XJTLU

Electrical and Electronic Engineering

EEE411 Advanced Signal Processing

MATLAB Individual Assignment II

Full Name:
Jing Li

Student ID number:
1301440

Supervisor:
Dr. Siyi Wang

Email Address:
Jing.Li17@student.xjtlu.edu.cn

December 11, 2017

1 Introduction

This lab requires students to deal with the simulation of a digital communication system with MATLAB, including digital modulator and demodulator. The lab has been divided into three parts. The first part contains five exercises, which simulates the process of digital modulator. The second part is about digital demodulator. It contains eight questions. The third part puts the modulator and demodulator together to simulate a digital transmission system. The corresponding codes, resulting plots and analyses are presented in the following parts.

2 Digital Modulator

Under normal circumstances, the channel cannot transfer the original signal generated by the source directly. The original signal need to be transferred to the signal which can be transmitted in the channel. This procedure is called modulation. Digital modulation can be considered as digital-to-analog transform. Figure 1 shows a block diagram of a digital transmitter.

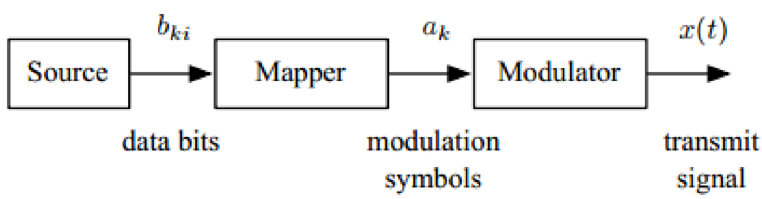


Figure 1: the process of a digital transmitter

2.1 Problem 1

2.1.1 Question

Consider 4ASK with the modulation alphabet $A = \{-4, -2, +2, +4\}$, and a mapping from bit pairs $b_{k1}b_{k2}$ to modulation symbols a_k, as given in the following table (the index k indicates the time):

$b_{k1}b_{k2}$	a_k
00	+4
01	+2
11	-2
10	-4

Write a MATLAB function that implements this mapping with a bit pair as the input and the corresponding modulation symbol as the output. Next, write a MATLAB function that maps a sequence of data bits to the corresponding sequence of modulation symbols.

2.1.2 Solution

The solution of mapping table is presented in the Question part. Listing 1 presents a MATLAB script which is used to solve this problem. Listing 2 shows the `mapping_sequence` function that is used to map data bits to symbols. It can not only map a bit pair, but also map a sequence of data bits.

Listing 1: MATLAB script

```
%create a sequence of data bits randomly
data=randi([0 1],1,8);
%maps bits to symbols
symbols=mapping_sequence(data);
```

Listing 2: `mapping_sequence` function

```
%mapping table
%00=+4;01=+2;11=-2;10=-4;
function symbols= mapping_sequence(bits)

pairs=reshape(bits,2,[]);
symbols(pairs(1,:)==0&pairs(2,:)==0)=+4;
symbols(pairs(1,:)==0&pairs(2,:)==1)=+2;
symbols(pairs(1,:)==1&pairs(2,:)==1)=-2;
symbols(pairs(1,:)==1&pairs(2,:)==0)=-4;

end
```

Data bits are created randomly. Three random examples are shown in Figure 2, Figure 3 and Figure 4. All examples have eight bits and and are mapped to four symbols. According to the mapping table, the corresponding produced symbols are correct.

data	[0,1,1,0,1,0,0,1]
symbols	[2,-4,-4,2]

Figure 2: random data bits and corresponding modulation symbols

data	[1,1,1,0,1,1,1,1]
symbols	[-2,-4,-2,-2]

Figure 3: random data bits and corresponding modulation symbols

data	[1,0,1,0,1,0,0,0]
symbols	[-4,-4,-4,4]

Figure 4: random data bits and corresponding modulation symbols

2.2 Problem 2

2.2.1 Question

Assume the transmit pulse $p(t) = rect(\frac{(t-T_s)/2}{T_s})$ and the symbol period $T_s = 2$ msec. Sketch the transmit pulse by hand (so that you know what to expect in MATLAB).
Plot the transmit pulse with MATLAB using a time resolution of 10 sample points per symbol period; pick a reasonable time interval. (You may use `rect.m` from the ICE.)

2.2.2 Solution

The sketch of the transmit pulse is shown in Figure 5. When $t \in (0, 2)$, $p(t)$ equals to 1. Otherwise, $p(t)$ equals to 0.

