

行動通訊

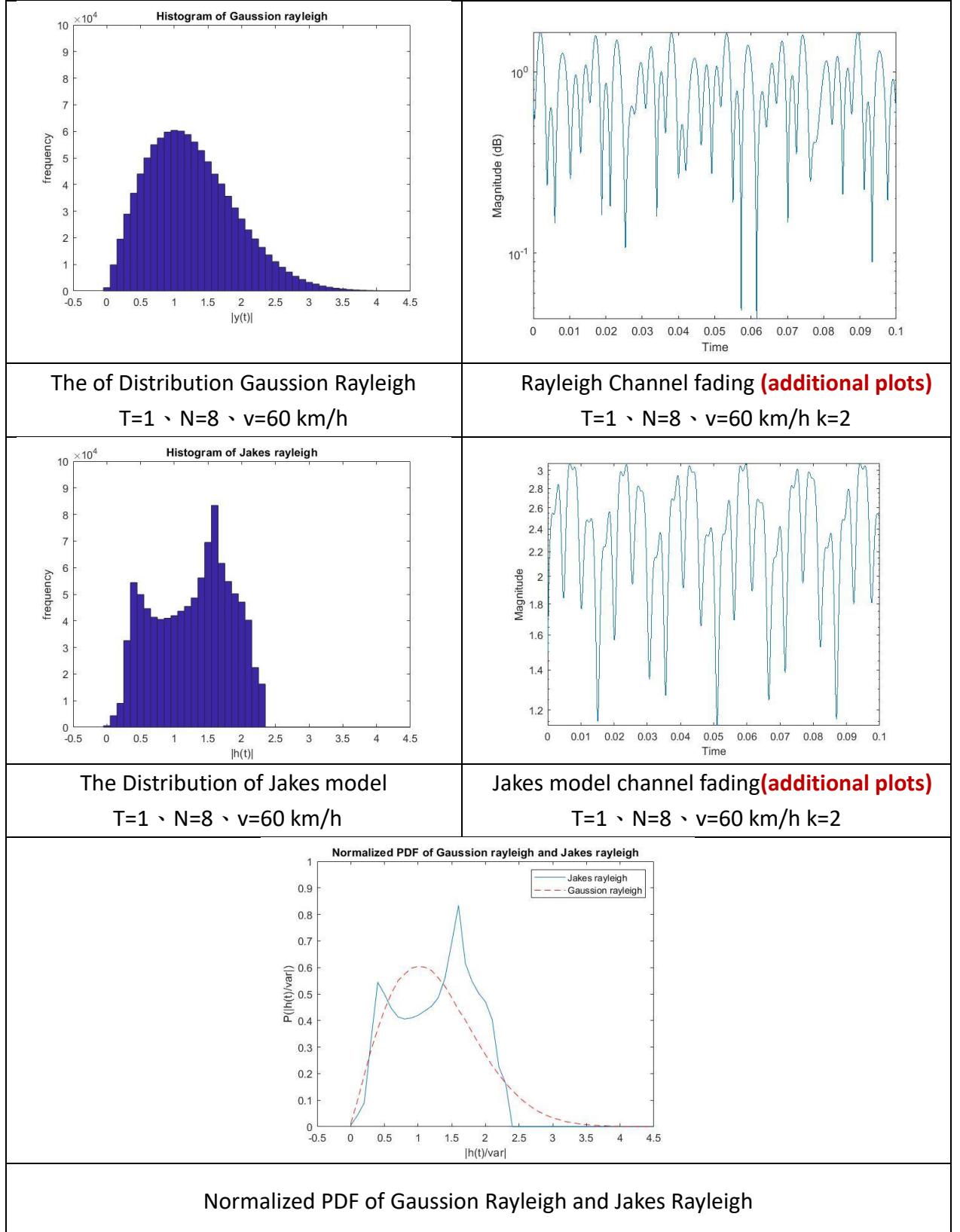
Project 1-Baseband Representation

系級：通訊四

學號：106503008

姓名：蔡嘉倫

(a)(i) In the first part of the project, we would like to analyze the characteristics of the channel model by comparing the theoretical PDF. Assume the carrier frequency is 3.5 GHz and the Let $\Delta t = 1\mu\text{s}$ is the sampling interval to generate Rayleigh fading signal.



討論：

1. Gauussion Rayleigh

首先說到瑞利分布的情況，由於是一個隨機二維向量的兩個分量呈獨立的、有著相同的方差、均值為 0 的常態分布。

因此會呈現一鐘形分布，即為此曲線的樣子。

2. Jakes model

在題目所給的 baseband 表示中，phase $\alpha_m = \frac{2\pi m}{N}$, $m=0,1,2...,N-1$

而在(i)小題中，我們所求的為 $N=8$ ，因此對於 scatters 的數量過少了導致所要討論的分散角數量也不夠，則再觀察 Gauussion Rayleigh 與 Jakes model 中，大概在 $|y(t)|=0.5$ 與 $|h(t)|=0.5$ 時有較為相近的數值而其他部分都相差過大，需要再經過取 N 的數值更大一些，才能使 Jakes model 的柱狀圖變得更加近似於雷利分布。

3. Channel fading

對於頻道衰落的快慢與發射端和接收端的相對運動速度的大小有關。相對運動導致接收訊號的都卜勒頻移。而都普勒頻移的導數即為同調時間。

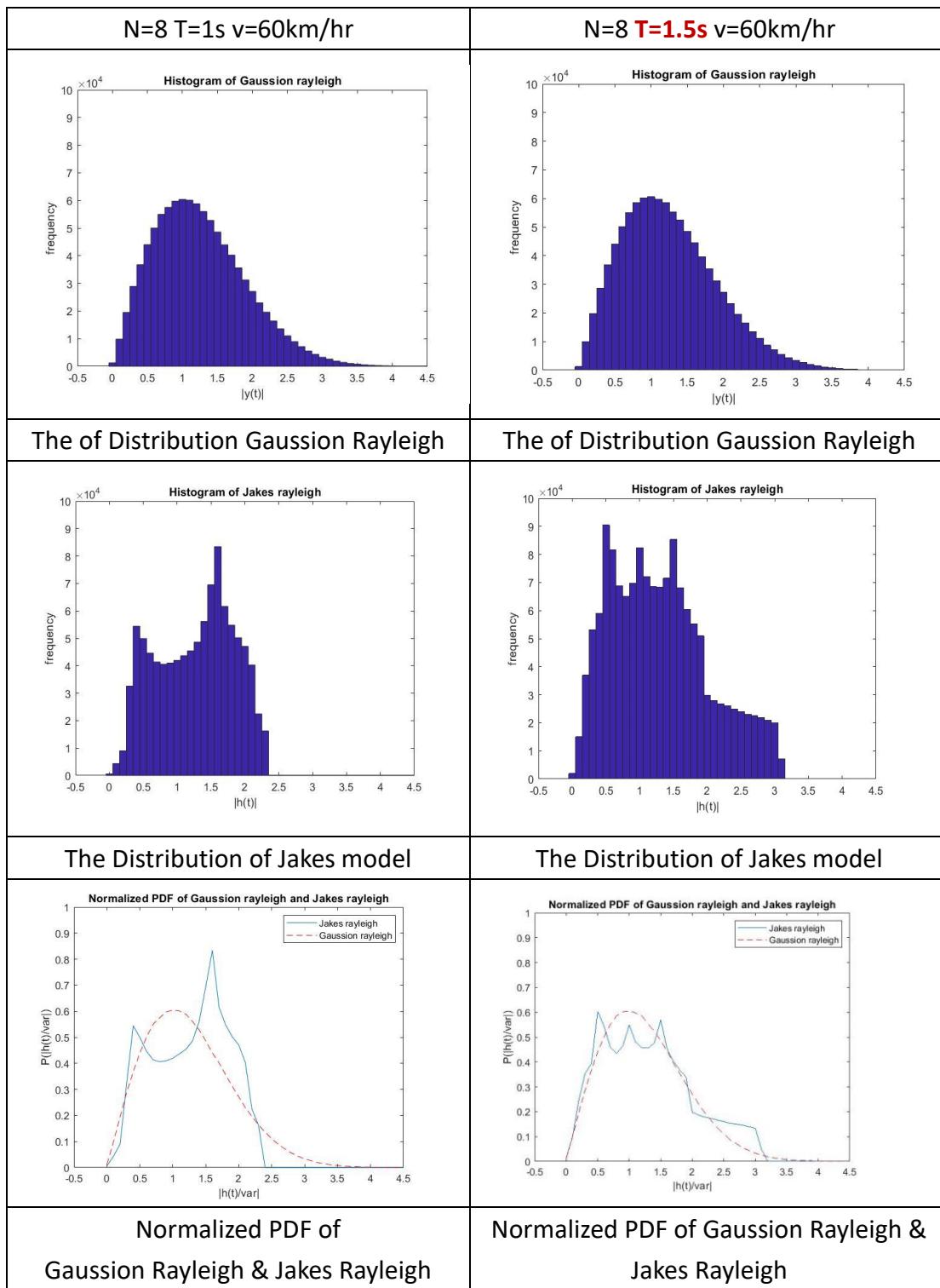
而在這兩個 channel 中的比較能發現若是使用 Gauussion Rayleigh 其衰減的能量在 0.06 秒的區間會掉至 10 倍左右。

而在 Jake model 中衰減的程度不多，我認為的原因是因為一開始 Jake Model 的設定，由於 Jake model 原本就是假設接收端能平均的接收來自四面八方的正弦波，因此能取得的數據可能較不符合實際狀況，因此可能獲得的數據會是比較理想化的。

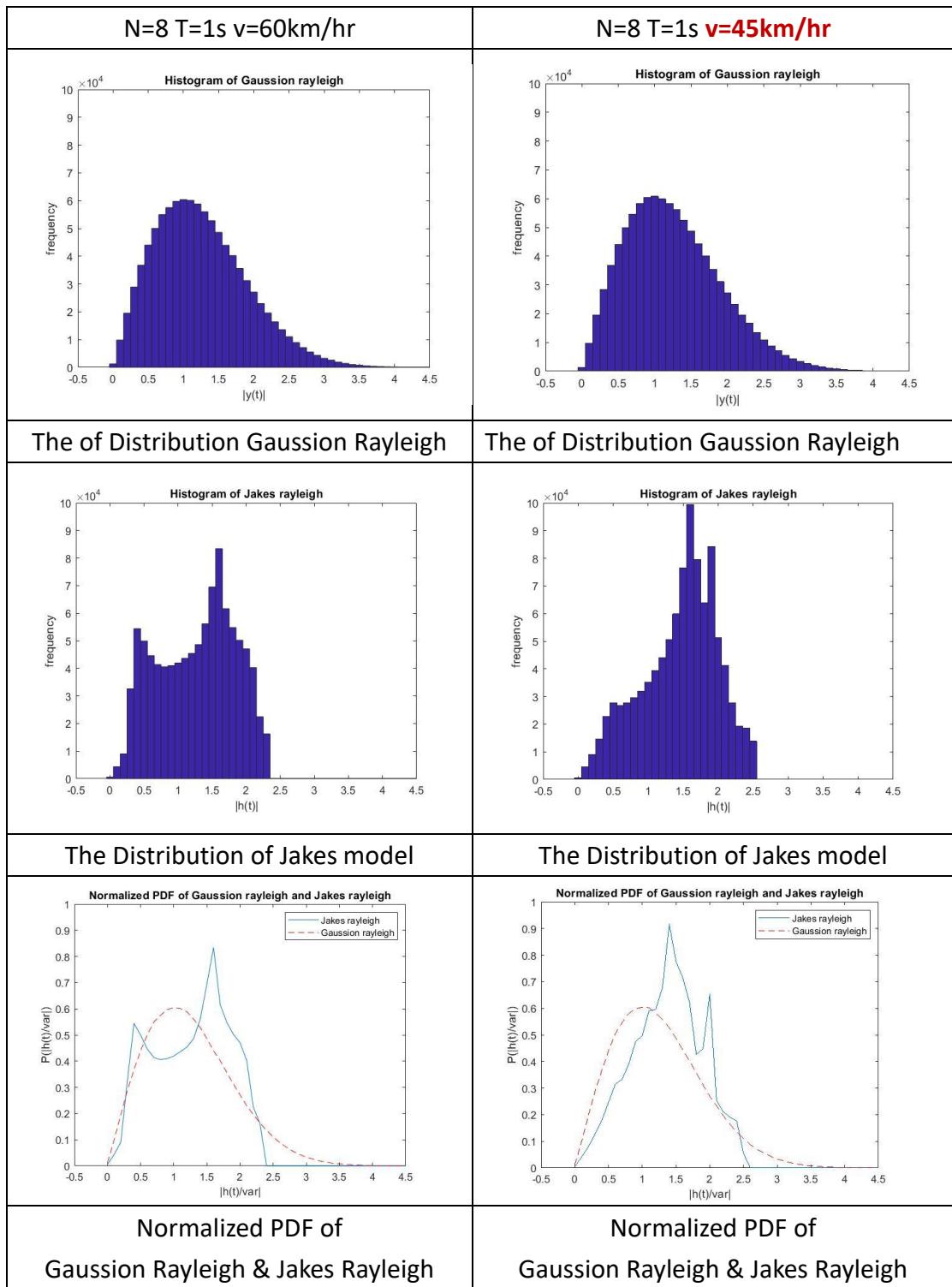
也能透過改變參數來改變衰減週期的長短，然後來觀察其他衰減的變化。

(ii)

