
&\left[\frac{1}{2}\delta(t) + \delta(t-T_0) + \ldots + \delta(t-9T_0) + \frac{1}{2}\delta(t\text{ - }10T_0)\right] + \ldots \\
&+ \frac{1}{2}\left[\frac{1}{2}\delta(t) + \delta(t-T_0) + \ldots + \delta(t-9T_0) + \frac{1}{2}\delta(t\text{ - }10T_0)\right] \\
&= \frac{1}{4}\delta(t) + \delta(t-T_0) + 2\delta(t-2T_0) + \cdots + 9\delta(t-9T_0) + 9.5\delta(t-10T_0) + 9\delta(t-11T_0) + \ldots \\
&+ 2\delta(t-18T_0) + \delta(t-19T_0) + \frac{1}{4}\delta(t-20T_0)
\end{aligned}
$$

As shown above, it is the derivation process. The time interval is selected as T0 = Ts / 10. The final image should be 21 pulse signals.

8.2 Matlab result

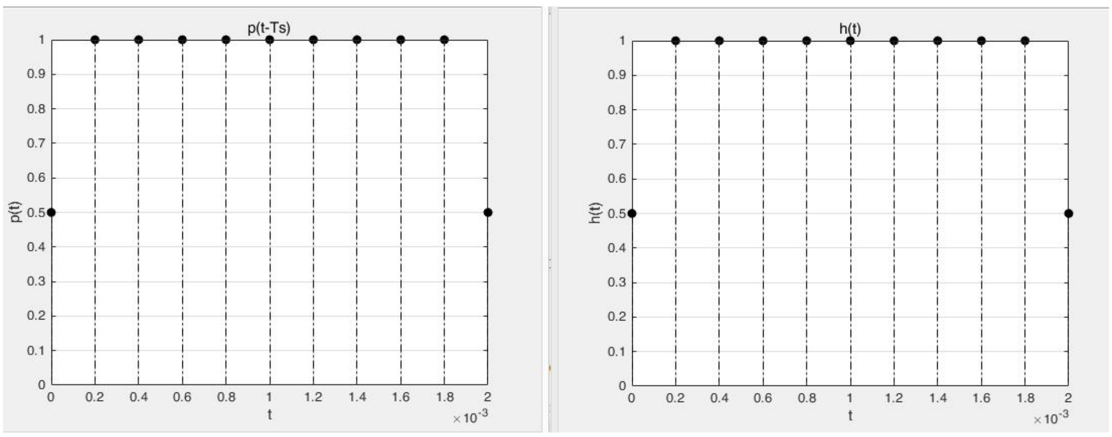

Figure 8 Result of p(t-Ts) and h(t)