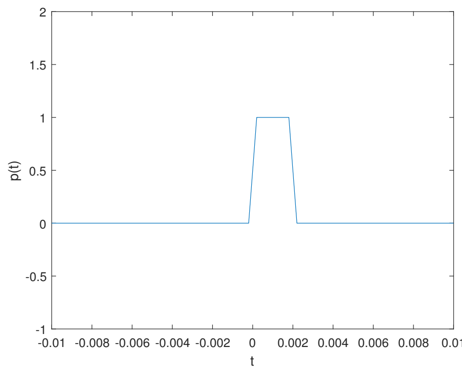


Figure 5: the sketch of the digital transmitter

It is easy for MATLAB to plot the transmit pulse. First, define $p(t)$ using `rect` function with its range and sampling period. Then, use `plot` function to plot. Listings below show the MATLAB script and `rect` function which is provided on the ICE.

Listing 3: MATLAB script

```
Ts=0.002;
T0=Ts/10;
t=-0.01:T0:0.01;
p=rect((t-Ts/2)/Ts);
plot(t,p);
xlabel('t');
ylabel('p(t)')
axis([-0.01 0.01 -1 2]);
```

Listing 4: `rect` function

```
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;
```

The resulting plot can be seen in Figure 6. Comparing Figure 5 and Figure 6, it is easy to see that the second plot has slopes in rising edge and falling edge. The reason is that in `rect` function, when $\frac{(t-T_s)/2}{T_s} = \frac{1}{2}\text{or}-\frac{1}{2}$, $p(t)$ equals to $\frac{1}{2}$ not 1. MATLAB is difficult to create a ideal rectangular pulse which exists in the real situation. In general, two figures are approximate.

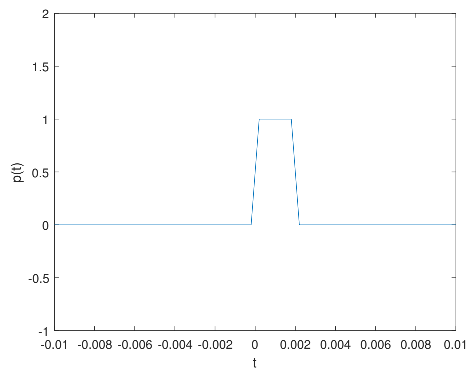


Figure 6: the process of a digital transmitter

2.3 Problem 3

2.3.1 Question

Assume a sequence of K modulation symbols: $a_1, a_2, ..., a_K$. The corresponding transmit signal is given by

$$x(t) = \sum_{K}^{k=1} a_k \cdot p(t - kT_s) \tag{1}$$

Sketch by hand the transmit signal for a sequence a_1, a_2, a_3, a_4 of four chosen modulation symbols.

2.3.2 Solution

The four chosen modulation symbols are $\{+4, +2, -2, -4\}$. Based on the equation given in the Question part, the sketch of the transmit signal is presented in Figure 7. In order to avoid the influence among symbols, the k-th symbol has the corresponding carrier wave with k time shift.

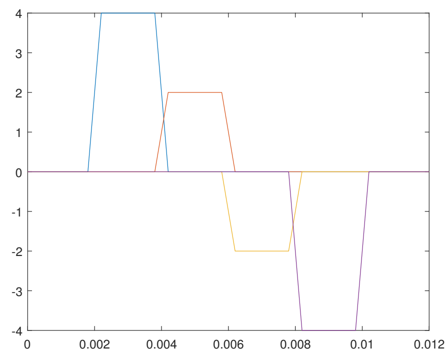


Figure 7: the sketch of the transmit signal

2.4 Problem 4

2.4.1 Question

Use the same modulation symbols as in the previous question. Write a MATLAB script to plot the signal $x_1(t) = a_1 \cdot p(t - T_s)$ with MATLAB. In a similar way, plot the signals $x_2(t), x_3(t)$, and $x_4(t)$ in the same figure.

2.4.2 Solution

MATLAB script of this question is shown in Listing 5. The range of t is $[0, 6T_s]$.

Listing 5: MATLAB script

```
%the same modulation symbols in the previous question
a=[+4,+2,-2,-4];

%the transmit pulse
Ts=0.002;
T0=Ts/10;
t=0:T0:6*Ts;
p=@(t)rect((t-Ts/2)/Ts);

```

```
%plot the signal x1(t)
x1=a(1,1)*p(t-1*Ts);
plot(t,x1);
hold on;

%plot the signal x2(t)
x2=a(1,2)*p(t-2*Ts);
plot(t,x2);
hold on;

%%plot the signal x3(t)
x3=a(1,3)*p(t-3*Ts);
plot(t,x3);
hold on;

%plot the signal x4(t)
x4=a(1,4)*p(t-4*Ts);
plot(t,x4);
```

The resulting plot can be seen in Figure 8. There are four signals with different colours. The first blue signal is from $a = +4$, the second red signal is from $a = +2$, the third orange signal is from $a = -2$ and the last purple signal is from $a = -4$.