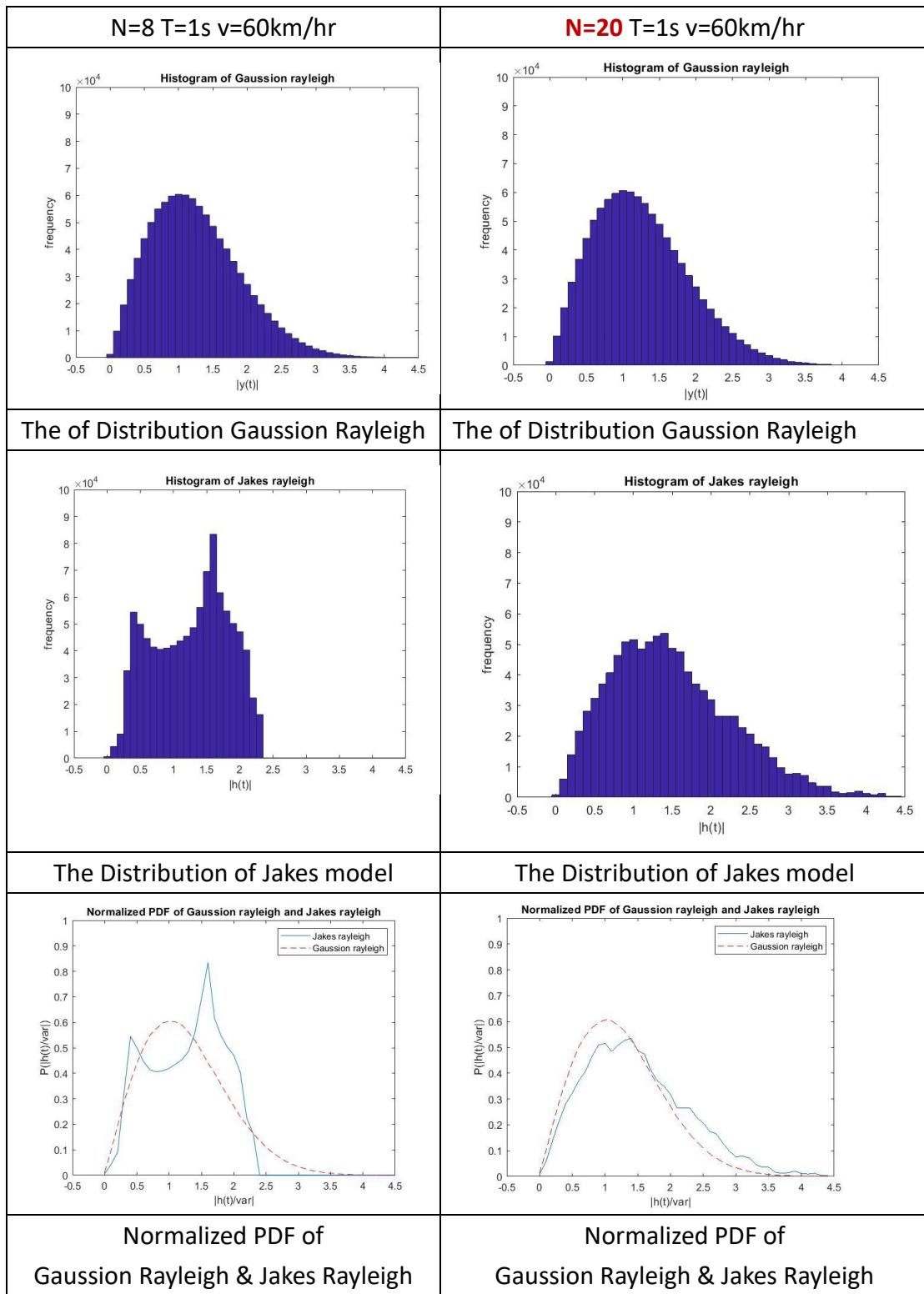
1. 首先，N 以及 V 固定，並改變 T，一邊維持原樣，一邊將 T 設為 1.5sec



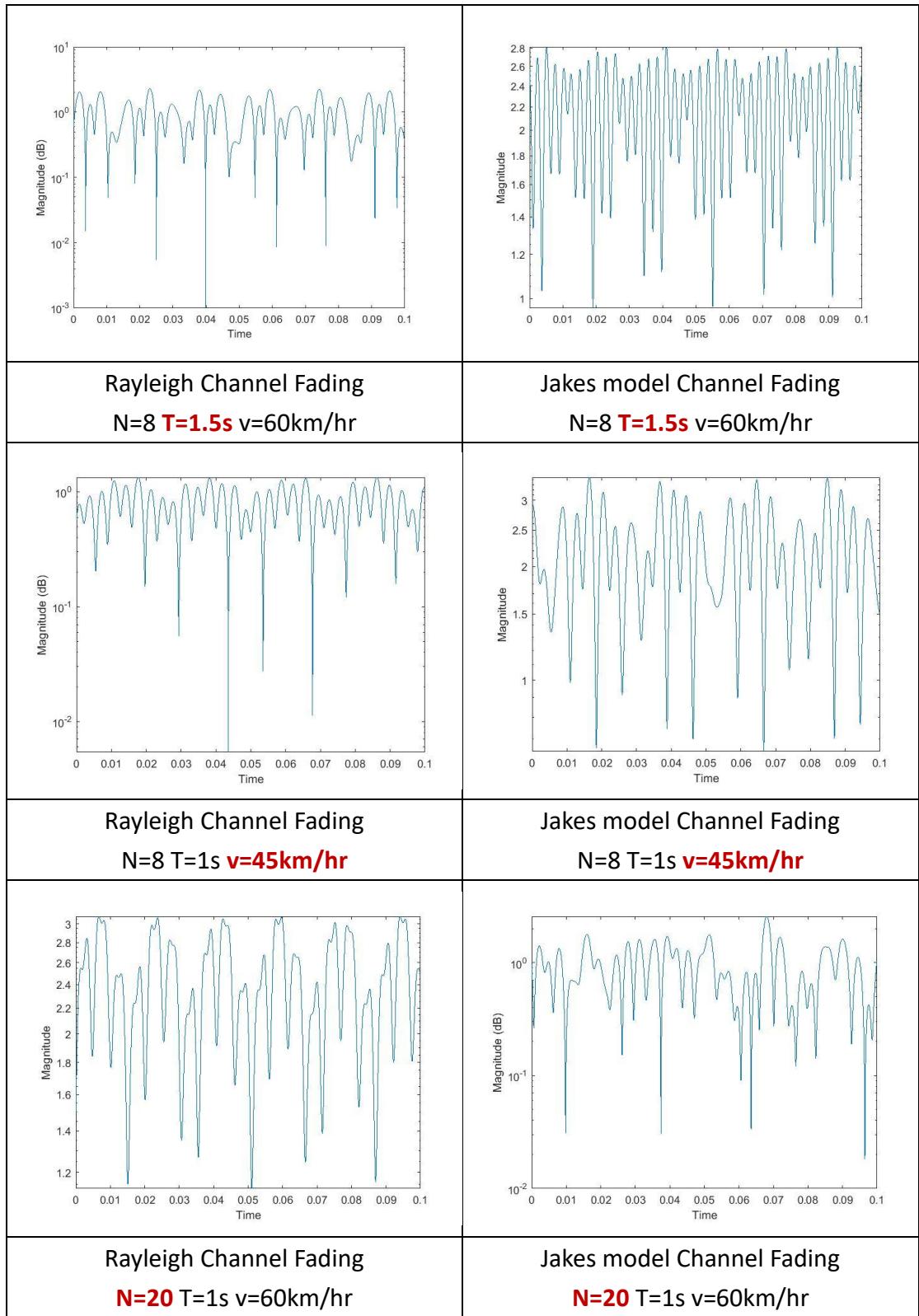
2..再來，N 以及 T 固定，並改變 V，一邊維持原樣，一邊將 V 設為 45km/hr



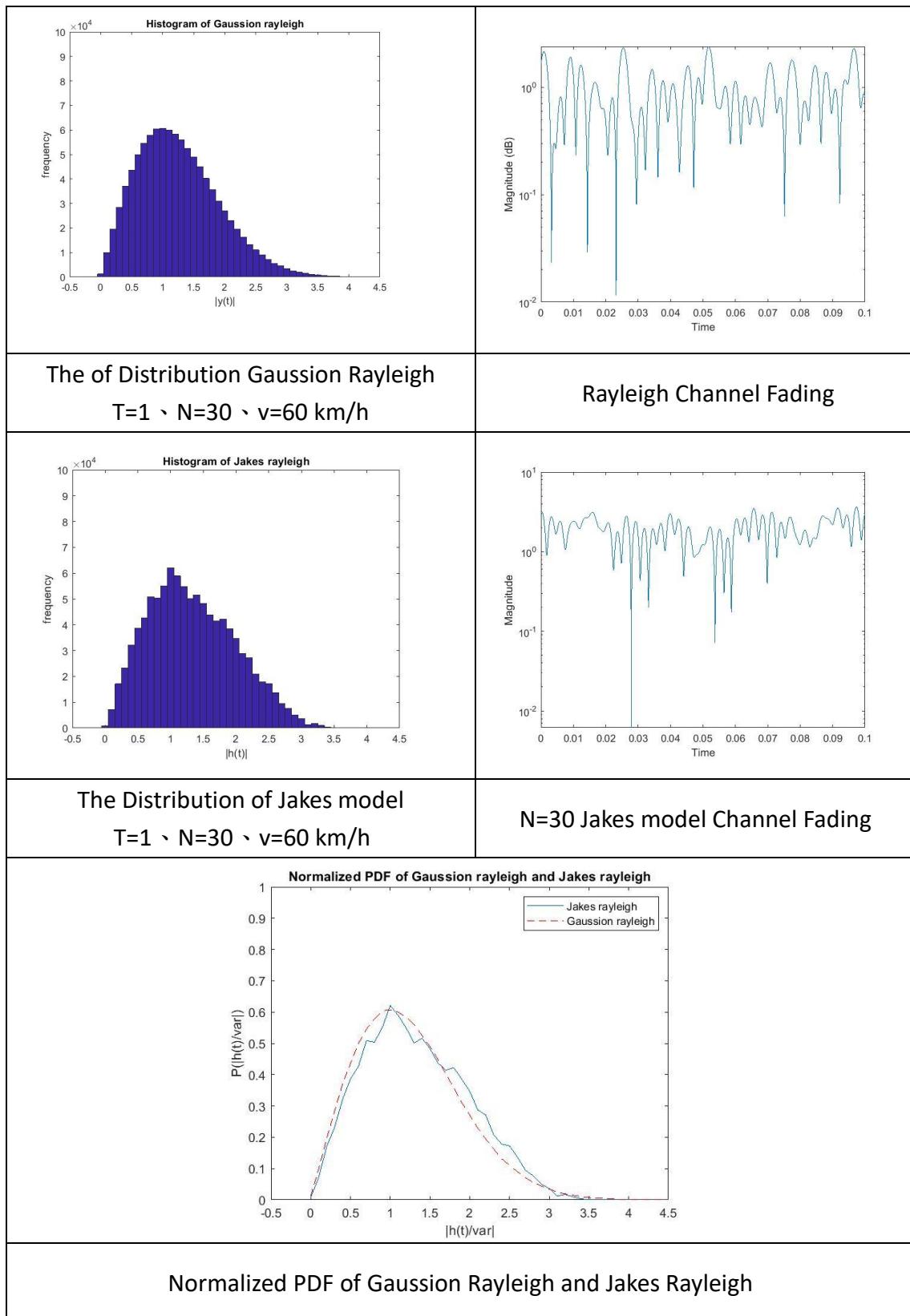
3. 再來，T 以及 V 固定，並改變 N，一邊維持原樣，一邊將 N 設為 20



4. 1~3 中的 fading($k=2$) (additional plots)



