

知能情報実験II(第8回):「無線通信における降
水減衰特性のモデリング」

175751C 宮城孝明

平成31年1月23日

目 次

1	実験の目的	2
2	実験の概要	2
3	実験の結果	2
3.1	10s ずつの変換と物理量の変換	2
3.1.1	18GHz コード	2
3.1.2	26GHz コード	4
3.2	降水強度のデータプログラム	5
3.2.1	降水時間を変換コード	5
3.2.2	降水間隔	6
3.2.3	コード	6
3.3	グラフ	8
3.3.1	コード	8
3.3.2	出力	8
4	考察	10

1 実験の目的

衛星通信や FWA(Fixed Wireless Access:固定無線通信)といった、10GHzを超えるような周波数帯を用いた無線通信では、降雨現象が大きな問題となる事が知られている。とりわけ、沖縄県のような亜熱帯地方に属する気候区では、スコールと呼ばれる大雨や台風による豪雨など、年間を通して頻繁に発生する。そのような地方で安定した無線通信を行うには、降雨による電波の減衰(降雨減衰と呼びます)を定量的に把握し、それに対処する必要がある。つまり、降雨減衰特性をモデリングすることが必須になる。本実験では、本学1号館の屋上に設置された FWA(18GHz帯、26GHz帯)の受信レベルと降雨強度(1分間の降雨量)の3つのデータを用いて、琉球大学周辺における18GHz帯、26GHz帯の無線通信における降雨減衰特性のモデリングを行う。なお、今回用いるデータはここだけにしか存在しない学術的に貴重なもの(研究論文のネタになる!)であり、宮里研究室の卒業研究テーマの一つに活用されている。本実験を通して、卒業研究の入り口を体験してみよう。

2 実験の概要

実験中に与えられたデータをもとに、手順1から7を通し、データを加工する。加工したデータを可視化し易いようにグラフにして示す。

3 実験の結果

3.1 10s ずつの変換と物理量の変換

3.1.1 18GHz コード

Listing 1: Rx18GHz_data

```
1 import os
2 import pandas as pd
3 import numpy as np
4 import datetime
5
6 data_path = "/Users/e175751/Desktop/programing課題
7 /2/experiment第八回/"
8 file_path = "/Users/e175751/Desktop/programing課題
9 /2/experiment第八回/"
10
11 month = ["06", "10", "11", "12"]
12
13 for i in month:
14     #print(i)
```

```

13     for j in range(1,32,1):
14         #print(j)
15         path = ["RxData/2009" + str(i) + "/2009" + str(i) +
16                 str(j).zfill(2) + "/192.168.100._csv.log"]
17         #print(path)
18         for k in path:
19             #print(k)
20             try:
21                 file = pd.read_csv(k,header=1)
22
23             except :
24                 continue
25
26             file_time = file[["time"]]
27             file = file[["_1803_RX_LEVEL"]]
28
29             try:
30                 f = lambda x: (x + 256)/2 - 121 if x < 0
31                     else x/2 - 121
32                 file_Rx = file[["_1803_RX_LEVEL"]].map(f)
33
34             except TypeError:
35                 continue
36
37             #との配列を列ごとに繋げるfile_timefile_rx
38             file_join = pd.concat([file_time, file_Rx], axis
39                                   =1)
40
41             #時間(columns)のという名前をつけるtime
42             file_join = file_join.set_index("time")
43
44             #を型に変換するtimedatettime
45             file_join.index = pd.to_datetime(file_join.index
46                                               )
47
48             #のを秒ごとに分割しfile_joointime10,の平均を取るRx
49             res = file_join.resample("10S").mean()
50
51
52             #でを指定しreset_indextime時間を整える,
53             df = res.reset_index("time")
54             #print(df6)
55
56             #結果をに書き込むfile
57             df.to_csv("data_Rx23.csv",mode="a",header=False)
58
59
60
61 print("終了しました")

