通訊系統電腦模擬與量測

Simulations and Measurements of Communication Systems

國立臺灣海洋大學 通訊與導航工程學系

Oral Report_2

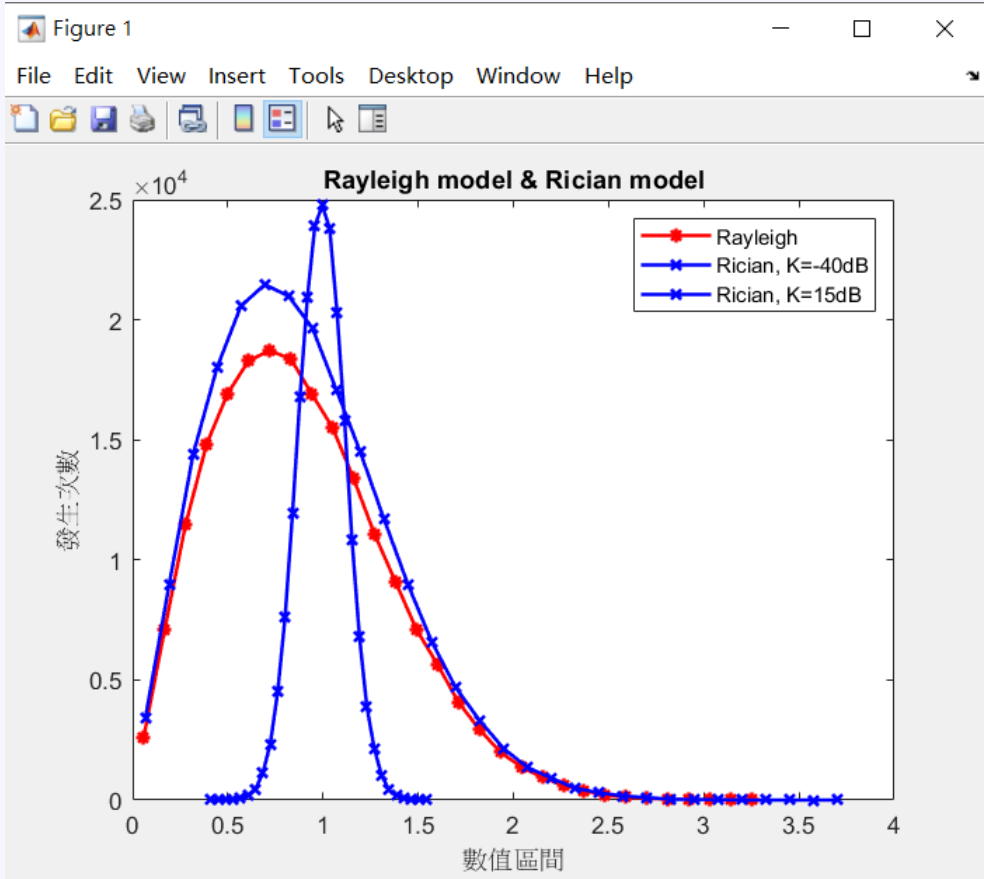
系級：通訊4A
學號：0086C035
姓名：余佳駿

Experiment # 5

Channel Model Simulations

Rayleigh model & Rician model

```
%% Rayleigh model
Rayleigh_ch=Ray(N);
cc=abs(Rayleigh_ch);
[temp,x]=hist(abs(Rayleigh_ch),level); % temp:個別區間個數 ; x: 區間間隔點

plot(x,temp,['-*r'],'LineWidth',1.5), hold on

%% Rician model
% 萊斯分佈實際上可以理解為主信號與服從瑞利分佈的多路徑信號分量的和
% K定義為主信號的功率與多徑分量功率之比
% K稱作萊斯因子，由K可以完全確定萊斯分佈。
% K=A^2/(2·σ^2)，當A->0時，萊斯分佈轉變為瑞利分佈。

for i=1:length(K_dB);
    Rician_ch(i,:)=Ric(K_dB(i),N);
    [temp x]=hist(abs(Rician_ch(i,:)),level);
    plot(x,temp,['-xb'],'LineWidth',1.5);
end

title('Rayleigh model & Rician model');
xlabel('區間'), ylabel('發生次數')
legend('Rayleigh','Rician, K=-40dB','Rician, K=15dB')

%% Rayleigh Channel Model
function model=Ray(L)

model = (randn(1,L)+j*randn(1,L))/sqrt(2);

end

%% Rician Channel Model

function model=Ric(K_dB,L)

K=10^(K_dB/10); % Rician factor
model = sqrt(K/(K+1)) + sqrt(1/(K+1))*Ray(L);

end
```