5. N=30, 其餘參數與(i)一樣



討論：

1. 改變 T 為 1.5s

此時都普勒偏移值會變得更小，而使得取樣點間距更密，讓整體波形更加接近 Gaussian Rayleigh 波形，原本因為偏移值較大，會造成誤差。

而看到衰減的圖形，Gaussian Rayleigh 的衰減變的比較平坦，而 Jake Model 衰減就會變得更密。

2. 改變 V 為 45km/hr

因為相對速度的下降，可能導致比在 60km/hr 時有著更高的接收頻率
例如在 $|h(t)|=2$ ，頻率高達 10^4 ，而形狀上也更加相似 Gaussian Rayleigh。

在通道衰減的部分，我們可以觀察到當 N 與 T 固定時，v 做改變，我們可以發現到當 v 越小，通道衰減的頻率也會越小，訊號衰弱之間隔則越大，反之 v 越大，通道衰減的頻率大，訊號衰弱之間隔則越小。

3. 改變 N 為 20/30

Jakes model 在 N=20 與 30 的時候，比 N=8 更接近 Rayleigh 分佈。

因為 Jakes model 函數中，phase $\alpha_m = \frac{2\pi m}{N}$, $m=0,1,2...,N-1$ ，當 N 變大

α_m 的個數增加，相位角數量也跟著增加，因此在取樣數量增加的情況 Jakes model 的分佈圖就越接近 Rayleigh 分佈。

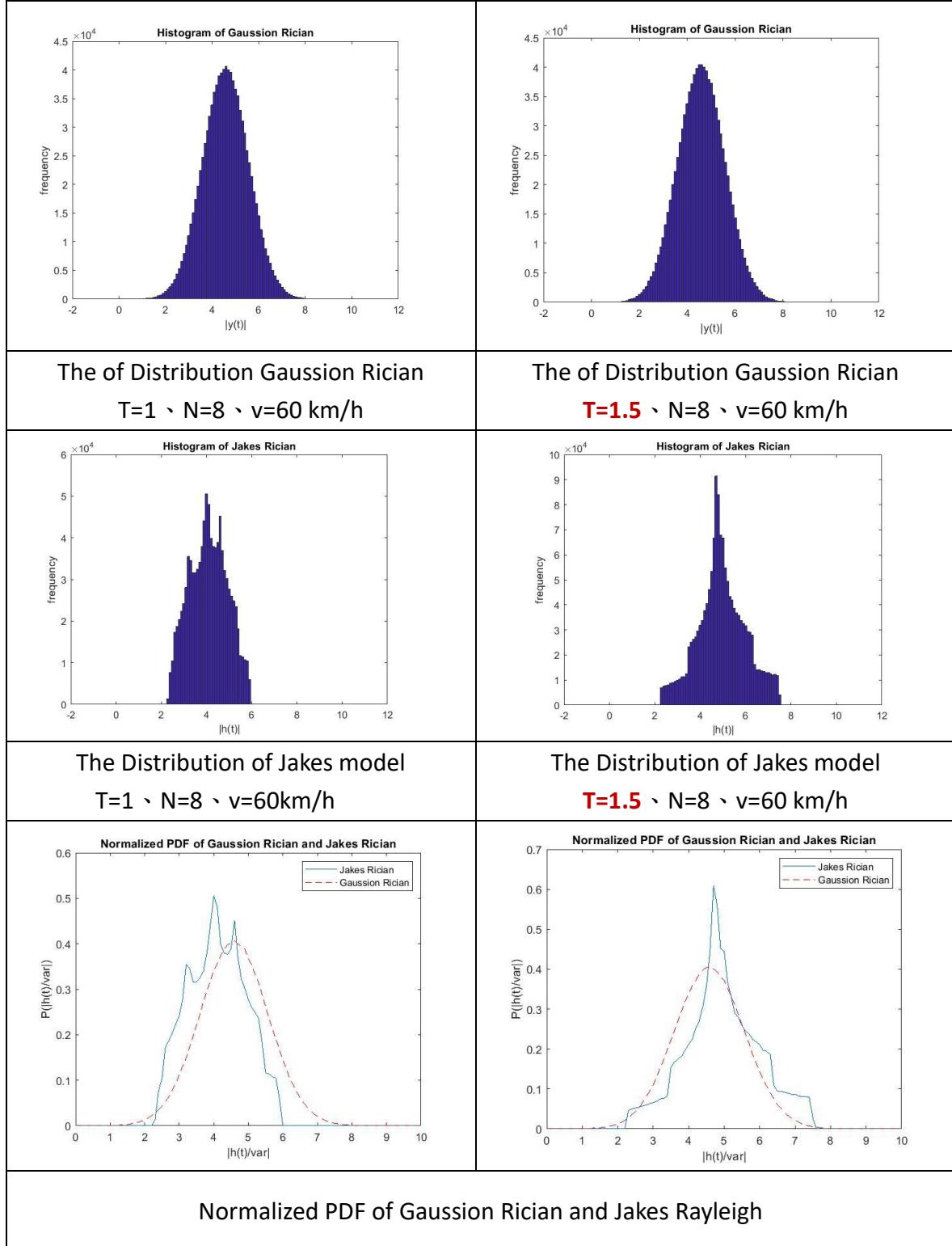
而在衰減通道的部分，我們也能察覺到較 Gaussian Rayleigh 分布 Jakes model 除了在 $t=0.03$ ，有相似 deep fading 的狀況，但其他接收情況更為平坦，即當取樣點更多時，能獲得更穩定的響應。

(iii)

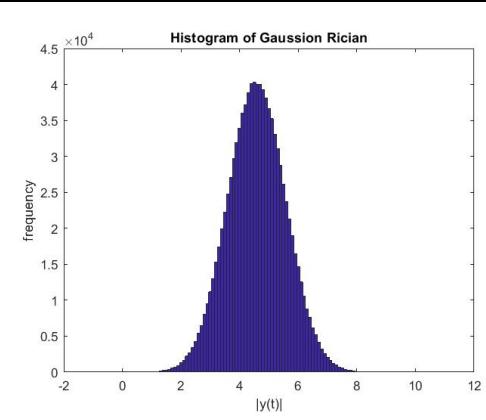
1.K=10

(i) $T=1$ 、 $N=8$ 、 $v=60 \text{ km/h}$

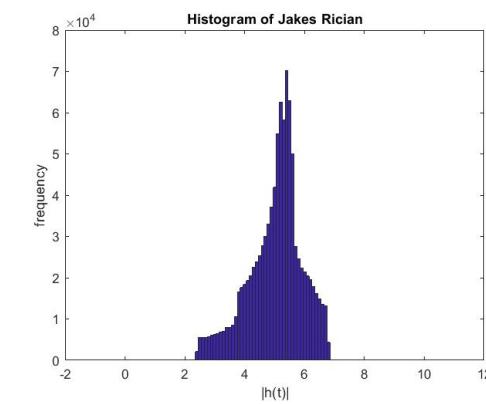
(ii) 改變 T 為 1.5sec



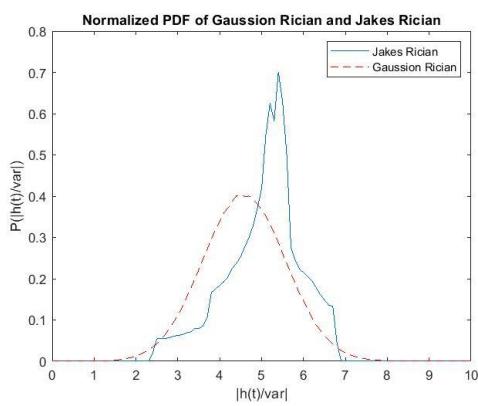
(ii) 改變 v 為 45km/hr



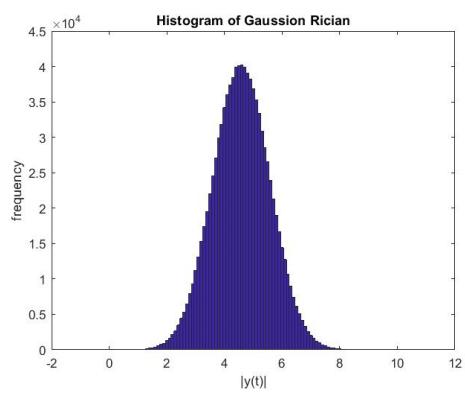
The Distribution of Gaussian Rician
 $N=8 T=1s v=45km/hr$



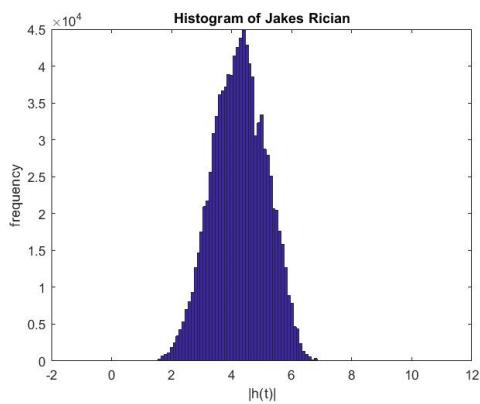
The Distribution of Jakes model
 $N=8 T=1s v=45km/hr$



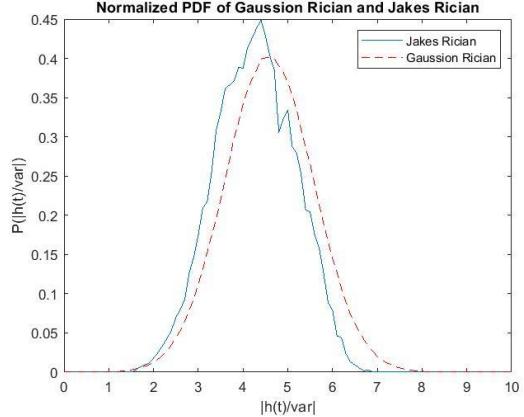
(ii) 改變 N 為 20



The Distribution of Gaussian Rician
 $N=20 T=1s v=60km/hr$

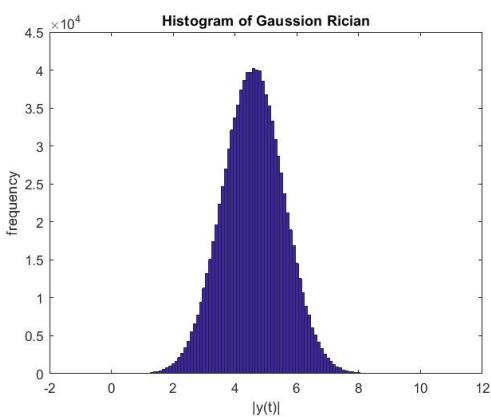


The Distribution of Jakes model
 $N=20 T=1s v=60km/hr$



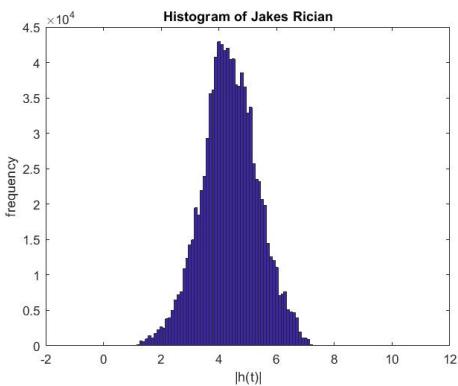
Normalized PDF of Gaussian Rician and Jakes Rayleigh