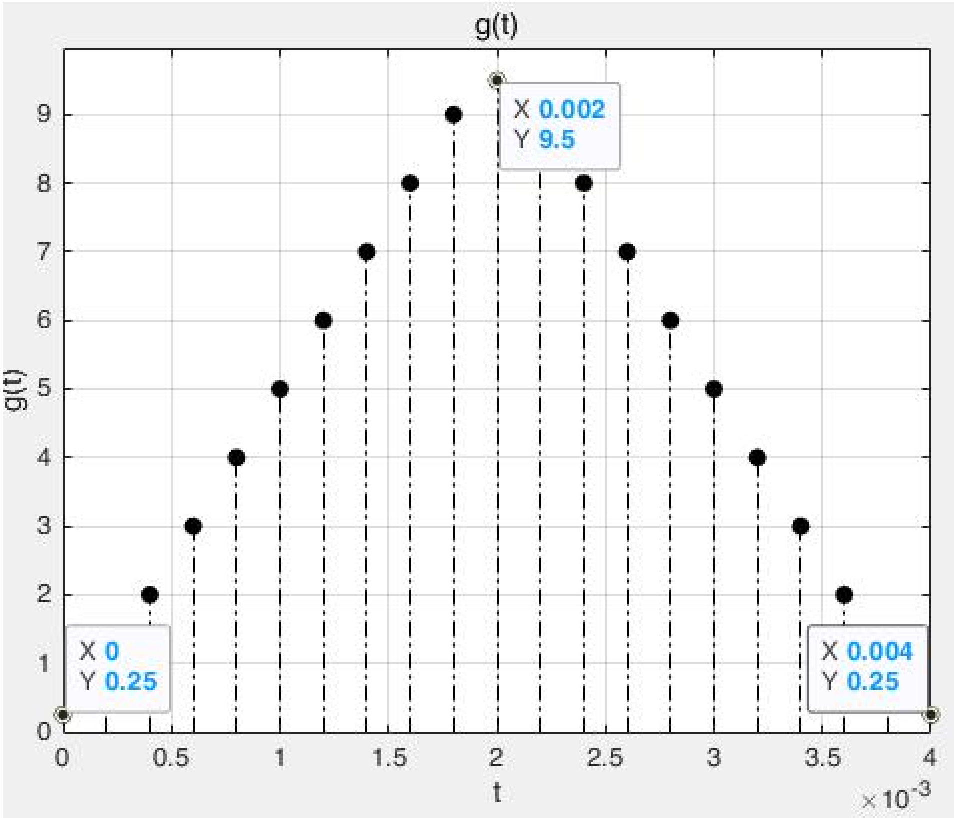

Figure 9 Result of g(t)

```
%Problem 8
clc;clear;close all;
Ts=0.002;
t=0:Ts/10:Ts;
P=@(t)rect((t-Ts/2)/Ts);
x=P(t);
h=P(Ts-t);
y=conv(x,h);
ty=0:Ts/10:2*Ts;
figure(1)
stem(t,x,'fill','k-.');title('p(t-Ts)');xlabel('t');yl
abel('p(t)');grid on;
figure(2)
stem(t,h,'fill','k-.');title('h(t)');xlabel('t');ylabe
l('h(t)');grid on;
figure(3)
stem(ty,y,'fill','k-.');title('g(t)');xlabel('t');ylab
el('g(t)');grid on;
```

Code 7 Main code of Problem 8

Comparing the results derived in Figure 9 and 9.1, we can find that the results of matlab are consistent with the expected results.

Problem 9

9.1 z1(t)

9.1.1 Result and code

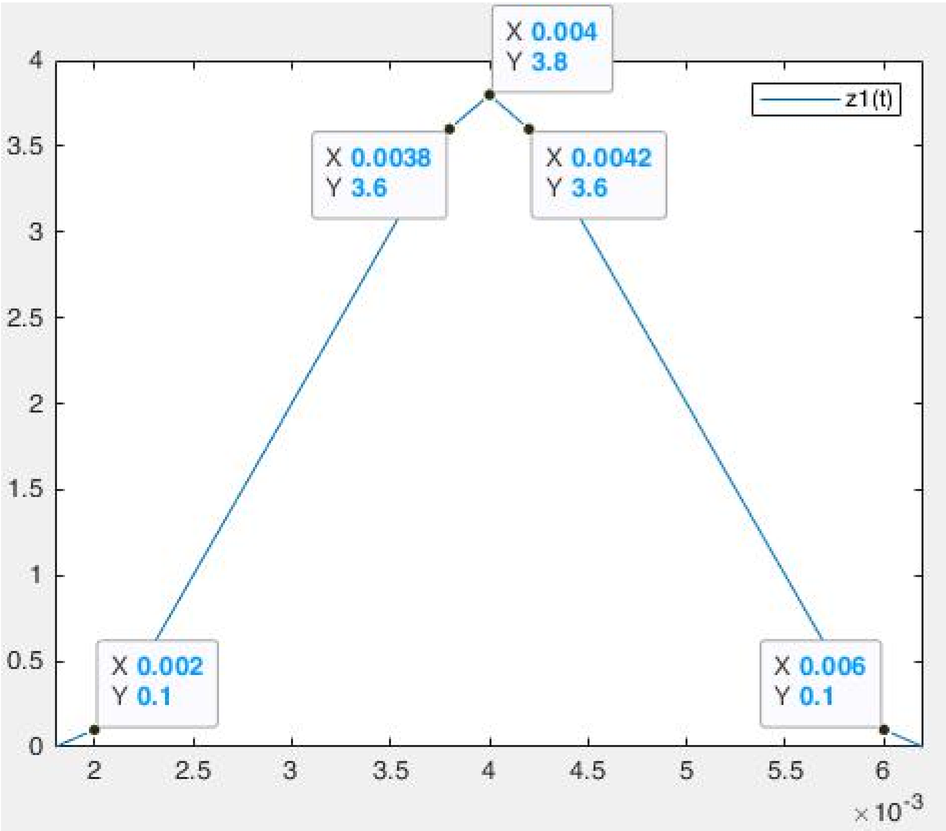

Figure 10 Result of z1(t)

```
clc;clear;close all;
Ts = 0.002;
T0 = Ts/10;
t = 0:T0:2*Ts;
a = [4 2 -2 -4];
p = @(t) rect((t-Ts/2)/Ts);
y1 = a(1).*p(t-1*Ts);
th = 0:T0:Ts;
h = p(Ts-th);
z1 = T0/Ts * conv(y1,h);
tz1 = T0 * (0:(length(z1)-1));
plot(tz1, z1,'DisplayName','z1(t)');hold on;
axis([0.0018 0.0062 0 4])
legend
```

Code 8 Plot code of z1(t)

9.1.2 Explanation

$$z_1(t) = \frac{T_0}{T_s} \cdot a_1 \cdot [p(t - T_s) * p(T_s - t)] \cdots\cdots\cdots\cdots (3)$$

The formula of z1 (t) is shown above, and compare Figure 10 and Figure 9, we can find that the image of the two signals is basically the same, but the amplitude is different. This is because the coefficient of z1 (t) is $\frac{T_0}{T_s} \cdot a_1 = \frac{0.0002}{0.00002} \times 4$.