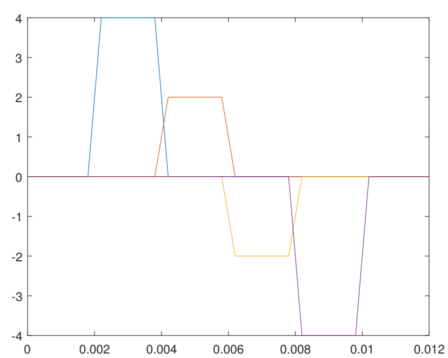


Figure 8: four signals in the same figure

2.5 Problem 5

2.5.1 Question

Write a MATLAB function that implements the modulator, *i.e.* takes a sequence of modulation symbols as the input and produces the transmit signal as the output.
Test your function by comparing it to the result of the previous two equations.

2.5.2 Solution

There are two listings below. Listing 6 is MATLAB script which run `modulator` function to achieve the transmit signal based on a sequence of symbols. Listing 7 shows the `modulator` function that is used to produce a transmit signal.

Listing 6: MATLAB script

```
symbols=[+4,+2,-2,-4];
[t,signal] = modulator(symbols);
plot(t,signal);
xlabel('t');
ylabel('x(t)');
```

Listing 7: `modulator` function

```
function [t,signal] = modulator(symbols)

signal=0;
col=size(symbols,2);

Ts=0.002;
T0=Ts/10;
t=0:T0:6*Ts;
p=@(t)rect((t-Ts/2)/Ts);

for i=1:col
    signal=signal+symbols(1,i)*p(t-i*Ts);
end

end
```

The final transmit signal is presented in Figure 9 with respect to the sequence of modulation symbols. Comparing it to the output plot of Problem 4, it is clear to see that Figure 8 is the decomposition of Figure 9 and Figure 9 is the integration of Figure 8. In addition, the transmit signals of Figure 7 and Figure 9 are same except MATLAB system error.

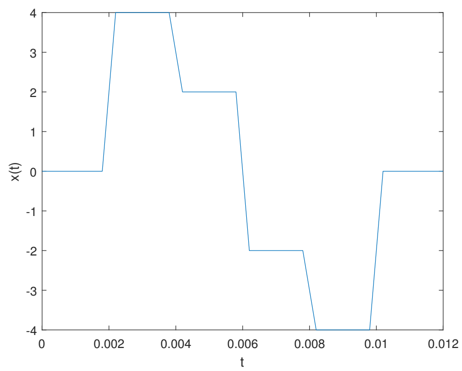


Figure 9: the final transmitted signal

3 Digital Demodulator

Digital demodulation is a method to recover the information content from the modulated signal, which can be considered as analog-to-digital conversion. Figure 10 shows a block diagram of a digital receiver.

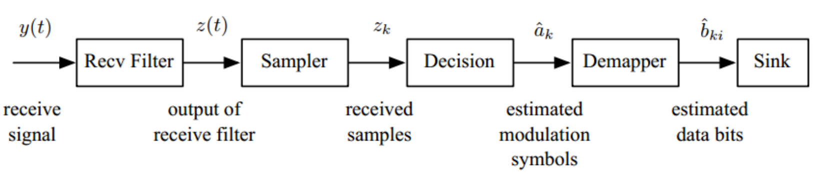


Figure 10: the process of a digital transmitter

3.1 Problem 6

3.1.1 Question

Assume that the receive filter is a matched filter, *i.e.*, the impulse response of the receive filter is $h(t) = p(T_s - t)$. Sketch by hand the impulse response of the receiver filter (with the correct time shift).

3.1.2 Solution

Figure 11 presents the sketch of the impulse response of the receiver filter. When the signal is transmitted through the channel, it will be some noise being added to the transmitted signal. The receiver filter is a matched filter. Matched filter is used to maximize the signal-to-noise ratio (SNR) in the presence of noise.

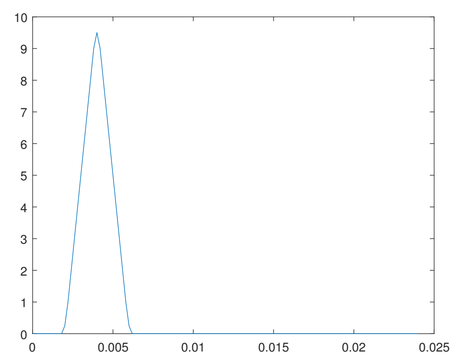


Figure 11: the sketch of the impulse response of the receiver filter

3.2 Problem 7

3.2.1 Question

For the time being, assume noise-free transmission. In this case, the received signal is $y(t) = x(t)$. The output signal of the receive filter is $z(t) = y(t)*h(t)$. Show analytically (by a short calculation) that

$$z(t) = \sum_{k=1}^{K} a_k \cdot [p(t - kT_s) * h(t)] \tag{2}$$

Which properties do you use to show that?

