通訊實驗

實驗八　第五組

電機112　林珮玉　E24084096
航太112　楊秉融　F44086181

實驗 8: Antipodal BPSK 錯誤率模擬

- 實驗目的
  - 本實驗主要探討接收端利用匹配濾波器，配合適當取樣時間來解調 AWGN 通道中之 antipodal BPSK 信號，並探討其錯誤率效能。

實驗 8: Antipodal BPSK 錯誤率模擬

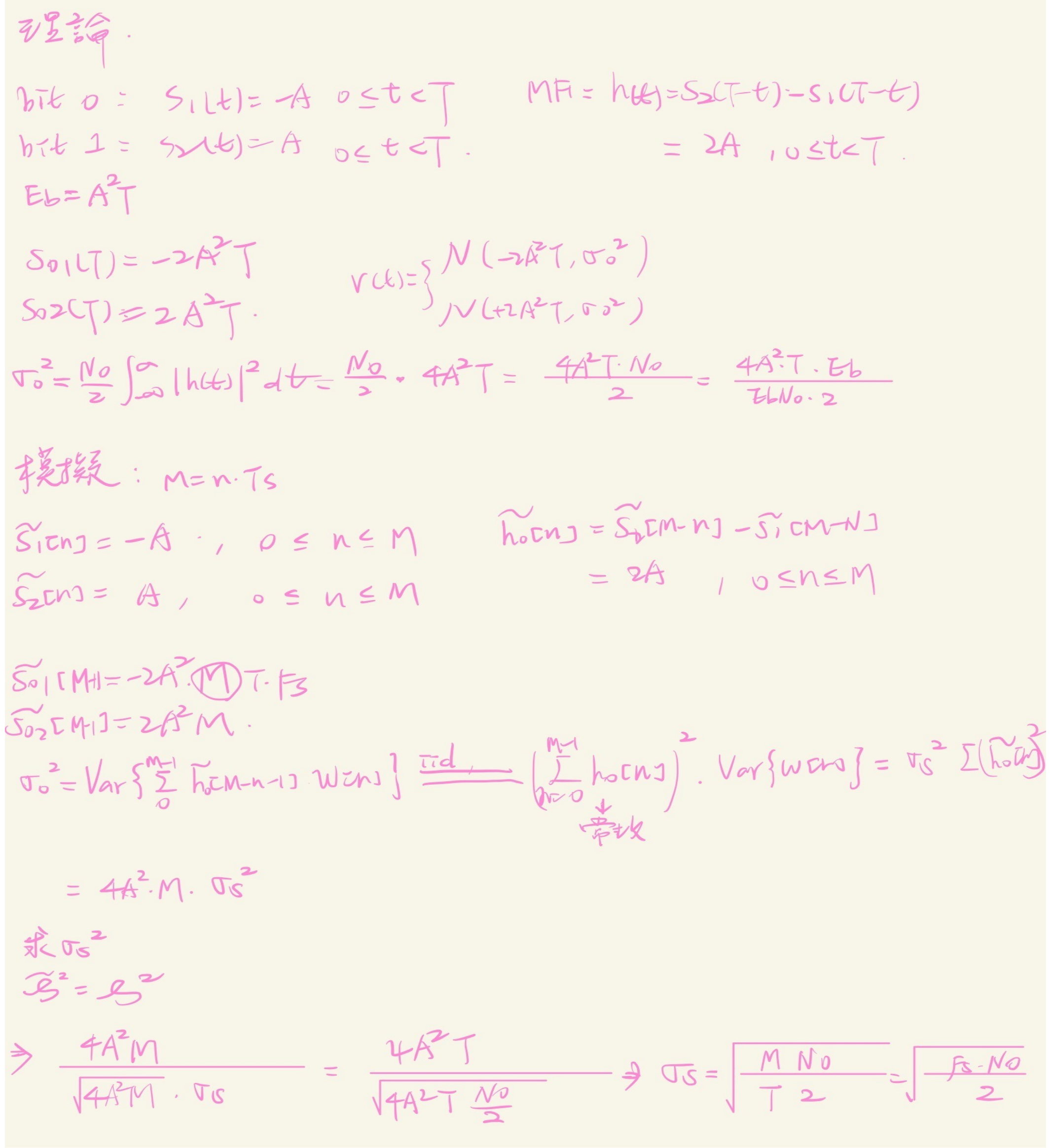
- BPSK推導

本次實驗與實驗七差別在於將輸入訊號由ook改成bpsk訊號輸入，詳細的理論公式推導以及輸入模擬如右圖所示:

推導時假設$s_1 s_2$振幅為$\pm A$、週期為T、取樣頻率為F_s

- 已知$Eb = \frac{A^2T+(-A^2)T}{2} = A^2T$
- 推導出理論值的$\sigma_0 = \sqrt{\frac{4A^2T*Eb}{EbN0*2}}$
- 模擬值的$\sigma_0 = \sqrt{4A^2M * {\sigma_s}^2}$、$M = T * F_s$
- 求出$\sigma_s = \sqrt{\frac{F_sN0}{2}}$

最後將公式所得導入Matlab進行測試



理論.

bit 0 : $s_1(t) = -A \quad 0 \le t < T$ $\quad$ MF = $h(t) = s_2(T-t) - s_1(T-t)$

bit 1 = $s_2(t) = A \quad 0 \le t < T$. $\quad = 2A \ , 0 \le t < T$.

$E_b = A^2T$

$S_{o1}(T) = -2A^2T$

$S_{o2}(T) = 2A^2T$.

$r(t) = \begin{cases} N(-2A^2T, \sigma_0^2) \\ N(+2A^2T, \sigma_0^2) \end{cases}$

$\sigma_0^2 = \frac{N_0}{2}\int_{-\infty}^{\infty} |h(t)|^2 dt = \frac{N_0}{2} \cdot 4A^2T = \frac{4A^2T \cdot N_0}{2} = \frac{4A^2 \cdot T \cdot E_b}{E_bN_0 \cdot 2}$

模擬 : $M = n \cdot T_s$

$\tilde{S}_1[n] = -A \ , \quad 0 \le n \le M$ $\quad$ $\tilde{h}_0[n] = \tilde{S}_2[M-n] - \tilde{S}_1[M-N]$

$\tilde{S}_2[n] = A \ , \quad 0 \le n \le M$ $\quad = 2A \ , \ 0 \le n \le M$

$\tilde{S}_{o1}[M-1] = -2A^2 \cdot \boxed{M} T \cdot F_s$

$\tilde{S}_{o2}[M-1] = 2A^2M$.

$\sigma_0^2 = Var\{\sum_0^{M-1} \tilde{h}_0[M-n-1] \cdot W[n]\} \xrightarrow{\text{iid}} (\underset{\substack{\downarrow \\ 常數}}{\sum_{n=0}^{M-1} h_0[n]})^2 \cdot Var\{W[n]\} = \sigma_s^2 \sum(\tilde{h}_0[n])^2$