9.2 z2(t)

9.2.1 Result and code

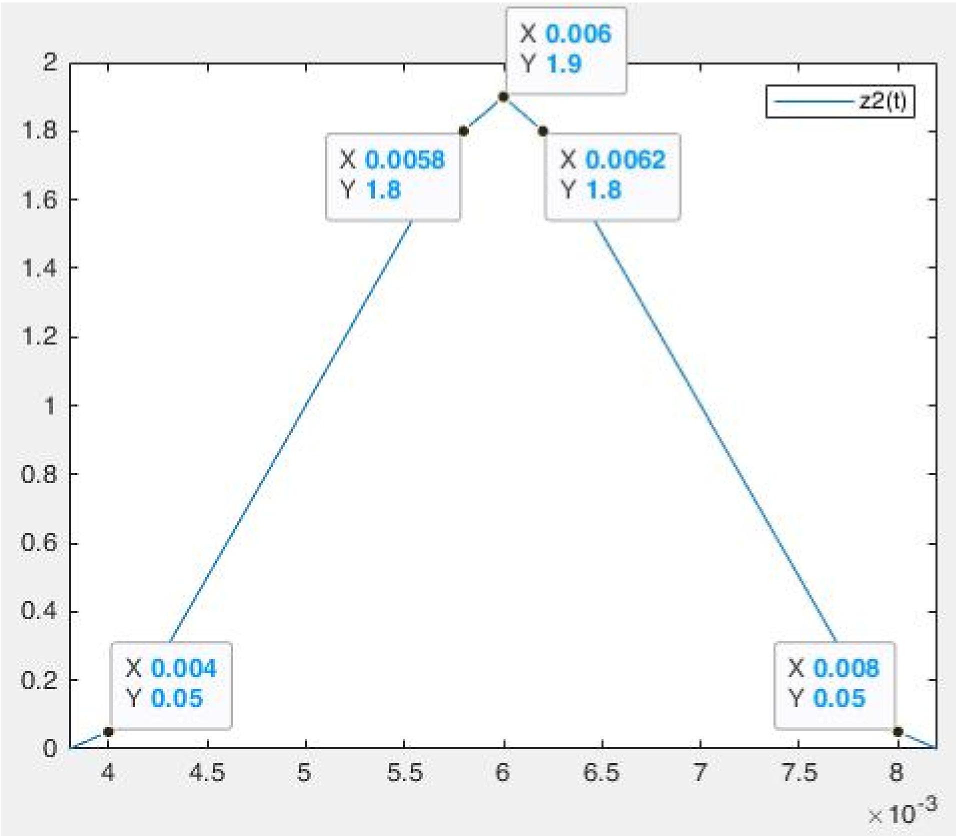

Figure 11 Result of z2(t)

```
clc;clear;close all;
Ts = 0.002;
T0 = Ts/10;
t = 0:T0:4*Ts;
a = [4 2 -2 -4];
p = @(t) rect((t-Ts/2)/Ts);
y2 = a(2).*p(t-1*Ts);
th = 0:T0:Ts;
h = p(Ts-th);
z2 = T0/Ts * conv(y2,h);
tz2 = T0 * (0:(length(z2)-1));
```

```
plot(tz2, z2,'DisplayName','z2(t)');hold on;
axis([0.0038 0.0082 0 2])
legend
```

Code 9 Plot code of z2(t)

9.2.2 Explanation

$$z_1(t)=\frac{T_0}{T_s}\cdot a_2\cdot[p(t-2T_s)*p(T_s-t)] \cdots\cdots\cdots \quad (4)$$

The formula of z1 (t) is shown above, and compare Figure 11 with Figure 9 and Figure 10, we can find that the image of the two signals is basically the same, but the amplitude is different. This is because the coefficient of z1 (t) is $\frac{T_0}{T_s}\cdot a_2=\frac{0.0002}{0.00002}\times 2$.

This result is expected.

9.3 z(t) by x1(t), x2(t), x3(t) and x4(t)