3.2.2 Solution

distribution law

3.3 Problem 8

3.3.1 Question

Use MATLAB to compute and plot the signal $g(t) = p(t - kT_s) * h(t)$ for $k = 1$. Discuss if the result is expected.

3.3.2 Solution

Listing 8 below shows a MATLAB script used to plot $g(t)$.

Listing 8: MATLAB script

```
Ts=0.002;
T0=Ts/10;
t=0:T0:6*Ts;
p=@(t)rect((t-Ts/2)/Ts);

y=p(t-Ts);
h=p(Ts-t);

figure;
t1=0:T0:12*Ts;
g=conv(y,h);
plot(t1,g);
```

The plot of the signal $g(t)$ could be seen in Figure 12. $h(t)$ is a matched filter. Based on the theory, the maximum output of the matched filter will obtain at symbol period T_s. Observing the signal in 12, it is the result expected.

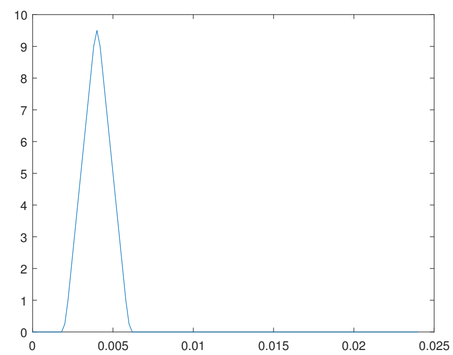


Figure 12: the plot of the signal $g(t)$

3.4 Problem 9

3.4.1 Question

Assume the transmit signal $x_1(t)$ from Problem 4. Compare and plot the corresponding signal $z(t)$. Relate your result to $g(t)$ from the previous question. Repeat this with $x_2(t)$. Is the result as expected? Now repeat this with the full transmit signal $x_(t)$ and interpret the result.

3.4.2 Solution

The Matlab script in Listing 9 is used to compute and plot signal $z(t)$ with multiple transmitted signal.

Listing 9: Matlab script

```
ak=[+4,+2,-2,-4];

Ts=0.002;
T0=Ts/10;
t=0:T0:6*Ts;
p=@(t)rect((t-Ts/2)/Ts);

h=p(t);

y1=ak(1,1)*p(t-1*Ts);
z1=T0/Ts*conv(y1,h);
```

```
tz1=T0*(1:(length(z1)));
plot(tz1,z1);

figure;
y2=ak(1,2)*p(t-2*Ts);
z2=T0/Ts*conv(y2,h);
tz2=T0*(1:(length(z2)));
plot(tz2,z2);

figure;
[time,y]=modulator(ak);
z=T0/Ts*conv(y,h);
tz=T0*(1:(length(z)));
plot(tz,z);
```

There are three resulting figures. Figure 13 presents the corresponding signal $z(t)$ when transmit signal is $x_1(t)$. Figure 14 plots the output of receive filter with $x_2(t)$. Figure 15 presents the output with the full transmit signal $x(t)$.