$= 4A^2 \cdot M \cdot \sigma_s^2$

求 σ_s^2

$\tilde{S}^2 = S^2$

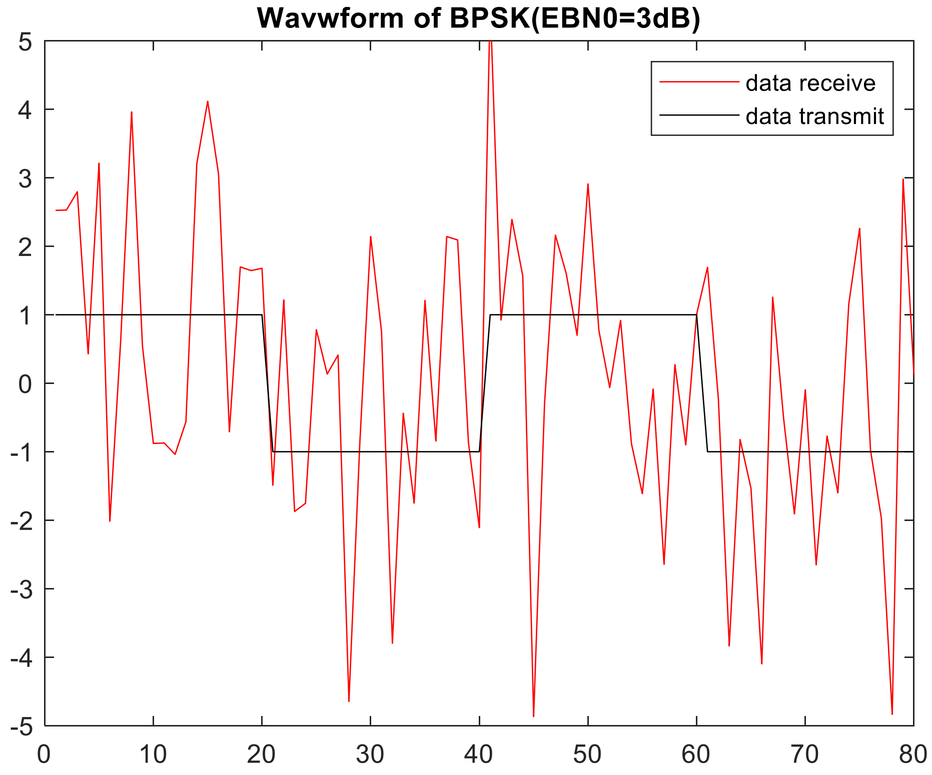
$\Rightarrow \frac{4A^2M}{\sqrt{4A^2M} \cdot \sigma_s} = \frac{4A^2T}{\sqrt{4A^2T \frac{N_0}{2}}} \Rightarrow \sigma_s = \sqrt{\frac{M}{T}\frac{N_0}{2}} = \sqrt{\frac{F_s \cdot N_0}{2}}$

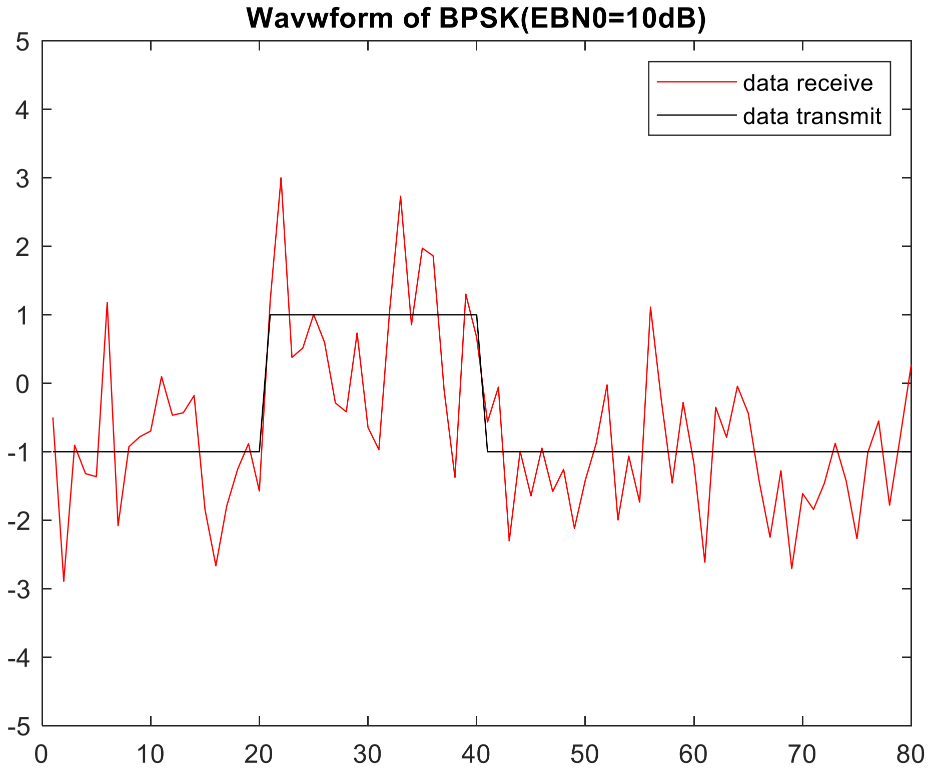
1. 修改範例程式，畫出 transmitted waveform 和 received waveform (duration = 4 symbol periods，$\frac{E_b}{N_0} = 10$ dB)，並比較之。