Experiment # 6

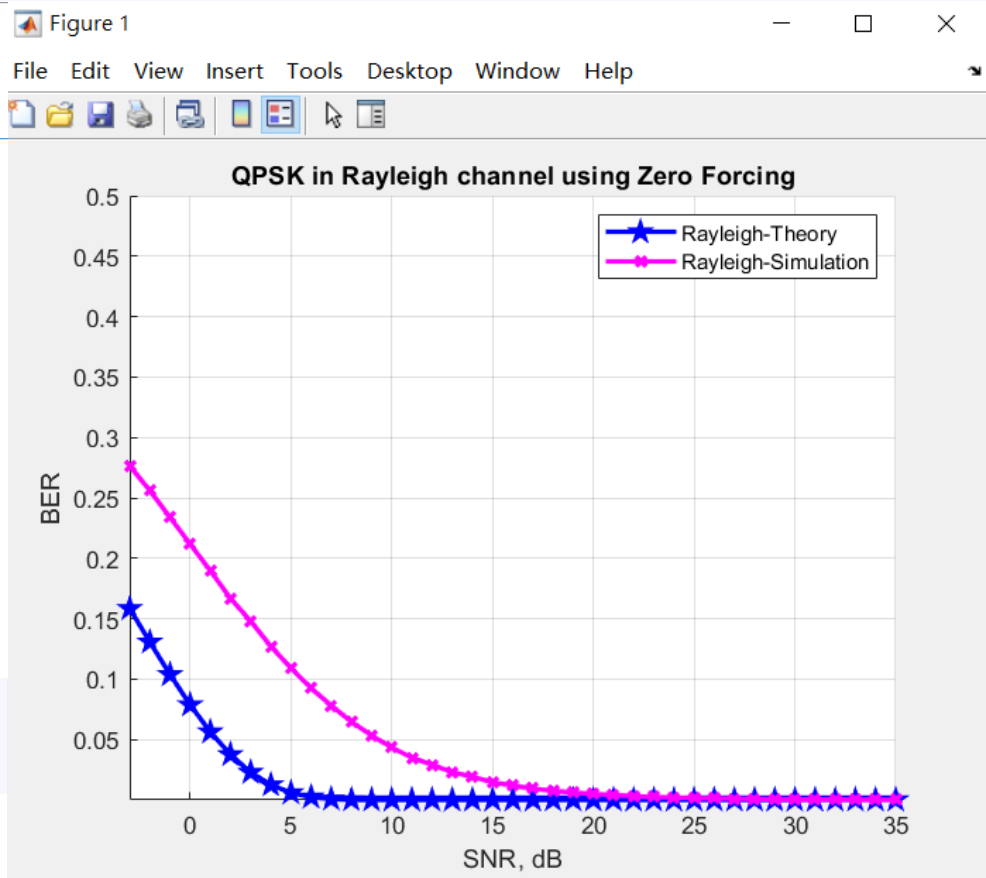
Inter-Symbol Interference (ISI) and Equalizer

Rayleigh channel using Zero Forcing

```
%% 生成QPSK
[mod,demod] = qpsk();

s = step(mod,ip);   % 做調變，length(s)=5000
Eb_N0_dB = [-5:35]; % multiple Eb/N0 values

%% Noise addition , equalization and error calculation
for i = 1:length(Eb_N0_dB)

   % white gaussian noise
   n = 1/sqrt(2)*[10^(-Eb_N0_dB(i)/20)]*[randn(length(s),1) + j*randn(length(s),1)];

   % Rayleigh channel
   h = 1/sqrt(2)*[randn(length(s),1) + j*randn(length(s),1)];
    % Rayleigh distribution: 當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的正態分佈時，這個向量的模呈瑞利分佈

   % Channel and noise Noise addition
   y = h.*s + n;
   % equalization，zero forcing
   yHat = y./h;

   %解調變輸出
   op = step(demod,yHat);

   % counting the errors
   nErr(i) = size(find([ip-op]),1);

end
```

```
%% BER CALCULATION

simBer = nErr/N; % simulated ber
%theoryBerAWGN = 0.5*(sqrt(10.^(Eb_N0_dB/10))); % theoretical ber
EbN0Lin = 10.^(Eb_N0_dB/10);
theoryBer = 0.5.*erfc(sqrt(EbN0Lin));
```