```

3.1.2 26GHz コード

Listing 2: Rx26GHz_data

```
1 import os
2 import pandas as pd
3 import numpy as np
4 import datetime
5
6
7 data_path = "/Users/e175751/Desktop/programing課題
8 /2/experiment第八回/"
9 file_path = "/Users/e175751/Desktop/programing課題
10 /2/experiment第八回/"
11
12 month = ["06","10","11","12"]
13
14 for i in month:
15     #print(i)
16     for j in range(1,32,1):
17         #print(j)
18         path = ["RxData/2009" + str(i) + "/2009" + str(i) +
19                 str(j).zfill(2) + "/192.168.100.11_csv.log"]
20         #print(path)
21         for k in path:
22             #print(k)
23             try:
24                 file = pd.read_csv(k,header=1)
25
26             except :
27                 continue
28
29             file_time = file[["time"]]
30             file = file[["_MX_RX_LEVEL"]]
31
32             try:
33                 f = lambda x: (x + 256)/2 - 121 if x < 0
34                     else x/2 - 121
35                 file_Rx = file["_MX_RX_LEVEL"].map(f)
36
37             except TypeError:
38                 continue
39
40             #との配列を列ごとに繋げるfile_timefile_rx
41             file_join = pd.concat([file_time, file_Rx], axis
42                                   =1)
43
44             #時間(columns)のという名前をつけるtime
45             file_join = file_join.set_index("time")
```

```

46         #を型に変換する timedatetime
47         file_join.index = pd.to_datetime(file_join.index
48         )
49
50         #のを秒ごとに分割し file_jovertime10, の平均を取る Rx
51         res = file_join.resample("10S").mean()
52
53
54         #でを指定し reset_indextime 時間を整える,
55         df = res.reset_index("time")
56         #print(df6)
57
58         #結果をに書き込む file
59         df.to_csv("data_Rx26.csv", mode="a", header=False)
60
61
62
63     print("終了しました")

```

3.2 降水強度のデータプログラム

3.2.1 降水時間を変換コード

Listing 3: Rain_data

```

1  import os
2  import pandas as pd
3  import numpy as np
4  import datetime
5
6  data_path = "/Users/e175751/Desktop/programing課題
7             /2/experiment第八回/"
8
9  month = ["06", "10", "11", "12"]
10
11  for i in month:
12      for j in range(1, 32, 1):
13          path = ["RainData/2009" + str(i) + str(j).zfill(2) +
14                  "/2009" + str(i) + str(j).zfill(2) + "_rain.csv
15                  "]
16          for k in path:
17              try:
18                  df = pd.read_csv(k)
19                  #print(file)
20              except:
21                  continue
22
23              try:
24                  f = lambda x: x * 60 * 0.0083333
25                  df2 = df["Recording_ started."].map(f)
26                  #print(file_Rx)

```

```

24         except:
25             continue
26
27         df3 = df2.fillna(0)
28         df3.to_csv("Rain_data.csv",mode="a",header=False
29                    )
30 print("終了です")

```

3.2.2 降水間隔

RainData の最大値が 150mm/h : 5mm とする
Rx18GHz の最大値が -25mm/h : -3mm とする
Rx26GHz の最大値が -10mm/h : -3mm とする

3.2.3 コード

Listing 4: Rain_result

```

1 import pandas as pd
2 data_ave = []
3 data_cu=[]
4
5 file = pd.read_csv("Rain_data.csv",header=None)
6 data = file[1]
7 cu=0
8 for i in range(150,-5,-5):
9     data_ave.append(i)
10    num=0
11    for k in range(0,len(file)):
12        if i-5< data[k] and data[k] <= i:
13            num += 1
14    num = num / len(file)
15    cu += num*100
16    data_cu.append(cu)
17
18 df = pd.DataFrame({
19     "frequency" : data_ave,
20     "Accumulation" : data_cu}
21 )
22 df.to_csv("Rain_result.csv",mode="w",header=False,index=
23          False)
24 print("finish")

```

Listing 5: Rx18GHz_result

```

1 import pandas as pd
2 file = pd.read_csv("data_Rx18.csv",header=None)
3 data_num = []

```

```

4 data_cu=[]
5 data_ave=[]
6 data = file[2]
7
8 cu=0
9 for i in range(-25,-100,-3):
10     data_ave.append(i)
11     num=0
12     for k in range(0,len(file)):
13         if i-3 < data[k] and data[k] <= i:
14             num+=1
15     num = num / len(file)
16     cu += num*100
17     data_cu.append(cu)
18
19 df = pd.DataFrame({
20     "frequency" : data_ave,
21     "Accumulation" : data_cu}
22 )
23 df.to_csv("Rx18_result.csv",mode="w",header=False,index=
24     False)
25 print("finish")