再畫 $\frac{E_b}{N_0} = 3$ dB 的圖，說明與 $\frac{E_b}{N_0} = 10$ dB 時之差異。

因為降低E_b，N_0保持一致，導致錯誤率上升$P_e = Q(\sqrt{SNR})$，因此$P_e(BER)$會變差。

下兩圖為模擬4bit在信躁比分別為3、10時的情況下，RX端接收到的訊號圖形分析。



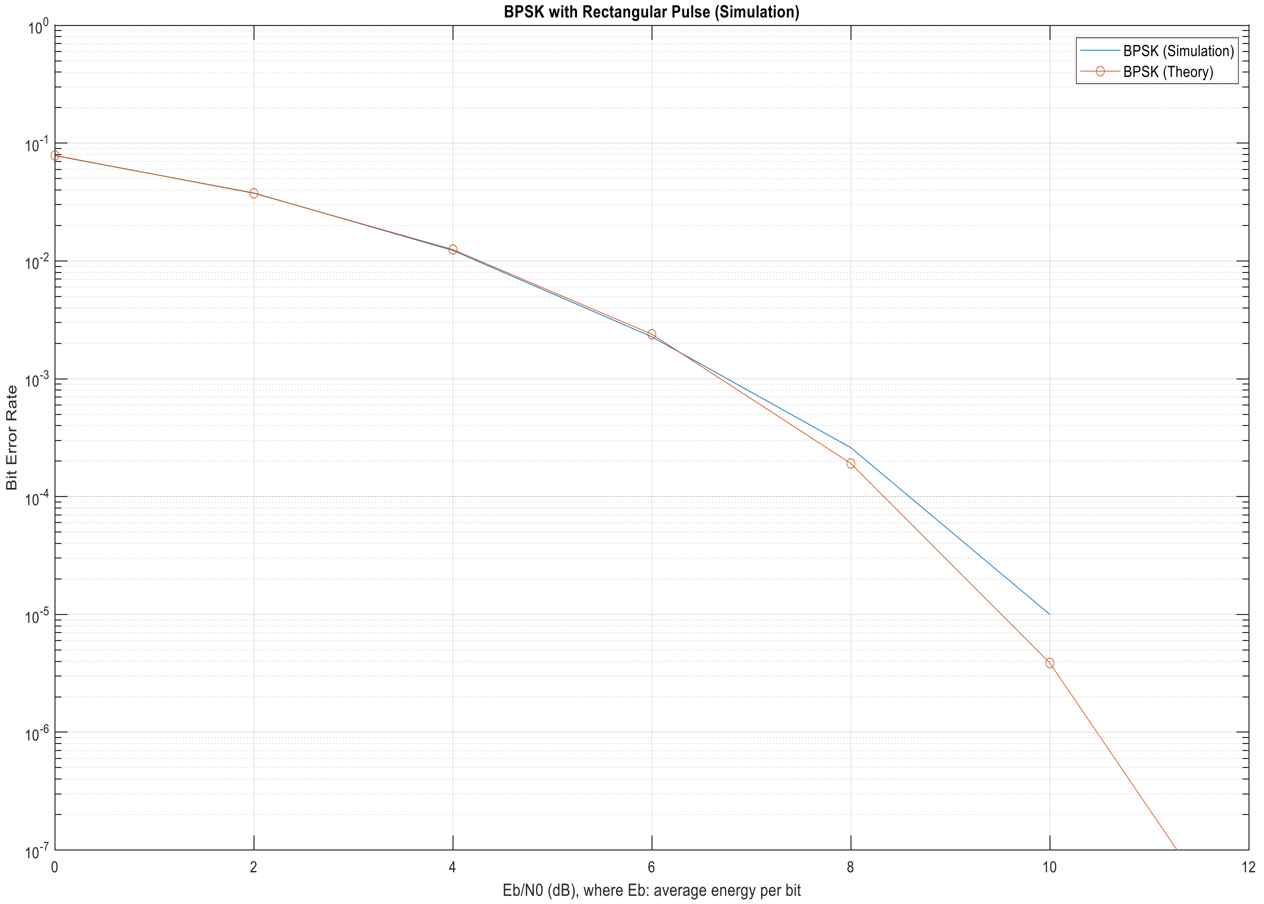


f. 畫出 OOK 之模擬錯誤率與 $\frac{E_b}{N_0}$ 之關係圖，並修改範例程式，另外畫出OOK 之理論錯誤率，比較與說明模擬結果與理論結果之差異。