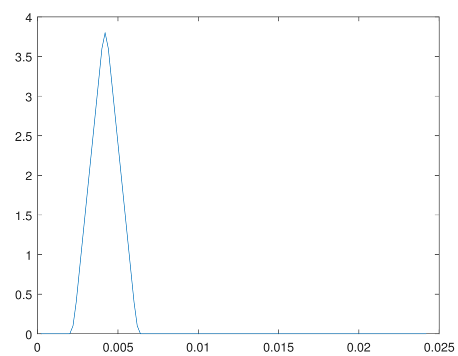


Figure 13: the process of a digital transmitter

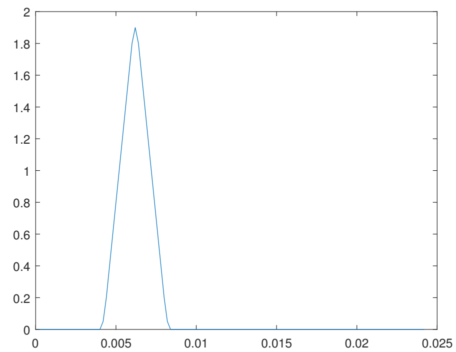


Figure 14: the process of a digital transmitter

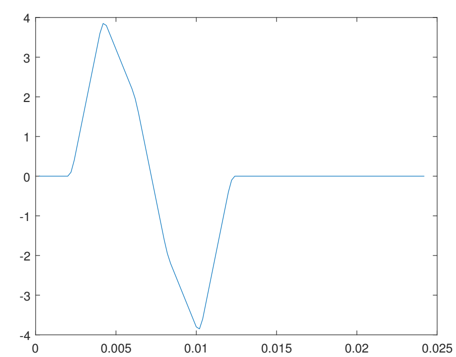


Figure 15: the process of a digital transmitter

The $h(t)$ in Problem 8 and $h(t)$ in this problem are different. But, comparing the plots, they are same. This means that two filters are both matched filters for the transmitted signal. Comparing $g(t)$ and $z_1(t)$, they all obtain the maximum value at the 2 times of T_s with different amplitudes. This results from different multiplier before convolution. Considering the original first symbol, the amplitude of $z_1(t)$ is expected which is 3.85. The second symbol is $+2$, so the result of $z_2(t)$ 1.95 is also expected. The transmitted signal for Figure 15 is full transmit $x(t)$ and the output is corresponding full signal $z(t)$.

3.5 Problem 10

3.5.1 Question

Based on the results of the previous problem, discuss what are the optimal sampling times for $z(t)$, to determine the samples z_k. Write a MATLAB function (or script) that obtains the samples z_k from $z(t)$.

3.5.2 Solution

Listing 9 is MATLAB script which run `obtain_sample` function to achieve samples z_k from $z(t)$. Listing 10 shows the `obtain_sample` function that is used to obtain samples.

Listing 10: MATLAB script

```
ak=[+4 +2 -2 -4];
[t,y]=modulator(ak);

Ts=0.002;
T0=Ts/10;
p=@(t)rect((t-Ts/2)/Ts);
h=p(t);

z=T0/Ts*conv(y,h);
tz=T0*(1:(length(z)));

z_k=obtain_sample(z,11,10,4);
```

Listing 11: `obtain_sample` function

```
%start_point: which point start to take sample
%Ts: sampling period
%number_sample: the number of sampling
function z_k= obtain_sample(z_t,start_point,Ts,number_sample)
z_k=[];

for i=start_point+Ts:Ts:11+number_sample*Ts
    z_k=[z_k z_t(i)];
end

end
```

In 16, ak is the original symbols and z_k is obtained samples. The output of matched filters merely can be used to make decisions at multiple of T_s, which is meaningful for receiver. Other information at other time is useless for receiver. Comparing ak and z_k, it is easy to see that samples are good enough to recover the original symbols. Therefore, the optimal sampling time for $z(t)$ is 0.002ms, which means 1 sample point per 10 points.

ak	[4,2,-2,-4]
z_k	[3.8500,1.9500,-1.9500,-3.8500]

Figure 16: the process of a digital transmitter

3.6 Problem 11

3.6.1 Question

Assume now transmission over an AWGN channel:$y(t) = x(t) + w(t)$. With MATLAB you can obtain a noisy receive signal.
The value of **varnoise** denotes the variance of the additive white Gaussian noise. Write a MATLAB script that plots (in subplots) the transmit signal, the noisy receive signal, the signal at the matched filter output, and prints out the samples z_k.
Do experiments with various noise variance, and discuss the effect of noise on $y(t)$, $z(t)$ and z_k. Is it easier to estimate the transmitted modulation symbol a_k from $y(t)$ or from $z(t)$?

3.6.2 Solution

Listing below presents a MATLAB script that can plot the transmit signal, the noisy receive signal and the signal at the matched filter output.

Listing 12: MATLAB script

```
%problem 11
varnoise=1;

symbols=[+4,+2,-2,-4];
[t,x] = modulator(symbols);
plot(t,x);
title('the transmit signal');
```

```

figure;
y=x+sqrt(varnoise)*randn(size(x));
plot(t,y);
title('the noisy receive signal');

Ts=0.002;
T0=Ts/10;
p=@(t)rect((t-Ts/2)/Ts);
h=p(Ts-t);

figure;
t1=0:T0:12*Ts;
z=conv(y,h)/10;
plot(t1,z);
title('the signal at the matched filter output');

figure;
z_k=obtain_sample(z,11,10,4);
plot(z_k);

%problem 12
figure;
a_k=determine_ak(z_k);
plot(a_k);

%problem 13
bits=demapping(a_k);
```

The resulting plots of signals can be seen below. Figure 17 shows three signal with `varnoise` $= 1$, the corresponding z_k is presented in Figure 18. Figure 19 shows the results with `varnoise` $= 3$ and the value of samples z_k can be seen in Figure 20. The `varnoise` in Figure 21 equals to 10. The relative samples z_k are shown in Figure 22.