```

Listing 6: Rx26GHz_result

```

1 import pandas as pd
2 file = pd.read_csv("data_Rx26.csv",header=None)
3 data_num = []
4 data_cu=[]
5 data_ave=[]
6 data = file[2]
7
8 cu=0
9 for i in range(-10,-50,-3):
10     data_ave.append(i)
11     num=0
12     for k in range(0,len(file)):
13         if i-3 < data[k] and data[k] <= i:
14             num+=1
15     num = num / len(file)
16     cu += num*100
17     data_cu.append(cu)
18
19 df = pd.DataFrame({
20     "frequency" : data_ave,
21     "Accumulation" : data_cu}
22 )
23 df.to_csv("Rx26_result.csv",mode="w",header=False,index=
24     False)
25 print("finish")

```


3.3 グラフ

3.3.1 コード

Listing 7: Rain_grahp

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 plt.xscale("log")
5 df = pd.read_csv("Rain_result.csv",header=None)
6 plt.grid(which="both")
7 plt.plot(df[1],df[0])
8 plt.savefig('Rain_cd.png')
```

Listing 8: Rx18_graph

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 plt.xscale("log")
5 df2 = pd.read_csv("Rx18_result.csv",header=None)
6 plt.grid(which="both")
7 plt.plot(df2[1],df2[0])
8 plt.savefig('Rx18_cd.png')
```

Listing 9: Rx26GHz_graph

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 plt.xscale("log")
5 df3 = pd.read_csv("Rx26_result.csv",header=None)
6 plt.grid(which="both")
7 plt.plot(df3[1],df3[0])
8 plt.savefig('Rx26_cd.png')
```

3.3.2 出力

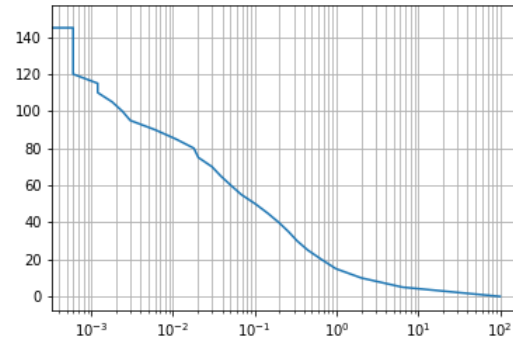


図 1: Rain データの累積分布図

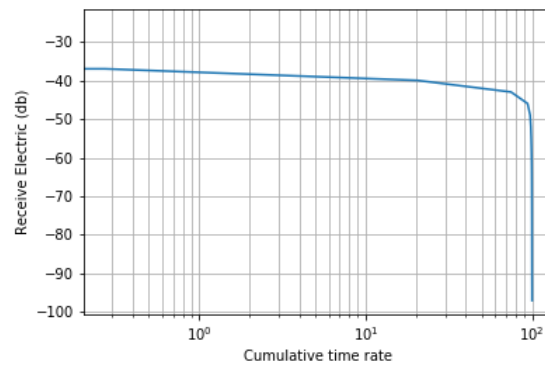


図 2: Rx18 の累積分布図

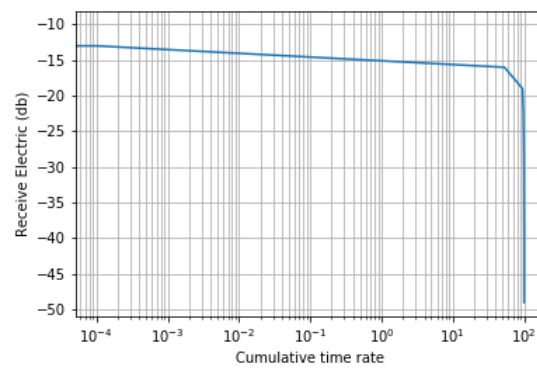


図 3: Rx26 の累積分布図

4 考察

Rain の累積分布は,100(mm/h) を超える確率は,100 分の 1 以下程度であるとわかる. つまり,1 年に 1 時間程度しか 100mm/h の雨が降らないことがこのグラフからわかる.

結果から見ると, 降雨強度が高ければ, 受信電界が低なる. つまり, 降雨量が上がれば電波通信に影響が起きることがわかる. Rx18GHz と Rx26GHz の累積分布図を見してみる. 一見どちらも似たようなグラフになっているが,Rx26GHz の方が累積時間率が細かくなっている. さらに, 受信電界も最大値は Rx26GHz が高い. つまり,Rx26GHz の方が Rx18GHz より降雨量の影響が受けないと予想される.