已知由BPSK傳訊時，錯誤率如下

$$P_e = Q\left(\sqrt{2\frac{E_b}{N_0}}\right)= Q\sqrt{2SNR}$$

透過觀察得知，模擬訊號比對所產生的錯誤率與理論錯誤率在信躁比較高的時候差異較大，主要原因是當信躁比較高的時候，由於模擬數據的量不足以支撐該錯誤率的模擬。特別可以觀察到當信躁比為11.5時，由於我們的資料量只有10^5，模擬顯示錯誤率為0，但當資料量提高到10^9甚至以上，仍然會有錯誤率的產生，但囿於電腦及程式限制，無法模擬出如此高量的數據分析。



g. Conditional pdf at MF output：

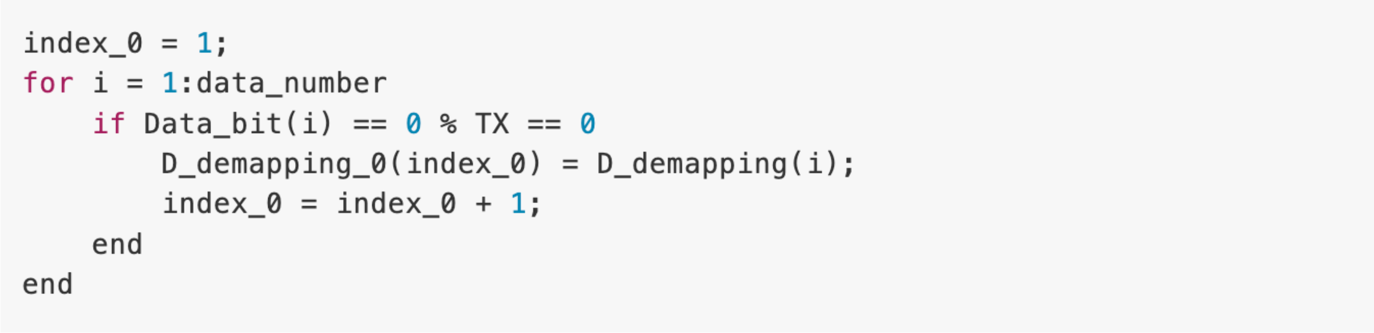
① 請修改範例程式，畫出

$$f\,(v(T)\mid \text{bit 0 sent})\ 與\ f\,(v(T)\mid \text{bit 1 sent})$$

於 $\frac{E_b}{N_0}=3$ dB 之理論值。利用 MF output 之模擬值（如變數 D_demapping，但你可能要適當 scale 此變數），畫出 $f\,(v(T)\mid \text{bit 1 sent})$ 之實驗值。

