浙江工业大学

MATLAB & Communication Simulations

Lab Report4

Name 凌智城

Teacher 张昱

Class 通信工程 1803 班

Student ID 201806061211

Department 信息工程学院

Date： May 5th,2021

I. Task_1

A. Code

lab4_1.m

```
%% Sampling the Antipodal Return to Zero baseband signal.

clear;
clc;

%% Generate a single symbol
Tb = 0.5;               % Symbol period.
fs = 1000;              % Sampling rate.
dt = 1/fs;              % Sampling interval.
N_sample = Tb*fs;       % Number of sampling points per symbol.
N = 7;                  % Number of symbols.
% Sequence transmission time.
t = 0 : dt : (N * N_sample - 1) * dt;
gt = [ones(1, N_sample / 2), zeros(1, N_sample / 2)]; % RZ.

%% Generate sequence '0 0 1 1 0 0 1'.
base = [0 0 1 1 0 0 1];    % 0 1 basic sequence.
st = [];
for i = 1 : N              % Generate sequence.
   if base(i)==1
       st = [st gt];
   else
       st = [st -1*gt];
   end
end

%% Draw result.
figure(1);
plot(t, st);grid on;
title('DBRZ¡®0011001¡¯');
```

B. Figure

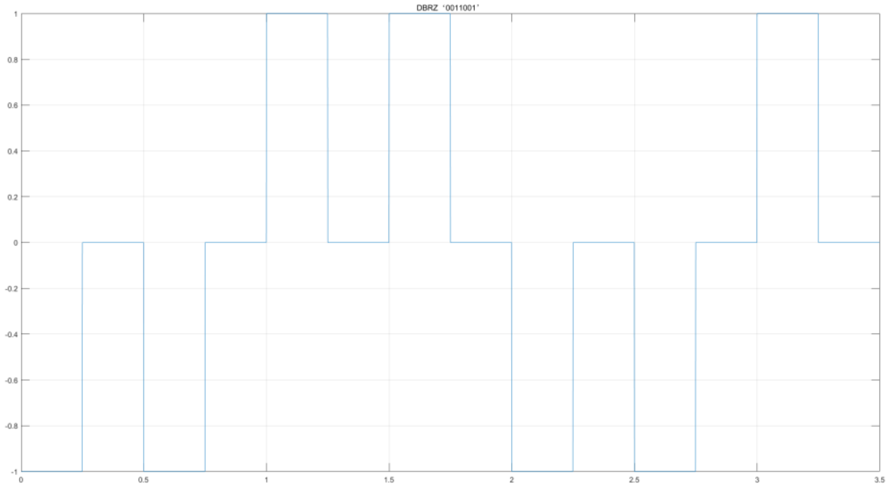


Fig.1 The Antipodal Return to Zero (RZ) data bit sequence '0011001'.

C. Discussion

The period of a single symbol signal is Tb=0.5s, and the duty cycle is 50%. Taking the first symbol as an example, 0 is represented as a transition from -1 to 0 in the bipolar return-to-zero signal, that is, it returns to zero before the next symbol arrives. The value is -1 within 0~0.25s, and the value is 0 within 0.25s~0.5s. Since there is a zero potential interval between adjacent symbols, the receiving end can easily identify the start and end time of each symbol, so that the sender and receiver can maintain correct bit synchronization.

II. Task_2

A. Code

lab4_2.m

```
%% Sampling the Antipodal Return to Zero baseband signal.
```

```matlab
clear;
clc;

%% Task
snr_in_dB=5.5;                  % Give the SNR (i.e., E/N0) in dB.
PSD = 0.1;
N0 = 2*PSD;

SNR=exp(snr_in_dB*log(10)/10);  % Signal-to-noise ratio.
E = SNR*N0;
% Sigma, standard deviation of noise.
sgma=E/sqrt(2*SNR);
N=1000;

% Generation of the binary data source follows.
for i=1:N
   temp=rand;                   % A uniform random variable over
(0,1).
   if (temp<0.5)
      dsource(i)=0;             % With probability 1/2, source output
is 0.
   else
      dsource(i)=1;             % With probability 1/2, source output
is 1.
   end
end
% The detection, and probability of error calculation follows.
numoferr=0;
for i=1:N
   % The matched filter outputs.
   if (dsource(i)==0)
      r=-E+gngauss(sgma);       % if the source output is "0".
   else
      r=E+gngauss(sgma);        % if the source output is "1".
   end
   % Detector follows.
   if (r<0)
      decis(i)=0;               % Decision is "0".
   else
      decis(i)=1;               % Decision is "1".
   end
   % If it is an error, increase the error counter.
   if (decis(i)~=dsource(i))
```

```
        numoferr=numoferr+1;
    end
end
p=numoferr/N;                          % probability of error e
```