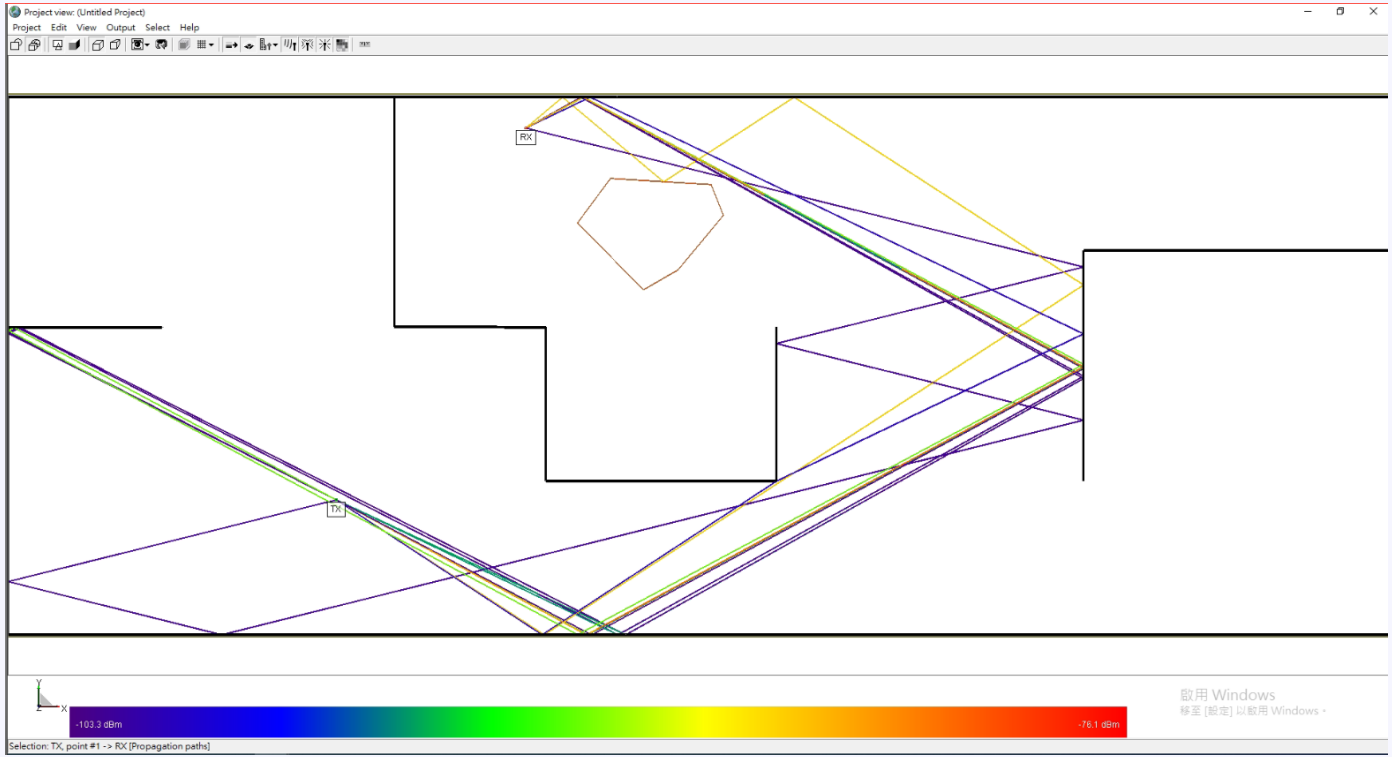
```
function [mod,demod] = qpsk()
mod = comm.QPSKModulator();
mod.BitInput = true;
demod = comm.QPSKDemodulator();
demod.BitOutput = true;
end
```