9.3.1 Result and code

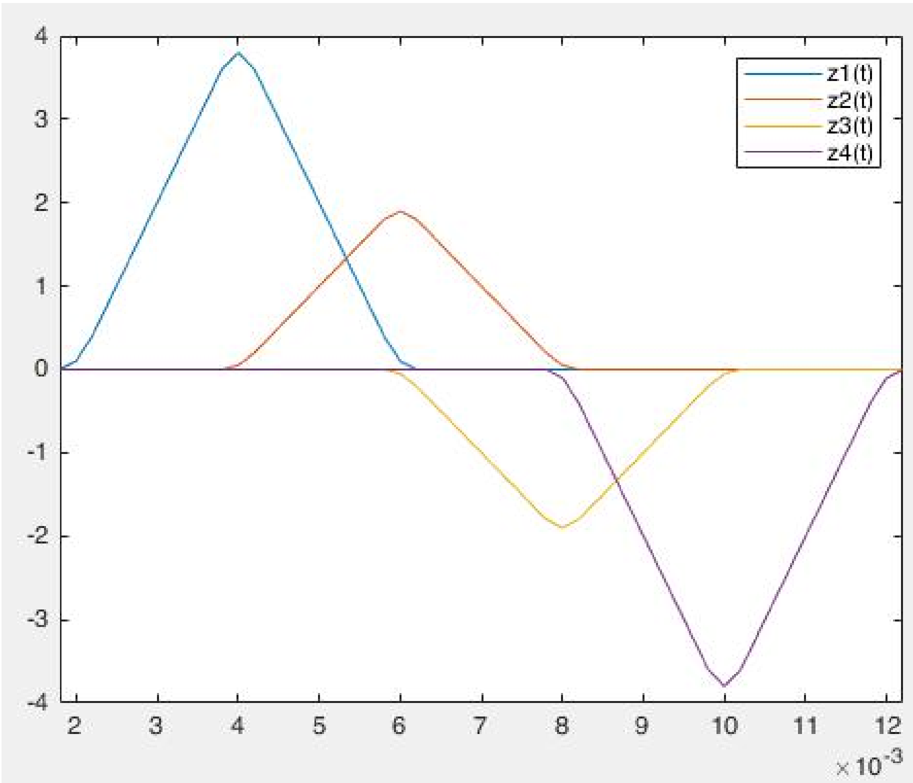

Figure 12 Result of z(t)

```
clc;clear;close all;
Ts = 0.002;
T0 = Ts/10;
t = 0:T0:6*Ts;
a = [4 2 -2 -4];
p = @(t) rect((t-Ts/2)/Ts);
```

```
y1 = a(1).*p(t-1*Ts);
y2 = a(2).*p(t-2*Ts);
y3 = a(3).*p(t-3*Ts);
y4 = a(4).*p(t-4*Ts);
th = 0:T0:Ts;
h = p(Ts-th);
z1 = T0/Ts * conv(y1,h);
z2 = T0/Ts * conv(y2,h);
z3 = T0/Ts * conv(y3,h);
z4 = T0/Ts * conv(y4,h);
tz1 = T0 * (0:(length(z1)-1));
plot(tz1, z1,'DisplayName','z1(t)');hold on;
plot(tz1, z2,'DisplayName','z2(t)');hold on;
plot(tz1, z3,'DisplayName','z3(t)');hold on;
plot(tz1, z4,'DisplayName','z4(t)');hold off;
axis([0.0018 0.0122 -4 4]);legend
```

Code 10 Plot code of z(t)

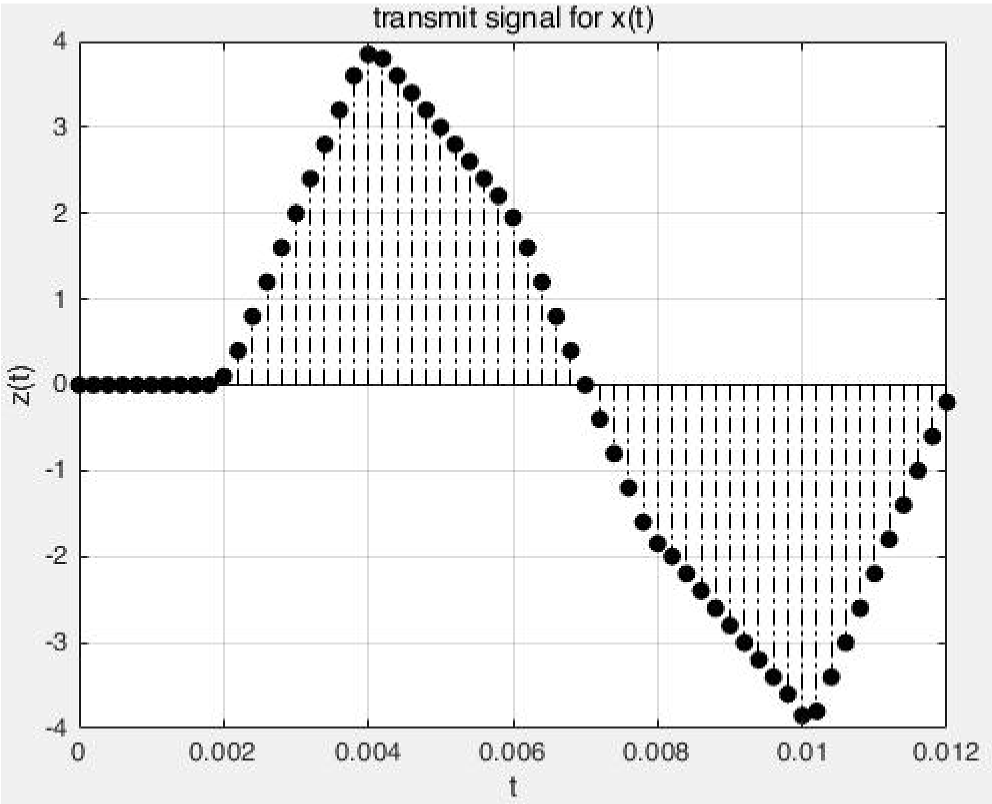

Figure 13 Result of transmit signal

```
clc;clear;close all;
%% 4ASK mapping
b=[0 0 0 1 1 1 1 0];%input
a=fourASK(b);
%% modulator
Ts=0.002;T0=Ts/10;
t=0:T0:Ts*(length(a)+1);
xt=modu(a,Ts,t);
%% after receive filter
```

```
p = @(t) rect((t-Ts/2)/Ts);
y = xt;
tp = 0:T0:Ts;
h = p(Ts-tp);
z = T0/Ts * conv(y,h);
tz = T0*(0:length(z)-1);
stem(tz,z,'fill','k-.'); grid on;
title('transmit signal for x(t)');
xlabel('t');ylabel('z(t)');
```

Code 11 Code of transmit signal

9.3.2 Explanation

The result of Figure 12 is plotting the signal z1(t), z2(t), z3(t) and z4(t) respectively, the shape of the image is based on the model discussed in Problem 9, except that the four signals have different amplitudes and phase differences (the phase differences are all Ts).

