實驗7：匹配濾波器(Matched Filter)模擬實驗結報

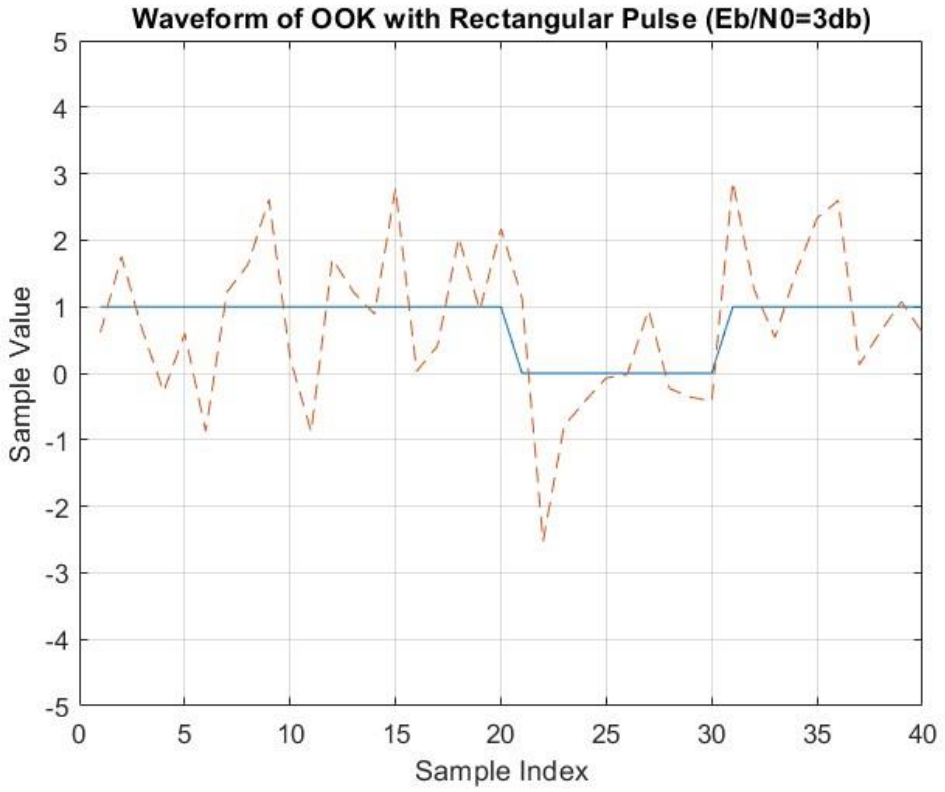
第二組

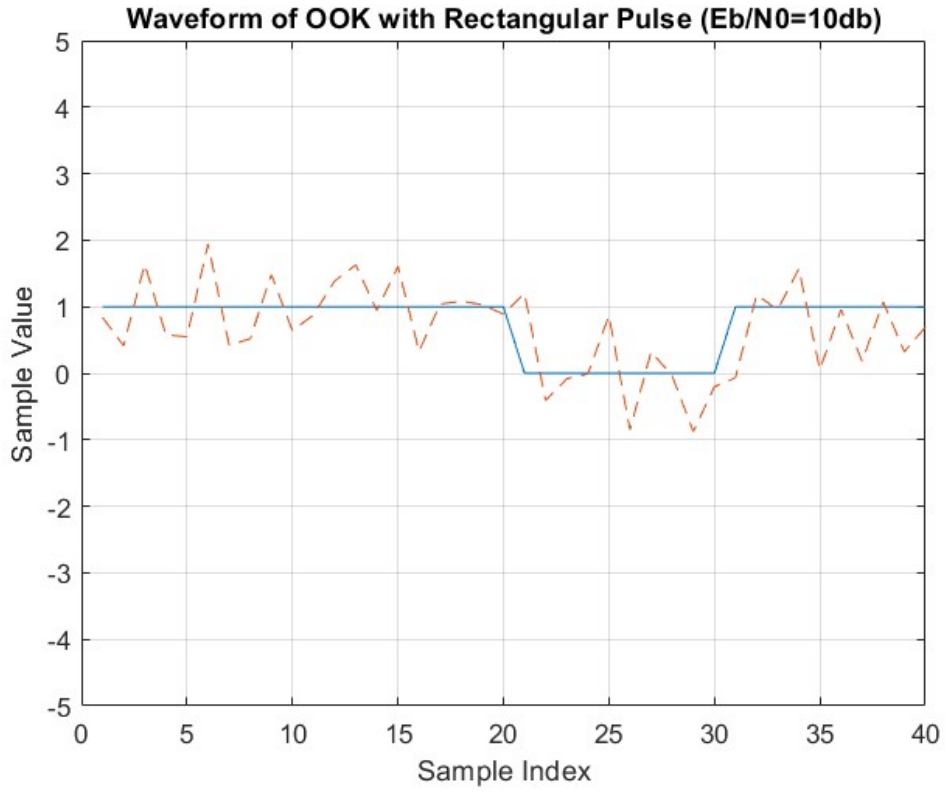
物理四 C24074031 劉嘉峰

系統三 F14081046 周呈陽

實驗結果討論1.

訊雜比越大，表示訊號功率在有雜訊的情況下較為顯著，較能看出原本訊號的波形





實驗結果討論1.-程式碼

```
clear all;
close all;
clc;
%%parameter
data_number = 10^5; % # of bits
EbN0dB_vec = [3 10] ;% Eb/N0 in dB
Fs=10; % sampling frequency (used to generate received samples
%% transmitter
Data_bit=(rand(1,data_number) > 0.5 ); % random bits
p1=ones(1,Fs);% discrete-time rectangular pulse that represents on
Data_pulse_array=(p1.')*Data_bit;
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( 0.5/EbN0/2*10 );
noise = normrnd(0, sgma, a, b ) ;
Data_receive=Data_pulse+noise ;% received samples
%% receiver
D_filtered=conv(Data_receive,p1);% MF output
index=1:length(Data_receive);
%plot(index(1:40),Data_pulse(1:40))
%ylim([-5 5])
%plot(index(1:40),Data_receive(1:40))
```

```
%接續左圖
x=index(1:40);
tx=Data_pulse(1:40);
rx=Data_receive(1:40);
figure(kk)
plot(x,tx)
hold on
plot(x,rx,'--')
grid on
ylim([-5 5])