(ii)N=30 時的情況



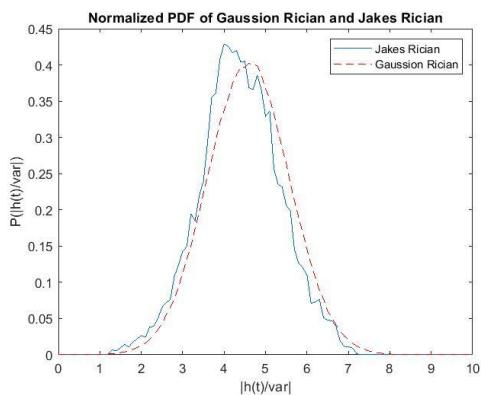
The of Distribution Gaussian Rician

T=1 、 N=30 、 v=60 km/h



The Distribution of Jakes model

T=1 、 N=30 、 v=60 km/h

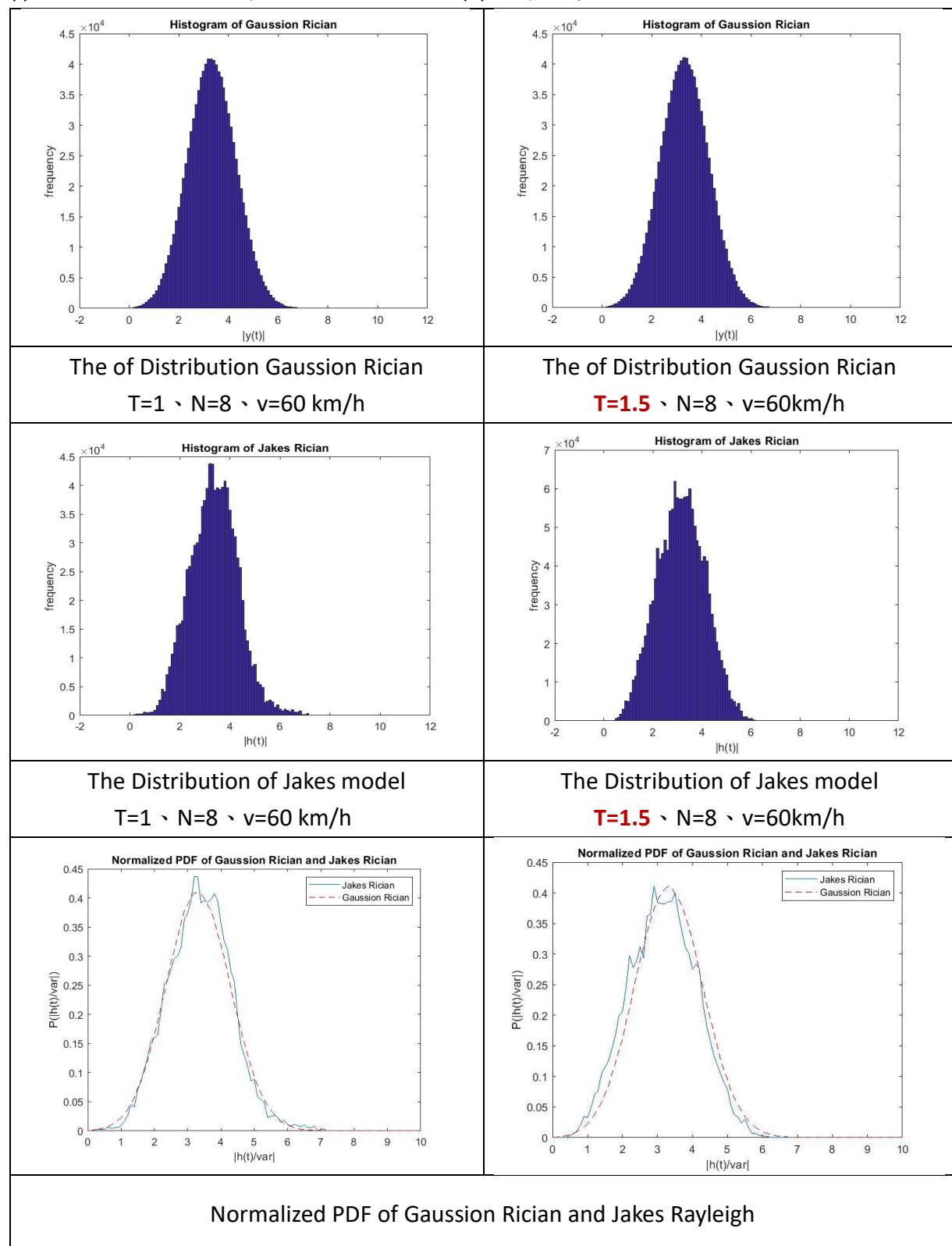


Normalized PDF of Gaussian Rician and Jakes Rayleigh

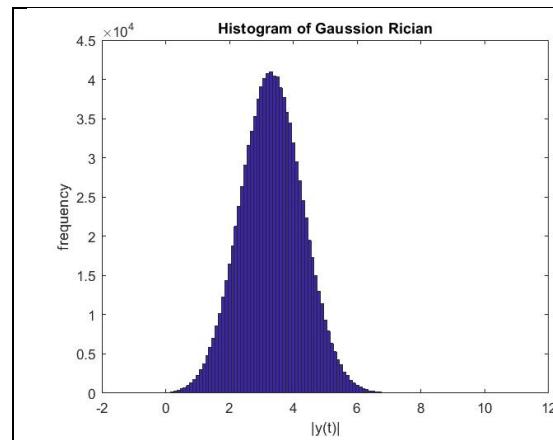
2.K=5

(i) T=1、N=8、v=60 km/h

(ii) 改變 T 為 1.5sec

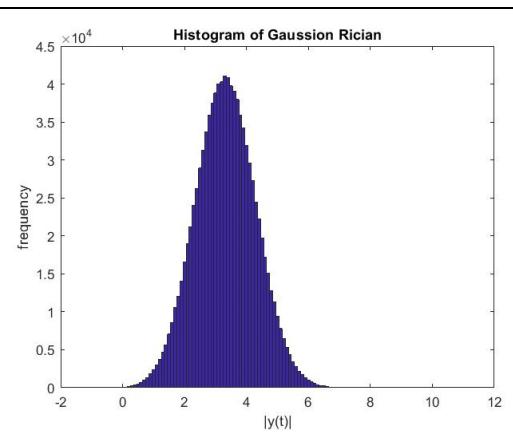


(ii) 改變 v 為 45km/hr

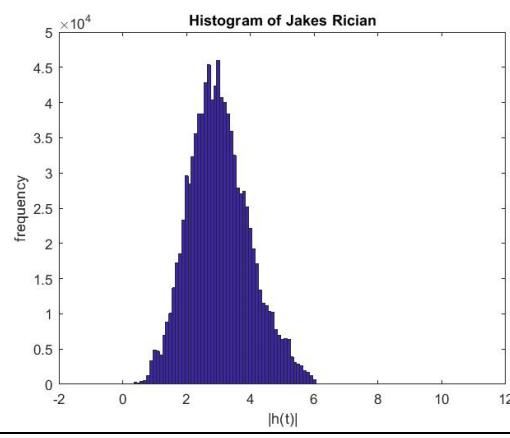


The of Distribution Gaussion Rician
 $T=1$ 、 $N=8$ 、 $v=45$ km/h

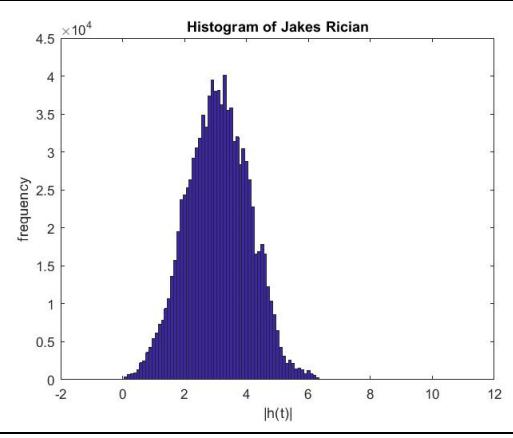
(ii) 改變 N 為 20



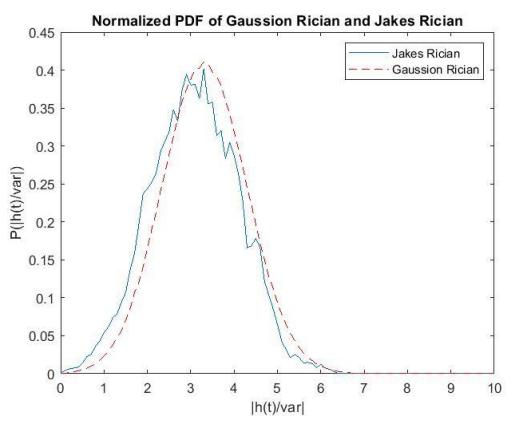
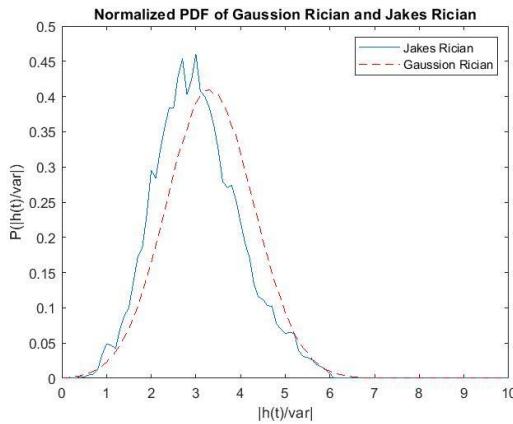
The of Distribution Gaussion Rician
 $T=1$ 、 $N=20$ 、 $v=60$ km/h



The Distribution of Jakes model
 $T=1$ 、 $N=8$ 、 $v=45$ km/h

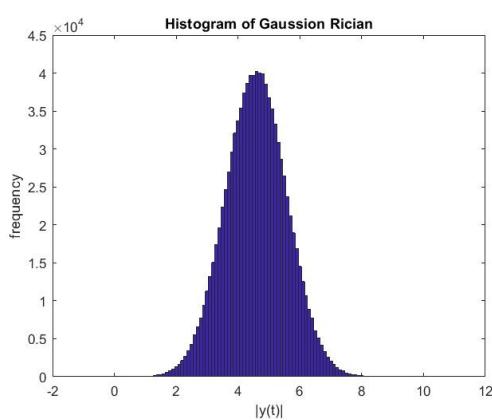


The Distribution of Jakes model
 $T=1$ 、 $N=20$ 、 $v=60$ km/h



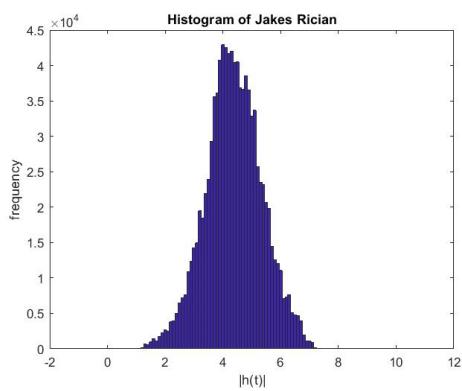
Normalized PDF of Gaussion Rician and Jakes Rayleigh