And the result of Figure 13 is plotting the signal z(t), which is the cumulative sum of z1(t), z2(t), z3(t) and z4(t).

Problem 10 The optimal sampling times

Comparing Figure 13 and Figure 12, we can find that the z(t) image changes at every two times when x(t) overlaps, which are Ts, 2Ts, 3Ts and 4Ts respectively, and the remaining time periods are uniformly increasing or decreasing. This can be used to describe the approximate image of the function through these 4 points, but include start and end points.

So that, should set the optimal sampling times to:

$$t = T_s, 2T_s, 3T_s, \cdots, (N+2)T_s$$

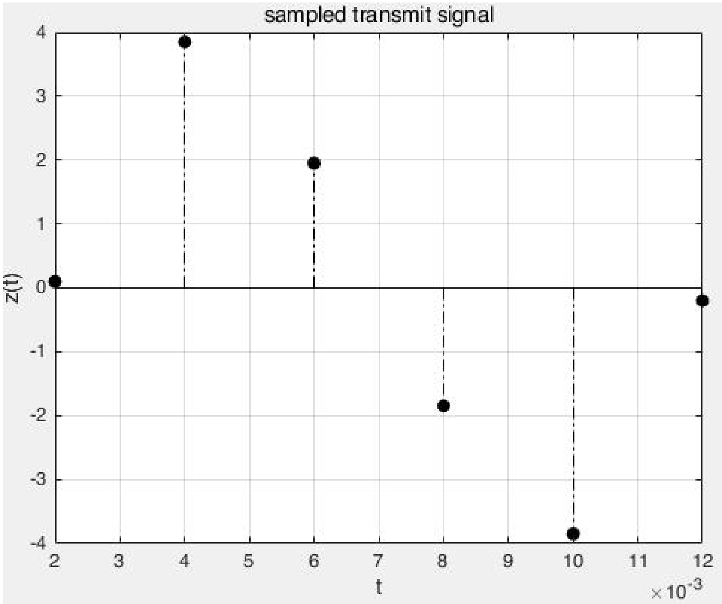

Figure 14 Sampled transmit signal

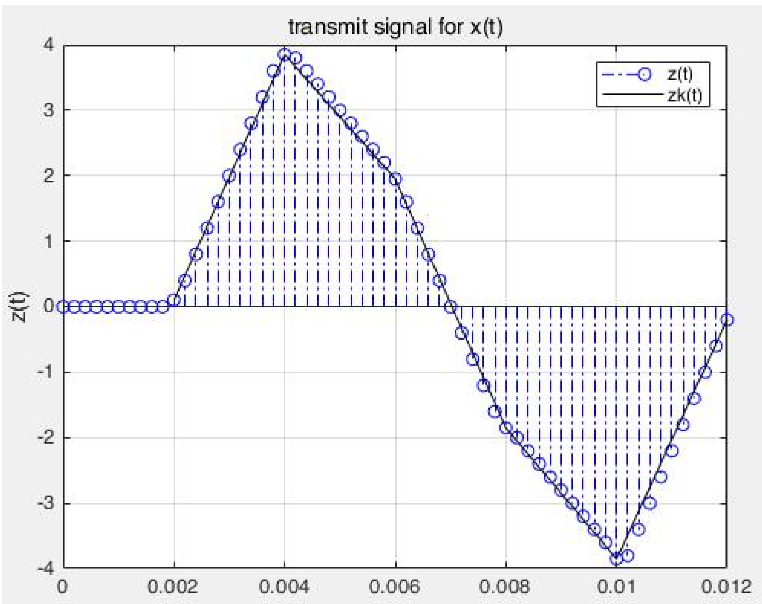

Figure 15 z(t) and zk(t)

```
%Problem 10
clc;clear;close all;
%% 4ASK mapping
b=[0 0 0 1 1 1 1 0];%input
a=fourASK(b);
%% modulator
Ts=0.002;T0=Ts/10;
t=0:T0:Ts*(length(a)+1);
xt=modu(a,Ts,t);
%% after receive filter
p = @(t) rect((t-Ts/2)/Ts);
y = xt;
tp = 0:T0:Ts;
h = p(Ts-tp);
z = T0/Ts * conv(y,h);
tz = T0*(0:length(z)-1);
figure(2)
stem(tz,z,'b-.','DisplayName','z(t)'); grid on; hold on;
title('transmit signal for x(t)');
xlabel('t');ylabel('z(t)');
%% sample
ts = Ts:Ts:(length(a)+2)*Ts;
zs = z(Ts/T0+1:Ts/T0:end);
plot(ts,zs,'k-','DisplayName','zk(t)');grid on;
legend;
figure(1)
stem(ts,zs,'fill','k-.'); grid on;
title('sampled transmit signal');
xlabel('t');ylabel('z(t)');
```