if kk==1
    title('Waveform of OOK with Rectangular Pulse (Eb/N0=3db)')
end
if kk==2
    title('Waveform of OOK with Rectangular Pulse (Eb/N0=10db)')
end
ylabel('Sample Value')
xlabel('Sample Index')
D_demapping=D_filtered(10:10:end)/10;% sampling at symbol rate
% decsion based on D_demapping
D_demap_N = (D_demapping > 0.5); % >0.5: 1; <=0.5: 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);
end
```

```
%接續上圖
%% generate plots
figure;
semilogy(EbN0dB_vec, BER,'o-');
xlabel('Eb/N0 (dB), where Eb: average energy per bit');
ylabel('Bit Error Rate')
legend('OOK (Simulation)');
grid
axis([0 12 1e-4 1e0])
title('OOK with Rectangular Pulse (Simulation)')
```

實驗結果討論2.a

a. 解釋為何（p. 16）

```
Data_bit=(rand(1,data_number) > 0.5 );
```

可產生 random bits?
此行所產生之 bit 0 和 bit 1，其發生機率理論上應為何？（說明之。）
實際於模擬中 bit 0 和 bit 1 發生的機率為何？是否符合理論預測？

1. 當data_number=N，rand(1,N)會產生1xN的矩陣，其值分布0~1之間，當rand(1,N)>0.5，回傳值為1，表示判斷式為True，反之回傳值為0，判斷為False，因此會產生1xN的矩陣，其值只包含0或1。
2. bit0和bit1的機率理論上會都是0.5
3. 在模擬中，bit1的機率為0.5019，bit0的機率為0.4981；與理論值差不多