② 於 $\frac{E_b}{N_0}=10$ dB 時，重複①步驟，並說明 conditional pdf 之變化。

Hint: 使用 MATLAB 指令：histogram。



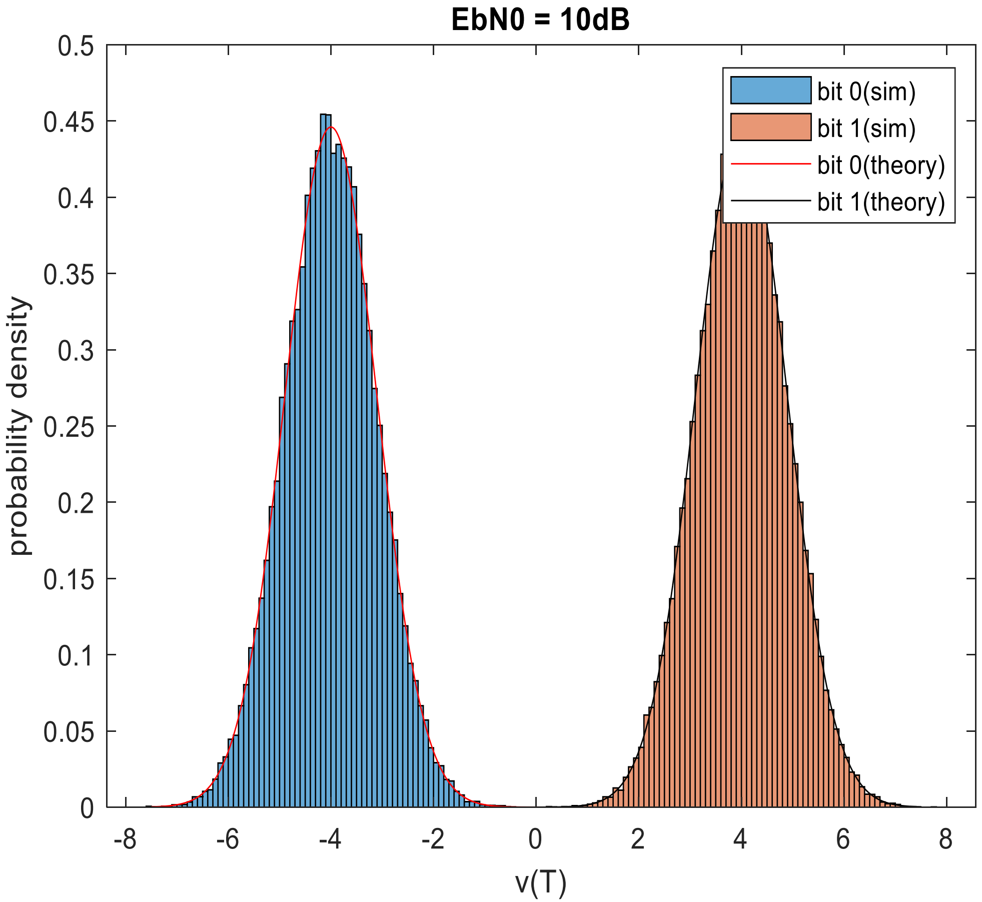
```
index_0 = 1;
for i = 1:data_number
    if Data_bit(i) == 0 % TX == 0
        D_demapping_0(index_0) = D_demapping(i);
        index_0 = index_0 + 1;
    end
end
```

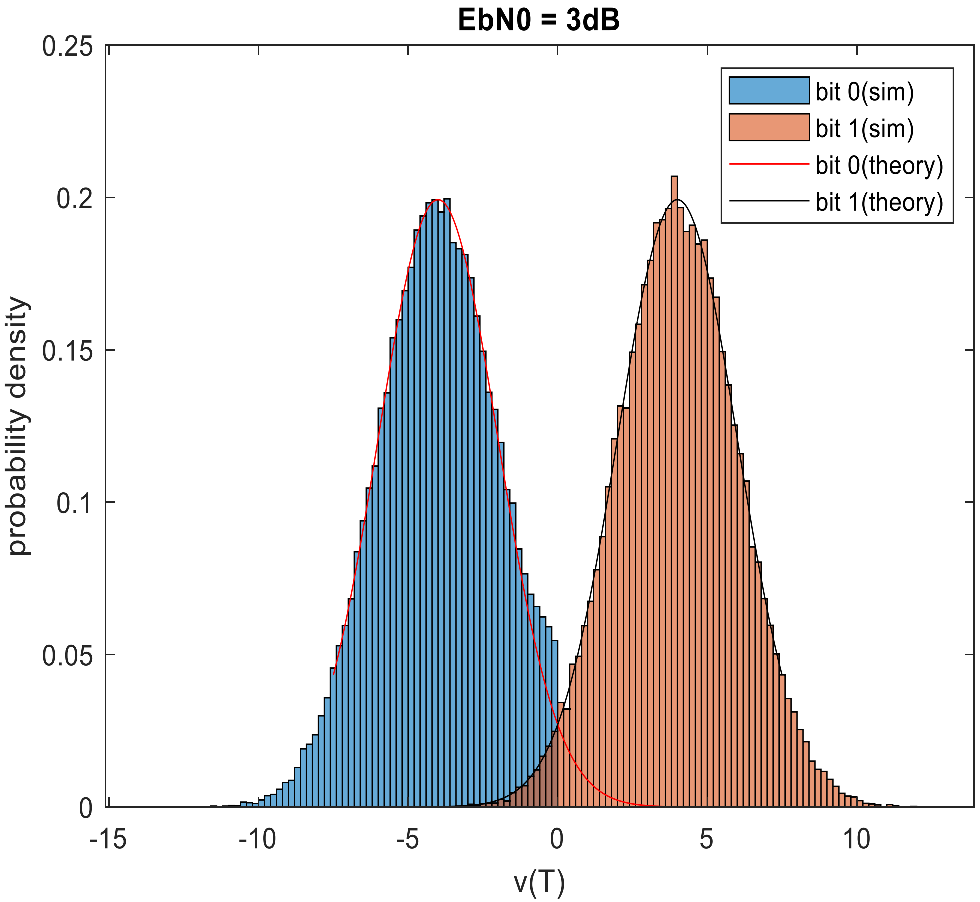
如果一開始傳送訊號是0，提取輸出訊號 $y(T)$ 並存入另一個自定義的array，代表**選取指定事前機率下的輸出數據**，傳送訊號為1亦是同理。