Code 12 Main code of Problem10

Figure 14 shows the sampled pulse signal. There are six points in total, including four inflection points, starting point and ending point. In Figure 15, the sampled image and the original image are compared, and the two basically overlap, indicating that the sampling effect is good

Problem 11 Add GWN

11.1 Result and code

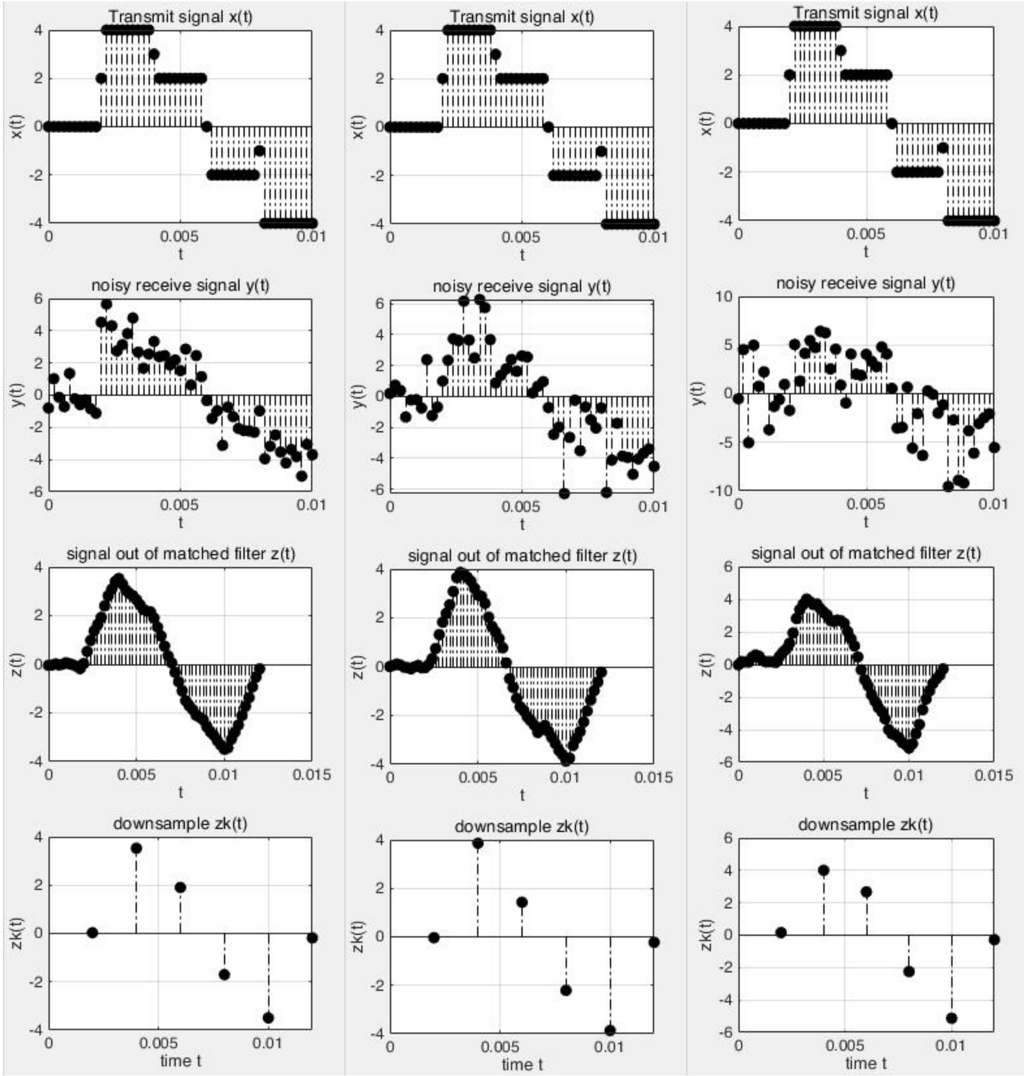

Figure 16 varinoise=1 Figure 17 varinoise=2 Figure 18 varinoise=5

```
%Prblem 11 Add GWN
clc;clear;close all;
%% 4ASK mapping
b=[0 0 0 1 1 1 1 0];%input
a=fourASK(b);
%% modulator
Ts=0.002;T0=Ts/10;
t=0:T0:Ts*(length(a)+1);
x=modu(a,Ts,t);
%% add GWN
varinoise=1;
y = x+sqrt(varinoise)*randn(size(x));
```

```
%% after receive filter
p = @(t) rect((t-Ts/2)/Ts);
tp = 0:T0:Ts;
h = p(Ts-tp);
z = T0/Ts * conv(y,h);
tz = T0*(0:length(z)-1);
%% sample
ts = Ts:Ts:(length(a)+2)*Ts;
zs = z(Ts/T0+1:Ts/T0:end);
%% plot
subplot(4,1,1);
stem(t,x,'fill','k-.'); grid on;
title('Transmit signal
x(t)');xlabel('t');ylabel('x(t)');
subplot(4,1,2);
stem(t,y,'fill','k-.'); grid on;
title('noisy receive signal
y(t)');xlabel('t');ylabel('y(t)');
subplot(4,1,3);
stem(tz,z,'fill','k-.'); grid on;
title('signal out of matched filter
z(t)');xlabel('t');ylabel('z(t)');
subplot(4,1,4);
stem(ts,zs,'fill','k-.'); grid on;
title('downsample zk(t)'); xlabel('time
t');ylabel('zk(t)');
```