實驗結果討論2.a-程式碼

```
clear all;
close all;
clc;
%%parameter
data_number = 10^5; % # of bits
EbN0dB_vec = [3 10] ;% Eb/N0 in dB
Fs=10; % sampling frequency (used to generate received samples
%% transmitter
Data_bit=(rand(1,data_number) > 0.5 ); % random bits
a=0;
b=0;
for k=1:data_number
    if Data_bit(k)==1
        a=a+1;
    else
        b=b+1;
    end
end
disp('bit1的機率'),a/data_number
```

```
bit1的機率
ans = 0.5019
```

```
disp('bit0的機率'),b/data_number
```

```
bit0的機率
ans = 0.4981
```

實驗結果討論2.b

b. 說明（p. 16）

Data_pulse=reshape(Data_pulse_array,1,length(p1)*data_number);

這行程式碼的作用為何？

Reshape(A,M,N)：A為原矩陣,m為列數,n為行數，將A轉換MxN的矩陣

Data_pulse=reshape(Data_pulse_array,1,length(p1)*data_number);

Data_pulse_array為10x100000，列數為1，行數為1000000，故將Data_pulse_array轉換成1x1000000的矩陣Data_pulse，

實驗結果討論2.c

c. 沒有雜訊時，matched filter output（即 p. 11 圖中之 $v(T)$）之理論值應為何？（以數學推理之）。而用此程式模擬時（請先不加雜訊），變數 D_demapping（於 p. 17 程式中代表 matched filter output）的值為何？理論值和模擬值相同嗎？解釋之。

1. $s_1(t) = 0, \quad 0 \le t \le 1$
   $s_2(t) = 1, \quad 0 \le t \le 1$ $\Rightarrow$ $y(t) \rightarrow \otimes \rightarrow \int_0^1 dt \rightarrow v(1)$

   $\otimes$ 上方輸入：$s_2(t) - s_1(t)$

   if $y(t) = s_1(t) \rightarrow v(1) = \int_0^1 s_1(t)[s_2(t) - s_1(t)]dt = "0"$

   if $y(t) = s_2(t) \rightarrow v(1) = \int_0^1 s_2(t)[s_2(t) - s_1(t)]dt = "1"$

2. 因為模擬值一個binary的1x100000的矩陣，其值只有0或1，與理論值相同。

```
D_demapping = 1×100000
     1     1     0     1     1     0     0     1     1     0     0     0     0     1     1     1     0     0 ...
```

實驗結果討論2.d

d. 解釋程式碼（p. 17）

```
D_demapping=D_filtered(10:10:end)/10;
```

為何需取 10:10:end ?

連續時間的符號區間為1秒，但電腦是用離散時間模擬。最一開始假設的Fs = 10 ，表示我們採用10個sample點充當1個連續時間的符號區間。在demapping時，需在每個符號結束處取樣，因此需要每10個sample取樣一次。

實驗結果討論2.e

e. p. 17 程式碼中，sgma^2 代表 noise sample 之（平均）功率。
解釋為何

```
sgma = sqrt( 0.5/EbN0/2*10 );
```

如此設定可達到所定義之 $\frac{E_b}{N_0}$ 值。

$EbN0dB = EbN0dB\_vec(kk) \Rightarrow$ 分別為 0, 2, 4, 6, 8, 10, 11.5 (dB)

$(SNR)_{dB} = 10\log_{10}(SNR) \Rightarrow SNR = 10^{\frac{(SNR)_{dB}}{10}}$

$EbN0 = 10^{\frac{EbN0dB}{10}} \Rightarrow$ 故 $EbN0 = SNR = \frac{Eb'}{N0'}$ (10進位)