Experiment # 7 # 8

RayTracing Simulations

RayTracing Indoor



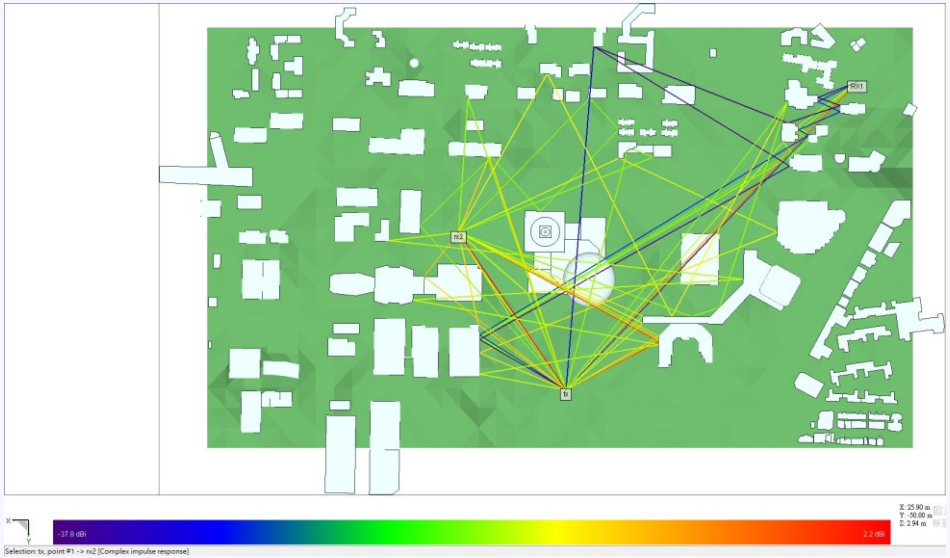
室內場景模擬

RayTracing

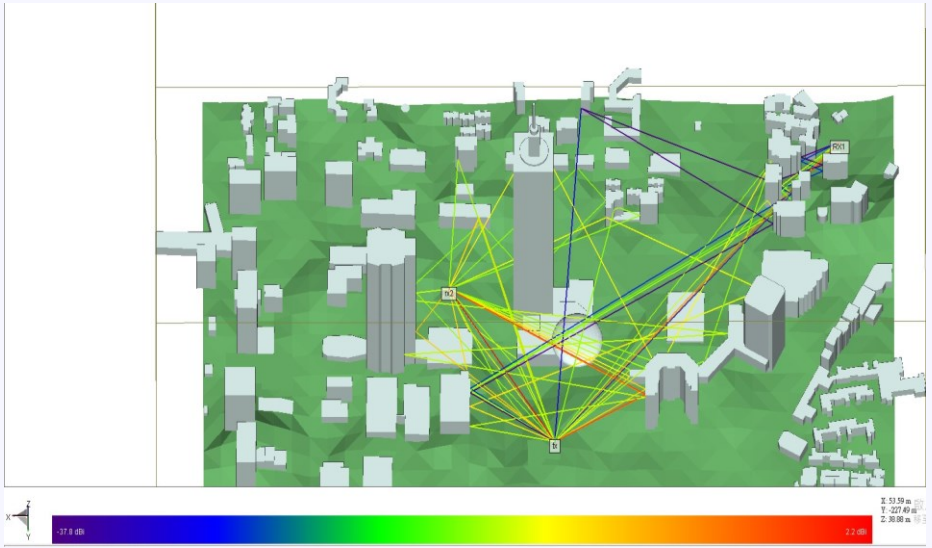
path_number	phase_value_deg	mean_time_of_arrival_sec	received_power_dBm
		1.0e-05 *	
1	112.0980		-80.8188
2	-80.1748	0.0644	-81.0100
3	-117.3310	0.0644	-81.2613
4	1.1366	0.0644	-82.2959
5	95.0446	0.0644	-84.0905
6	-16.4072	0.0645	-85.3347
7	-75.2700	0.0645	-85.3354
8	58.0946	0.0683	-85.7464
9	-77.6735	0.0683	-90.2012
10	-172.3930	0.0966	-93.3004
11	-28.6490	0.0644	-94.3684
12	-167.6860	0.0644	-94.6443
13	101.9400	0.0644	-94.8159
14	-37.1020	0.0644	-95.0909
15	-40.6911	0.0644	-95.2747
16	136.3020	0.0644	-100.4030
17	-55.9456	0.0645	-100.5940
18	-93.0254	0.0645	-100.8430
19	7.5597	0.0646	-102.4000
20	159.9570	0.1153	-102.6170
21	-147.7890	0.1153	-102.7140
22	7.4686	0.0966	-102.8060
23	-0.7403	0.0966	-102.9840
24	154.5190	0.0966	-103.0760
25	169.4250	0.0966	-103.3240
		0.0645	