(ii) 當 $N=30$ 時



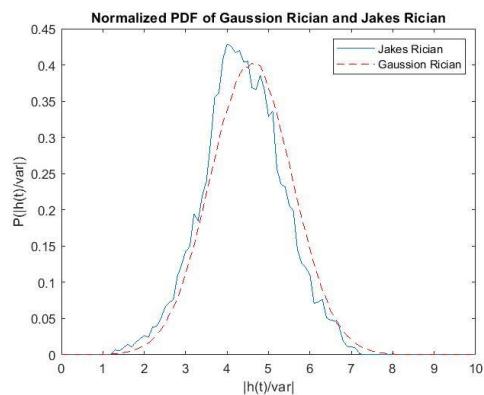
The of Distribution Gaussian Rician

$T=1$ 、 **N=30** 、 $v=60$ km/h



The Distribution of Jakes model

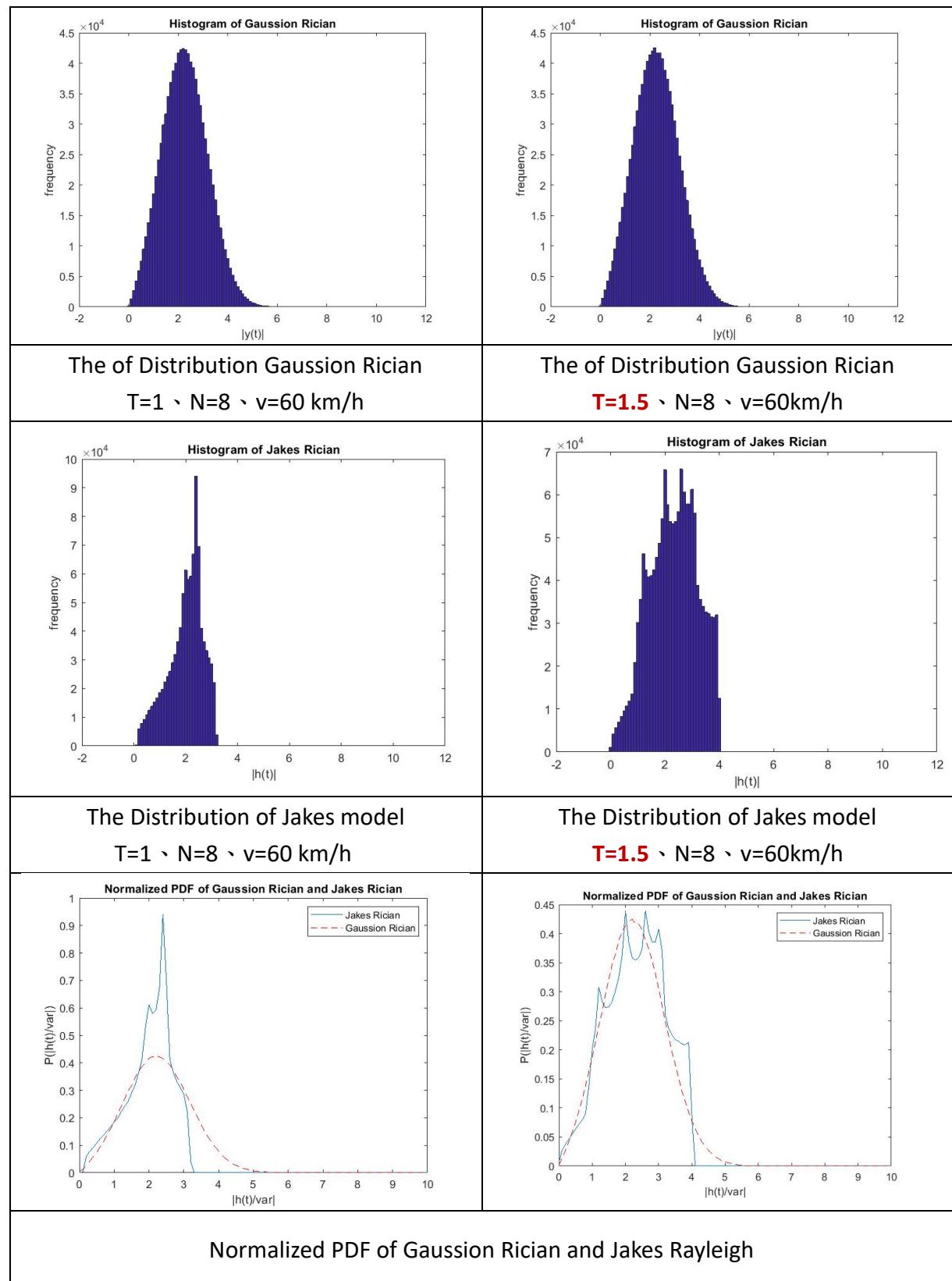
$T=1$ 、 **N=30** 、 $v=60$ km/h



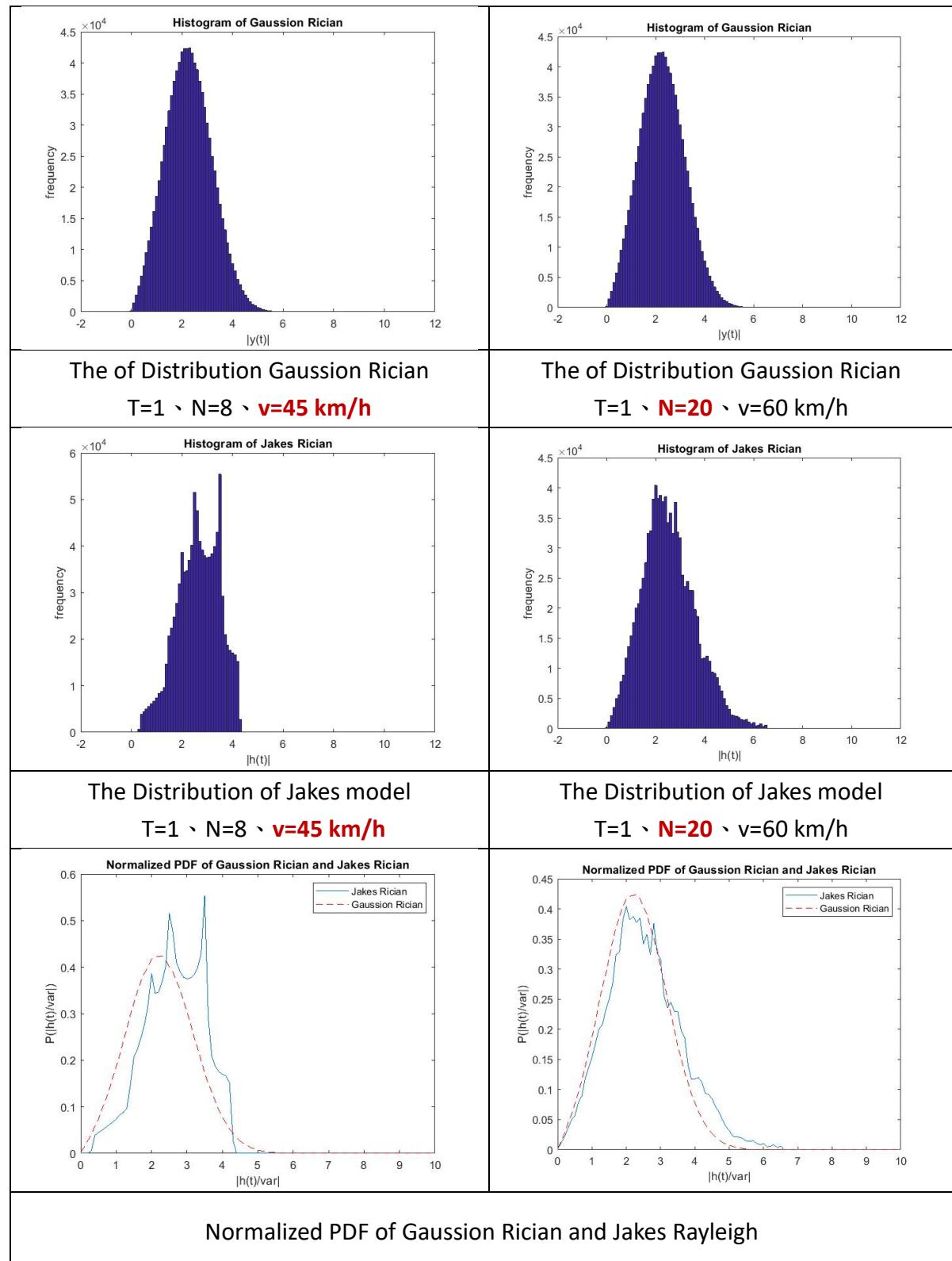
3.K=2

(i) $T=1$ 、 $N=8$ 、 $v=60$ km/h

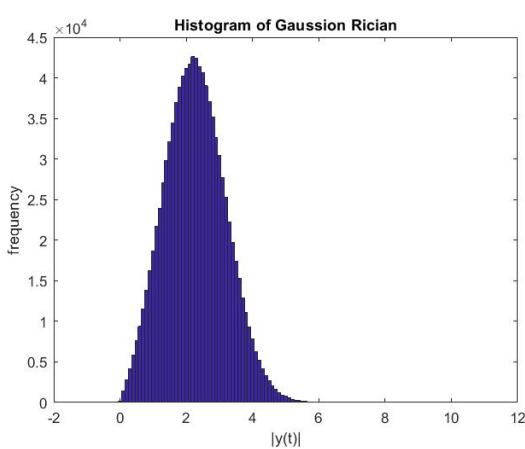
(ii) 改變 T 為 1.5sec



(ii) 改變 v 為 45km/hr

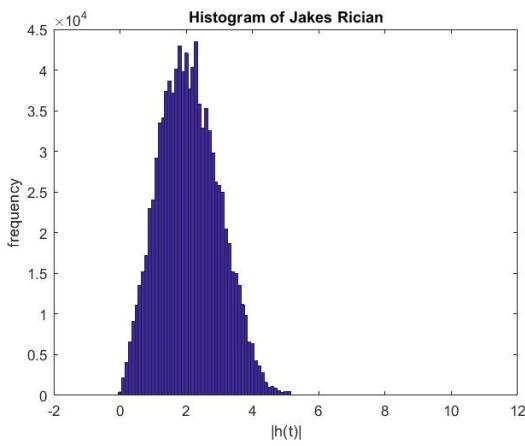


(ii)N=30



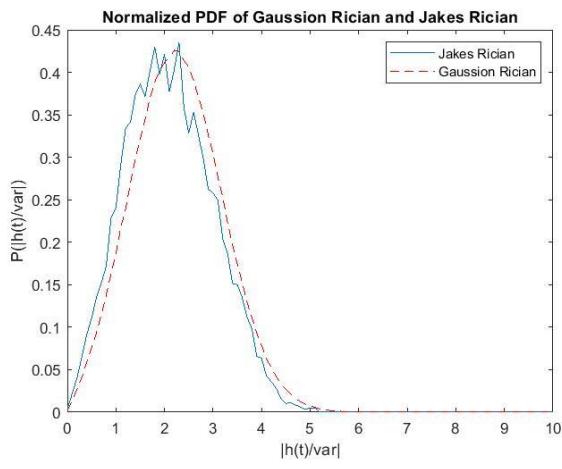
The of Distribution Gaussian Rician

T=1 、 N=30 、 v=60 km/h



The Distribution of Jakes model

T=1 、 N=30 、 v=60 km/h



討論：

- 在無線信道中，萊斯分佈是一種最常見的用於描述接收信號包絡統計時變特性的分佈類型。其中萊斯因子是反映信道質量的重要參數。

- Rician fading channel Impulse response

$$h(t) = Ae^{j\omega nt} + E_o C_m \sum_{m=1}^N e^{j(\omega nt + \psi_m)}$$

- 萊斯因子

$$K = \frac{A^2}{\sigma^2} = A^2 = K\sigma^2 \text{ 來求出振幅，因此能得知 } K \text{ 與 } A \text{ 正相關。}$$

1. 振幅：

將 $h(t)$ 取絕對值比較，可得知 K 越大，振幅也越大，圖形也會越往右移動。

2. 頻率響應：

可以從實驗後的圖片中了解到，在 $k=5$ 的時候，最符合高斯分布的頻率響應，其他可能因為又 $k=10$ 使圖形右移過多，response 經過 $h(t)$ 後也過大，即是最突出的 A ；而在 $k=2$ 時，又因為 response 不同，也超過了高斯分布。

3. T

除了 $k=10$ 時，導致頻率響應過大之外，其他的圖形中，我們都能觀察到與高斯分布的改變差異不大，有可能是我們所改的週期變化不夠大，因此沒有明顯的變化。

4. V

除了在 $k=5$ 時，因為與高斯分布相似，因此偏移沒過大之外，我們能發現當相對速度減少，在 $k=2$ 與 $k=10$ 時會導致 $|h(t)|$ 右移過多。

5. k

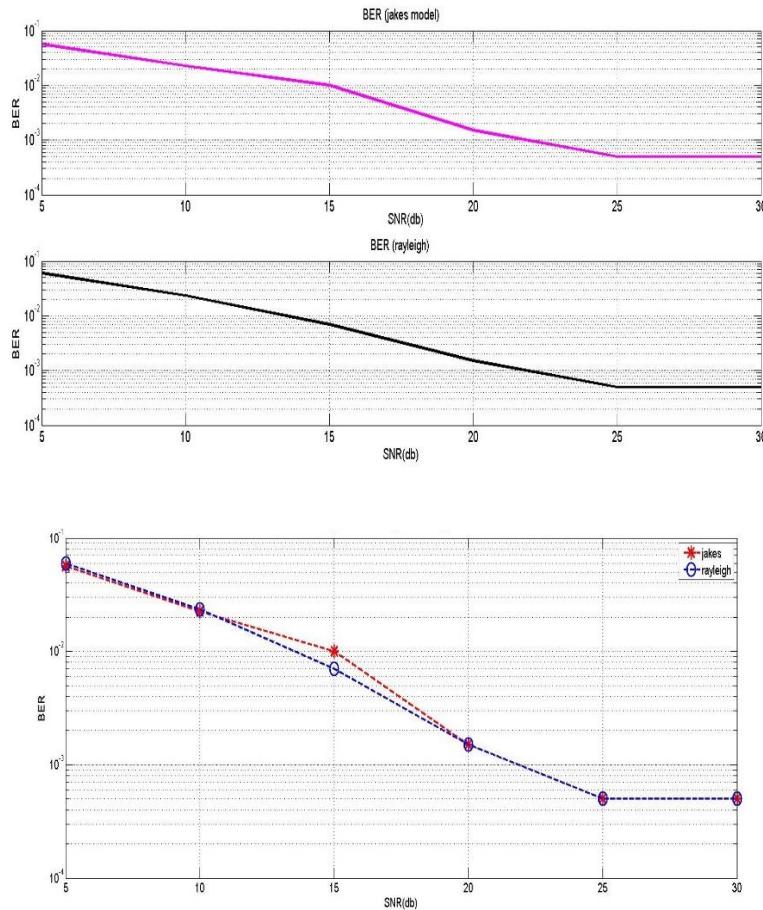
而在 $k=10$ ，模擬分布圖會越像 Rician 分佈，原因在於 $Z = [(X + A)^2 + Y^2]^{1/2}$ ， K 越大 A 也越大， $(X + A)^2$ 在 Rician 分佈所佔的成份也越大。當 A^2 遠大於 X 及 Y 的變異數時， Y^2 就可以省略，則 Z 會趨近於平均值為 A 的高斯分佈。

6. N

在經過 N 數目的增加， $C_m = \sqrt{1/N}$ ，若 N 更大則能使 C_m 變小，讓萊斯分布的多重路徑影響降低，剩下直射訊號 A 的影響，因此更能接近於高斯分布。

(b)

- (i) By assuming perfect estimation of the complex channel gain $c[k]$ in the receiver, please generate the BER curves vs. Eb/No by using the Jake's model $c[k]=h(t)|_{t=kt}$ and the model of complex Gaussian random process $c[k]=n_R[k]+j n_I[k]$. Please show both BER curves with the two different channel models for the comparison and provide your comment on the results.