$\Rightarrow N0' = \frac{Eb'}{EbN0}$

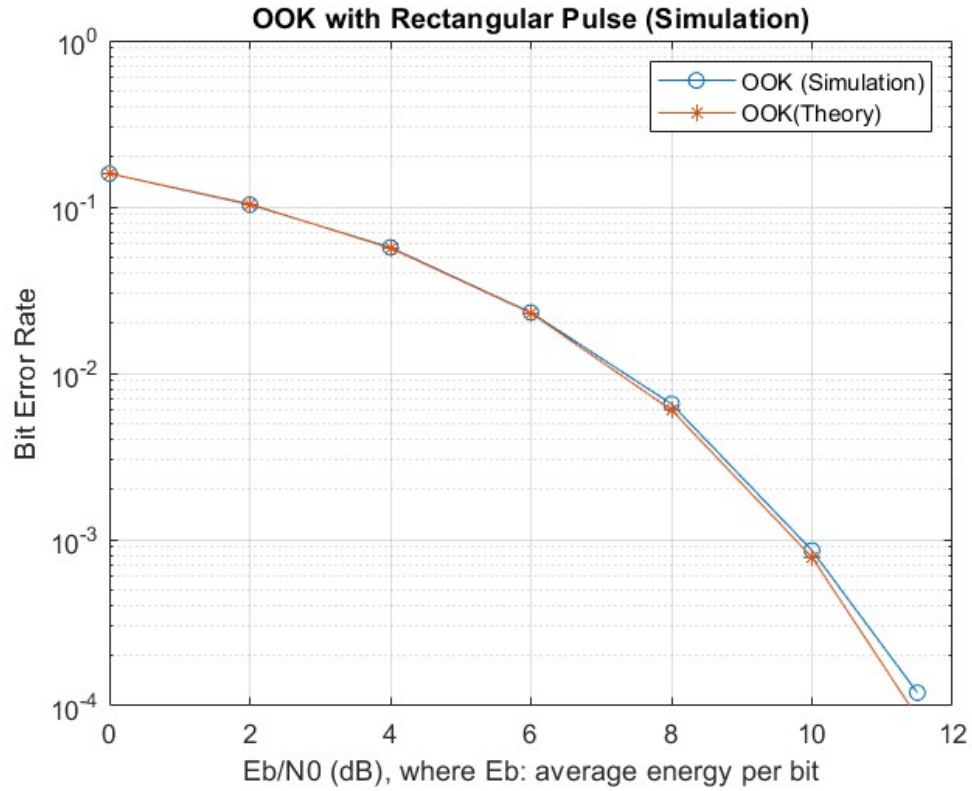
$\sigma_N^2 = \frac{N0'T}{2} = \frac{Eb' \cdot T}{2 \cdot EbN0} = \frac{5}{2\,EbN0}$ ，$\left(Eb' = \frac{E1+E2}{2} = 5,\ T = 1\right)$

$= \frac{0.5 \cdot 10^5}{EbN0 \cdot 2} = \frac{5}{2\,EbN0}$ ↗ 相等

實驗結果討論2.f

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。

BER算的是平均來看每個位元的錯誤率。由於雜訊是隨機產生，當位元數大，平均來說的位元錯誤率會與理論值較相近;位元數小，則與理論值誤差較大。(實驗採用的data_number為10^5)



實驗結果討論2.f -程式碼

```
clear all;
close all;
clc;
%%parameter
data_number = 10^5; % # of bits
EbN0dB_vec = [0 2 4 6 8 10 11.5] ;% Eb/N0 in dB
Fs=10; % sampling frequency (used to generate received samples
%% transmitter
Data_bit=(rand(1,data_number) > 0.5 ); % random bits
a=0;
b=0;
for k=1:data_number
    if Data_bit(k)==1
        a=a+1;
    else
        b=b+1;
    end
end
%disp('bit1的機率'),a/data_number
%disp('bit0的機率'),b/data_number


p1=ones(1,Fs);% discrete-time rectangular pulse that represents on
Data_pulse_array=(p1.')*Data_bit;
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( 0.5/EbN0/2*10 );
noise = normrnd(0, sgma, a, b ) ;
Data_receive=Data_pulse+noise;% received samples
%Data_receive=Data_pulse;
%% receiver
D_filtered=conv(Data_receive,p1);% MF output
index=1:length(Data_receive);
%plot(index(1:40),Data_pulse(1:40))
%ylim([-5 5])
%plot(index(1:40),Data_receive(1:40))



D_demapping=D_filtered(10:10:end)/10;% sampling at symbol rate
% decsion based on D_demapping
D_demap_N = (D_demapping > 0.5); % >0.5: 1; <=0.5: 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);
end
```

```
%% generate plots
figure;
semilogy(EbN0dB_vec, BER,'o-');
hold
xlabel('Eb/N0 (dB), where Eb: average energy per bit');
ylabel('Bit Error Rate')
axis([0 12 1e-4 1e0])
title('OOK with Rectangular Pulse (Simulation)')


ebnodB = [0 2 4 6 8 10 11.5];
ebno = 10.^(ebnodB/10);
Pb = qfunc(sqrt(ebno));
semilogy(ebnodB,Pb,'*-')
legend('OOK (Simulation)','OOK(Theory)')
grid on
```