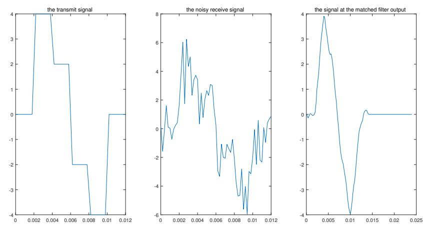
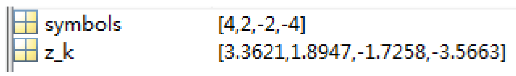


Figure 17: the process of a digital transmitter



symbols	[4,2,-2,-4]
z_k	[3.3621,1.8947,-1.7258,-3.5663]

Figure 18: the process of a digital transmitter

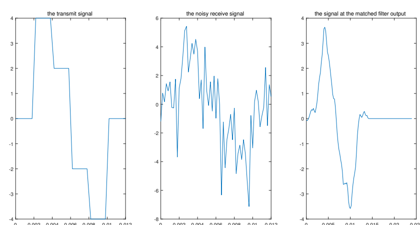
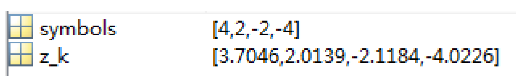


Figure 19: the process of a digital transmitter



symbols	[4,2,-2,-4]
z_k	[3.7046,2.0139,-2.1184,-4.0226]

Figure 20: the process of a digital transmitter

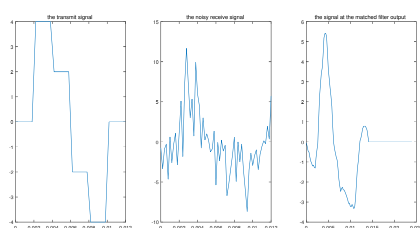


Figure 21: the process of a digital transmitter

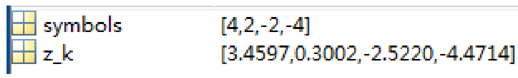


Figure 22: the process of a digital transmitter

In the Problem 10, the transmission is noise-free. However, in this problem, transmission goes through AWGN channel. So, AWGN noise is added to the transmit signal and the receive signal become noisy. Comparing the first and the second subplots in Figure 17, 19 and 21, the noisy receive signals become more chaotic as a increase of `varnoise`. With the change of receive signals, the third signals at the matched filter output in figures are also changed, which contains several distortion. Based on the characteristic of matched filters that filters merely focus on amplitudes at multiple of T_s, z_k can be obtained successfully, but the values might be changed. Compared with $y(t)$, it is easier to estimate the transmitted modulation symbols from $z(t)$. Because the effects of noise on $y(t)$ are more severe that on $z(t)$. After matched filter, $z(t)$ can recover the most important information.

3.7 Problem 12

3.7.1 Question

Consider now the decision block. Determine the decision regions for each modulation symbol. Then write a MATLAB function that implements the decision function, *i.e.*, that maps z_k to $\widehat{a_k}$.

3.7.2 Solution

Listing 13 shows a `determine_ak` function used to map z_k to $\widehat{a_k}$. Calculate the distance between z_k and the original modulation symbol, find the closet one and assign this symbol to $\widehat{a_k}$.

Listing 13: `determine_ak` function

```
function a_k = determine_ak(z_k)
a_k=[];
%calculate the distance between samples of z_k and the real symbols
%find the closet one and then maps z_k to a_k
for i=1:length(z_k)
    if z_k(i)>0
```

```
        dis1=abs(4-z_k(i));
        dis2=abs(2-z_k(i));
        if dis1<dis2
            a_k=[a_k 4];
        else
            a_k=[a_k 2];
        end
    else
        dis3=abs(-4-z_k(i));
        dis4=abs(-2-z_k(i));
        if dis3<dis4
            a_k=[a_k -4];
        else
            a_k=[a_k -2];
        end
    end
end

end
```

The answers can be seen below. The modulation symbols are $[+4, +2, -2, -4]$ and the samples are $[3.3942, 1.8294, -1.8379, -3.9977]$. After mapping, the final $\widehat{a_k}$ are $[+4, +2, -2, -4]$ that are the same as the modulation symbols.

a_k	[4,2,-2,-4]
symbols	[4,2,-2,-4]
z_k	[3.3942,1.8294,-1.8379,-3.9977]

Figure 23: the process of a digital transmitter

3.8 Problem 13

3.8.1 Question

Write a MATLAB function that implements the Demapper. (This is very similar to your Mapper, written before.)

3.8.2 Solution

Listing 14 below presents a `demapping` function which maps a sequence of modulation symbols to the corresponding sequence of data bits based on the previous mapping table.