RayTracing Outdoor

地區: 台北101



Propagation 2D



Propagation 3D

室外場景模擬

接收端資訊

```
# <Transmitter Set: Tx: 3 tx - Point 1>
# <Receiver Set: Rx: 7 RX1 >
# <number of receiver points>
# <receiver point number> <number of paths for this point>
# <path number> <phase value(deg)> <mean time of arrival(sec)> <received power(dBm)>
 1
1 11
1 -103.83 2.18722e-06 -103.364
2 -163.085 2.18767e-06 -116.019
3 46.624 2.2649e-06 -131.882
4 76.2179 2.35971e-06 -136.958
5 -60.2449 2.96439e-06 -137.054
6 -139.57 2.97003e-06 -140.143
7 -18.3771 3.01106e-06 -145.926
8 61.4075 2.466e-06 -172.713
9 141.348 3.16549e-06 -173.423
10 126.806 3.48398e-06 -184.725
11 -80.8828 5.24286e-06 -199.436
```

RX1(較遠)

```
# <Transmitter Set: Tx: 3 tx - Point 1>
# <Receiver Set: Rx: 8 rx2 >
# <number of receiver points>
# <receiver point number> <number of paths for this point>
# <path number> <phase value(deg)> <mean time of arrival(sec)> <received power(dBm)>
 1
1 25
1 -78.1984 9.84348e-07 -88.429
2 123.576 1.71802e-06 -101.135
3 79.3609 1.71875e-06 -112.664
4 47.113 1.22549e-06 -123.698
5 -83.025 1.69177e-06 -123.837
6 -24.8853 1.71882e-06 -124.943
7 131.733 1.68887e-06 -125.622
8 -140.446 9.96798e-07 -126.046
9 -28.5455 1.05099e-06 -126.991
10 85.054 3.96363e-06 -128.251
11 -73.53 9.98456e-07 -128.773
12 110.087 1.58572e-06 -131.482
13 128.796 5.49612e-06 -132.084
14 -87.1777 3.96364e-06 -134.971
15 116.359 1.69207e-06 -135.028
16 -14.2978 1.06726e-06 -137.615
17 94.6839 1.59586e-06 -137.946
18 131.207 6.60473e-06 -138.231
19 -32.0338 1.0627e-06 -138.856
20 173.378 1.77153e-06 -139.623
21 -77.7435 1.03435e-06 -140.319
22 55.942 2.27423e-06 -141.701
23 -108.57 2.50021e-06 -142.148
24 -69.8603 4.43731e-06 -142.265
25 -155.921 3.8029e-06 -142.289
```

RX2(較近)

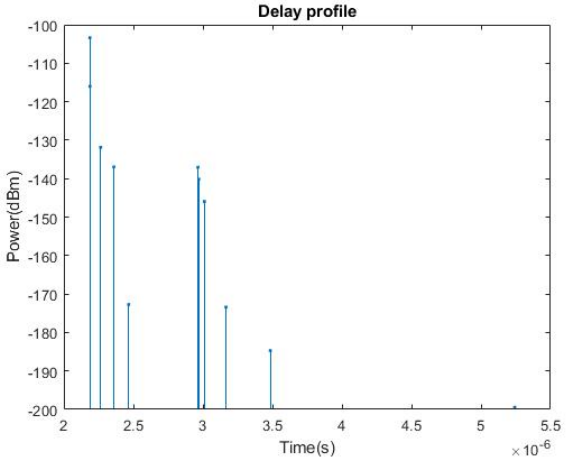
Delay Profile

```
clear all;
close all;
```

```
%% 讀取數據
filename=['C:\Users\User\Downloads\0086C035\x3d\TEST.cir.t001_03.r008.p2m'];
%從第7列開始讀取
DP=textread(filename, '' , 'headerlines', 7) ;
%依照抵達時間排序
DP=sortrows(DP,3);
```