實驗結果討論2.g

g. Conditional pdf at MF output：

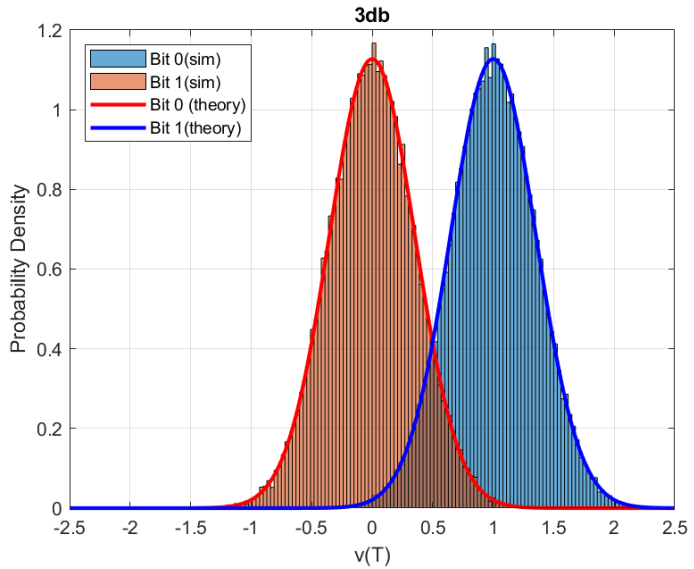
① 請修改範例程式，畫出

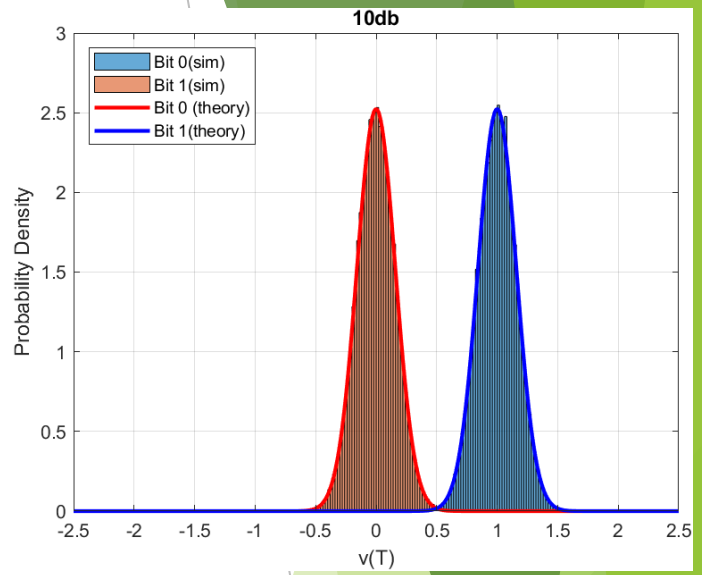
$$f(v(T) \mid \text{bit 0 sent}) \text{ 與 } f(v(T) \mid \text{bit 1 sent})$$