透過觀察下頁模擬程式結果可以發現傳送訊號為0、1時，呈現出在±4的位置的高斯分布，也符合我們先前所推導的$\pm 2A^2T=\pm 4$。

模擬所需的σ_S在先前亦有所推導，可參考前頁。

模擬結果如下：


EbN0 = 10dB
probability density
v(T)
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
-8
-6
-4
-2
0
2
4
6
8
bit 0(sim)
bit 1(sim)
bit 0(theory)
bit 1(theory)


EbN0 = 3dB
probability density
v(T)
0
0.05
0.1
0.15
0.2
0.25
-15
-10
-5
0
5
10
bit 0(sim)
bit 1(sim)
bit 0(theory)
bit 1(theory)

附件：Q1

```
% matched filter
% - antipodal keying using rectangular pulse (T=2)
% That is, when OFF, assuming A=-1, thus E1 = 2
% when ON: assuming A=1, thus E2 = 2
% The "average bit energy" (Eb) = (E1+E2)/2 = 2 %
clear all; close all;
%% parameters
data_number = 10^5; % # of bits
EbN0dB_vec = [0:2:10 11.5]; % Eb/N0 in dB
Fs=10; % sampling frequency (used to generate received samples)
T=2;
%% transmitter
Data_bit=(rand(1,data_number) > 0.5 ); % random bits
p1=ones(1,T*Fs);% discrete-time rectangular pulse that represents one symbol
Data_pulse_array=2*((p1.')*Data_bit)-1;
Data_pulse=reshape(Data_pulse_array,1,length(p1)*data_number);
[a b] = size(Data_pulse);
EbN0dB = 10;
EbN0 = 10^(EbN0dB/10); % EbN0 is now in linear scale
sgma = sqrt( 10/EbN0 );
noise = normrnd(0, sgma, a, b ) ;
Data_receive=Data_pulse+noise; % received samples
```

```
%% receiver
D_filtered=conv(Data_receive,2*p1); % MF output
D_demapping=D_filtered(20:20:end)/Fs; % sampling at symbol rate
% decsion based on D_demapping
D_demap_N = (D_demapping > 0); % >0: 1; <=0: 0
% BER computation
Error_num=sum(xor(D_demap_N,Data_bit));
BER = Error_num/data_number;
fprintf('EbN0 in dB is %g\n',EbN0dB);
fprintf('Bit error rate is %g\n',BER);
%% cut the sequence
n = 1:80;
for i = 1:80;
Data_receive_cut(i) = Data_receive(i);
Data_bit_cut(i) = Data_pulse(i);
end
%% figure
figure;
plot(n,Data_receive_cut,'r');hold on
plot(n,Data_bit_cut,'k');
axis([0 80 -5 5]);
legend('data receive','data transmit');
```