Code 13 Main code of Problem 11

11.2 Discussion

Comparing the three graphs in Figure 16, Figure 17, and Figure 18, we can find that as the variance of the Gaussian white noise increases, y(t), z(t), and zk(t) all change to different degrees. Among them, y(t) has the largest change and the difference between the figure and the original signal is larger and larger, z(t) has a change in the middle and is close to the original figure, zk(t) has the smallest change and the figure has basically no change.

It is easier to estimate the transmitted modulation symbol ak from z(t). This is because $z(t)=y(t)*h(t)$, both y (t) and h (t) contain the same noise signal. After calculation by convolution, the noise has little effect on the final signal. Therefore, in Figure 16 and Figure 17 and Figure 18, although As the variance of Gaussian white noise increases, it can also be seen that the approximate curve of the graph does not change much. In the y (t) signal, the noise has a very serious effect on the graph, and the estimation bias will also be very large. and so it is easier to estimate the transmitted modulation symbol ak from z(t).

Problem 12 Decision block

```
function [z] = DB(zk)
%Decision block;
z = [1;1;1;1]*zk;
z_standard = [4;2;-2;-4]*ones(1,length(zk));
z_distance = abs(z -z_standard);
[~,N] = min(z_distance);
z = z_standard(N);
end
```

Code 14 Function of decision block

In this problem, we should determine the decision regions for each modulation symbol, the formula is shown in Code14. The specific principle is to use each zk to subtract the standard ak(4, 2, -2, -4), and find the smallest absolute value. The standard ak at the corresponding position is required ak.

For example, assume that zk is [3.5 1.59 -5 -1.654 2.356 3.569], the processing of the function is as follows:

Name	Numbers					
zk	3.5	1.59	-5	-1.654	2.356	3.569
z	3.5	1.59	-5	-1.654	2.356	3.569
	3.5	1.59	-5	-1.654	2.356	3.569
	3.5	1.59	-5	-1.654	2.356	3.569
	3.5	1.59	-5	-1.654	2.356	3.569
z_standard	4	4	4	4	4	4
	2	2	2	2	2	2
	-2	-2	-2	-2	-2	-2
	4	4	4	4	4	4
z_distance	0.5	2.41	9	5.654	1.644	0.431
	1.5	0.41	7	3.654	0.356	1.569
	5.5	3.59	3	0.346	4.356	5.569
	7.5	5.59	1	2.346	6.356	7.569
N	1	2	4	3	2	1
ak	4	2	-4	-2	2	4

Problem 13 Demapper

```
function [b] = Demapper(zk)
b=zeros(length(zk),2);          % initialize the
matrix [k,2]
b(:,1)=~(sign(zk)+1);           % get the first bit
b(:,2)=~(zk-2*sign(zk));        % get the second bit
b=reshape(b',1,2*length(zk),[]); % reshape to [1,2k]
matrix
end
```

Code 15 Function of demapper

The above code is processed by 4ASK demapper. Through the input ak, the corresponding binary output groups bk1 and bk2 are output. The specific operation process is shown in the following table:

ak	Sign of ak	bk1= ~(Sign of ak - 1)	bk2= ~(2 - 2*Sign of ak)
4	1	0	0
2	1	0	1
-2	-1	1	1
-4	-1	1	0

~() is a logical calculation of an NOT gate, determine whether the calculation result in parentheses is 0. If it is not 0, the output is False, the output is 0. If the result is True, the output is 1.

Problem 14 Digital transmission system

```
%Prblem 14 Digital transmission system
clc;clear;close all;
%% 4ASK mapping
b=[1 0 0 0 1 1 0 1 0 1];%input
a=fourASK(b);
%% modulator
Ts=0.002;T0=Ts/10;
t=0:T0:Ts*(length(a)+1);
x=modu(a,Ts,t);
%% add GWN
varinoise=1;
y = x+sqrt(varinoise)*randn(size(x));
%% after receive filter
p = @(t) rect((t-Ts/2)/Ts);
tp = 0:T0:Ts;
h = p(Ts-tp);
z = T0/Ts * conv(y,h);
tz = T0*(0:length(z)-1);
%% sample
ts = Ts:Ts:(length(a)+2)*Ts;
zs = z(Ts/T0+1:Ts/T0:end);
%% Decision block
z_DB = DB(zs(2:end-1));
%% Demapper
result = Demapper(z_DB);
%% plot each signal
subplot(4,1,1);
stem(t,x,'fill','k-.'); grid on;
title('Transmit signal
x(t)');xlabel('t');ylabel('x(t)');
subplot(4,1,2);
stem(t,y,'fill','k-.'); grid on;
title('noisy receive signal
y(t)');xlabel('t');ylabel('y(t)');
subplot(4,1,3);
stem(tz,z,'fill','k-.'); grid on;
title('signal out of matched filter
z(t)');xlabel('t');ylabel('z(t)');
subplot(4,1,4);
stem(ts,zs,'fill','k-.'); grid on;
title('downsample zk(t)'); xlabel('time
t');ylabel('zk(t)');
```