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。





1. 用離散模擬連續的情形，需要調整sigma才能呈現理論的情況

$$\tilde{\sigma}_0^2 = 10 \cdot \tilde{\sigma}^2_{sample}$$

2. 當訊雜比變高時，則雜訊相對於訊號的影響變小，接收到的訊號更加集中在期望值附近，因此位元0、1的分布交疊區域變少，更容易正確辨別位元，所以錯誤率下降

實驗結果討論2.g-程式碼

```
close all; clc;
%%parameter
data_number = 10^5; % # of bits
EbN0dB_vec = 3 ;% Eb/N0 in dB
Fs=10; % sampling frequency (used to generate received samples
%% transmitter
Data_bit=(rand(1,data_number) > 0.5 ); % random bits
p1=ones(1,Fs);% discrete-time rectangular pulse that represents on
Data_pulse_array=(p1.')*Data_bit;
Data_pulse=reshape(Data_pulse_array,1,length(p1)*data_number);
%% AWGN channel
EbN0dB = EbN0dB_vec;
EbN0 = 10^(EbN0dB/10); % EbN0 is now in linear scale
sgma = sqrt( 0.5/EbN0/2*10 );
num1=0;
num0=0;
for eig=1:1000000
    if Data_pulse(eig)==1
        num1=num1+1;
    else
        num0=num0+1;
    end
end
bit1=ones(1,num1)+normrnd(0, sgma, a, num1 ) ;
bit0=zeros(1,num0)+normrnd(0, sgma, a, num0 );
b1_filtered=conv(bit1,p1);% MF output
b1_demapping=b1_filtered(10:10:end)/10;
b0_filtered=conv(bit0,p1);% MF output
b0_demapping=b0_filtered(10:10:end)/10;
b1_h=histogram(b1_demapping,'Normalization','pdf');
hold on
b0_h=histogram(b0_demapping,'Normalization','pdf');
xlim([-2.5 2.5])
hold on
x = -2.5:.01:2.5;
ave0 = normpdf(x,0,sgma/sqrt(10));
ave1 = normpdf(x,1,sgma/sqrt(10));
plot(x,ave0,'r','LineWidth',2)
hold on
plot(x,ave1,'b','LineWidth',2)
legend('Bit 0(sim)','Bit 1(sim)','Bit 0 (theory)','Bit 1(theory)','Location','northwest')
grid on
title('3db')
xlabel('v(T)')
ylabel('Probability Density')
```

```
close all;clc;
%%parameter
data_number = 10^5; % # of bits
EbN0dB_vec = 10 ;% Eb/N0 in dB
Fs=10; % sampling frequency (used to generate received samples
%% transmitter
Data_bit=(rand(1,data_number) > 0.5 ); % random bits
p1=ones(1,Fs);% discrete-time rectangular pulse that represents on
Data_pulse_array=(p1.')*Data_bit;
Data_pulse=reshape(Data_pulse_array,1,length(p1)*data_number);
%% AWGN channel
EbN0dB = EbN0dB_vec;
EbN0 = 10^(EbN0dB/10); % EbN0 is now in linear scale
sgma = sqrt( 0.5/EbN0/2*10 );
num1=0;
num0=0;

for eig=1:1000000
    if Data_pulse(eig)==1
        num1=num1+1;
    else
        num0=num0+1;
    end
end
bit1=ones(1,num1)+normrnd(0, sgma, a, num1 ) ;
bit0=zeros(1,num0)+normrnd(0, sgma, a, num0 );
b1_filtered=conv(bit1,p1);% MF output
b1_demapping=b1_filtered(10:10:end)/10;
b0_filtered=conv(bit0,p1);% MF output
b0_demapping=b0_filtered(10:10:end)/10;
b1_h=histogram(b1_demapping,'Normalization','pdf');
hold on
b0_h=histogram(b0_demapping,'Normalization','pdf');
xlim([-2.5 2.5])
hold on
x = -2.5:.01:2.5;
ave0 = normpdf(x,0,sgma/sqrt(10));
ave1 = normpdf(x,1,sgma/sqrt(10));
plot(x,ave0,'r','LineWidth',2)
hold on
plot(x,ave1,'b','LineWidth',2)
legend('Bit 0(sim)','Bit 1(sim)','Bit 0 (theory)','Bit 1(theory)','Location','northwest')
grid on
title('10db')
xlabel('v(T)')
ylabel('Probability Density')
```