B. Figure

工作区

名称	值
decis	*1x1000 double*
dsource	*1x1000 double*
E	0.7096
i	1000
N	1000
N0	0.2000
numoferr	6
p	0.0060
PSD	0.1000
r	-0.7743
sgma	0.2664
SNR	3.5481
snr_in_dB	5.5000
temp	0.1186

Fig.2 Lab4_2_database.

C. Discussion

It can be seen that if the noise of PSD=0.1 is added, after Monte Carlo simulation, there will be several misjudgments of the signal, resulting in an increase in the bit error rate. Obviously, if the power spectral density of the noise becomes larger, the bit error rate will continue to increase.

III. Task_3

A. Code

lab4_3.m

```
%% Calculate the symbol error rate by Monte Carlo simulation.

clear;
clc;

%% Task(1):Plot Antipodal RZ baseband signal constellation
diagram.
x1=1;
y1=0;
x2=-1;
y2=0;
figure(1)
subplot(1,2,1)
plot(x1,y1,'o',x2,y2,'*')
axis('square')

%% Task(2):Suffering noise with PSD of 0.1,constellation diagram.
PSD = 0.1;
N0 = 2*PSD;
n0=N0*randn(100,1);
n1=N0*randn(100,1);
n2=N0*randn(100,1);
n3=N0*randn(100,1);
x1=1+n0;
y1=n1;
x2=-1+n2;
y2=n3;
subplot(1,2,2)
plot(x1,y1,'o',x2,y2,'*')
axis('square')
```

B. Figure

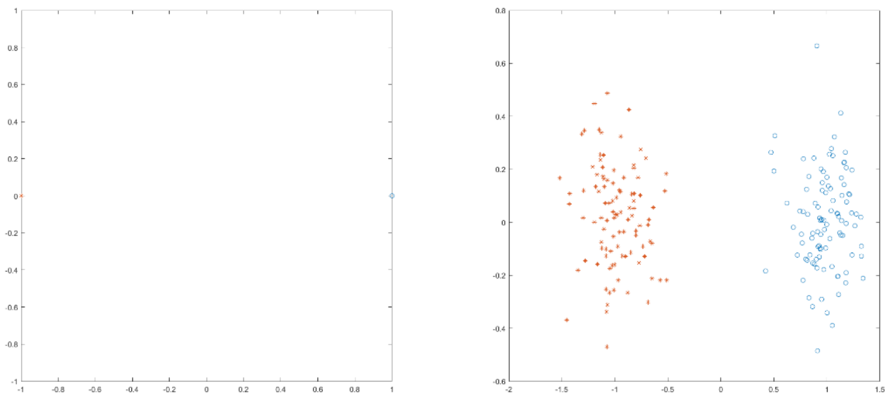


Fig.3 Antipodal RZ baseband signal constellation diagram.

C. Discussion

In Figure 3, we can see that the signal constellation with PSD=0.1 noise is added, and the signal points are centrally distributed with no noise added. Similarly, if the noise is large enough, the constellation diagrams of bipolar signals will overlap to a certain extent, which will lead to misjudgment in signal discrimination and increase the bit error rate.