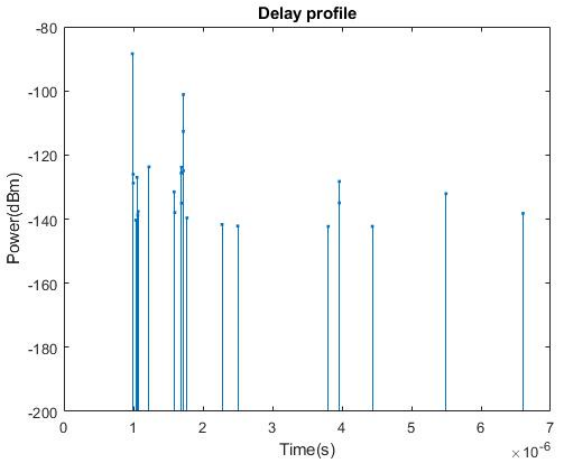
```
%% 繪製圖表
Time=DP(:,3);  % mean time of arrival(sec)
Power=DP(:,4); % received power(dBm)
% BaseValue，圖表中的所有條形序列具有相同的基準線
% 坐標軸從-200開始
stem(Time,Power,'.','BaseValue',-200);
title('Delay profile');
xlabel('Time(s)');ylabel('Power(dBm)');
```

RX1



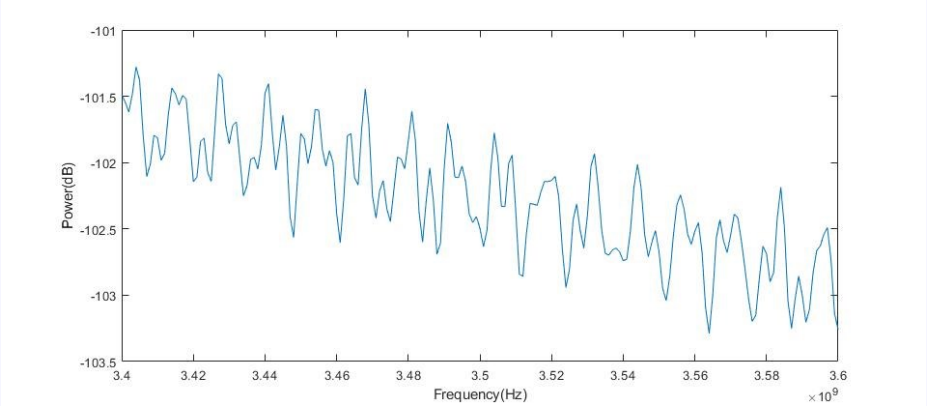
較差

RX2



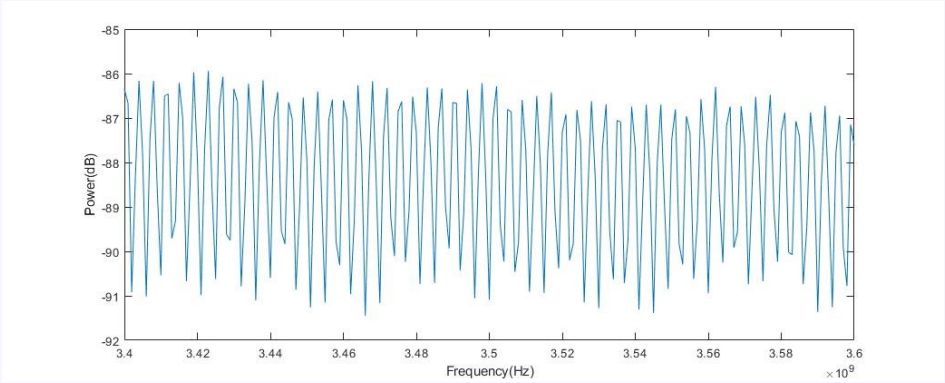
較優

Frequency Response



Frequency response RX1

訊號品質較差



Frequency response RX2

訊號品質較優

Thanks