Listing 14: `demapping` function

```
function bits = demapping(symbols)
len=length(symbols);
for i=1:len
    if symbols(i)==+4
        bits(:,i)=[0;0];
    elseif symbols(i)==+2
        bits(:,i)=[0;1];
    elseif symbols(i)==-2
        bits(:,i)=[1;1];
    elseif symbols(i)==-4
        bits(:,i)=[1;0];
    end
end
bits=reshape(bits,1,[]);
end
```

The answers can be seen below. Figure 24 shows the symbols and the corresponding data bits.

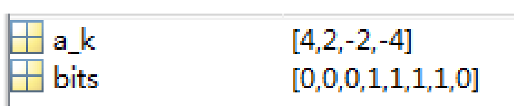


Figure 24: the process of a digital transmitter

4 Digital Transmission System

In the previous questions we have investigated the individual components required for a baseband digital communication system. We now put all the components together.

4.1 Problem 14

4.1.1 Question

Write a MATLAB script that implements the transmitter, the channel and the receiver.
The start is a sequence of data bits (you may choose them or pick them randomly), and the end is the sequence of estimated data bits.

4.1.2 Solution

Listing below presents a MATLAB script which simulates the whole procedure from digital modulation to digital demodulation. The transmission is noise-free in this problem and the sequence of data bits are picked randomly.

Listing 15: MATLAB script

```
%the transmitter
data=randi([0 1],1,8);
symbols=mapping_sequence(data);
[t,x] = modulator(symbols);
plot(t,x);
title('the transmitter');

%the channel
Ts=0.002;
T0=Ts/10;
p=@(t)rect((t-Ts/2)/Ts);
h=p(Ts-t);
figure;
plot(t,h);
title('the channel');

%the receiver
figure;
t1=0:T0:12*Ts;
z=conv(x,h)/10;
plot(t1,z);
title('the receiver');

%the estimated data bits
```

```
z_k=obtain_sample(z,11,10,4);
a_k=determine_ak(z_k);
bits=demapping(a_k);
```

The transmitted signal of random data bits is plotted in Figure 25. The matched filter can be seen in Figure 26. Figure 27 presents the output of matched filter. In figure 28, the start is data $[1, 1, 0, 1, 1, 0, 1, 0]$ and the end is estimated bits $[1, 1, 0, 1, 1, 0, 1, 0]$. Two sequence of data bits are the same, which means that data bits are transmitted and received successfully.

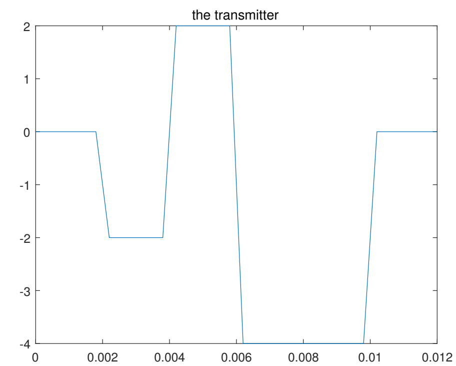


Figure 25: the process of a digital transmitter

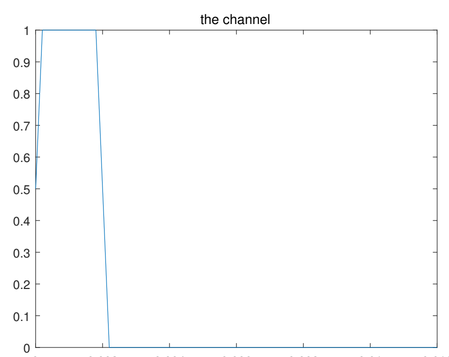


Figure 26: the process of a digital transmitter

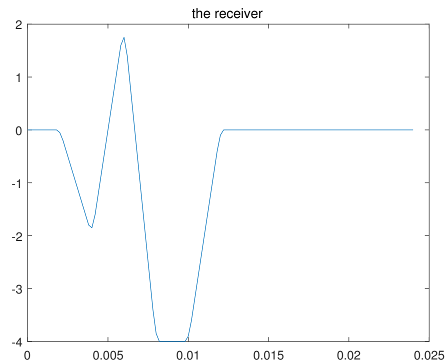


Figure 27: the process of a digital transmitter

bits	[1,1,0,1,1,0,1,0]
data	[1,1,0,1,1,0,1,0]

Figure 28: the process of a digital transmitter

4.2 Problem 15

4.2.1 Question

Test your script for very low noise and for higher noise. Does the system react as expected?

4.2.2 Solution

Listing 16 shows a MATLAB script which adds AWGN noise based on the script in Problem 14.