IV. Task_4

A. Code

lab4_4.m

```
%% Calculate the symbol error rate by Monte Carlo simulation.

clear;
clc;

%%Realize the optimum receiver and
```

```
%verify the code using the antipodal baseband signal.

% Generate a single symbol.
Tb = 0.5;              % Symbol period.
fs = 1000;             % Sampling rate.
dt = 1/fs;             % Sampling interval.
N_sample = Tb*fs;      % Number of sampling points per symbol.
N = 7;                 % Number of symbols.
t = 0 : dt : (N * N_sample - 1) * dt; % Sequence transmission
time.
gt = [ones(1, N_sample / 2), zeros(1, N_sample / 2)]; % RZ.
% Generate baseband sequence '0 0 1 1 0 0 1'.
base1 = [0 0 1 1 0 0 1];      % 0 1 basic sequence.
s1t = [];
for i = 1 : N                 % Generate original sequence.
   if base1(i)==1
      s1t = [s1t gt];
   else
      s1t = [s1t -1*gt];
   end
end

% Generate another decision signal sequence '1 1 1 1 1 1 1'.
base2 = [1 1 1 1 1 1 1];      % 0 1 decision sequence.
s2t = [];
for i = 1 : N                 % Generate decision sequence.
   if base2(i)==1
      s2t = [s2t gt];
   else
      s2t = [s2t -1*gt];
   end
end

nt = zeros(1,N*Tb*fs);        % Noise signal which there is zero.
rt = s1t + nt;                % Received signal.

numoferr=0;

for k = 1 : N                 % Correlator.
    s1 = s1t(1:k*Tb*fs/2);    % Correlation signal sequence.
    s2 = s2t(1:k*Tb*fs/2);
    r = rt(1:k*Tb*fs/2);      % Received signal sampling at Tb.
    % Input of the detector for the kth data bit.
    y1(k) = sum(r.*s1)*dt;
```

```
    y2(k) = sum(r.*s2)*dt;
    if (y1(k)>y2(k))
        decis(k)=base1(k);      % Decision is "base1".
    else
        decis(k)=base2(k);      % Decision is "base2".
    end
    % If it is an error, increase the error counter.
    if (decis(k)~=base1(k))
        numoferr=numoferr+1;
    end
end

p=numoferr/N;                   % probability of error estimate.
```

B. Figure

工作区

名称	值
base1	[0,0,1,1,0,0,1]
base2	[1,1,1,1,1,1,1]
decis	[0,0,1,1,0,0,1]
dt	1.0000e-03
fs	1000
gt	*1x500 double*
i	7
k	7
N	7
N_sample	500
nt	*1x3500 double*
numoferr	0
p	0
r	*1x1750 double*
rt	*1x3500 double*
s1	*1x1750 double*
s1t	*1x3500 double*
s2	*1x1750 double*
s2t	*1x3500 double*
t	*1x3500 double*
Tb	0.5000
y1	[0.2500,0.2500,0.5000,0.5000,0.7500,0.7500,1]
y2	[-0.2500,-0.2500,-0.5000,-0.5000,-0.2500,-0.2500,0]

Fig.4 Lab4_4_database.