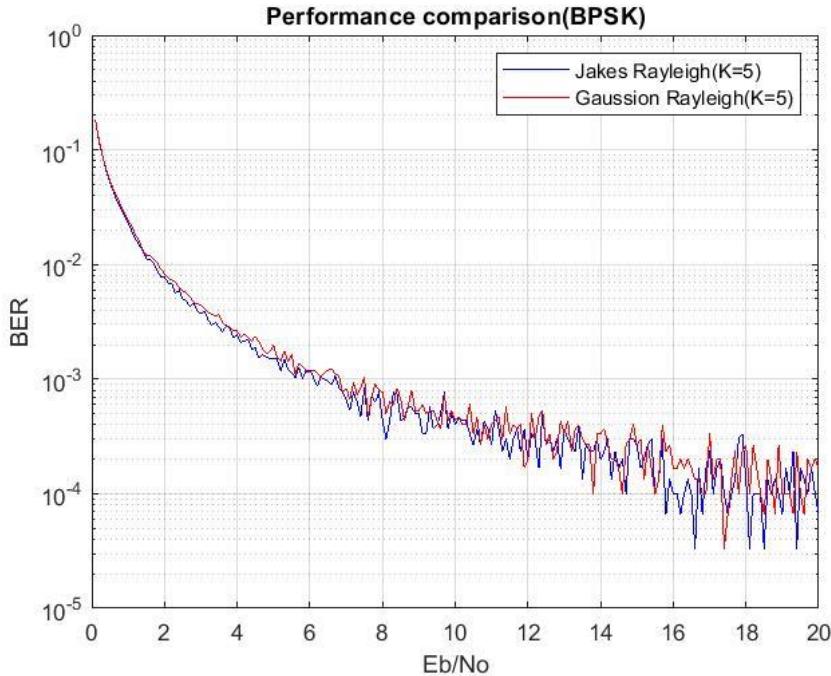
討論:若是我們固定 $BER=10^{-2}$ ，此時能透過最後一張的比較圖，

看出 Jakes model 能對應到 $SNR=15$ ，在對應 Gaussian Rayleigh 的 $SNR=14$ ，因此能推斷出 Jakes model 的效能會優於 Gaussian Rayleigh 但在其他部分，效能其實是差不多的。

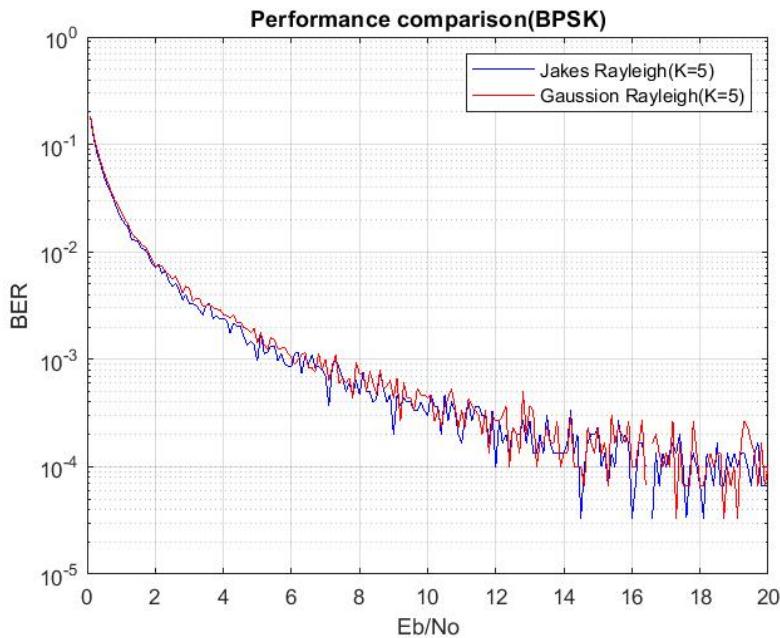
而我認為會有稍微略高於 Gaussian Rayleigh 的表現是因為在 Jakes model 時由於對於各個接收訊號有更小的相關性，因此會使 channel gain 在不同時間點有更多的增益，而不會受到其他接收端有所影響，因此可能有較好的效能。

2. (ii) Repeat the experiment of step (i) with the K factor equal to 5.

- N=50



- N=100(**additional plots**)



說明:

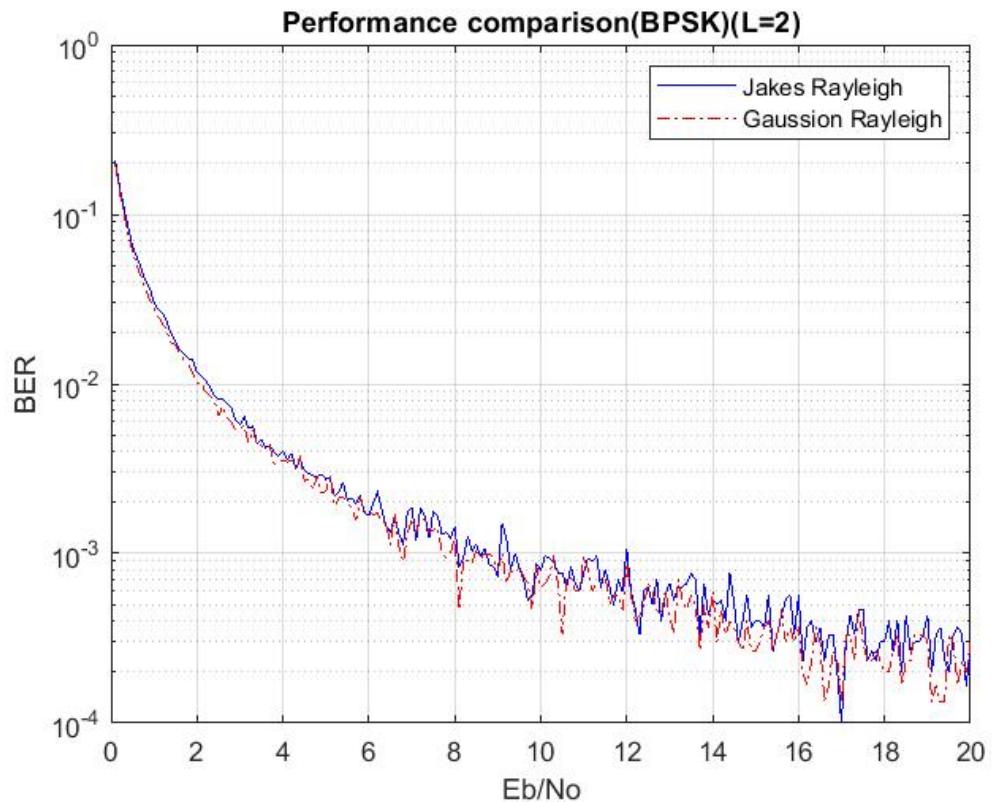
同樣先經過圖形上的比較，能發現 Jake Rayleigh 比原本的 Gaussian Rayleigh 效能更好。

相較於(i)而言，由於有加了 $k=5$ ，因此提供了 Rician channel 一個直射路徑，使得整體效能更好。

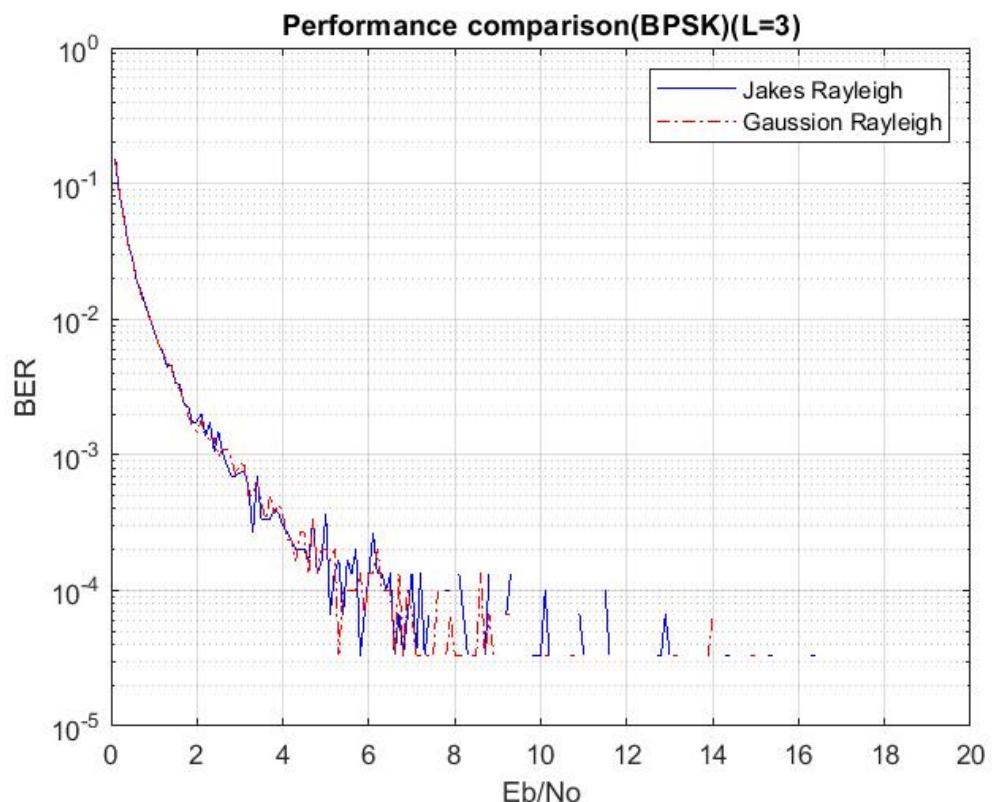
而對於 N 更大的圖形而言，整體效能更加靠近 Rayleigh 分布

(iii)

- L=2

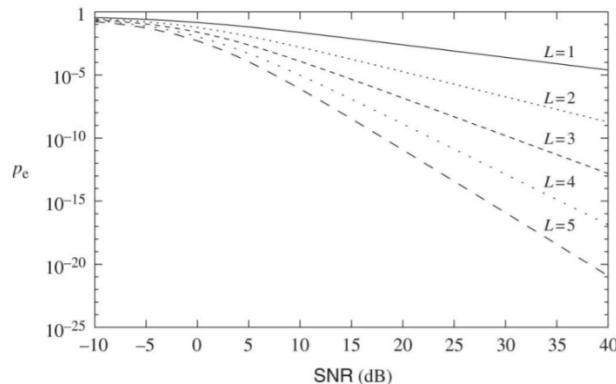


- L=3



說明：

首先 diversity order 依照大部分的定義而言，是指斜率。



因此透過 $L=2$ 與 $L=3$ 的圖形，能觀察到當 L 越大則效能越好。

並且效能應該是大於以上兩種的。

此外，若效能過差的時候可能會產生 deep fading，即當錯誤率太高時。但是若是使用 diversity 能增加更多的訊號路徑來增加 L ，而採用此策略則能避免掉通道內的 fading，增加通道傳輸時的可靠性。
因此就有更好，更穩定的效能。