Code 16 Code of digital transmission system

The above code is the code required by this Problem, which includes the transmitter, the channel and the receiver. The processing of the code is shown below:

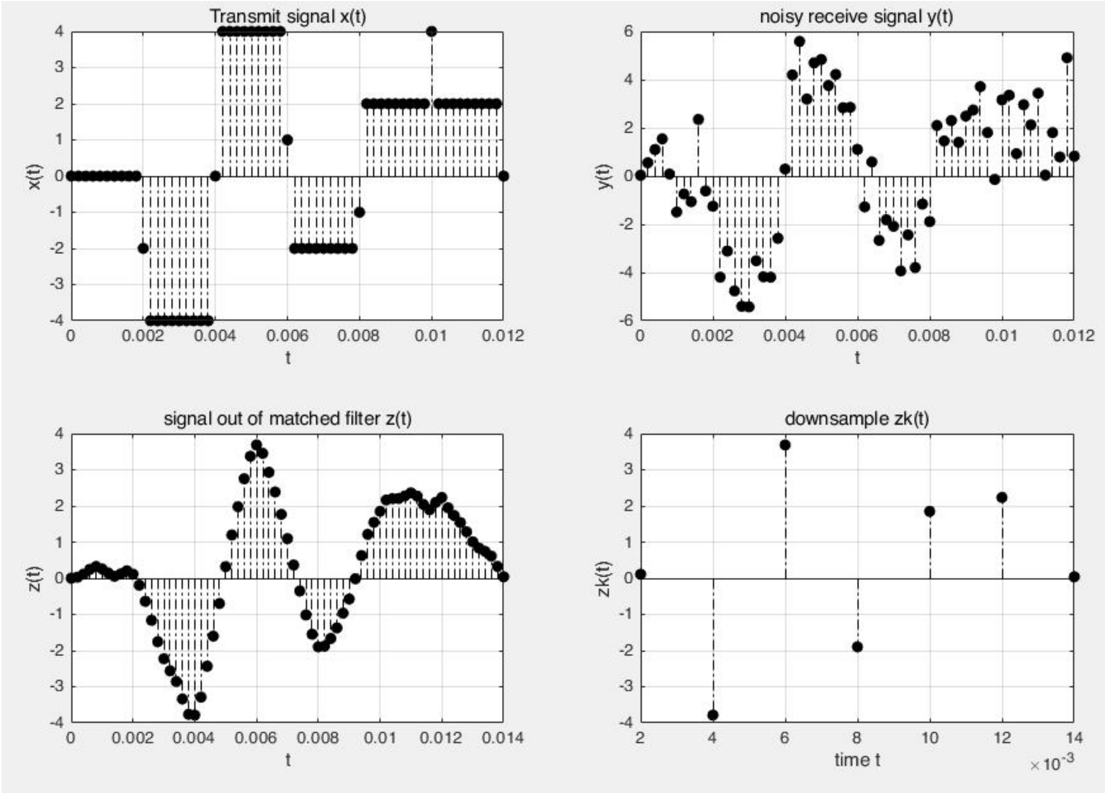

Figure 19 Processing of Code 16

According to the requirements, the start is a sequence of data bits:

b [1,0,0,0,1,1,0,1,0,1]

And the end is the sequence of estimated data bits:

result [1,0,0,0,1,1,0,1,0,1]

We can find that start and end are the same. It can be shown that under the influence of adding Gaussian white noise with a variance of 1, the system can correctly estimate the input and output it as the result.

Problem 15 Test

This section needs to test the previously designed system for low noise and high noise. The input is set to [1 0 0 0 1 1 0 1 0 1], the variance range for low noise is 1--5, and the variance range for high noise is 6--10. The test results are shown in the following table:

Variance		Input	Result	Same or not
Low noise	1	[1 0 0 0 1 1 0 1 0 1]	[1 0 0 0 1 1 0 1 0 1]	Same
	2		[1 0 0 0 1 1 0 1 0 1]	Same
	3		[1 0 0 0 1 1 0 1 0 1]	Same
	4		[1 0 0 0 1 1 0 1 0 1]	Same
	5		[1 0 0 0 1 1 0 1 0 1]	Same
Higher noise	6		[1 0 0 0 1 1 0 1 0 1]	Same
	7		[1 0 0 0 1 1 0 1 0 1]	Same
	8		[1 0 0 0 1 1 0 1 0 1]	Same
	9		[1 1 0 0 1 1 0 0 0 1]	not
	10		[1 0 0 0 0 1 0 0 0 1]	not

According to the above table, we can find that the results obtained under low noise are always consistent with our expected results, and when the noise increases, the results deviate from the expected conditions.

This is because when the noise increases, the processing effect of the filter h (t) designed in the system is not ideal, resulting in a subsequent series of error information being transmitted, and the final result also deviates from expectations.