V. Task_5

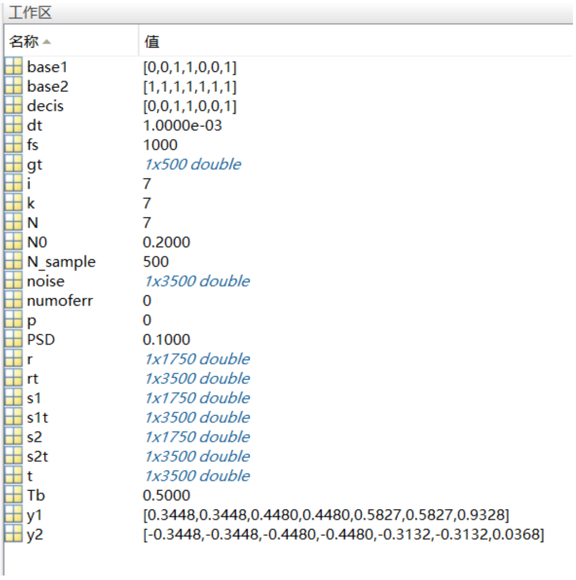
A. Code

lab4_5.m

```
%% Calculate the symbol error rate by Monte Carlo simulation.

clear;
clc;

%%Realize the optimum receiver and
%verify the code using the antipodal baseband signal.

% Generate a single symbol.
Tb = 0.5;             % Symbol period.
fs = 1000;            % Sampling rate.
dt = 1/fs;            % Sampling interval.
N_sample = Tb*fs;     % Number of sampling points per symbol.
N = 7;                % Number of symbols.
% Sequence transmission time.
t = 0 : dt : (N * N_sample - 1) * dt;
gt = [ones(1, N_sample / 2), zeros(1, N_sample / 2)]; % RZ.
% Generate baseband sequence '0 0 1 1 0 0 1'.
base1 = [0 0 1 1 0 0 1];   % 0 1 basic sequence.
s1t = [];
for i = 1 : N              % Generate original sequence.
   if base1(i)==1
      s1t = [s1t gt];
   else
      s1t = [s1t -1*gt];
   end
end

% Generate another decision signal sequence '1 1 1 1 1 1 1'.
base2 = [1 1 1 1 1 1 1];   % 0 1 decision sequence.
s2t = [];
for i = 1 : N              % Generate decision signal sequence.
   if base2(i)==1
      s2t = [s2t gt];
   else
      s2t = [s2t -1*gt];
   end
```

```
end

PSD = 0.1;
N0 = 2*PSD;
noise = randn(1,length(s1t))*sqrt(N0/2*fs);    % Add the noise.
rt = s1t + noise;                              % Received signal.

numoferr=0;

for k = 1 : N                 % Correlator.
    s1 = s1t(1:k*Tb*fs/2);  % Correlation signal sequence.
    s2 = s2t(1:k*Tb*fs/2);
    r = rt(1:k*Tb*fs/2);       % Received signal sampling at Tb.
    % Input of the detector for the kth data bit.
    y1(k) = sum(r.*s1)*dt;
    y2(k) = sum(r.*s2)*dt;
    if (y1(k)>y2(k))
        decis(k)=base1(k);      % Decision is "base1".
    else
        decis(k)=base2(k);      % Decision is "base2".
    end
    % If it is an error, increase the error counter.
    if (decis(k)~=base1(k))
        numoferr=numoferr+1;
    end
end

p=numoferr/N;                    % probability of error estimate.
```

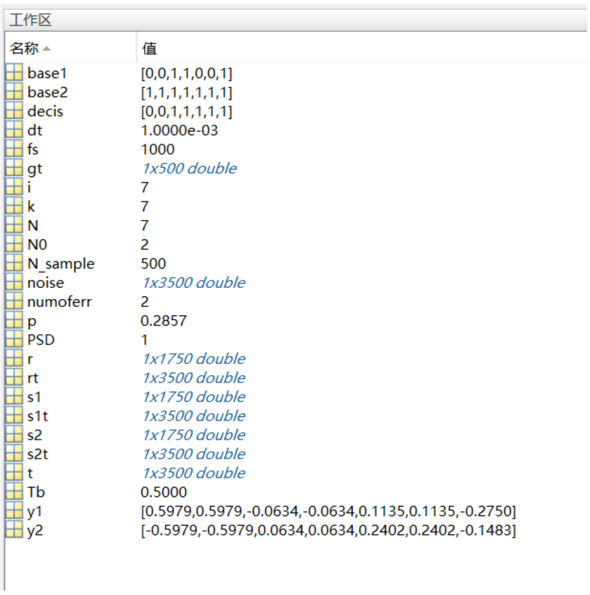
B. Figure



工作区

名称 ▲	值
base1	[0,0,1,1,0,0,1]
base2	[1,1,1,1,1,1,1]
decis	[0,0,1,1,0,0,1]
dt	1.0000e-03
fs	1000
gt	1x500 double
i	7
k	7
N	7
N0	0.2000
N_sample	500
noise	1x3500 double
numoferr	0
p	0
PSD	0.1000
r	1x1750 double
rt	1x3500 double
s1	1x1750 double
s1t	1x3500 double
s2	1x1750 double
s2t	1x3500 double
t	1x3500 double
Tb	0.5000
y1	[0.3448,0.3448,0.4480,0.4480,0.5827,0.5827,0.9328]
y2	[-0.3448,-0.3448,-0.4480,-0.4480,-0.3132,-0.3132,0.0368]

Fig.5 lab4_5_database(PSD=0.1).



工作区

名称 ▲	值
base1	[0,0,1,1,0,0,1]
base2	[1,1,1,1,1,1,1]
decis	[0,0,1,1,1,1,1]
dt	1.0000e-03
fs	1000
gt	1x500 double
i	7
k	7
N	7
N0	2
N_sample	500
noise	1x3500 double
numoferr	2
p	0.2857
PSD	1
r	1x1750 double
rt	1x3500 double
s1	1x1750 double
s1t	1x3500 double
s2	1x1750 double
s2t	1x3500 double
t	1x3500 double
Tb	0.5000
y1	[0.5979,0.5979,-0.0634,-0.0634,0.1135,0.1135,-0.2750]
y2	[-0.5979,-0.5979,0.0634,0.0634,0.2402,0.2402,-0.1483]

Fig.6 lab4_5_database(PSD=1).

C. Discussion

After adding the noise with PSD=0.1, through the relevant receiver and timing decision device, the lower bit error rate can still be guaranteed, and the original sequence can be restored; if the power spectral density of the noise is increased, the probability of misjudgment will increase proportionally Big.