對於 $L=2$ 而言，Jakes model 與 Gaussian Rayleigh 效能差不多。

而在 $L=3$ 時可以很明顯 Jakes model 效能優於 Gaussian Rayleigh。

Source Code

a(i)產生 Jakes model 與 Gaussian Rayleigh 的直方圖

```
1 - clc;clear;close all;
2 - %-----先給定參數-----%
3 - T=1; %時間取T=1，若取太多，例如100則電腦無法負荷計算量
4 - delta_t=0.000001; %以下為題目所給的數值
5 - N=30;
6 - fc=3.5*10^9;
7 - c=3*10^8;
8 - Eo=sqrt(2);
9 - v=(60000)/3600; %先換成(m/s)較為常用的單位
10 - landa=c/fc;
11 - t=0:delta_t:T; %以下為所使用的間隔(t/m)
12 - m=0:1:N-1;
13 - c_m=sqrt(1/N); %以下為baseband representation的參數表示
14 - w_n=(2*pi*v)/landa;
15 - rfa_m=(2*pi*m)/N;
16 - %-----計算Jake's model的h(t)-----%
17 - for i=1:N %用for來進行sigma運算
18 - phi_m=2*pi*rand(1,1); %代表的是uniform distribution
19 - h(i,:)=c_m*exp(j*(cos(rfa_m(i))*w_n*t+phi_m));
20 - end
21 - h_t=Eo*sum(h);
22 - %-----計算Jake's rayleigh fadding pdf -----%
23 - A_h=abs(h_t); %取絕對值
24 - y=[0:0.1:10];
25 - [a,y]=hist(A_h,y); %hist指令將整個資料依其大小分為多組，各組所得之數目作為繪圖之根據。
26 - p_A_h=(a./sum(a)/(0.1)); %句號的意思是一個一個相乘 [2 3].*[2 3]=[4 9] 本行是要算密度
27 -
28 - %-----計算 Gaussian rayleigh fadding pdf -----%
29 - var=1;
30 - c=sqrt(var)*(randn(1,1000000)+j*randn(1,1000000));%計算一般的complex Gassion
31 - A_c=abs(c);
32 - y_c=[0:0.1:10];
33 - [a_c,y_c]=hist(A_c,y_c);
34 - p_A_c=(a_c./sum(a_c)/0.1);
35 - %-----此處為計算平均值與變異數-----%
36 - mean_c=mean(c);mean_h_t=mean(h_t);
37 - var_c=std(c)^2;var_h_t=std(h_t)^2;
38 - mean_A_c=mean(A_c);mean_A_h=mean(A_h);
39 - var_A_c=std(A_c)^2;var_A_h=std(A_h)^2;
40 - %-----繪出直方圖-----%
41 - figure(1)
42 - plot(y,p_A_h,y_c,p_A_c,'r--'); %將Gaussian Rayleigh使用紅色與-表示
43 - title('Normalized PDF of Gassion rayleigh and Jakes rayleigh');
44 - xlabel('|h(t)|/var|');ylabel('P(|h(t)|/var|)');
45 - legend('Jakes rayleigh','Gassion rayleigh') %用legend函數做出圖示
46 - axis([-0.5 4.5 0 1]);
47 - figure(2)
48 - hist(A_h,y);title('Histogram of Jakes rayleigh');
49 - xlabel('|h(t)|');ylabel('frequency');
50 - axis([-0.5 4.5 0 10^5]);
51 - figure(3)
52 - hist(A_c,y_c);title('Histogram of Gassion rayleigh');
53 - xlabel('|y(t)|');ylabel('frequency');
54 - axis([-0.5 4.5 0 10^5]);
```

a. Rayleigh Fading Channel

```
1 - clc;clear;close all;
2 - %-----先給定參數-----%
3 - n = 30;                                % the number of the scatters
4 - c = sqrt(1/n);                           % Cm
5 - Eo = 1;                                 % Scatter power
6 - v = (60000)/3600;                        % Mobile Velocity
7 - fc = 3.5*10^9;                           % Carrier Frequency(3.5*10^9)
8 - landa = 3*10^8/fc;                        % Wavelength
9 - d = (1:n);
10 - wn = 2*pi*v/landa;
11 - phi = 2*pi*(d-1) / n;                  % angle of scatters
12 - t=(0:0.000001:1);
13 - theta = (2*pi).*rand(1,n);              % 0~2*pi Uniform distributed
14 - xc = Eo * c * cos(wn*t);
15 - xs = zeros(1,length(xc));               % 利用zeros來將數值化為矩陣方便計算
16 - %-----計算signal虛部與實部-----%
17 - for k=1:n
18 -     xc = Eo * c * cos(cos(phi(k))* wn*t+theta(k)) + xc; % 實部
19 -     xs = Eo * c * sin(cos(phi(k))* wn*t+theta(k)) + xs; % 虛部
20 - end
21 - ag = abs(xc+ sqrt(-1) * xs);           % 為了計算Magnitude，因此加上絕對值
22 - %-----繪出圖形-----%
23 - hist(ag, 300);
24 - ylabel('Number of Hits');
25 - xlabel('Magnitude');
26 - figure(2);

27 - semilogy(0.000001*(1:100000),ag(1:100000))%x軸為線性刻度，y 軸為對數刻度
28 - xlabel('Time');
29 - ylabel('Magnitude (dB)');
```

a. Rician Fading Channel

```
1 - clc;clear;close all;
2 - %-----先給定參數-----
3 - n = 30;                                % the number of the scatters
4 - c = sqrt(1/n);                          % Cm
5 - km=2;                                  % K-factor (in dB)
6 - kf=10^(km/10);                         % K-factor
7 - Eo=1;                                   % Scatter Power
8 - a=sqrt(2*kf*(Eo^2));                  % formula of Amplitude
9 - v = 60000/3600;                         % Mobile Velocity
10 - fc = 3.5*10^9;                        % Carrier Frequency
11 - landa = 3*10^8/fc;                   % Wavelength
12 - d = (1:n);
13 - wn = 2*pi*v/landa;                  % angle of scatters
14 - t=(0:0.000001:1);
15 - theta = (2*pi).*rand(1,n);           % 0~2*pi Uniform distributed
16 - xc = Eo * c * cos(wn*t);
17 - xs = zeros(1,length(xc));
18 - xm = zeros(1,length(xc));            % 利用zeros來將數值化為矩陣方便計算
19 - %-----計算signal虛部與實部-----
20 - for k=1:n
21 -     xc = Eo * c * cos(cos(phi(k))* wn*t+theta(k)) + xc; % 實部
22 -     xs = Eo * c * sin(cos(phi(k))* wn*t+theta(k)) + xs; % 虛部
23 - end
24 -
25 - xm = a * cos(wn*t) + a * sqrt(-1)*sin(wn*t);    % 茶斯分布即差在此A*e^j*cos(wn*t)
26 - ag = abs(xm + xc + sqrt(-1) * xs);              % Magnitude
27 - %-----繪出圖形-----
28 - hist(ag, 300);
29 - ylabel('Number of Hits');
30 - xlabel('Magnitude');
31 - figure(2);
32 - semilogy(0.000001*(1:100000),ag(1:100000))
33 - xlabel('Time');
34 - ylabel('Magnitude');
```

b. BER curves vs. Eb/No

```

1 - clc;clear all;close all;
%-----先給定參數-----
2 - n=32;
3 - T=2*le-3; % time
4 - Ts=le-6; % sampling time
5 - Wl=0.15; % wl=wavelength
6 - V=60000/3600;
7 - N=8; % number of scatters
8 - %-----產生binary source-----
9 - x=randi(1,n); % 隨機產生
10 - d=x*2-1;
11 - Cm=sqrt(1/N);
12 - Wn=2*pi*V/Wl % wn=2*pi*v/landa;
13 - e=l; % Scatter Power
14 - h_t=zeros(1,T/Ts); % 變為矩陣方便計算
15 - % -----計算h[k]:complex channel gain by using jake's model-----
16 - for m=0:N-1
17 -     am=2*pi*m/N;
18 -     q=rand(1,T/Ts)*2*pi;
19 -     h_tmp=e*Cm*exp(j*(cos(am)*Wn*[Ts:Ts:T]+q));
20 -     h_t=h_t+h_tmp;
21 - end
22 - h=h_t/max(abs(h_t));
23 - %-----設定complex Gaussian random process-----
24 - ray_tmp=randn(1,T/Ts)+j*randn(1,T/Ts);
25 - %-----先計算r[k]:complex Gaussian channel gain-----
26 - r=ray_tmp/max(abs(ray_tmp));
27 - %-----在接收端接收離散訊號-----
28 - xht=h.*d;
29 - %-----complex channel-----
30 - xrt=r.*d;
31 - for nn=1:6
32 -     snr=nn*5;
33 -     %-----加上AWGN訊號-----
34 -     xh=awgn(xht,snr,'measured');
35 -     xr=awgn(xrt,snr,'measured');
36 -     %-----在接收端所收到的訊號-----
37 -     yh=conj(h).*xh; % receive signal with perfect estimation h[k] and r[k]
38 -     yr=conj(r).*xr;
39 -     %-----解調-----
40 -     yhd=sign(real(conj(h).*xh));
41 -     yrd=sign(real(conj(r).*xr));
42 -     %-----計算BER= 送傳錯誤/總傳送-----
43 -     [number_h, ratio_h]= symerr(d,yhd) % number: number of symbol errors
44 -     [number_r, ratio_r]= symerr(d,yrd) % ratio : symbol error rate
45 -     ber_h(nn,:)=ratio_h;
46 -     ber_r(nn,:)=ratio_r;
47 - end
48 - %-----繪製圖形-----
49 - figure
50 - subplot(2,1,1)
51 - x_axle=l5:5:30];
52 - semilogy(x_axle,ber_h,'m');
53 - title('BER (jakes model)');
54 - xlabel('SNR(db)');
55 - ylabel('BER');
56 - grid on;
57 - subplot(2,1,2)
58 - semilogy(x_axle,ber_r,'k');
59 - title('BER (rayleigh)');
60 - xlabel('SNR(db)');
61 - ylabel('BER');
62 - grid on;
63 - figure
64 - semilogy(x_axle,ber_h,'r--*',x_axle,ber_r,'b--o');
65 - xlabel('SNR(db)');
66 - legend('jakes','rayleigh');
67 - grid on;

```

b.K factor=5

```
1 - clc;clear;
2 - %-----先給定參數-----
3 - T=3;                                % time
4 - delta_t=0.0001;
5 - N=100;                               % number of scatters
6 - fc=3.5*10^9;                         %(注意此處數值不能過大，大約500左右負荷不了)
7 - c=3*10^8;                            % Carrier Frequency(3.5*10^9)
8 - Eo=sqrt(10);
9 - v=(60000)/3600;
10 - landa=c/fc;
11 - c_m=sqrt(1/N);
12 - w_n=(2*pi*v)/landa;
13 - g=0;
14 - t=0:delta_t:T;
15 - m=0:1:N-1;
16 - rfa_m=(2*pi*m)/N;                  % angle of scatters
17 - %-----計算Jakes model的h(t)-----
18 - for i=1:N
19 - phi_m=2*pi*rand(1,1);
20 - h1(i,:)=c_m*exp(j*(cos(rfa_m(i))*w_n*t+phi_m));    %公式表示
21 - end
22 - for i=1:N
23 - phi_m=2*pi*rand(1,1);
24 - h2(i,:)=c_m*exp(j*(cos(rfa_m(i))*w_n*t+phi_m));
25 - end
26 -  

27 - h_t1=Eo*sum(h1);
28 - h_t2=Eo*sum(h2);
29 - %-----產生c[k]-----%
30 - c_k1=h_t1;                           %Rayleigh
31 - c_k2=h_t2;
32 -
33 - c_k1_Gauss=(randn(1,length(c_k1))+j*randn(1,length(c_k1))):%Gaussian
34 - c_k2_Gauss=(randn(1,length(c_k2))+j*randn(1,length(c_k2)));
35 - %-----產生d[k]-----%
36 - bit_num=length(c_k1);
37 - d_k=source(bit_num);
38 - %-----計算Eb且建立BPSK model-----
39 - for SNR=0.1:0.1:20
40 - mean_c_k1=mean(c_k1);mean_c_k1_Gauss=mean(c_k1_Gauss);
41 - var_c_k1=std(c_k1)^2;var_c_k1_Gauss=std(c_k1_Gauss)^2;
42 - %-----此處Eb只計算一個diversity channel 因為其他channel應會相同-----
43 - Eb=mean(abs(c_k1).^2);
44 - No=Eb/SNR;
45 - Eb_Gauss=mean(abs(c_k1_Gauss).^2);
46 - No_Gauss=Eb_Gauss/SNR;
47 - g=g+1;
48 - var_w=No/2;
49 - bit_err=0;
50 - bit_Gauss_err=0;
51 - var_w_Gauss=No_Gauss/2;
52 - w_k1=sqrt(var_w)*randn(1,length(d_k));
```

```

53 - w_k2=sqrt(var_w)*randn(1,length(d_k));
54
55 - w_k1_Gauss=sqrt(var_w_Gauss)*randn(1,length(d_k));
56 - w_k2_Gauss=sqrt(var_w_Gauss)*randn(1,length(d_k));
57
58 - x_k1=c_k1.*d_k+w_k1;
59 - x_k2=c_k2.*d_k+w_k2;
60
61 - x_k1_Gauss=c_k1_Gauss.*d_k+w_k1_Gauss;
62 - x_k2_Gauss=c_k2_Gauss.*d_k+w_k2_Gauss;
63 %-----+x_k2.*conj(c_k2)+x_k3.*conj(c_k3));此處在做MRC combining-----%
64 - decision_k=real(x_k1.*conj(c_k1)+x_k2.*conj(c_k2));
65 - decision_k_Gauss=real(x_k1_Gauss.*conj(c_k1_Gauss)+x_k2_Gauss.*conj(c_k2_Gauss))
66 %-----計算錯多少bit及BER-----%
67 - for i=1:length(x_k1)
68 -     if decision_k(i)>=0 %decission rule上大於0
69 -         d_k_esti(i)=1; %則在此處偵測為1
70 -         if d_k_esti(i)~=d_k(i) %若不等於所偵測的值
71 -             bit_err=bit_err+1; %設bit錯誤此counter+1
72 -         end
73 -     end %以下大致相似
74 -     if decision_k(i)<0
75 -         d_k_esti(i)=-1;
76 -         if d_k_esti(i)~=d_k(i)
77 -             bit_err=bit_err+1;
78 -         end
79 -     end
80 - end
81 - for i=1:length(x_k1_Gauss)
82 -     if decision_k_Gauss(i)>=0
83 -         d_k_Gauss_esti(i)=1;
84 -         if d_k_Gauss_esti(i)~=d_k(i)
85 -             bit_Gauss_err=bit_Gauss_err+1;
86 -         end
87 -     end
88 -     if decision_k(i)<0
89 -         d_k_Gauss_esti(i)=-1;
90 -         if d_k_Gauss_esti(i)~=d_k(i)
91 -             | bit_Gauss_err=bit_Gauss_err+1;
92 -         end
93 -     end
94 - end
95 - Pb(g)=bit_err/bit_num; %計算錯誤率
96 - Pb_Gauss(g)=bit_Gauss_err/bit_num;
97 - end
98 %-----繪製圖形-----%
99 - SNR=0.1:0.1:20;
100 - semilogy(SNR,Pb,'b-',SNR,Pb_Gauss,'r-');
101 - xlabel('Eb/No');
102 - ylabel('BER');
103 - legend('Jakes Rayleigh(K=5)', 'Gaussian Rayleigh(K=5)');
104 - title('Performance comparison(BPSK)');
105 - grid on;