附件：Q1f

```
% matched filter
% - antipodal keying using rectangular pulse (T=2)
% That is, when OFF, assuming A=-1, thus E1 = 2
% when ON: assuming A=1, thus E2 = 2
% The "average bit energy" (Eb) = (E1+E2)/2 = 2 %
clear all; close all;
%% parameters
data_number = 10^5; % # of bits
EbN0dB_vec = [0:2:10 11.5]; % Eb/N0 in dB
Fs=10; % sampling frequency (used to generate received samples)
T=2;
%% transmitter
Data_bit=(rand(1,data_number) > 0.5 ); % random bits
p1=ones(1,T*Fs);% discrete-time rectangular pulse that represents one symbol
Data_pulse_array=2*((p1.')*Data_bit)-1;
Data_pulse=reshape(Data_pulse_array,1,length(p1)*data_number);
```

```
for kk=1:length(EbN0dB_vec)
%% AWGN channel
[a b] = size(Data_pulse);
EbN0dB = EbN0dB_vec(kk);
EbN0 = 10^(EbN0dB/10); % EbN0 is now in linear scale
sgma = sqrt( 10/EbN0 );
noise = normrnd(0, sgma, a, b ) ;
Data_receive=Data_pulse+noise; % received samples
%% receiver
D_filtered=conv(Data_receive,2*p1); % MF output
D_demapping=D_filtered(20:20:end)/Fs; % sampling at symbol rate
% decsion based on D_demapping
D_demap_N = (D_demapping > 0); % >0: 1; <=0: 0
% BER computation
Error_num=sum(xor(D_demap_N,Data_bit));
BER(kk) = Error_num/data_number;
fprintf('EbN0 in dB is %g\n',EbN0dB);
fprintf('Bit error rate is %g\n',BER);
%% theory BER for BPSK
ber_theory(kk)=qfunc(sqrt(2*EbN0));
end

%% generate plot
figure;
semilogy(EbN0dB_vec, BER, EbN0dB_vec,ber_theory,'o-');
hold on;
xlabel('Eb/N0 (dB), where Eb: average energy per bit');
ylabel('Bit Error Rate')
legend('BPSK (Simulation)', 'BPSK (Theory)','FontSize',10);
grid
axis([0 12 10^-7 1])
title('BPSK with Rectangular Pulse (Simulation)')
```

附件：Q1g

```
% matched filter
% - antipodal keying using rectangular pulse (T=2)
% That is, when OFF, assuming A=-1, thus E1 = 2
% when ON: assuming A=1, thus E2 = 2
% The "average bit energy" (Eb) = (E1+E2)/2 = 2 %
clear all; close all;
%% parameters
data_number = 10^5; % # of bits
Fs=10; % sampling frequency (used to generate received samples)
T=2;
%% transmitter
Data_bit=(rand(1,data_number) > 0.5 ); % random bits
p1=ones(1,T*Fs);% discrete-time rectangular pulse that represents one symbol
Data_pulse_array=2*((p1.')*Data_bit)-1;
Data_pulse=reshape(Data_pulse_array,1,length(p1)*data_number);
[a b] = size(Data_pulse);
EbN0dB = 10;
EbN0 = 10^(EbN0dB/10); % EbN0 is now in linear scale
sgma = sqrt( 10/EbN0 );
noise = normrnd(0, sgma, a, b ) ;
Data_receive=Data_pulse+noise; % received samples
%% receiver
D_filtered=conv(Data_receive,2*p1); % MF output
D_demapping=D_filtered(20:20:end)/Fs; % sampling at symbol rate
% decsion based on D_demapping
D_demap_N = (D_demapping > 0); % >0: 1; <=0: 0v
% BER computation
Error_num=sum(xor(D_demap_N,Data_bit));
BER = Error_num/data_number;
fprintf('EbN0 in dB is %g\n',EbN0dB);
fprintf('Bit error rate is %g\n',BER);
index_0=1;
for i = 1:data_number
if D_demap_N(i) == 0
D_demap_0(index_0)=D_demapping(i);
index_0 = index_0+1;
end
end
index_1 = 1;
for i = 1:data_number
if Data_bit(i) == 1
D_demap_1(index_1) = D_demapping(i);
index_1 = index_1 + 1;
end
end
sgma_theory = sqrt((4*2*2)/(EbN0*2));
%% generate plots
histogram(D_demap_0,'Normalization','pdf');
hold on
histogram(D_demap_1,'Normalization','pdf')
xaxis = -7.5:0.01:7.5;
plot(xaxis,normpdf(xaxis,-4,sgma_theory),'r')
hold on
plot(xaxis,normpdf(xaxis,4,sgma_theory),'k')
legend('bit 0(sim)','bit 1(sim)','bit 0(theory)','bit 1(theory)');
xlabel('v(T)');ylabel('probability density')
title('EbN0 = 10dB')
```