Listing 16: MATLAB script

```
%the transmitter
data=randi([0 1],1,8);
symbols=mapping_sequence(data);
[t,x] = modulator(symbols);
plot(t,x);
title('the transmitter');

```

```
%add noise to the transmitter
figure;
varnoise=1;
y=x+sqrt(varnoise)*randn(size(x));
plot(t,y);
title('the noisy signal');

%the channel
figure;
Ts=0.002;
T0=Ts/10;
p=@(t)rect((t-Ts/2)/Ts);
h=p(Ts-t);
plot(t,h);
title('the channel');

%the receiver
figure;
t1=0:T0:12*Ts;
z=conv(y,h)/10;
plot(t1,z);
title('the receiver');

%the estimated data bits
z_k=obtain_sample(z,11,10,4);
a_k=determine_ak(z_k);
bits=demapping(a_k);
```

There are four kinds of results with different `varnoise`. The value of `varnoise` represent the level of noise. In Figure 29, `varnoise` equals to 1 and the final data bits are estimated successfully which is presented in Figure 30. When `varnoise` is 5, the data bits are also recovered. As for higher noise, this script can not estimate data bits completely with `varnoise` = 50 or 100.

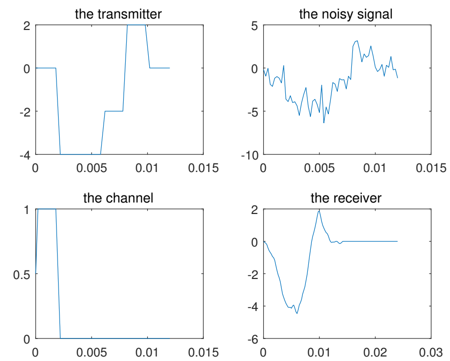


Figure 29: the process of a digital transmitter

bits	[1,0,1,0,1,1,0,1]
data	[1,0,1,0,1,1,0,1]

Figure 30: the process of a digital transmitter

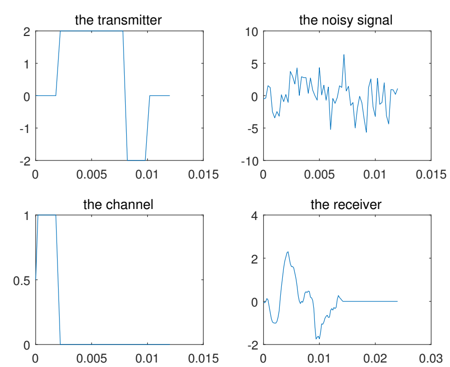


Figure 31: the process of a digital transmitter

bits	[0,1,0,1,0,1,1,1]
data	[0,1,0,1,0,1,1,1]

Figure 32: the process of a digital transmitter

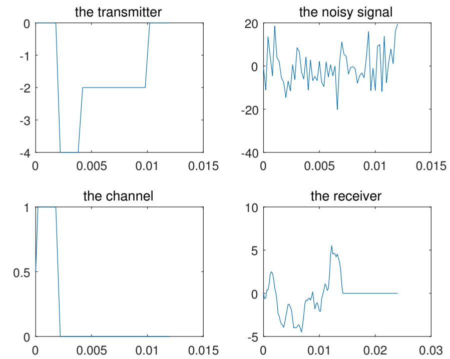


Figure 33: the process of a digital transmitter

Figure 34: the process of a digital transmitter

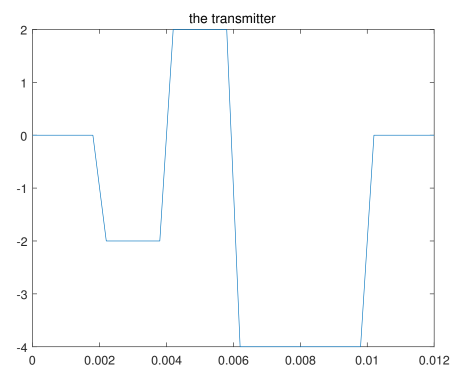


Figure 35: the process of a digital transmitter

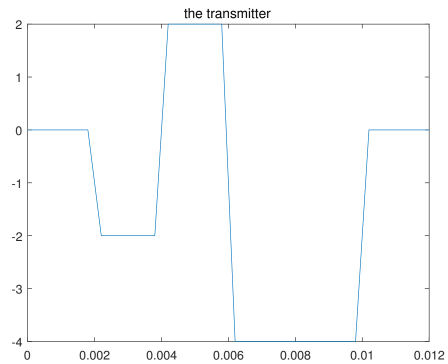


Figure 36: the process of a digital transmitter

5 Conclusion

In conclusion, through this experiment, students master general knowledge about digital modulation and demodulation. In the real situation, there always exist noise when transmitting signals, so it is important to find the optimal filter to remove noise and increase SNR.