```

b.diversity(L=2)

```
1 - clc;clear;
2 - %-----先給定參數-----%
3 - T=3;                                % time
4 - delta_t=0.0001;
5 - N=100;                               % number of scatters
6 - %注意此處數值不能過大，大約500左右負荷不了)
7 - fc=3.5*10^9;                         % Carrier Frequency(3.5*10^9)
8 - c=3*10^8;
9 - Eo=sqrt(10);                          % Scatter power
10 - v=(60000)/3600;
11 - landa=c/fc;
12 - c_m=sqrt(1/N);
13 - w_n=(2*pi*v)/landa;
14 - g=0;
15 - t=0:delta_t:T;
16 - m=0:1:N-1;
17 - rfa_m=(2*pi*m)/N;                  % angle of scatters
18 - %-----計算Jakes model的h(t)-----%
19 - for i=1:N
20 - phi_m=2*pi*rand(1,1);
21 - h1(i,:)=c_m*exp(j*(cos(rfa_m(i))*w_n*t+phi_m));
22 - end
23 - for i=1:N
24 - phi_m=2*pi*rand(1,1);
25 - h2(i,:)=c_m*exp(j*(cos(rfa_m(i))*w_n*t+phi_m));
26 - end

28 - h_t1=Eo*sum(h1);
29 - h_t2=Eo*sum(h2);
30 - %-----產生c[k]-----%
31 - c_k1=h_t1;                           %Rayleigh
32 - c_k2=h_t2;
33 -
34 - c_k1_Gauss=(randn(1,length(c_k1))+j*randn(1,length(c_k1)));%Gaussian
35 - c_k2_Gauss=(randn(1,length(c_k2))+j*randn(1,length(c_k2)));
36 - %-----產生d[k]-----%
37 - bit_num=length(c_k1);
38 - d_k=source(bit_num);
39 - %-----計算Eb且建立BPSK model-----%
40 - for SNR=0.1:0.1:20
41 - mean_c_k1=mean(c_k1);mean_c_k1_Gauss=mean(c_k1_Gauss);
42 - var_c_k1=std(c_k1)^2;var_c_k1_Gauss=std(c_k1_Gauss)^2;
43 - %-----此處Eb只計算一個diversity channel 因為其他channel應會相同-----%
44 - Eb=mean(abs(c_k1).^2);
45 - No=Eb/SNR;
46 - Eb_Gauss=mean(abs(c_k1_Gauss).^2);
47 - No_Gauss=Eb_Gauss/SNR;
48 - g=g+1;
49 - var_w=No/2;
50 - bit_err=0;
51 - bit_Gauss_err=0;
52 - var_w_Gauss=No_Gauss/2;
53 - w_k1=sqrt(var_w)*randn(1,length(d_k));
```

```

54 - w_k2=sqrt(var_w)*randn(1,length(d_k));
55
56 - w_k1_Gauss=sqrt(var_w_Gauss)*randn(1,length(d_k));
57 - w_k2_Gauss=sqrt(var_w_Gauss)*randn(1,length(d_k));
58
59 - x_k1=c_k1.*d_k+w_k1;
60 - x_k2=c_k2.*d_k+w_k2;
61
62 - x_k1_Gauss=c_k1_Gauss.*d_k+w_k1_Gauss;
63 - x_k2_Gauss=c_k2_Gauss.*d_k+w_k2_Gauss;
64 %-----+x_k2.*conj(c_k2)+x_k3.*conj(c_k3));此處在做MRC combining-----%
65 - decision_k=real(x_k1.*conj(c_k1)+x_k2.*conj(c_k2));
66 - decision_k_Gauss=real(x_k1_Gauss.*conj(c_k1_Gauss)+x_k2_Gauss.*conj(c_k2_Gauss));
67 %-----計算錯多少bit及BER-----%
68 - for i=1:length(x_k1)
69 -     if decision_k(i)>=0           %decision rule上大於0
70 -         d_k_esti(i)=1;           %則在此處偵測為1
71 -         if d_k_esti(i)~=d_k(i)    %若不等於所偵測的值
72 -             bit_err=bit_err+1;    %設bit錯誤此counter+1
73 -         end
74 -     end                         %以下大致相似
75 -     if decision_k(i)<0
76 -         d_k_esti(i)=-1;
77 -         if d_k_esti(i)~=d_k(i)
78 -             bit_err=bit_err+1;
79 -         end
80 -     end
81 - end
82 - for i=1:length(x_k1_Gauss)
83 -     if decision_k_Gauss(i)>=0
84 -         d_k_Gauss_esti(i)=1;
85 -         if d_k_Gauss_esti(i)~=d_k(i)
86 -             bit_Gauss_err=bit_Gauss_err+1;
87 -         end
88 -     end
89 -     if decision_k(i)<0
90 -         d_k_Gauss_esti(i)=-1;
91 -         if d_k_Gauss_esti(i)~=d_k(i)
92 -             bit_Gauss_err=bit_Gauss_err+1;
93 -         end
94 -     end
95 - end
96 - Pb(g)=bit_err/bit_num;
97 - Pb_Gauss(g)=bit_Gauss_err/bit_num;
98 - end
99 %-----繪製圖形-----%
100 - SNR=0.1:0.1:20;
101 - semilogy(SNR,Pb,'b',SNR,Pb_Gauss,'r-');
102 - xlabel('Eb/No');ylabel('BER');legend('Jakes Rayleigh','Gaussian Rayleigh')
103 - title('Performance comparison(BPSK)(L=2)');
104 - grid on;

```

b.diversity(L=3)

```
1 - clc;clear;
2 - %-----先給定參數-----%
3 - T=3;                                % time
4 - delta_t=0.0001;
5 - N=50;                                 % number of scatters
6 - %注意此處數值不能過大，大約500左右負荷不了
7 - fc=3.5*10^9;
8 - c=3*10^8;
9 - Eo=sqrt(10);
10 - v=(60000)/3600;
11 - landa=c/fc;
12 - c_m=sqrt(1/N);
13 - w_n=(2*pi*v)/landa;
14 - g=0;
15 - t=0:delta_t:T;
16 - m=0:1:N-1;
17 - rfa_m=(2*pi*m)/N;                  % angle of scatters
18 - %-----計算Jakes model的h(t)-----%
19 - for i=1:N
20 - phi_m=2*pi*rand(1,1);
21 - h1(i,:)=c_m*exp(j*(cos(rfa_m(i))*w_n*t+phi_m));
22 - end
23 - for i=1:N
24 - phi_m=2*pi*rand(1,1);
25 - h2(i,:)=c_m*exp(j*(cos(rfa_m(i))*w_n*t+phi_m));
26 - end
27 - for i=1:N                         %由於L=3，因此要有第三個path的signal
28 - phi_m=2*pi*rand(1,1);
29 - h3(i,:)=c_m*exp(j*(cos(rfa_m(i))*w_n*t+phi_m));
30 - end
31 - h_t1=Eo*sum(h1);
32 - h_t2=Eo*sum(h2);
33 - h_t3=Eo*sum(h3);
34 - %-----產生c[k]-----%
35 - c_k1=h_t1;                          %Rayleigh
36 - c_k2=h_t2;
37 - c_k3=h_t3;
38 -
39 - c_k1_Gauss=(randn(1,length(c_k1))+j*randn(1,length(c_k1)));//Gaussian
40 - c_k2_Gauss=(randn(1,length(c_k2))+j*randn(1,length(c_k2)));
41 - c_k3_Gauss=(randn(1,length(c_k1))+j*randn(1,length(c_k1)));
42 - %-----產生d[k]-----%
43 - bit_num=length(c_k1);
44 - d_k=source(bit_num);
45 - %-----計算Eb且建立BPSK model-----%
46 - for SNR=0.1:0.1:20
47 - mean_c_k1=mean(c_k1);mean_c_k1_Gauss=mean(c_k1_Gauss);
48 - var_c_k1=std(c_k1)^2;var_c_k1_Gauss=std(c_k1_Gauss)^2;
49 - %-----此處Eb只計算一個diversity channel 因為其他channel應會相同-----%
50 - Eb=mean(abs(c_k1).^2);
51 - No=Eb/SNR;
```

```

52 - Eb_Gauss=mean(abs(c_k1_Gauss).^2);
53 - No_Gauss=Eb_Gauss/SNR;
54 - g=g+1;
55 - var_w=No/2;
56 - bit_err=0;
57 - bit_Gauss_err=0;
58 - var_w_Gauss=No_Gauss/2;
59 - w_k1=sqrt(var_w)*randn(1,length(d_k));
60 - w_k2=sqrt(var_w)*randn(1,length(d_k));
61 - w_k3=sqrt(var_w)*randn(1,length(d_k));
62 -
63 - w_k1_Gauss=sqrt(var_w_Gauss)*randn(1,length(d_k));
64 - w_k2_Gauss=sqrt(var_w_Gauss)*randn(1,length(d_k));
65 - w_k3_Gauss=sqrt(var_w_Gauss)*randn(1,length(d_k));
66 -
67 - x_k1=c_k1.*d_k+w_k1;
68 - x_k2=c_k2.*d_k+w_k2;
69 - x_k3=c_k3.*d_k+w_k3;
70 -
71 - x_k1_Gauss=c_k1_Gauss.*d_k+w_k1_Gauss;
72 - x_k2_Gauss=c_k2_Gauss.*d_k+w_k2_Gauss;
73 - x_k3_Gauss=c_k3_Gauss.*d_k+w_k3_Gauss;
74 - %-----x_k2.*conj(c_k2)+x_k3.*conj(c_k3));此處在做MRC combining-----%
75 - decision_k=real(x_k1.*conj(c_k1)+x_k2.*conj(c_k2)+x_k3.*conj(c_k3));
76 - decision_k_Gauss=real(x_k1_Gauss.*conj(c_k1_Gauss)+x_k2_Gauss.*conj(c_k2_Gauss)+x_k3_Gauss.*conj(c_k3_Gauss));
77 - %-----計算錯多少bit及BER-----%

```

<pre> 78 - for i=1:length(x_k1) 79 - if decision_k(i)>=0 80 - d_k.est(i)=1; 81 - if d_k.est(i)~=d_k(i) 82 - bit_err=bit_err+1; 83 - end 84 - end %以下大致相似 85 - if decision_k(i)<0 86 - d_k.est(i)=-1; 87 - if d_k.est(i)~=d_k(i) 88 - bit_err=bit_err+1; 89 - end 90 - end 91 - end 92 - for i=1:length(x_k1_Gauss) 93 - if decision_k_Gauss(i)>=0 94 - d_k_Gauss.est(i)=1; 95 - if d_k_Gauss.est(i)~=d_k(i) 96 - bit_Gauss_err=bit_Gauss_err+1; 97 - end 98 - end </pre>	<pre> 99 - if decision_k(i)<0 100 - d_k_Gauss.est(i)=-1; 101 - if d_k_Gauss.est(i)~=d_k(i) 102 - bit_Gauss_err=bit_Gauss_err+1; 103 - end 104 - end 105 - end 106 - Pb(g)=bit_err/bit_num; 107 - Pb_Gauss(g)=bit_Gauss_err/bit_num; 108 - end 109 - %-----繪製圖形-----% 110 - semilogx(SNR,Pb,'b',SNR,Pb_Gauss,'r-.'); 111 - xlabel('Eb/No'); 112 - ylabel('BER'); 113 - legend('Jakes Rayleigh','Gaussian Rayleigh'); 114 - title('Performance comparison(BPSK)(L=3)'); 115 - grid on; </pre>
---	--